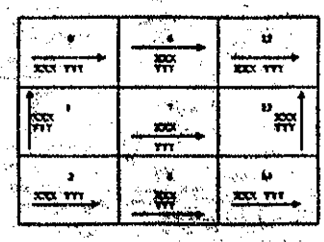
2D/3D MODE
no code #
(Section 7.3.4)

Toggles between 2D and 3D input modes. 2D mode suppresses elevation prompting in POINT LOCATION commands. 3D mode enables prompting for elevations of points to be defined and vertical angles for points to be located.

ACTIVE FEATURE
no code #
(Section 7.2.5)

Toggles between having an active feature and not having an active feature for point annotation operations.

ACTIVE JUSTIFICATION
no code #
(Section 7.3.6)



Allows the current text justification set from the IGDS TEXT JUSTIFICATION to override the current feature justification or the justification stored in the database with a point. ACTIVE JUSTIFICATION is used with ANNOTATE commands, PLOT POINTS, and STORE.

ACTIVE OFFSET
no code #
(Section 7.2.4)

Toggles between using active offsets and not using active offsets.

ACTIVE OFFSET
no code #
(Section 7.3.1)

Defines values for the active offsets *off1* and *off2*.

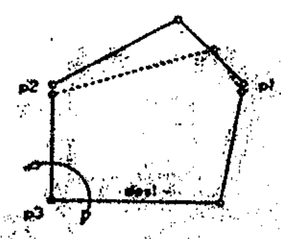
ADJUST AREA des1 area p1 p2 p3 p4 /des2
ROTATE
NORMAL
PARALLEL
code # 42
(Section 10.1.3)

Adjusts the area described by *des* to equal *area*. Three adjustment options are available:

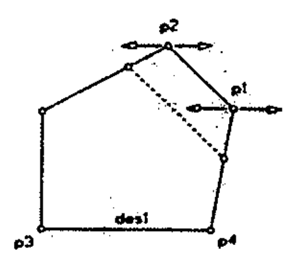
1) Rotate – rotates sides between *p1* and *p2* about *p3* (p4=0).

2) Normal – moves sides between *p1* and *p2* in a direction perpendicular to the line defined by *p1* and *p2* (p3=0, p4=0).

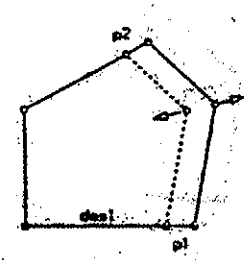
3) Parallel – moves sides between *p1* and *p2* in a direction parallel to the line defined by *p3* and *p4*.




AREA ROTATE

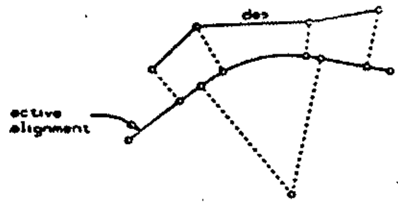



AREA PARALLEL




AREA NORMAL

**ALIGNMENT OFFSET** des /x /n /sBEG sEND
code #97
(Section 9.2.6)




Locates points on the active alignment perpendicular to *des*. The optional interval *x* begins at the first point in *des*.

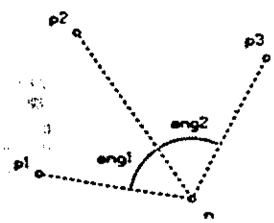
---

**ALPHA SCROLL**
no code #
(Section 7.2.3)

Toggles between displaying and not displaying output on the tutorial screen while running ICS in graphics.

---

**ANGLE RESECTION** p1 p2 p3 ang1 ang2 /n
code #68
(Section 8.2.8)




Locates point *n* by resection, given three points and two angles to the points.

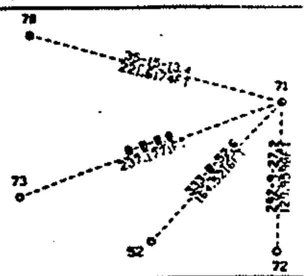
---

**ANGLES** pOC pBS desFS
code #62
(Section 11.4.3)

Computes and reports angles from *pBS* at *pOC* to points in *desFS*.

---

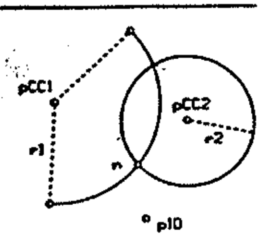
**ANGLES & DISTANCES**
no code #
(Section 13.4.1)




Annotates the line segments between the occupied point *pOC* and the foresight points in *des*. The angle right (from *pBS*) and distance annotation are placed on the lines between the *pOC* and the points in *des*.

---

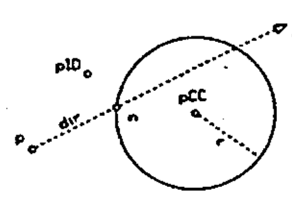
**ARC ARC INTERSECT** n PCC1 r1 pCC2 r2 pID
code #25
(Section 8.4.6)




Calculates the intersection (closest to *pID*) of:
(1) Circle 1 with center point *pCC1* and radius *r1*
(2) Circle 2 with center point *pCC2* and radius *r2*.

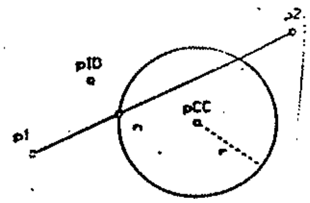
---

**ARC LINE DIRECTION** n pCC p dir pID /off
code # 46
(Section 8.4.5)




Calculates the intersection (closest to *pID*) of:
(1) The circle with center point *pCC* and radius *r*
(2) The line through the point *p* at direction *dir*.

**ARC LINE POINTS** n pCC r p1 p2 pID /off
code #22
(Section 8.4.4)




Calculates the intersection (closest to *pID*) of:
(1) The circle with center point *pCC* and radius *r*
(2) The line *p1 p2*.

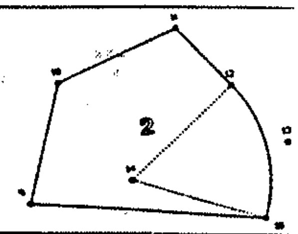
---

**AREA** des
code # 26
(Section 13.2.4)

In graphics, this command places the area of a polygon described in *des* in the center of the polygon. In alpha it displays the area on the screen. The first and last points in *des* must match.

---

**AREA DIRECTION** des
code #27
(Section 11.3.1)




Computes and reports the area and boundary courses of the closed figure given by *des*.

---

**AUTO PLOT** fnum /z /ffname
code #79
(Section 7.2.6)

Points and figures computed by ICS will be automatically plotted in the design file according to the feature selected. If *fname* is used, *fnum* is 0. If *z* is specified, all points will be plotted at that elevation in a 3D design file. EXCEPTION: points placed with a data point will be placed at active *z*. To turn AUTO PLOT OFF, *fnum*=0.

---

**AUTO PLOT**
no code #
(Section 7.2.6)

Toggles between automatically placing or not placing graphics for all points and figures created.

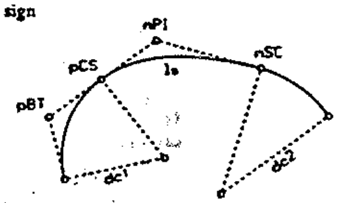
---

**CLEAR ALIGNMENT**
code #120
(Section 9.1.6)

Erases your current vertical alignment.

---

**COMPOUND SPIRAL** cid pBT pCS nSIT nSC ls dc1 dc2 sign
code #58
(Section 9.2.12)




Locates the SIT and SC of the spiral arc connecting two circular curves.

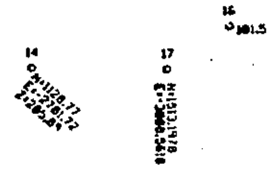
**COORDINATE TABLE** des
no code #
(Section 12.1.1)

Writes the coordinates of points you specify in *des* to an ASCII file containing the coordinate table. After creating the coordinate table, you can plot it to the design file using the graphics command PLOT FILE.

---

**COORDINATES**
no code #
(Section 13.2.1)

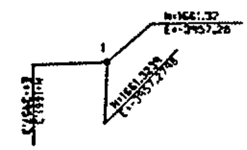
Annotates coordinates of points according to specifications in the feature table and active parameters. ACTIVE FEATURE, ACTIVE JUSTIFICATION, and ELEVATION ANNOTATION affect the outcome of this command.



---

**COORDINATES ANGLED**
no code #
(Section 13.3.1)

Places the coordinates of specified points along angled leaders. If you want to place identical leaders on a group of points, use a fence. Affected by ACTIVE ANGLE.



---

**COORDINATES STRAIGHT**
no code #
(Section 13.3.2)

Places the coordinates of specified points along straight leaders. If you want to place identical leaders on a group of points, use a fence.

---

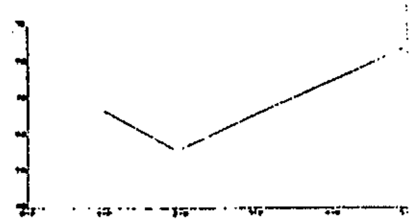
**COPY POINTS** des /n /dx /dy /dz
code # 134
(Section 8.1.3)

Copies the coordinates and elevations of points in *des* to new point ID's.

---

**CREATE PROFILE** des /Vfg /fnum or ffname
no code #
(Section 9.3.3)

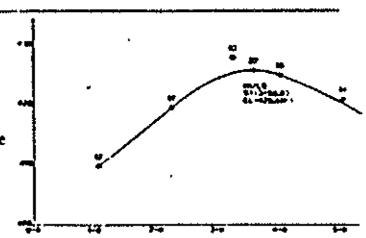
Plots a centerline profile given a description *des* of points with defined elevations.



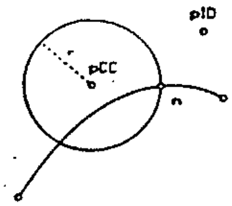
---

**CURVE DRAIN** sBEG /sEND /n
code # 117
(Section 9.3.6)

Computes and labels all high and low elevations on the vertical alignment between *sBEG* and *sEND*.



**CURVE SPIRAL** n r pCC pID
code #60
(Section 8.4.11)




Calculates the intersection (closest to *pID*) of:
(1) Previously defined spiral by SIMPLE SPIRAL or SPIRAL LENGTH and
(2) Curve defined by radius *r* and *pCC*.

**CURVE TABLE**
no code #
(Section 12.1.2)

Toggles between generating and not generating a curve table. If ON, when running the LABEL LINE command, CURVE TABLE writes curve data to an ASCII file containing the curve table.

**DEFINE Z** p z
no code #
(Section 8.1.5)

Defines the elevation of point *p*; *p* must be input with a data point.

**DELETE COORDS** des
code #87
(Section 8.1.2)

Deletes points in *des* from the database.

**DELETE FIGURES** des
code #88
(Section 9.1.4)

Deletes figure definitions (but not points) defined in *des* from the database.

**DESCRIBE ALIGNMENT** Hfg /x /sBEG /sEND
code #49
(Section 11.2.1)

Lists the following alignment data:

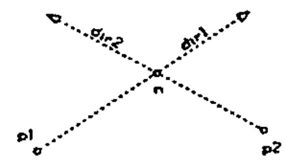
- o coordinates & sta of PC, PI, PT, CC
- o distances and directions of tangents
- o PI deflection angles
- o curve data.

**DESCRIBE VERTICAL ALIGNMENT** sBEG /sEND
code #112
(Section 11.2.2)

Describes a vertical alignment, including points of vertical inflection, curvature, and tangency and topographic high and low points, and grades.

3

**DIRECTION INTERSECT** n p1 dir1 p2 dir2 /off1 /off2 /va1 va2
code #48
(Section 8.4.2)




Calculates the intersection of:
(1) The line through *p1* at direction *dir1*
(2) The line through *p2* at direction *dir2*.
**Use SET HEIGHT to use vertical angles to calculate the mean elevation of *n*.

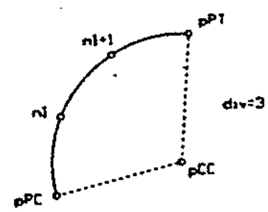
---

**DISTANCE** des
code #10
(Section 11.4.1)

Computes and reports distances between points in *des*.

---

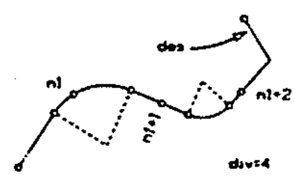
**DIVIDE ARC** pPC pPT pCC div /n
code #41
(Section 8.3.2)




Divides a clockwise arc with center *pCC* into *div* equal segments between *pPC* and *pPT*. Stores the resultant coordinates as *n*, *n+1*, etc.

---

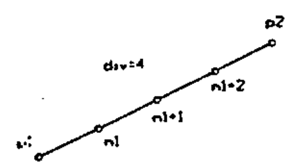
**DIVIDE FIGURE** des div /n
code #103
(Section 8.3.3)




Divides the figure *des* into *div* equal parts. Stores the resultant coordinates as points *n*, *n+1*, etc.

---

**DIVIDE LINE** p1 p2 div /n
code #40
(Section 8.3.1)




Divides line *p1 p2* into *div* equal parts. Stores the resultant coordinates as points *n*, *n+1*, etc.

---

**ELEVATION ANNOTATION**
no code #
(Section 13.1.1)

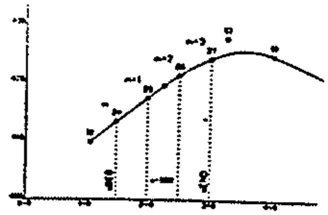
Sets the type of elevation annotation that ICS will use when labeling points (x, y, and z; x and y only; z only).

---

**END OF RUN**
no code #
(Section 6.2)

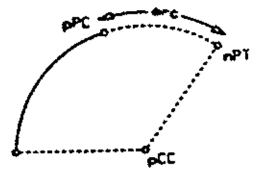
Identical to the alpha END OF JOB command except that an END OF RUN command terminates ICS. This command closes the ICS database, completes any plotting, sends the message ICS PROCESSING COMPLETE, and terminates ICS.

**EVEN STATIONS** x /sBEG /sEND /n
code #114
(Section 9.3.4)



Computes and plots points $n$, $n+1$, $n+2$, etc. at intervals of $x$ between *sBEG* and *sEND*. You can assign zero to *sBEG* and *sEND* to indicate *from beginning to end.*

**EXTEND ARC** pPC pCC nPT arc
code #67
(Section 8.2.7)




Given known points *pPC*, *pCC*, and *arc*, locates the *nPT* along the extension of the arc from *pPC*.

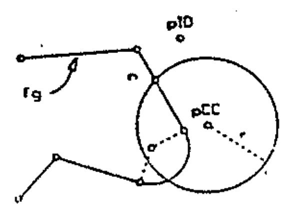
**FENCE**
no code #
(Section 7.2.1)

Toggles the active fence to on or off.

**FENCE CONTENT**
no code #
(Section 7.3.9)

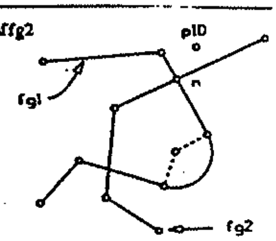
Toggles between points and figures for the fence content. When set to points, only points are located for a fence operation. When set to figures, figures are located for a fence operation.

**FIGURE ARC INTERSECTION** n fg pCC r pID /offg
code #99
(Section 8.4.8)



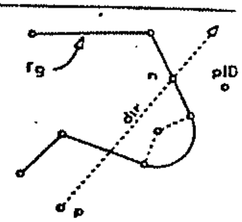

Calculates the intersection (closest to *pID*) of:
(1) The figure *fg*
(2) The arc with center point *pCC* and radius *r*.

**FIGURE FIGURE INTERSECTION** n fg1 fg2 pID /offg1 /offg2
code #100
(Section 8.4.9)




Calculates the intersection (closest to *pID*) of:
(1) The figure *fg1*
(2) The figure *fg2*.

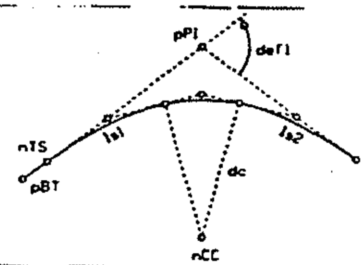
**FIGURE LINE INTERSECTION** n fg p dir pID /offg /off
code #98
(Section 8.4.7)




Calculates the intersection (closest to *pID*) of:
(1) The figure *fg*
(2) The line through *p* at direction *dir*.

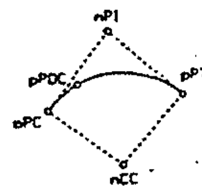
**FIT ALIGNMENT** nCC pBT pPI dc ls1 ls2 defl sign
code #51
(Section 9.2.13)

Fits a spiral–curve–spiral section into an alignment. Locates the points: *nTS*, *nSIT* (*in*), *nSC*, *nPI*, *nCS*, *nSIT* (*out*), *nST*, *nCC*, and *n* (mid curve).



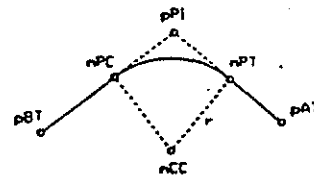

**FIT CURVE EDGE** pPC pPOC pPT
code # 106
(Section 9.2.8)

Calculates critical points of the curve through the *pPC*, a point on the curve *pPOC*, and the *pPT*.



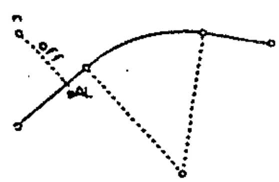

**FIT CURVE TANGENTS** pBT pPI pAT nPC nCC nPT /r
code # 106
(Section 9.2.8)

Calculates critical points of a curve tangent to the line connecting *pBT* to *pPI* and the line connecting *pAT*. Point *nPC* is the point of curvature, *nPT* is the point of tangency, and point *nCC* is the center of the curve. If radius *r* is not input, *pBT* is taken to be *pPC*.




**FROM ALIGNMENT** n sAL /off /rod
code # 96
(Section 9.2.5)

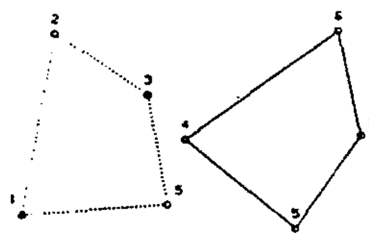
Locates point *n* at station *sAL* and optional offset *off*. Calculates the elevation of *n* if you specify *rod*.




**HELMERT TRANSFORM**
**APPLY TRANSFORM** des /distrib /verify
code #1
(Section 10.2.2)

Transforms points in *des* based on the transformation computed by COMPUTE TRANSFORM or defined in the graphics tutorial.

| | |
|---|---|
| Verify=0: | Adjust all points. |
| Verify=1: | Accept/reject each point on individual basis. |
| Distrib=0: | Do not apply residual distribution. |
| Distrib=1: | Apply residual distribution. |



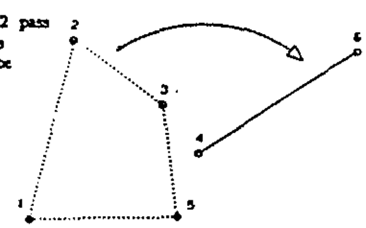

**HELMERT TRANSFORM**
**COMPUTE TRANSFORM** desFROM desTO /sca /3D
code #71
(Section 10.2.2)

These two commands are used together to perform a four-parameter conformal transform of the coordinate system of *desFROM* to the coordinate system of *desTO* as follows:

1) *desFROM* and *desTO* must contain a minimum of 2 pass points common to both coordinate systems. The pass points in each *des* must be in the same order and be equal in number.

2) Input of scale *sca* fixes the transformation scale.

3) A 3D (seven parameter) transformation can be selected by entering 1 for *3D*.



**ICS. PAR**
no code #
(Section 7.3.5)

Displays a tutorial that allows you to access much of the parameter file ICS.PAR and interactively edit the parameters while you are working in a design file. Changes you make to the parameter file through the ICS.PAR command are maintained between design sessions if the UPDATE option is selected.

---

**INPUT FILE** mode *filename
code #64
(Section 7.1.2)

Invokes batch operating mode and runs ICS commands from *dev:[udc]filename.ext.*
mode=0 lists input commands;
mode=1 does not list input.

---

**INPUT MODES**
code # 130
(Section 7.3.12)

Defines active units for curve mode, height mode, and unit mode or resets the modes to default values. Allows you to specify an input mode so that you input values can be in units other than the command default.

---

**INSERT CURVE**
no code #
(Section 9.1.2)

Inserts a curve with center *pCC* in a figure in graphics only.

---

**INSERT SPIRAL**
no code #
(Section 9.1.3)

Inserts a spiral in a figure in graphics only.

---

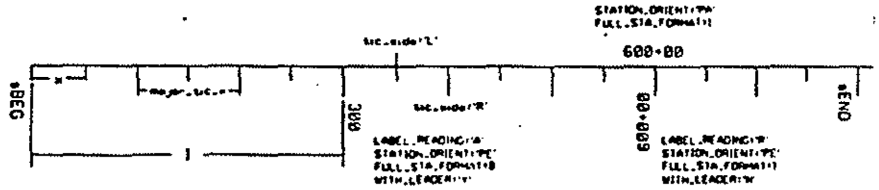
**INVERSE DIRECTION** des
code #16
(Section 11.4.2)

Computes and reports distances and directions between points in *des.*

---

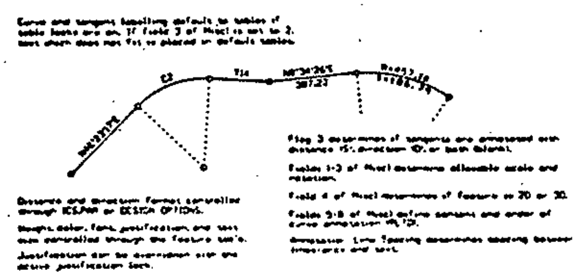
**LABEL ALIGNMENT** Hfg /x /1 /fnum /sBEG /sEND /lfname
code #83
(Section 13.2.2)

Places tic marks at each *x* interval and labels stationing at each *1* interval and transition points (*PC, PT, etc.*). If *lfname* is used, *fnum=0.* If only a portion will be labeled, specify begin and end station with *sBEG & sEND.* If x = 0, only points defining the alignment are labeled.



**LABEL LINES** des /fnum or Ifname
code #74
(Section 13.2.5)

Labels lines defined by *des* in the Feature Table.



**LABEL VERTICAL ALIGNMENT** sBEG /sEND /des /fnum or Ifname
code #119
(Section 13.2.3)

Labels the *PVI*, *PVC*, *PVT*, and the grade of tangents between *sBEG* and *sEND*. You can assign zero to *sBEG* and *sEND* to indicate *from beginning to end* .

**LABEL VERTICAL STATIONS** des /fnum or Ifname
code #121
(Section 13.2.6)

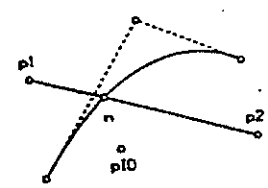
Labels points in *des* with stations and elevations.

**LEVEL RUN** pBS rBS /nFS rFS
code # 123
(Section 8.2.10)

Calculates the height of the instrument (H.I.) and, optionally, the elevation of *nFS*.

**LINE SPIRAL** n p1 p2 pID
code #57
(Section 8.4.10)

Calculates the intersection (closest to *pID*) of:
(1) Previously defined spiral by SIMPLE SPIRAL or SPIRAL LENGTH.
(2) The line *p1 p2*.




**LINE TABLE**
no code #
(Section 12.1.2)

Toggles between generating and not generating a line table. If ON, when running the LABEL LINE command, LINE TABLE writes line data to an ASCII file containing the curve table.

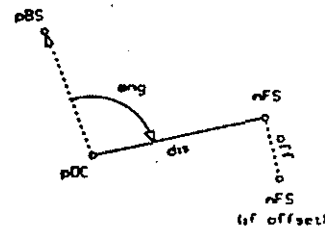
**LIST COORDS** des
code #81
(Section 11.1.1)

Lists coordinates of each point in *des*.

**LIST FIGURES** des
code #82
(Section 11.1.2)

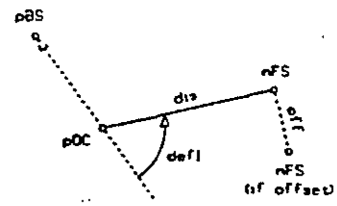
Lists figures and figure descriptions.

**LOCATE ANGLE** pBS pOC nFS ang dis /vaFS /off /vaBS pVB
code #13
(Section 8.2.3)



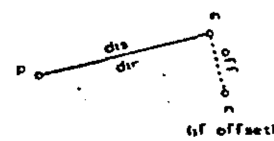

From the occupied point *pOC*, locates *nFS* at an angle *ang* from *pBS* to *n*, and a distance of *dis*. Optional *off* is the perpendicular offset distance from the *FS* line to *n*. Optional *vaBS* requires that the *BS* elevation and backsight target height be pre-defined with SET HEIGHT *htFS hIOC htBS*.

**LOCATE DEFLECTION** pBS pOC nFS defl dis /vaFS /off /vaBS pVB
code #14
(Section 8.2.4)



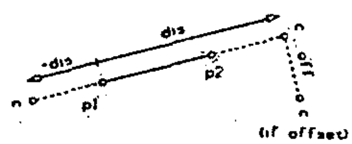

From the occupied point *pOC*, locates *nFS* at a deflection angle *defl* from *pBS* and at a distance *dis*. See LOCATE ANGLE for notes on *off*, *vaBS*, and *pVB*.

**LOCATE DIRECTION** p n dir dis /va /off
code #51
(Section 8.2.2)




From *p*, locates *n* at direction *dir* and distance *dis* at optional offset *off*.

**LOCATE LINE** p1 p2 n dis /va /off /Tdis
code #15
(SECTION 8.2.1)




From *p1* along the (extension of) line *p1*, *p2*, locates *n* at *dis* from *p1* (and optionally, offset *off*). If *Tdis* is specified, then *dis* will be scaled by the ratio (*p1 to p2*) *Tdis*.

**MOVE POINTS** des dx dy dz
code # 135
(Section 8.1.4)

Moves points in *des* to new locations and updates the database.

**NEW FEATURE TABLE**
code # 129
(Section 7.1.3)

Changes the feature table while you are in an active job.

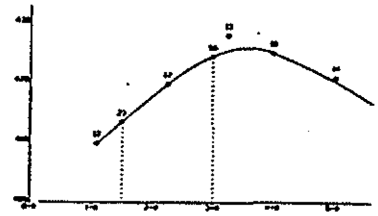
**NEW JOB** j1 j2 dev[UIC]fn.dbs *heading
code # 2
(Section 6.1)

Same as the START OF JOB command in alpha. This command instructs ICS to reset all indicators and to prepare for a new job. When running ICS in graphics, NEW JOB is performed automatically during initialization.

---

**ODD STATIONS** sAL /n
code #113
(Section 9.3.5)

Stores and plots a point *n* at station *sAL* on the vertical alignment.



---

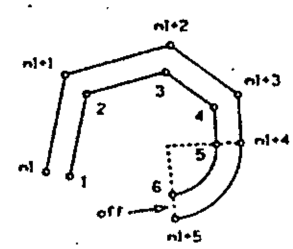
**OUTPUT FILE** mode *filename
code #65
(Section 7.1.1)

Creates formatted report file named *filename.ext* (in addition to .ATR file).
mode=0 list input commands;
mode=1 does not list input;
mode=-1 closes report file.

---

**PARALLEL FIGURE** des off n /nfg /dz
code #101
(Section 9.1.5)

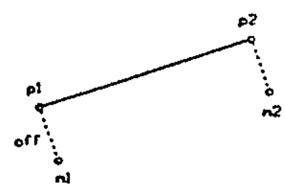
Locates a new figure parallel to *fig* at an offset distance *off*, and stores points of the new figure as *n1*, *n1+1*, etc. and the figure as *nfg*.
For a closed figure:
+off = external
-off = internal




---

**PARALLEL LINE** p1 p2 off /n1 n2 /deltaz
code # 18
(Section 8.2.6)

Locates points parallel with the line *p1 p2* at an offset distance *off*, and stores them as *n1* and *n2*.




---

**PLOT ALIGNMENT**
code # 125
(Section 12.1.7)

Places complex shapes or connected line strings for the points described in *des*. Use PLOT SHAPE if you are going to use the PLOT XSECTION COMMAND.

---

**PLOT CELL**
no code #
(Section 12.1.3)

Places cells on the points described in *des*. You have the option to place the active cell or use information in the feature table to determine the cells to be placed.

**PLOT CURVES** des /fnum or lfname
code #94
(Section 12.1.6)

Plots a curve string through the points listed in *des.* Useful for plotting lake or stream boundaries and plotting a spiral by plotting a curve through points on a spiral. At least 3 points, none of which lie in a straight line, must exist for this command to work properly.

**PLOT FILE**
no code #
(Section 12.1.10)

Plots ASCII files to a design file. This command is useful for plotting coordinate, line, and curve tables to your design file. The active weight, color, angle, text node justification, level, font, and text size all affect this command.

**PLOT LINES** des /fnum or lfname
code # 77
(Section 12.1.4)

Places lines between the points described in *des.*

**PLOT POINTS** des /fnum or lfname
code #76
(Section 12.1.1)

Plots point symbols and numbers. Flag 2 in the Feature Table affects the results of this command. If Flag 2 is set to:

E - plots elevation only
X - plots northing and easting
Z - plots northing, easting, and elevation
N - plots feature name only
e - plots elevation and feature name
x - plots northing, easting, and feature name
z - plots northing, easting, elevation, and feature name

**PLOT POV** des /sBEG /sEND /fnum or lfname
code # 115
(Section 12.1.2)

Plots all points in *des* on the vertical alignment between *sBEG* and *sEND.* You can assign zero to *sBEG* and *sEND* to indicate *from beginning to end.*

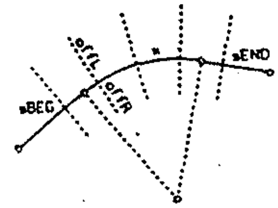
**PLOT SHAPES** des /fnum or lfname
code # 7
(Section 12.1.5)

Plots connected line strings and/or curves between points in *des.*

**PLOT VERTICAL ALIGNMENT** sBEG /sEND /fnum or lfname
code #116
(Section 12.1.8)

Plots vertical alignment between *sBEG* and *sEND.* You can assign zero to *sBEG* and *sEND* to indicate *from beginning to end.*

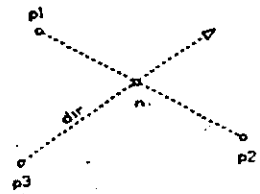
**PLOT XSECTIONS** Hfg x [illegible] /sBEG /sEND
code #84
(Section 12.1.9)




Places cross-section lines in the design file at all points of transition along an alignment described by *Hfg* and at intervals of distance *x*. If you want to place only one cross-section line, set *x* to zero and *sBEG* and *sEND* to the station where you want the cross-section line.

---

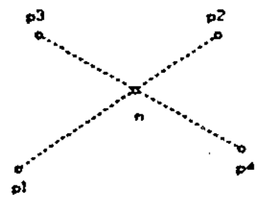
**POINTS DIRECTION INTERSECT** n p1 p2 p3 dir /off1 /off2 /va
code #92
(Section 8.4.3)




Calculates the intersection of:
(1) The line *p1 p2*
(2) The line through *p3* at direction *dir*.
**Use SET HEIGHT to use vertical angles.

---

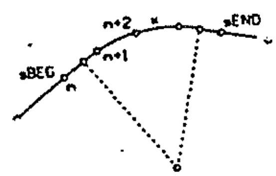
**POINTS INTERSECT** n p1 p2 p3 p4 /off1 /off2
code #19
(Section 8.4.1)




Calculates the intersection of:
(1) The line *p1 p2*
(2) The line *p3 p4*.

---

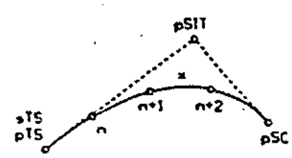
**POINTS ON ALIGNMENT** x n1 /off /sBEG /sEND
code #95
(Section 9.2.4)




Locates and lists coordinates of points *n1*, *n1+1*, etc. at each interval *x* along the alignment *des* at an optional offset *off* between *sBEG* and *sEND*.

---

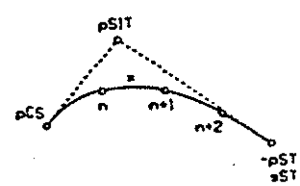
**POINTS ON SPIRAL IN** pTS sTS pSIT pSC x n /off /sBEG /sEND
code #56
(Section 9.2.11)




Locates points *n*, *n+1*, etc. along a spiral (or at an offset distance *off*) at even stationing, separated by a distance *x*.

---

**POINTS ON SPIRAL OUT** pTS sST pSIT pCS x n /off /sBEG /sEND
code #56
(Section 9.2.11)




Locates points *n*, *n+1*, etc. along a spiral (or at an offset distance *off*) at even stationing separated by a distance *x*.

---

**QUIET MODE**
no code #
(Section 7.2.2)

Toggles the prompting for new point and figure ID's on and off.

8

**RELOCATION OFFSET** desAL2 /x /sBEG /sEND
no code #
(Section 11.2.4)

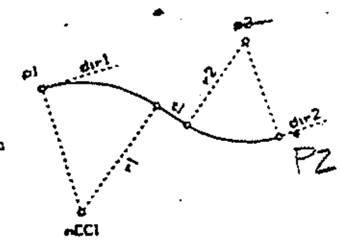
Same as the STATIONS AND OFFSETS command except that the interval *x* is defined along the baseline, not the centerline (*desAL2*). In alpha, you must key in STATIONS AND OFFSETS instead of RELOCATION OFFSET and precede the interval *x* with a negative sign (-).

**RETURN TO ICS**
no code #
Section 6.3)

Returns control back to ICS from the IGDS menu.

**REVERSE CURVE** p1 dir1 r1 p2 dir2 r2 nCC1 /tl
code # 124
(Section 9.2.3)

Finds the point of intersection of two reverse curves, the tangent point of the second curve, and the location of the center of each curve. An optional tangent of length *tl* can be inserted between the two curves.

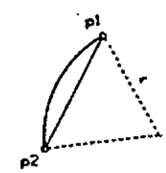



**REVIEW**
no code #
(Section 7.3.10)

Displays a tutorial that allows you to view the values of current parameters from ICS.PAR, ICS_ANNOTATE.PAR, the feature table, and current active parameters.

**SEGMENT** p1 p2 r
code #29
(Section 11.3.2)

Computes and reports the area of the circular segment (radius=*r*) bounded by the arc and chord *p1 p2*.




**SET ALIGNMENT: ACTIVATE** Hfg /pAL sAL /Vfg
code # 109
(Section 9.2.1)

Activates a horizontal alignment without altering stationing.

**SET ALIGNMENT: DEFINE STATIONING** Hfg /pAL sAL /Vfg
code # 109
(Sectiuon 9.2.1)

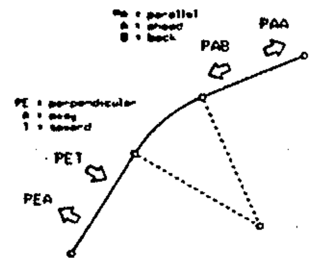
Activates a horizontal alignment and defines stationing.

**SET HEIGHT** htFS1 /hi1 /htBS1 /htFS2 /hi2 /htBS2
code #9
(Section 7.3.3)

Sets height of instrument *hi** and height of target *ht**
for LOCATE and DIRECTION INTERSECT commands.

**SET ORIENTATION**
code # 133
(Section 7.3.7)

Sets the orientation for symbol placement along an alignment. You can place symbols parallel or perpendicular to a figure, with the symbol facing toward or away from the figure.




---

**SET PARAMETER** dir /falc1 /falc2 /grid
code #63
(Section 7.3.2)

Sets direction mode for tabular output and LABEL LINES commands.
dir = 0 for azimuth
dir = 1 for bearing
Sets false coord 1 *falc1*, false coord 2 *falc2*, offsets, and grid scale factor for PLOT commands.

---

**SET PROFILE** vEX sORG sta zORG elev
code #122
(Section 9.3.1)

Establishes the parameters ICS uses to place a vertical alignment: *vEX* (ratio of vertical exaggeration), *sORG* (stationing origin coordinate), *sta* (station of this origin), *zORG* (elevation origin coordinate), *elev* (elevation of this origin). These parameters are definable in graphics with the SET PROFILE tutorial.

---

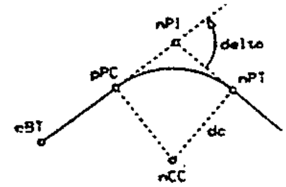
**SHOW FEATURE** 0 fname or 1 fnum
code #44
(Section 11.1.3)

Displays information about a feature in the feature table. 0 indicates access by feature name; 1 indicates access by feature number. Graphic ID of point with the desired feature can also be used.

---

**SIMPLE CURVE** nCC pBT pPC nPI nPT dc delta sign
code #33
(Section 9.2.9)

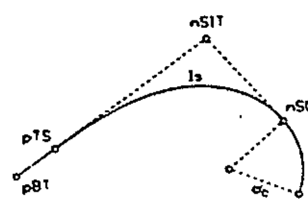
Given known points *pBT* and *pPC* and the *dc*, *delta*, and *sign*, locates *nCC*, *nPI* and *nPT*.




---

**SIMPLE SPIRAL IN** cid pBT pTS nSIT nSC ls sign
code #53
(Section 9.2.10)

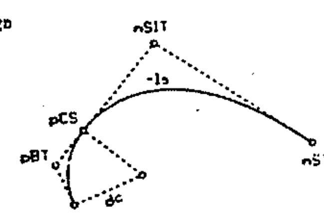
Given *pBT*, *pTS*, *ls*, *dc*, and sign, locates *nSIT* and *nSC*.



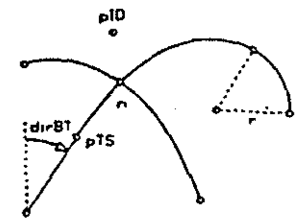

---

**SIMPLE SPIRAL OUT** cid pBT nST nSIT pCS -ls dc sign
code #53
(Section 9.2.10)

Given *pBT*, *pCS*, *ls*, *dc* and sign, locates *nST* and *nSIT*.



SPIRAL [illegible] n pTS dirBT ls r
code #59
(Section 8.4.12)

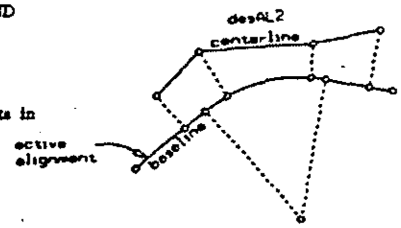



Calculates the intersection of:
(1) Previously defined spiral by SIMPLE SPIRAL or SPIRAL LENGTH.
(2) The spiral defined by its *pTS, dirBT, ls,* and *r.*

STATION EQUATION Bksta Ahsta /Hfg
code # 128
(Section 9.2.2)

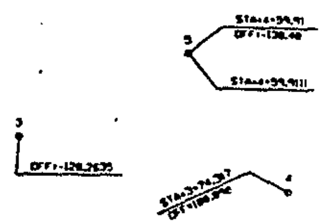
Defines the station equation for the active horizontal alignment, beginning at a specified point on the alignment, where the previous station equation ends (*Bksta*) and the new station equation begins (*Ahsta*). If *Ahsta<Bksta(f=1), then Ahsta=Ahsta+*incr. The increment *Incr* is defined in ICS.PAR.

STATIONS & OFFSETS desAL2 /x /sBEG /sEND
code #91
(Section 11.2.3)



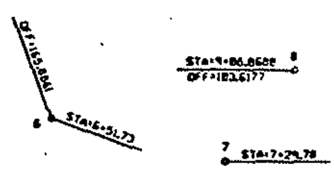

Computes and lists the stations and offsets of points in *desAL2* from the active alignment.

STATIONS ANGLED
no code #
(Section 13.3.3)



Places the stations of specified points along angled leaders. If you want to place identical leaders on a group of points, use a fence. Affected by ACTIVE ANGLE.

STATIONS STRAIGHT
no code #
(Section 13.3.4)



Places stations of specified points along a straight leaders. If you want to place identical leaders on a group of points, use a fence.

STATIONS/OFFSETS
no code #
(Section 13.1.2)

Sets the type of elevation annotation that ICS will use when labeling stations/offsets (station and offset, station only, or offset only).

STORE n c1 c2 z p
code #5
(Section 8.1.1)

Stores point *n* with coordinates *c1, c2,* and *z.* If optional *p* is specified, *n* takes its values. If optional *-cno* is specified as -1, -2, or -3 the coordinate *c1, c2, or z* is modified, respectively.

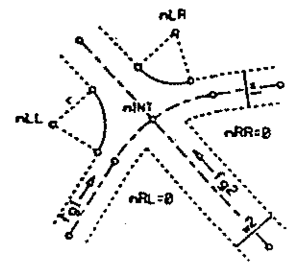
**STORE FIGURE** nfg des /pAL sAL
code # 70
(Section 9.1.1)

Defines *nfg* in the database as the list of point numbers used in *des*.

---

**STREETS INTERSECT** r fg1 w1 fg2 w2 pID nINT nLL nLR nRL nRR
code # 102
(Section 9.2.7)

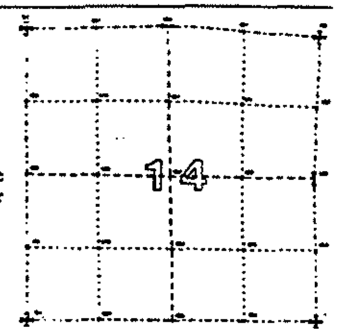
Locates the intersection *nINT* and street return points (*CC*, *nPC*, and *nPT*) of two streets, a street and a property line, or a street and a right-of-way. The first street is described by centerline *fg1* and width *w1*; the second street is described by centerline *fg2* and width *w2*. A 0 input for any of *nINT*, *nLL*, *nLR*, *nRL*, and *nRR* implies that the point/group of points is not be be computed. A 0 input for *r* indicates that only PI's should be computed.




---

**SUBDIVIDE SECTION** nSec des1 /n /des2
code # 131
(Section 8.3.4)

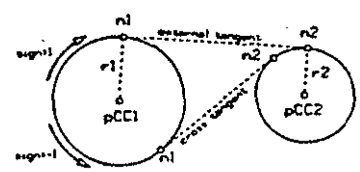
Subdivides a section of land into sixteen parcels according to requirements of the Public Land Survey System (PLSS). The section of land is described by the section number *nSec* and a description *des1* of the four section corner points. Use *des2* to include 1 or more quarter corners in the subdivision.




---

**TANGENT** n1 pCC1 r1 n2 pCC2 r2 /sign /ext
code #32
(Section 8.2.9)

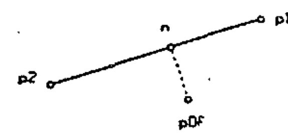
Locates the point of tangency n1 on a circle with center pCC1 and radius r1, and the point of tangency n2 on a circle with center pCC2 and radius r2. You can use one of two methods. Set *ext*=1 for an external tangent and *ext*=-1 for cross tangent. (See the *ICS User's Guide* for details.)




---

**TANGENT OFFSET** n des p1 p2
code #52
(Section 8.2.5)

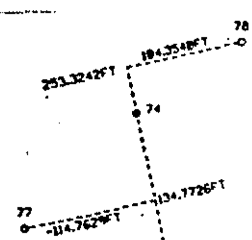
Locates *n* on the line *p1 p2* as the perpendicular offset from *des*.



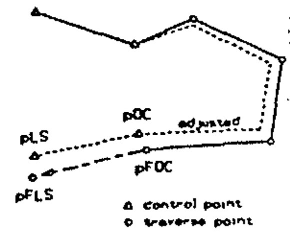

---

**TANGENTS & OFFSETS**
no code #
(Section 13.4.2)

Annotates the baseline between the backsight point *pBS* and the occupied point *pOC* with the distances and offsets. The command places the annotation perpendicular to the baseline.



**TRAVERSE ADJUSTMENT** des pFOC Dtol /pLS pFLS Atol /Tdis Mdis
COMPASS
CRANDALL
TRANSIT
code #6
code #7
code #8
(Section 10.1.1)




Transform coordinates in *des* (according to command's adjustment rule) to close a traverse.

NOTES:

1) *pFOC* is the field location of the last p# in *des*

2) Command will abort if the computed angular closure is greater than *Atol*, or if the computed positional error is greater that *Dtol*.

---

**TRAVERSE ANGLES** des
code #89
(Section 11.4.4)

Computes and reports the lengths and the interior angles of lines in the traverse defined in *des*.

**TRAVERSE DEFLECTION** des
code #90
(Section 11.4.5)

Computes and reports the lengths and angular deflections of lines in the traverse defined in *des*.

---

**TRAVERSE/SIDESHOT**
no code #
(Section 7.3.8)

Toggles between traverse mode and sideshot mode. While in traverse mode, ICS assumes that you are inputting a traverse and does not prompt for the occupied and backsight points. With the traverse mode OFF, ICS assumes that you are inputting sideshots (radial survey data).

---

**VERTICAL ADJUST** des pFOC tol /prop
code #45
(Section 10.1.2)

For adjustment of level loops. Misclosure equals the difference between last point in *des* and *pFOC*. If misclosure is greater than *tol*, this command does nothing. If misclosure is less than *tol*, the misclosure is distributed evenly unless *prop* = 1 (to distribute proportionally to lengths of lines). After adjustment, *pFOC* is deleted from the database.

---

**VERTICAL END** nPVI sAL elev Vfg /n
code #111
(Section 9.3.2)

Enters the final *PVI* and figure ID *Vfg* for a vertical alignment.

---

**VERTICAL START** nPVI sAL elev /Vcl1 /Vcl2
code #110
(Section 9.3.2)

First command used to create vertical alignment. Enters all points of vertical inflection (*PVI*) except for last point.

INTERGRAPH

December 31, 1987

# Interactive Coordinate Geometry Subsystem (ICS) User's Guide

DMAP135
Edition 2

**Intergraph Corporation**

One Madison Industrial Park
Huntsville, Alabama 35807-4201
Tel. (205) 772-2000 TLX 4992084
FAX (205) 772-2461

SLA*741 8.8, rev. 4

## Warranties and Liabilities

All warranties given by Intergraph Corporation about equipment or software are set forth in your purchase contract.

The information and the software described in this document are subject to change without notice and should not be construed as commitments by Intergraph Corporation. Every effort is made to insure technical accuracy; however, Intergraph Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Intergraph or its affiliated companies.

## Copyright



## Trademarks

Intergraph is a registered trademark of Intergraph Corporation. DMRS, IGDS, ICS, ETI, GNA, SAC, and LAD are trademarks of Intergraph Corporation.

Intergraph Corporation
One Madison Industrial Park
Huntsville, AL 35807-4201

# Document Delivery Notice (DDN)

| | |
|---|---|
| *Software Part No.* | SLA*741 ICS |
| *Document No.* | DMAP135<br>Edition 2 |
| *Title* | Interactive Coordinate Geometry Subsystem (ICS) User's Guide |
| *Software Release Version* | 8.8, rev. 4 |
| *Date* | December 31, 1987 |
| *Replaces* | DIXD7950 |
| *Comments* | The complete document consists of the following:<br><br>o DMAP13520 (Document)<br><br>o DMAP1351S (Spine)<br><br>o DMAP1351T (Tabs)<br><br>o DIXGDMRS5 (3" Full Size Generic Binder)<br><br>Also availabe is DMAP170 ICS Quick Reference Guide.<br><br>Please take a moment to complete and return the preaddressed, postage-paid form at the front of this document. Intergraph encourages customers to submit any comments, suggestions, or criticisms they might have about this document. |

## What do you think about this Intergraph Mapping/Energy Document?

**DMAP135 Edition 2**
**Interactive Coordinate Geometry Subsystem (ICS) User's Guide**

After you have used this document, please help us to improve it by responding to the questions below and circling the appropriate answer. Include additional pages of comments if you wish, and then fold and mail this form to Intergraph. Postage is prepaid.

1. Was this document easy to read and understand? YES NO
   If no, please specify why.
   ______________________________
   ______________________________
   ______________________________

2. Are additional or improved illustrations needed? YES NO
   If yes, please specify where.
   ______________________________
   ______________________________

3. Were you able to locate information quickly? YES NO
   If no, please specify why.
   ______________________________
   ______________________________
   ______________________________

4. Did you encounter any errors or inconsistencies? YES NO
   If yes, please specify and note page numbers below.
   ______________________________
   ______________________________
   ______________________________

5. Did this document adequately meet your needs? YES NO
   If no, please specify how it could be improved.
   ______________________________
   ______________________________
   ______________________________

6. Have you used Intergraph documentation before? YES NO

7. Have you used a previous edition of this document? YES NO

8. How would you rate this document overall? POOR FAIR GOOD VERY GOOD EXCELLENT

*Optional:*
Name ______________________ Date ____________
Company/Address ______________________
City ______________ State ________ Zip Code ________

Position (check all that apply) :
___Operator ___Programmer ___System Manager ___Application Manager ___Other:__________

Thank you for your time and effort.

INTERGRAPH

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 9079 HUNTSVILLE, ALABAMA

Postage will be paid by the addressee

**Intergraph Corporation**

Attention: Manager of Mapping / Energy
Technical Documentation
IW17A2
One Madison Industrial Park
Huntsville, Alabama 35807-9985

## Document Conventions

1. Command names are in uppercase letters.

2. Note: is used to emphasize important information or potential errors.

3. Lowercase letters appearing within a prompt indicate a name or value that varies. A small x represents an alphabetic character; a small n represents a numeric character.

4. System prompts and messages (screen text generated by the software) are shown flush left as they appear on the screen.

5. System responses (explanations of what the software is doing) are in *italic* type.

6. Various symbols appear in the text. They represent certain keys or cursor buttons as outlined below:

| | |
|---|---|
| <C> | Command button |
| <CTRL> | Control key |
| <CTRL Z> | Unless otherwise noted in the text, you must hold down the <CTRL> key while pressing the second key. |
| <D> | Data button |
| <R> | Reset/Reject button |
| <RETURN> | Return key |

7. *Identify* means to input a value by keying in the value or by placing data point(s).

8. *Key in* means to key in input at the keyboard and press <RETURN>.

## Reference Bar Legend

*Reference bars are provided on this page and at the beginning of each section to aid in locating a particular section. To use the bars, find the section listing on this page. Look at the edge of the document, find the corresponding bar, and open the document to that page.*

# Contents

## Appendices

# Figures

# Tables

# 1. INTRODUCTION TO ICS

As an engineering professional, you are probably familiar with the purpose and capabilities of coordinate geometry (COGO) programs. In the Interactive Coordinate Geometry Subsystem (ICS™), Intergraph® combined the specialized capabilities of traditional coordinate geometry routines, the graphical capabilities of IGDS™ (Interactive Graphics Design Software), and the database capabilities of DMRS™ (Database Management and Retrieval System). The result is a software package that combines coordinate geometry routines and computer graphics to solve surveying and engineering problems in a more flexible and integrated environment than traditional programs. ICS creates a database that has the potential of serving as a basis for other programs or users who require intelligent databases.

ICS uses a DMRS database to store information. Points are stored in the database with a specific point ID, coordinates, and an associated feature. (Features contain symbology information such as level, color, and symbol font character.) Figures are associations of points such as a subdivision lot or alignment. Figures are stored in another section of the database with a specific figure ID, the group of points, and stationing information (optional). The database can be extended to incorporate attributes specific to the needs of the client.

ICS operates in both the interactive and batch modes, and in the graphics and alpha environments. You have the option to work in any one of these environments, and you can switch back and forth between them.

Using ICS, the surveying and civil engineer can solve complex problems in such fields as cadastral surveying, bridge and roadway design, and right-of-way surveys. The purpose of this manual is to show you how to use ICS tools to solve such problems.

Typically, you perform these general steps for using ICS. First, you may collect field data and then load it into ICS. You can load the data automatically using Intergraph's Electronic Theodolite Interface (ETI™) software, or you can input the data manually. Then you can define groupings of points as figures and locate new points based on points you have. You might locate new points by intersecting figures you have defined, by offsetting from alignments, or by dividing figures. You might then do some traverse adjustments, vertical adjustments, area adjustments, and Helmert transformations.

## 1.1 Changes to ICS Since the Last Release

Since the last release of ICS, these changes have been made:

- the ICS matrix menu has been reorganized
- some ICS commands have been modified
- new commands have been added
- an additional parameter file has been added
- enhancements have been made to command line inputs
- a program has been added to provide patterning for elements
- a user command to define linkage generation mode

If you are familiar with ICS already, you may want to read only the portions of this manual that address the changes. Table 1-1 lists the modified commands, and Table 1-2 lists the added commands. Each of these tables gives section references and brief descriptions. The new parameter file is ICS_ANNOTATE.PAR, and it is described in Section 3.7. The enhancements to the command line are described in Section 2.3. The program that allows you to pattern elements is entitled FCPELT (Feature Code Patterning by Element List), and it is described in Appendix C. The user command added is LINKAGE.UCM, and it is also described in Appendix C.

*Table 1-1. Modified ICS Commands Since the Last Release*

| *Command* | *Section* | *Description* |
|---|---|---|
| LOCATE ANGLE | 8.2 | added optional elevation/rod input |
| LOCATE DEFLECTION | 8.2 | " " |
| LOCATE DIRECTION | 8.2 | " " |
| LOCATE LINE | 8.2 | " " |
| TANGENT OFFSET | 8.2.5 | added option to input multiple perpendicular offset points |
| PARALLEL LINE | 8.2.6 | added z translation |
| PARALLEL FIGURE | 9.1.5 | added z translation |
| SIMPLE CURVE | 9.2.9 | added optional radius input |
| SIMPLE SPIRAL | 9.2.10 | " " |
| COMPOUND SPIRAL | 9.2.12 | " " |
| FIT ALIGNMENT | 9.2.13 | " " |
| SET PROFILE | 9.3.1 | added tutorial in graphics |
| TRANSFORM COORDINATES | 10.2.2 | added tutorial in graphics |
| STATIONS AND OFFSETS | 11.2.3 | added the option to define interval along the baseline instead of centerline |
| DESCRIBE VERTICAL ALIGNMENT | 12.1.8 | added stopping sight distance |
| LABEL ALIGNMENT | 13.2.2 | added variables in ICS_ANNOTATE.PAR that affect how labels are placed |
| LABEL VERTICAL ALIGNMENT | 13.2.3 | added % suffix to grade readout |
| LABEL LINES | 13.2.5 | modified so that LINE and CURVE TABLE locks and ICS_ANNOTATE.PAR affect how labels are placed |
| PLOT TEXT | 14.5 | added flag to indicate view dependent or independent for placing the text |

*Table 1-2. New ICS Commands Since the Last Release*

| *Command* | *Section* | *Description* |
|---|---|---|
| FENCE ON/OFF | 7.2.1 | toggles fence mode on and off |
| SET ORIENTATION | 7.3.7 | sets orientation for symbol placement along an alignment |
| FENCE CONTENTS | 7.3.9 | toggles fence content between points and figures |
| REVIEW | 7.3.10 | displays tutorial that allows you to view the current values of parameters, modes, and settings from ICS.PAR, ICS_ANNOTATE.PAR, the feature table, and the ENVIRONMENT menu block |
| INPUT MODES | 7.3.12 | defines active units for curve mode, height mode, and unit mode |
| COPY POINT | 8.1.3 | copies the coordinates of a known point to another point ID |
| MOVE POINT | 8.1.4 | moves existing points to new locations and updates the database |
| SUBDIVIDE SECTION | 8.3.4 | subdivides a section of land into sixteen parcels |
| CREATE PROFILE | 9.3.3 | plots elevations of centerline control points in the design file |
| RELOCATION OFFSETS | 11.2.4 | same as STATIONS AND OFFSETS except that the interval is defined along the baseline |
| COORDINATE TABLE | 12.1.1 | writes coordinates of points to an ASCII file |
| CURVE TABLE | 12.1.2 | writes curve data to an ASCII file |
| LINE TABLE | 12.1.2 | writes line data to an ASCII file |
| PLOT CELL | 12.1.3 | places cells |
| PLOT FILE | 12.1.10 | plots ASCII files |
| STATIONS/OFFSETS | 13.1.2 | sets the type of elevation annotation for labeling stations/offsets |
| RENUMBER POINTS | 14.6 | renumbers points in the database |

## 1.2 Use and Organization of this Guide

This user's guide is written for engineers and design professionals. To fully use ICS and this manual, you should be familiar with the concepts of coordinate geometry. You should know how to combine simple geometric commands to solve complex practical problems. This guide assumes you are familiar with standand engineering terms, although a glossary of some terms is provided.

Additionally, to use ICS you should have a working knowledge of basic operations of the Intergraph system, IGDS, and DMRS. This guide does not describe these operations. Refer to Section 1.3 for a list of references.

This user's guide is divided into two main parts. The first five chapters provide you with general information you need to run ICS. You should read these sections thoroughly before running ICS. Sections 6 through 14 describe how to use each command. The commands are grouped together according to logical function, and each section covers one menu block of commands. These sections are intended to be used as a reference guide to ICS commands.

| *Read* | *For* |
|---|---|
| Section 1 | an overview of ICS |
| Section 2 | basic information about ICS: operating characteristics in the alpha and graphics modes and structure (syntax) of the commands |
| Section 3 | a description of ICS directories and files, including files that are delivered and files you create when running ICS |
| Section 4 | a description of ICS databases and how to create them |
| Section 5 | steps to follow when running ICS |
| Section 6 | a description of the CONTROL commands |
| Section 7 | a description of the ENVIRONMENT commands |
| Section 8 | a description of the POINTS/COORDINATES commands |
| Section 9 | a description of the FIGURES/ALIGNMENTS commands |
| Section 10 | a description of the ADJUST/TRANSFORM commands |
| Section 11 | a description of the REPORTS commands |
| Section 12 | a description of the PLOT commands |
| Section 13 | a description of the ANNOTATE commands |
| Section 14 | a description of alpha commands that are not available on the graphics menu |
| Appendix A | a description of the feature editor EFEED, and instructions on how to use it |
| Appendix B | instructions on how to modify menus |
| Appendix C | a description of ICS utilities including I2SDB, ICS Customer Support Library, the digital terrain modeling interface (ICSTODTM), the database upgrade utility, the Linkage User Command, and Feature Code Patterning by Element List (FCPELT) |
| Appendix D | instructions on how to create symbols |
| Appendix E | plots of example projects |
| Appendix F | a list of ICS error messages, each with reason and recovery |

## 1.3 List of References

Refer to the following documents for further information:

- *Interactive Graphics Design Software (IGDS) User's Guide*
- *DMRS Data Definition Language (DDL) User's Guide*
- *IGDS Font Librarian User's Guide*
- *Electronic Theodolite Interface (ETI) User's Guide*
- *Font Utilities User's Guide*

# 2. GETTING STARTED: BASICS ABOUT ICS

You can operate ICS in either the alpha or graphics mode. The commands for these environments are basically the same, but you have to input information differently, and some commands that are available in one mode are not available in the other. The following sections describe the structure of the commands, the differences between the two working environments, and some basic data entry fields.

## 2.1 The Structure of ICS Commands

You input information to ICS in the form of commands that you either select on the menu or key in from an alphanumeric or graphics terminal. Within the command line, you specify the command you want ICS to execute, the input(s) ICS needs to execute the command, and the output(s) you want ICS to compute. An example of a command line is:

```
STORE n c1 c2 /z /p
```

where STORE is the command name, and the other entries are input or output data fields (or parameters). Data fields that follow a forward slash (/) such as *z* and *p* are optional parameters; you *do not* have to input values for these fields for the command to run. Commands that are not preceded by a slash are required entries; you must specify these entries, or ICS will not execute the command. The slashes are displayed on the screen when you select a command to let you know those fields are optional, but you do not key-in the slashes when you input information. Required data fields always precede optional data fields in a command line.

Command lines can have more than one data field after a slash such as:

```
COPY POINT des /n /dx dy dz
```

In this case, *des* is the only required data entry field. Notice that the *dy* and *dz* fields are not preceded by a forward slash, but they follow a field, *dx*, which is preceded by a forward slash. This means that all three fields—*dx*, *dy*, and *dz*—are optional data fields, *and they must all be specified if one is specified*. This rule holds true for all groups of data fields that follow a forward slash.

ICS reads the information you provide for the data fields consecutively along the command line. Therefore, you must input information in the order specified by the command line. If you want to skip an optional field to get to the next one, you can do that with a zero (0) (exceptions are listed with specific commands). For instance, for the COPY POINT command shown above, you might want to use the optional fields *dx*, *dy*, and *dz* without specifying *n*. You can do this by assigning a value of zero (0) to *n*:

```
COPY POINT 4 0 90.0 90.0 90.0
```

## 2.2 ICS Operating Characteristics: Alpha and Graphics Modes

You can operate ICS in either alpha or graphics modes. Alpha mode is necessary when you do not have a graphics terminal available, and it is useful when you do not need to see graphics displayed. If you need to see graphics displayed as you work, you should run ICS interactively in graphics.

You can submit batch jobs in both alpha and graphics modes. Running ICS in batch mode is useful for running large files of ICS commands. To run ICS in batch mode, you create a text file that serves as input to the ICS run. The file can contain many commands with input data. There is usually no interaction between you and ICS from the beginning to the end of the input file. If ICS detects an error while running in batch mode, an error message is either displayed to the screen and/or written to the report file. See Section 7.1.2 for details on how to run a batch job.

You can also work interactively in both alpha and graphic modes. In both modes you can execute one command at a time and see the results. In graphics mode you see the graphic results, and in alpha and graphics you see the prompts and messages ICS sends to the screen.

### 2.2.1 Working in the Alpha Mode

If you are at an alphanumeric terminal, you run ICS by keying in commands. There are two ways to key in the command name:

- Key in the entire command name (in full or abbreviated)

  Note: The abbreviation must be unique. When a command name is more than one word, at least one space or exactly one slash must appear between the abbreviation of each name in the command. Example: the abbreviation for DIRECTION INTERSECT can be DIR/INT or DIR INT, but not DIRINT.

- Key in the code number

  Note: See the *ICS Quick Reference Guide* for a list of command names and corresponding code numbers.

  **Numeric codes are accepted by ICS only if the START OF JOB command is a numeric code.**

You key in the command name, its abbreviation, or its numeric code in the first record in the series. You can enter a command name or its abbreviation in any column of an input record. The remaining records in the series can begin with data.

Numeric codes must begin in column one. The first column of the remaining records in the series must be blank. If column one begins with an alphabetic character, ICS interprets this character as the first character of a command name.

When you enter a command name, it remains active until you key in another command name. Therefore, you can input the command name once for several runs of the command. Most of the commands available in graphics are also available in alpha mode (see Section 2.2.2 for an overview of the commands available). Additionally, there are some commands available in alpha mode only. These commands are described in Section 14.

Entering Data

When you enter a command name, a data field description appears, which prompts you for point ID's, distances, and other types of information that the command requires. As described in Section 2.1, a forward slash (/) appears in some data field prompts to indicate optional entries; *do not key in the slash.* You key in the required data and any optional entries you want to include, leaving at least one space between data entries. No space is required between the command name and the first data entry. See Section 2.3 for more detailed information on entering data.

Note: Do not use tabs when keying in data.

---

### 2.2.2 Working in the Graphics Mode

If you are running ICS at a graphics workstation, you can run ICS in the alpha or graphics environment. If you are running ICS in the graphics environment, you can select commands graphically, or you can key in alpha commands. However, to key in alpha commands you must key in a backslash (\) prior to keying in the first character of the command. If you are at a graphics workstation and are in the alpha environment, you cannot make selections graphically.

Figure 2-1 shows the ICS matrix menu. The commands are grouped logically according to function:

- CONTROL commands (Section 6)
  Allow you to get in and out of ICS.

- ENVIRONMENT commands (Section 7)
  Allow you to define input and output files, choose default settings, and toggle settings on and off.

- POINTS/COORDINATES commands (Section 8)
  Allow you to store and delete points and coordinates.

- FIGURES/ALIGNMENTS commands (Section 9)
  Allow you to store and delete figures and alignments.

- ADJUST/TRANSFORM commands (Section 10)
  Allow you to alter the values of point coordinates.

- REPORTS commands (Section 11)
  Allow you to display information stored in the database to the screen and to print this information to an output file.

- PLOT commands (Section 12)
  Allow you to place graphic elements in IGDS design files and create coordinate, line, and curve tables.

- ANNOTATE commands (Section 13)
  Allow you to place text in a design file to provide information on points, lines, areas, and alignments.

In addition to these menu blocks, there is a DELIMITERS box on the matrix menu. This box has characters that you need to use frequently in ICS. Pressing <D> on one of these characters is equivalent to keying in the character from the keyboard. Refer to Section 2.3.2 for more information on ICS delimiters.

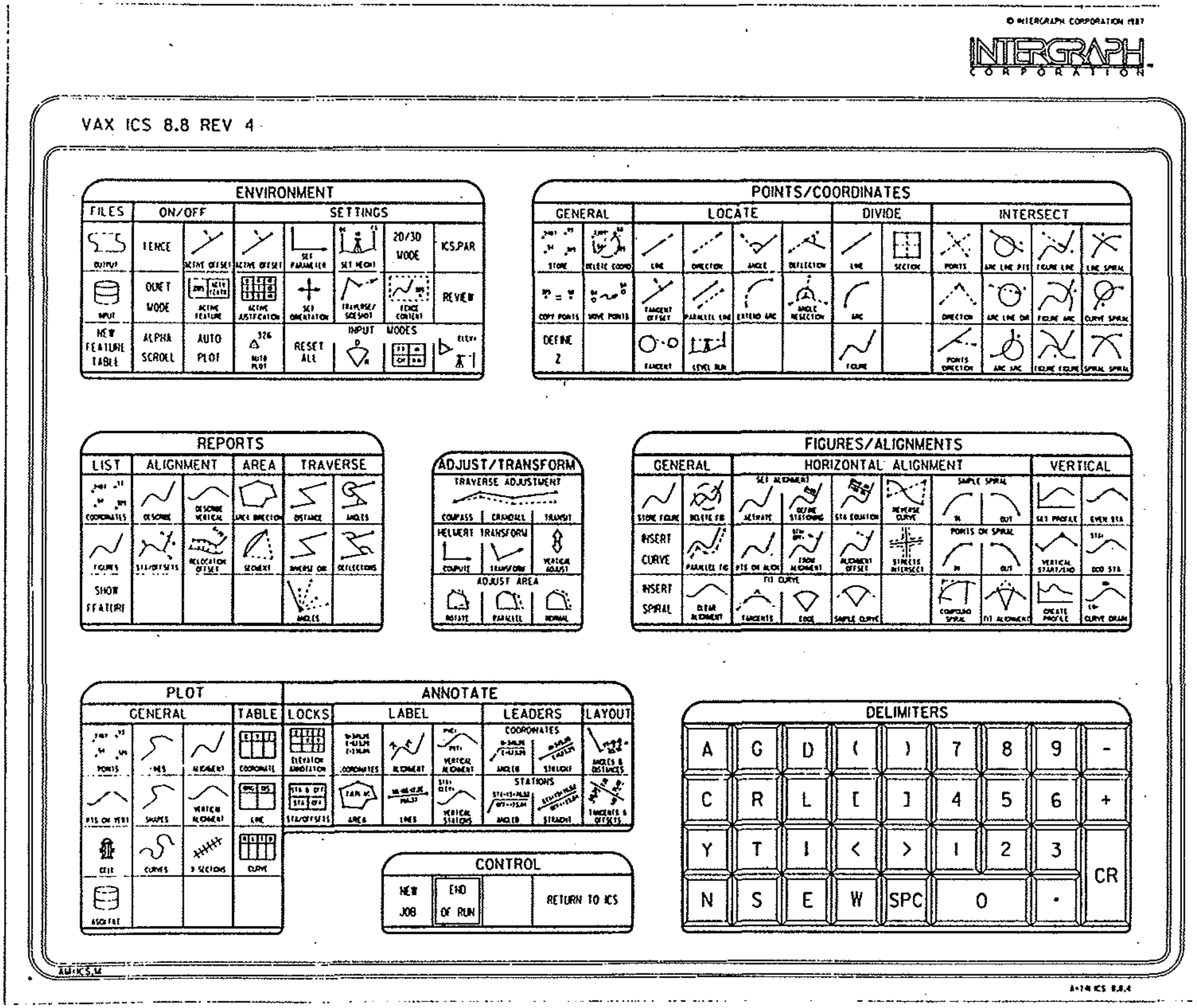



*Figure 2-1. The ICS Matrix Menu*

## Entering Data

At a graphics terminal you have the option to enter data using key-ins, data points via the cursor, or a combination of the two. However, ICS in the graphics environment is primarily designed to accept and check input by data point. For instance, distances can be defined by giving an initial point and terminal point. Likewise, angles can be defined by graphically indicating three points or a point and a direction. These cursor inputs are computed into distances and/or angles and displayed in the status field (the left side of the left field) for acceptance or rejection.

In the graphics environment, key-ins are allowed. For example, when prompted for the intial point of a distance, you can key in the distance, overriding the graphic input sequence.

When keying in data, if you key in more than one data field item and press <RETURN>, ICS attempts to execute the command. If you have not input all the required data fields, ICS suspends execution and prompts you for the additional field(s).

In order to enter data for the optional fields, you must either key in the whole command line (including the optional parameters you want), or key in the last required data item with the optional fields. If you use data points for all the required data fields, ICS assumes you do not want to enter data in the optional fields. See Section 2.3 for more detailed information on entering data.

<u>Running ICS on Dual Screen Workstations (InterAct's and InterView's)</u>

While running ICS in graphics at a dual screen graphics terminal, you can use the ICS matrix menu as a paper or screen menu. You can select IGDS commands simply by moving the hand-held cursor over to the IGDS menu. To return to ICS from IGDS, you select RETURN on the ICS matrix menu. The cursor buttons are described below.

Command Button: To select commands, place the cross-hairs of the hand-held cursor over the desired command on the matrix menu, and press <C> (or <D> for screen menu).

Data Button: To identify points, place the cross-hairs of the screen cursor in the desired location and press <D>. Press <D> a second time (in rapid succession) to accept the point. If you are identifying a series of points, you can accept a point when you press <D> to identify the next point in the series. When you have identified the last point in the series, remember to press <D> one extra time to accept it.

Reset Button: To move back a single step, press <R>. In some commands, reset moves you back more than one step if that is more logical for data input.

If the workstation is Unix-based, ICS can be accessed through a control strip (see Section 5). Due to display truncation, use of ICS screen menus on Unix-based workstations is not recommended.

<u>Running ICS on Single Screen Workstations (InterPro's)</u>

Running ICS in graphics at an InterPro differs from running ICS on a dual screen workstation in two ways. Paper menus cannot be attached without a digitizing tablet, and the hand-held input device is a mouse. If you want to use IGDS while running ICS, select the IGDS option from the menubar. The mouse buttons are described below.

Left Button: <C> To select the IGDS hierarchical menu, place the arrow on the menubar option and tap the left button.

Middle Button: <D> or <r> To make ICS menu selections and identify points, place the cross-hairs of the screen cursor in the desired location and tap the middle button. If identifying points, tap the middle button a second time to accept the point. If you are identifying a series of points, you can accept a point when you tap the middle button to identify the next point in the series. When you have identified the last point in the series, remember to tap the middle button one extra time to accept it.

Right Button: <R> To move back a single step, press the right button. In some commands, this action moves you back more than one step in the process, if that is more logical for data input.

## 2.3 How To Enter Data

Entering data in alpha and graphic modes is basically the same. The main difference is that in graphics you can identify graphic elements and certain input values with data points via the cursor, and in alpha mode you cannot. You can enter data in several different ways:

- keying in the information directly
- identifying graphic elements, default values, and certain angle/distance input with data points (graphics mode only)
- using delimiters
- using special operators
- using input modifiers.

Delimiters are certain alphabetic characters that you can use to instruct ICS to calculate a value. For instance, if a data field requires a distance value, you can key in the distance, or you can use a delimiter to instruct ICS to calculate the distance between two points. Refer to Section 2.3.2 for more detailed information on delimiters.

Special operators are symbol characters, including the asterisk (*), period (.), greater-than symbol (>), and less-than symbol (<). You can use these characters to place comments in the report file or to instruct ICS to refer to the previous command line for data values. Refer to Section 2.3.3 for more detailed information on special characters.

Input modifiers are specific prefixes and suffixes you can use when entering values. By default, ICS commands expect a certain unit for parameters; however, some of these commands allow you to input another unit if you prefix/suffix the value. For example, INTERSECTION commands that require a degree of curvature to define a curve accept a radius if you prefix the radius value with an *R*. See Section 2.3.4 for more detailed information on input modifiers.

### 2.3.1 General Data Fields

General data fields that you will encounter frequently in ICS command lines include: points, figures, descriptions, coordinates, directions, distances, and angles. Each of these is described in detail below.

Points: *p* and *n*

When *p* or *n* appear in a data field, you are being prompted to identify a point. A lowercase *p* indicates a predefined point with a previously assigned ID; in other words, the point you identify must be a point that is already defined in the database. A lowercase *n* indicates an undefined point with no previously assigned point ID. ICS will calculate the coordinates of *n* and define the point ID for it when the command executes.

You can identify point *p* by keying in the ID, or by pressing <D> on the point symbol in graphics (after you graphically select a point, you must accept it by pressing <D> again). You can identify point *n* by keying in the ID, by skipping the field with a zero (0), or by skipping the field in graphics by pressing <D>. When you skip the *n* field, ICS finds the next available point ID and assigns that to the new point.

By default, if you key in a point ID that is already assigned, ICS issues a warning. Then you can choose to redefine the point or not. If you decide not to redefine the point, ICS prompts you again for *n*. If you do not want to be warned, you can change a parameter in ICS.PAR (see Section 3.6). If you want ICS to assign the point ID for you, key in zero (0) or press <D> (in graphics).

Figures: *fg* and *nfg*

When *fg* or *nfg* appears in the data field, you are being prompted to identify a figure. A figure is an association of points; it can include curve and spiral information. A *fg* indicates a predefined figure with a previously assigned ID; in other words, the figure you identify must be a figure that is already defined in the database. A *nfg* indicates a figure that has not been previously defined. When the command executes, ICS will store the figure ID (with the association of points) in the database.

You can identify figure *fg* by keying in the ID, or by pressing <D> on the graphic element in graphics (after you graphically select a figure, you must accept it by pressing <D> again). You can identify figure *nfg* by keying in the ID, by skipping the field with a zero (0), or by skipping the field in graphics by pressing <D>. When you skip the *nfg* field, ICS finds the next available figure ID and assigns that to the new figure.

By default, if you key in a figure ID that is already assigned, ICS issues a warning. Then you can choose to redefine the figure or not. If you decide not to redefine the figure, ICS prompts you again for *nfg*. If you do not want to be warned, you can change a parameter in ICS.PAR (see Section 3.6). If you want ICS to assign the figure ID for you, key in zero (0) or press <D> (in graphics).

Descriptions: *des*

When *des* appears in the data field, you are being prompted to identify a group of related points or figure(s). You can transpose the figure you identify in the figure description.

You identify the list of points or figure(s) by keying in the ID's, or by pressing <D> on the point symbols or figure elements in graphics (after you graphically selet a point or figure, you must accept it by pressing <D> again). If you key in a list of point ID's, you must enclose that list in parentheses. If you key in a list of figure ID's, you must enclose that list in brackets. If you key in a single figure ID, you do not need to use brackets. Additionally, you can indicate a range of points or figures using a hyphen.

You can also indicate circular curves, vertical curves, and spirals in a description. To indicate that two points are connected by a curve rather than a straight line, you separate the two points in the data field by three additional items of information: the letter C which indicates a curve, the point number of the center of curvature, and the letter L or R to indicate left or right. For example, (63 C4R 2) is a description of curve to the right from point 63 to point 2. The curve is centered at point 4.

To indicate that two points are connected by a vertical curve (parabola), input the first point, the center of vertical curvature prefixed with the letter V, and the second point. For example, (51 V47 52) is a description of a vertical curve from 51 to 52 with a PVI at point 47.

To indicate that two points are connected by a spiral, input the point ID of the point of intersection of the tangents with an S before it. The input for storing a spiral in a figure is either (pTS pSIT pSC) or (pCS pSIT pST). For example, (5 S6 4 C12L 15) describes a spiral from point 5 to point 4 with the intersection of the tangents at point 6. For a compound spiral, replace *pTS* and *pST* with *pCS* and *pSC*, respectively.

To transpose a figure, you can identify the points in reverse order or append a T to the description or figure ID. See Section 2.3.2 for more details on the T delimiter. Table 2-1 shows sample key-ins for *des*.

*Table 2-1. Sample Key-ins for Descriptions*

| *Key-in* | *Explanation* |
|---|---|
| (1 2 3 4) | Point numbers 1, 2, 3, and 4 |
| (1 - 4) | Point numbers 1, 2, 3, and 4 |
| (1 - 4 1) | A closed loop; the first point is the same as the last point |
| (904 C12R 61) | Curve from point 904 to point 61 centered at point 12, to the right |
| (500 - 264 480 -486) | Points 500 to 264 and points 480 to 486; a range of numbers may be in increasing or decreasing order. |
| [1-10] | Figures 1 through 10 |
| 364 | Figure 364 |
| (1-10 S11 12 C13L 14) | Spiral from point 10 to point 12 with curve; TS is at 10, SC is at 12 |
| (1 2)T | This expression means the same as (2 1). |
| (175 - C178R 179)T | This expression means the same as (179 C178L - 175). |
| 364T | The transposition of stored figure 364. |

Coordinates: *c1, c2* and *z*

When *c1, c2,* and *z* appear in the data field, you are being prompted to identify the coordinates of a point. A point can be two or three dimensional; the *z* coordinate is always optional (see Section 7.3.4 for information on 2D/3D input modes).

You can identify the coordinates by keying in the coordinates, or by pressing <D> in graphics. By default, if you key in the coordinates, you must key in Northing followed by Easting followed by the elevation. If you want to change the sequence and direction of coordinates (N and E, S and E, X and Y, etc.), you can modify the ICS.PAR parameter file (see Section 3.6).

Direction: *dir*

When *dir* appears in the data field, you are being prompted to identify the direction. You can identify the direction by key-in or by graphically indicating the direction.

In graphics, when a direction is necessary for a command, ICS uses the data point given to accept the last known point preceding the *dir* data field to calculate the direction. Therefore, when *dir* appears in the prompt field, ICS is asking you if you like the direction already calculated. If you like the direction, press <D> to accept. If you do not, press <R> once to establish a new direction; press <R> twice to redefine the initial point to be used in calculating the direction.

Directions can be input as azimuths or bearings. Acceptable units include sexagesimal degrees, decimal degrees, and gons. Directions can also be input as the sum or difference of other directions; such expressions are evaluated left to right and must be enclosed in parentheses.

Note: You can change the grid orientation and angular units in the parameter file ICS.PAR.

Also, directions can be input using the A delimiter (see Section 2.3.2). See Table 2-2 for input examples for *dir*.

*Table 2-2. Sample Key-ins for Directions*

| *Key-in* | *Explanation* |
|---|---|
| BEARINGS | |
| N 30 5 58.OW | North 30 degrees, 5 minutes, 58 seconds West |
| N 30 5 58W | North 30 degrees, 5 minutes, 58 seconds West |
| N 30-05-58.OW | North 30 degrees, 5 minutes, 58 seconds West |
| S90 0 0 E | Due East |
| S 0 0 0 W | Due South |
| A 7 30 | Direction from point 7 to point 30 |
| (S50.W+A7 8-A12 3) | Bearing S 50-0-0W plus angle between course 7 8 and minus angle between course 12 3. |
| AZIMUTHS | |
| 253-42-30 | 253 degrees, 42 minutes, 30 seconds |
| 253 42 30.0 | 253 degrees, 42 minutes, 30 seconds |
| 0 0 0 | Due North |
| 45. | 45 degrees |
| 45 30. | 45 degrees, 30 minutes |
| -90 0 0 | Due West |
| 90-0-0 | Due East |
| 192.3267 | 192.3267 degrees |
| A482 483 | Direction from point 482 to point 483 |
| (S50.0 W+A 7 8 - A 12 3) | Bearing south 50 degrees west plus azimuth from 7 to 8 minus azimuth from 12 to 3. |

Distances: *dis*

When *dis* appears in a data field, you are being prompted to identify a distance. You can identify the distance by keying it in, or by graphically indicating the distance. You can have a positive or negative distance.

In graphics, when a distance is necessary for a command, ICS uses the data point given to accept the last known point preceding the *dis* data field to calculate the distance. Therefore, when *dis* appears in the prompt field, ICS is asking you if you like the distance already calculated. If you like the distance, press <D> to accept. If you do not, press <R> once to establish a new terminal point; press <R> twice to redefine the initial point to be used in calculating the distance.

Also, the distance can be expressed using the D delimiter (see Section 2.3.2 for details) or using input modifiers (see Section 2.3.4 for details). Table 2-3 shows sample key-ins for distances.

*Table 2-3. Sample Key-ins for Distances*

| *Key-in* | *Explanation* |
|---|---|
| 342.1 | Distance of 342.1 units |
| 50 | Distance of 50.0 units |
| (50 + 100) | Distance of 150 units |
| (342.1 + 50 - 100) | Distance of 292.1 units |
| D 1.500 | Distance from point 1 to 500 |
| (D 1 500 - D 3 8 + 14) | Distance from point 1 to 500 minus the distance from point 3 to 8 plus 14.0 units |

Angles: *a* and *va*

When *a* or *va* appears in the data field, you are being prompted for the angle or vertical angle, respectively. You can identify angles by keying them in, or by graphically identifying them. Graphically identifying them works the same way as the *dir* and *dis* fields.

You can key in angles as degrees, minutes, and seconds (dms), or as centesimal degrees, minutes, and seconds (gon). Separate the degrees minutes and seconds with either a space or a dash (-). Also, you can express angles as decimal degrees. For example, the following key-ins for *a* or *va* are equivalent:

35.5
35-30.
35-30-00.

Only the degrees portion of the angle can carry a positive or negative sign. If the minutes contain a decimal point, then seconds are omitted. If the degrees have a decimal point, then both the minutes and seconds are omitted. Degrees, minutes, and seconds input can be greater than 360, 60, and 60, respectively. For example, 0-156-0 is equivalent to 2-36-0.

Note: If you do not input minutes and seconds, you must terminate the angle field with a period. Otherwise, ICS will consider it an incomplete field, and an error will result. See Table 2-4 for sample key-ins that have periods to terminate the angle field.

An angle to the right (clockwise) is positive. An angle to the left (counterclockwise) is negative.

Angles can be defined as the sum or difference of other angles; these expressions must be enclosed in parentheses. Also, angles can be defined using the G delimiter (see Section 2.3.2 for details). Table 2-4 shows sample key-ins for angles.

*Table 2-4. Sample Key-ins for Angles*

| *Key-in* | *Explanation* |
|---|---|
| 75 0 5.1 | 75 degrees, 0 minutes, 5.1 seconds right |
| -75 0 5.1 | 75 degrees, 0 minutes, 5.1 seconds left |
| 075-00-05.1 | 75 degrees, 0 minutes, 5.1 seconds right |
| 90-00-00 | 90 degrees right |
| 90 0 0 | 90 degrees right |
| +90 0 0 | 90 degrees right |
| 90. | 90 degrees right |
| 230 20 0 | 230 degrees, 20 minutes right |
| -0 30 0 | 30 minutes left |
| -70-30.5 | 70 degrees, 30 minutes, 30 seconds left |
| (30-0-0) | 30 degrees, 0 minutes, 0 seconds right |
| (A 5 6 - 30.) | Azimuth from 5 to 6 minus angle of 30 degrees, 0 minutes, 0 seconds |
| G 7 8 9 | Angle at point 8 from point 7 to point 9 right |
| -G25 30 6 | Angle at point 30 from point 25 to point 6 left |

### 2.3.2 Using Delimiters in the Command Line

Delimiters are certain alphabetic characters that you can use to instruct ICS to calculate a value. For instance, if a data field requires a distance value, you can key in the distance, or you can use a delimiter to instruct ICS to calculate the distance between two points. Table 2-5 shows the delimiters ICS allows. You can key these delimiters in from the keyboard, or you can select these on the ICS matrix menu by pressing <D> over the desired delimiter in the DELIMITERS menu block. You do not have to put a space between the delimiter and the point ID you key in.

Note: You can use delimiters while working in graphics. For example, if you want to calculate a direction using the A delimiter, you could place the cursor over A on the DELIMITERS portion of the menu and press <D>. Then you could graphically select two points. ICS calculates the direction from the first point you select to the second point you select.

Limitation of the T delimiter

Using the T delimiter with an alignment does *not* result in stationing decreasing ahead. Rather, it transposes the defined stationing from one end of the alignment to the other.

*Table 2-5. ICS Delimiters*

| *Delimiter* | *Data Fields* | *Examples and their Meanings* | |
|---|---|---|---|
| A | directions: *dir* | A 12 56 | direction from point 12 to point 56 |
| | | (A 1 2-A3 4) | direction from point 1 to point 2 minus the direction from point 3 to point 4 |
| D | distances: *dis* | D 67 54 | distance from point 67 to point 54 |
| | | (D 5 4+D 3 1) | distance from point 5 to point 4 plus the distance from point 3 to point 1 |
| G | angles: *a* and *va* | G 12 20 23 | angle at point 20 from point 12 to point 23 right |
| | | -G7 8 9 | angle at point 8 from point 7 to point 9 left |
| T | descriptions: *des* | 15T | transposition of stored figure 15 |
| | | (3 4)T | same as (4 3) |

### 2.3.3 Using Special Operators in the Command Line

Special operators are symbol characters, including the asterisk (*), period (.), greater-than symbol (>), and less-than symbol (<). You can use these characters to place comments in the report file or to instruct ICS to refer to the previous command line for data values. The following sections describe each character in detail.

The Asterisk Symbol (*): Placing Comments in the Output File

If you want to place comments in the output file, you can do so by using the asterisk symbol (*). Comments must be preceded by an asterisk (*) in the input records and must be placed after the last data field in a command line. Also, in alpha mode you can place a comment on a line by itself by placing the asterisk in column 1.

Note: Commands that contain an asterisk in the parameter list *cannot* contain a comment on the same line. For example, INPUT FILE, NEW FEATURE TABLE, OUTPUT FILE, and START OF JOB are commands that cannot contain an asterisk on the same line.

The Period Symbol (.): Copying Data Fields from Previous Data Fields

In alpha, if you are using the same command repeatedly, you can use a period (.) in place of a value for a data field to instruct ICS to insert the same value as that of the previous command line. For example, to define two coordinates at the exact same location, you can key in:

```
STORE
  1 1 1 1      * store point 1 at [1 1 1]
  2 . . .      * store point 2 there as well
```

The Greater-Than Symbol (>): Incrementing Previous Data Fields

You can also use the greater-than symbol (>) when using the same command repeatedly in alpha. This symbol instructs ICS to add one to the value of the previous command line. You *cannot* use this operator for angular quantities. For example, to store two consecutive points at the exact same location, you can key in:

```
STORE
  1 1 1 1      * store point 1 at [1 1 1]
  > . . .      * store point 2 there as well
```

The Less-Than Symbol (<): Decrementing Previous Data Fields

You can also use the less-than symbol (<) when using the same command repeatedly in alpha. This symbol instructs ICS to subtract one from the value of the previous command line. You *cannot* use this operator for angular quantities. For example, to store two consecutive points at the exact same location, you can key in:

```
STORE
  2 1 1 1      * store point 2 at [1 1 1]
  < . . .      * store point 1 there as well
```

Sample Traversing Sequence Using Special Characters

Consider the following traversing sequence. For information on the LOCATE ANGLE command, refer to Section 8.2.3.

```
LOCATE ANGLE
  1000 1001 1002 10-15-25 102.05
  > > > 65-15-25 100        *bs 1001, at 1002, fs 1003
  . . > . (. +100)          *bs 1001, at 1002, fs 1004
                            *1003 is a pol; 1004 is 200 from 1002
  > > > 9-15-15 75.92       *bs 1002, at 1003, fs 1005
  . . > 44.5   16.5         *fs 1006-1008 as sideshots
  . . > 28.7   75.5         *fs 1007
  . . > 76.2   56.3         *fs 1008
  > > > 36-9-15 98.65       *continue traversing
              .
              .
              .
```

Note: **Special operators must be used with discretion. In particular, the special characters should only appear after a command in which all data items have been entered for the command WITHOUT using any operators. Also, the previous command must be the same command as the one with the special operators. Even if two commands have the same input sequence and data types, this does NOT guarantee proper handling of special operators.**

### 2.3.4 Using Input Modifiers in the Command Line

Input modifiers are specific prefixes and suffixes you can use when entering values. By default, ICS commands expect a certain unit for their parameters; however, some of these commands allow you to input another unit if you prefix/suffix the value. The parameters (data fields) that have allowable alternate units are distance (*dis*), elevation (*z*), and degree of curvature (*dc*)/radius (*r*).

If you do not want to prefix/suffix values, you can use the INPUT MODES command in the ENVIRONMENT menu block to define active units for curve mode, height mode, and unit mode. When an active unit is set, ICS interprets the values you input according to the active unit, not the default unit the command expects. Refer to Section 7.3.12 for more details.

The precedence rules for command line inputs and modes set by the INPUT MODES command are:

1. An input modifier overrides *both* a mode setting and the default unit a command expects.

2. A mode setting *only* overrides the default unit a command expects. It does *not* override an input modifier.

Note: Using an input modifier to specify interpretation according to the default command is *not* recommended when input modes are set. The preferred procedure is to clear input modes prior to selecting the command.

<u>Distances</u>

By default all ICS commands require you to input distances in the linear units defined in the database (ICS_HEADER—entity 53). With suffixes, however, you can input alternate units, and ICS converts the values you give into the linear units defined in the database. Conversion factors are based on:

1 survey foot = 12/39.37 meters

1 survey chain = 66 survey feet

You define the database units when you create the database using SURVEYDB.COM. The database units can be feet, meters, or chains.

The database units can be specified with the characters shown below.

| | |
|---|---|
| Feet | ', f, F, ft, fT, Ft, or FT |
| Meters | m or M |
| Kilometers | km, Km, or KM |
| Chains | ch or CH |

Allowable *suffixes* for distance values include:

| | |
|---|---|
| ', f, F | convert from feet to database units |
| m, M | convert from meters to database units |
| c, C | convert from chains to database units |

Note: Every distance that requires conversion *must* be suffixed. The converted distance is *not* explicitly reported, but the results are evident in the ICS computations.

Elevations

By default, locate commands expect a vertical angle for defining the z coordinate. If you want to give an elevation or rod reading instead, you can do so by using the *prefixes* shown below.

| | |
|---|---|
| a,A | define heights by vertical (zenith) angle |
| e,E | define heights by direct elevation value |
| r,R | define heights with rod readings |

Degrees of Curvature or Radii

By default, SIMPLE CURVE, SIMPLE SPIRAL IN, SIMPLE SPIRAL OUT, COMPOUND SPIRAL, and FIT ALIGNMENT commands expect a degree of curvature for defining curves. The FIT CURVE command expects a radius by default. Using prefixes, you have the option to define curves using the degree of curvature or radius for these commands. The *prefixes* are shown below.

| | |
|---|---|
| d,D | define curve by degree of curvature |
| r,R | define curve by radius |

# ICS Directories and Files

3

# 3. ICS DIRECTORIES AND FILES

ICS uses three directories: PRO_DD_ICS, WRK_DD_ICS, and USR_DD_ICS. The PRO_DD_ICS directory contains all Intergraph Interactive Coordinate Geometry Subsystem product files. *This product directory is required. Do not modify this directory.* The logical name PRO_DD_ICS is assigned to the [IGR.ICS.PRO] directory.

The WRK_DD_ICS directory is the numeric equivalent of PRO_DD_ICS. The numeric directory for ICS is [15,50]. The USR_DD_ICS directory can contain site or user-defined files, such as the feature table file and the parameter files. If ICS does not find these files in USR_DD_ICS, it looks in PRO_DD_ICS. This directory is optional.

ICS uses several files: the database files, design file, audit trail file, report file, command file, feature table file, general parameter file, and annotation parameter file. Table 3-1 lists these files with descriptions. Some of the files are named by ICS; others you can name yourself. The files that are ASCII files can be printed on a line printer. The following sections describe each file in detail except for the database file which is described in detail in Section 4.

The features table file and the two parameters files are in the PRO_DD_ICS directory. You can copy any or all of these files to your directory and edit them to meet your needs. In order to have ICS use the feature table and parameter files in your directory, you must make the job assignment by keying in

ASSIGN/JOB dev:[UIC] USR_DD_ICS

where *dev* is the logical name of your disk drive, and *UIC* is your user identification code. Use ASSIGN/GROUP dev:[UIC] USR_DD_ICS to assign these to a group of users with [nnn,xxx] directories with a common nnn.

*Table 3-1. ICS Files*

| *File Extension* | *Filename* | *Purpose* | *ASCII (Y/N)* |
|---|---|---|---|
| .ATR | same as the .DGN file (default for alpha: first four letters of username) | recovery: contains input records | Y |
| .CMD | any valid filename (you define the name) | batch mode input | Y |
| .DBS | any valid filename (you define the name) | storage of points and figures | Y |
| .DGN | should be different than .DBS filename | placing graphics | N |
| .PAR | ICS_ANNOTATE.PAR | defining default parameters that control annotation and coordinate, line, and curve tables | Y |
| .PAR | ICS.PAR | defining default general parameters | Y |
| .RPT | any valid filename (you define the name) | documenting results, errors, and optionally input records | Y |
| .TBL | FEATURES.TBL | defining features (symbology) | N (can convert) |

In addition to the files listed in Table 3-1, the ICS product delivery directory contains files with default point and pattern cells. To support the use of these cells, a font library, feature menu, and a couple of patterning files are provided. Each of these files is described briefly below. You can pattern elements using an ICS batch patterning program called FCPELT (Feature Code Patterning by Element List). See Appendix C.

SURSYM.CEL — This is an IGDS cell file containing a set of standard survey point cells. You can modify these cells or add additional cells. To use these cells as symbols, you can install them in a font library or use the delivered font library, SURFONT.FLB.

SURPAT.CEL — This is an IGDS cell file containing a set of standard survey cells for line and area patterning. You can use these pattern cells for IGDS interactive patterning or with the ICS batch patterning program FCPELT. You can modify these cells or add additional cells to this file.

SURFONT.FLB — This file contains all the standard IGDS fonts available in PRO_DD_IGDS:FONTLIB as well as all point cells from SURSYM.CEL. **These point cells have been installed as symbol font 90 in SURFONT.FLB.** All other symbols required for Intergraph surveying software products such as Geodetic Network Analysis (GNA™) and Localization and Analysis of Deformations (LAD™) are also included in this symbol font.

SURFEA.DGN — This is an IGDS design file containing a feature matrix menu cell that has been installed in ICS.CEL. The menu selections represent the standard features in FEATURES.TBL. They provide a quick means to set an active feature without having to remember specific feature numbers or names. When you select from the menu, an AUTO PLOT command with the appropriate feature is passed to ICS. To attach the menu from within ICS, key in:

SURFEA,Mn -OR- SURFEA,Sn

where *n* equals 1, 2, or 3. At a workstation the menu is available directly from the ICS control strip. See Appendix A for information on the Feature Table and the Feature Table Editor (EFEED), and see Appendix B for information on modifying this feature menu.

FCPELT.CLD — This is the command language definition file. It defines the FCPELT command line to the system.

FCPELT_DR.EXE — This is an executable, the ICS feature table-driven patterning pre-processor.

## 3.1 Design File

If you are going to use ICS in graphics, you must create a 2D or 3D design file. This is a standard Intergraph design file, and it contains any graphics elements placed by an ICS command. This file is optional in the alphanumeric environment.

## 3.2 Audit Trail File

The Audit Trail File is an ASCII file created or appended to each time you run ICS. It records all inputs, serving as a historical record of user/program interaction. You can edit the audit trail file using any VMS system editor, and you can use it as input later to recover what you have done. You can disable the creation of this file by changing a parameter in ICS.PAR (see Section 3.6).

If you are writing to a design file, by default the audit trail file is assigned the same name as the design file. If you do not write to a design file, by default the filename takes the first four characters of the username and adds the extension .ATR. You can change the default filename for the audit trail file by keying in:

    ASSIGN/JOB filename ICS$ATR

where *filename* is the desired audit trail filename.

You can also specify the device and UIC in the form

    ASSIGN/JOB dev:[UIC]filename ICS$ATR

To change the name of the default audit trail file, store this logical assignment in your LOGIN.COM file. Otherwise, you must key in this logical assignment every time you log in.

With consecutive runs of ICS, the audit trail file will be appended to the end of the previous .ATR file of the same name. A header listing the date and time is placed in the .ATR file as a comment at the beginning of each ICS run. You can use the .ATR file for batch input.

## 3.3 Report File

This file is optional and can be created each time ICS is invoked. It records all input (optional) and output written to the terminal, as well as any error messages. ICS formats this file as a report, and you can print this report on a line printer.

In alpha, a code is available for inserting form feeds into report files at locations you specify. This code is two asterisks (**); you key in these asterisks in columns 1 and 2 of the command line or input file line where you want to place the page break.

## 3.4 Input File

If you want to run ICS in the batch mode, you must create a bulk input file. Create and edit this file using a system editor. When you have completed entering your information, select or key in INPUT FILE (see Section 7.1.2).

Note: If you are running an input file in alpha mode and you want to abort the run, press the <ESC> key.

## 3.5 Features Table File

The default feature table FEATURES.TBL contains a set of graphical characteristics (color, line weight, text size, etc.) used in plotting points, plotting lines, and labeling drawings and charts developed with ICS and IGDS.

If you wish to edit the product FEATURES.TBL or create your own feature table, copy this table to your directory. Use the IGDS FONT LIBRARIAN to create more symbols (see the *IGDS Font Librarian User's Guide*). Use EFEED, the feature table editor, to edit feature tables (see Appendix A for a description of EFEED).

Note: In order to have ICS use the feature table from your directory, you must make the job assignment described in Section 3.

## 3.6 Parameter File

The ICS parameter file (ICS.PAR) contains start up values for ICS. A copy of the ICS.PAR file with the possible values for each parameter appears below. The default values in the left column are automatically assigned by ICS. You can change these values as described below.

Editing the ICS.PAR File

You can modify the ICS.PAR file using one of two methods:

- a standard VAX editor
- the ICS ENVIRONMENT Tutorial (in graphics only).

In order to use either of these methods, you first copy ICS.PAR from PRO_DD_ICS to your working directory, and then make the job assignment as described in Section 3. Using a standard VAX editor, you can modify any parameters by replacing the current values with new ones.

Note: When you use EDT to edit ICS.PAR, do not use tabs.

In graphics you have the option to display the ICS ENVIRONMENT Tutorial to change a subset of the parameters in ICS.PAR. To display this tutorial, you select the ICS.PAR command in the ENVIRONMENT portion of the menu (see Section 7.3.5).

Note: The ICS.PAR command will attempt to change the product directory file if you have not previously created one in your user directory.

The ICS.PAR File

A copy of the ICS.PAR file is shown on the following two pages. Spaces between some of the lines are included for ease of reading. For more information on the use of variables within the namelist tables in ICS_ANNOTATE.PAR, refer to Section 12.1.1 (coordinate table variables), Section 12.1.2 (line and curve table variables), and Section 13.2.2 (alignment variables).

```
D       !Angular mode: D=0-360d/G=0-400g
40      !Maximum number of lines/output page
100     !Maximum number of errors/job

R       !+ve direction of horizontal axis  (R=right, L=left)
U       !+ve direction of vertical axis  (U=up, D=down)
V       !coordinate sequence (H=horizontal 1st, V=vertical 1st)
U       !direction of zero azimuth (R, L, U, D)
R       !direction of azimuth normal to zero (R, L, U, D)

N       !character for zero azimuth
E       !character for normal to zero azimuth
S       !character for opposite to zero azimuth
W       !character for opposite to normal to zero azimuth
=       !character delimiter for annotation label prefix
R       !character for radius of curve annotation
L       !character for length of curve annotation
T       !character for tangent length of curve annotation
D       !character for central angle of curve annotation
L       !character for length of spiral annotation
S       !character for central angle of spiral annotation
a       !character for change in degree of spiral curvature annotation

Z       !vertical reference direction of theodolite (Z=zenith, H=horizontal)
4       !decimal accuracy readout
10      !vertical exaggeration for profiles (exag:1)
1       !1= prompt if point/fig defined; 0=reassign with no prompt
1       !create complex elements (0= no, 1 = yes)
0       !create/write audit trail file (0 = no, 1 = yes)
1       !list feature name with coordinates (0 = no, 1 = yes)
B       !Default angular output mode; (B=bearings, A=azimuth)

PI      !Alignment annotation for line to line (3 char max/pad with blanks)
PC      !Alignment annotation for line to curve (3 char max/pad with blanks)
TS      !Alignment annotation for line to spiral (3 char max/pad with blanks)
PT      !Alignment annotation for curve to line (3 char max/pad with blanks)
PCC     !Alignment annotation for curve to curve (3 char max/pad with blanks)
CS      !Alignment annotation for curve to spiral (3 char max/pad with
        blanks)
ST      !Alignment annotation for spiral to line (3 char max/pad with blanks)
SC      !Alignment annotation for spiral to curve (3 char max/pad with
        blanks)
STS     !Alignment annotation for spiral to spiral (3 char max/pad with
        blanks)
```

| | |
|---|---|
| PVC | !Alignment annotation for line to vertical curve (3 char max/pad with blanks) |
| PVT | !Alignment annotation for vertical curve to line (3 char max/pad with blanks) |
| PRV | !Alignment annotation for vertical curve to vertical curve (3 char max/pad with blanks) |
| PVI | !Alignment annotation for vertical intersection point (3 char max/pad with blanks) |
| | |
| CC | !Alignment annotation for center of curvature (3 char max/pad with blanks) |
| PI | !Alignment annotation for spiral intersection of tangents (3 char max/pad with blanks) |
| POL | !Alignment annotation for point on line (3 char max/pad with blanks) |
| POC | !Alignment annotation for point on circular curve (3 char max/pad with blanks) |
| POS | !Alignment annotation for point on spiral (3 char max/pad with blanks) |
| | |
| 10 | !point/figure ID level (1-63) |
| 1 | !decimal seconds readout for angular values |
| 1000000 | !increment for overlap station equations |
| 2 | !station format: 2=xxx+xx.xx; 3=xxx+xxx.xx |
| | |
| BK | !annotation for back station at equation |
| AH | !annotation for ahead station at equation |
| STA | !annotation prefix for station label (3 char max/pad with blanks) |
| OFF | !annotation prefix for offset label (3 char max/pad with blanks) |
| 2 | !area annotation: 0=master, 1=area, 2=master and area |
| AREA | !prefix for area annotation (6 char max/pad with blanks) |
| | |
| METRE | !(0) master units suffix (10 char max/pad with blanks) |
| HECTARE | !(1) area units suffix (10 char max/pad with blanks) |
| | |
| G1 | !vertical alignment annotation prefix for back tangent (3 char max/pad with blanks) |
| G2 | !vertical alignment annotation prefix for forward tangent (3 char max/pad with blanks) |
| HI/LOW | !vertical alignment annotation for drain/crest point (6 char max/pad with blanks) |
| | |
| 2.5 | !place holder for future enhancement ... necessary, but not currently used |

## 3.7 Annotation Parameter File

The ICS Annotation Parameter File (ICS__ANNOTATE.PAR) contains parameters that control the content, order, and spacing of data written to coordinate, line, and curve tables. (You create these tables using the TABLE commands in the PLOT portion of the menu. See Section 12.2). This file also contains parameters that affect the annotation of horizontal alignments.

Similar to ICS.PAR, ICS searches your directory (USR__DD__ICS) for this file. If no file is found, ICS uses the default parameter file in the ICS product directory (PRO__DD__ICS). The ICS__ANNOTATE.PAR file is organized differently than the standard ICS.PAR file. Parameters in ICS.PAR are not listed in any particular order, whereas related parameters in ICS__ANNOTATE.PAR are grouped into namelist tables. The following namelist tables are included:

    $COORDINATE__TABLE
    $CURVE__TABLE
    $LINE__TABLE
    $LABEL__ALIGN

The individual parameters for each group follow the namelist table name and are delimited by a $END.

Note: **It is essential that you do not change nor delete any of the namelist table names, individual parameter names, and the corresponding delimiters ($END).**

<u>Editing the ICS ANNOTATE.PAR File</u>

You can modify the parameter values by editing the ICS__ANNOTATE.PAR file with a standard VAX editor. To customize the parameters, copy ICS__ANNOTATE.PAR from PRO__DD__ICS to your working directory. Then modify any parameters by replacing the current values with a new ones. For example, to change the filename associated with a coordinate table, you would edit the line

    filename = 'usr__dd__ics:coord.tbl'

which immediately follows $COORDINATE__TABLE. If the filename you wanted was XYZ.TBL, the new line would read

    filename = 'usr__dd__ics:xyz.tbl'

Similarly, to set the next available curve number to equal 100, you would edit the next__curve parameter in the $CURVE__TABLE namelist table. The new line would read

    next__curve = 100

Note: The equal sign is required. Anything following an exclamation mark (!) is an optional comment.

The ICS ANNOTATE.PAR File

A copy of the ICS_ANNOTATE.PAR file is shown below. Spaces between some of the lines are included for ease of reading.

```
PARAMETERS TO DEFINE COORDINATE TABLE FORMAT.
Note that the total table width is limited to 80 characters.
Specify order to control column order and contents.
EN for easting/northing , NE for northing/easting.
ENH for easting/northing/height , NEH for northing/easting/height.
ID is ALWAYS column 1 , Height is ALWAYS column 4.

 $COORDINATE_TABLE
        filename = 'usr_dd_ics:coord.tbl'
        order = 'ENH'                   !
        just = 'R'                      ! 'C' (centre) or 'R' (right) justification
        spacing = 3                     ! # of spaces between columns
        names(1) = 'Point Number'                  ! ID column name
        fieldwidths(1) = 10             ! ID fieldwidth
        names(2) = 'Northing (Ft)'                 ! N column name
        fieldwidths(2) = 15             ! N fieldwidth
        names(3) = 'Easting (Ft)                   ! E column name
        fieldwidths(3) = 15             ! E fieldwidth
        names(4) = 'Z (Ft)'                        ! H column name
        fieldwidths(4) = 10             ! H fieldwidth
 $end

PARAMETERS TO DEFINE CURVE TABLE FORMAT.
ICS updates next curve number after writing curve data or default curve data.
Maximum curve number is 9999.
Table data is written to 'filename' when the CURVE TABLE lock is ON.
Specify 'order' to control column order and contents. Where:
R for radius
D for delta
T for tangent
t for tangent bearing
L for length
C for chord
c for chord bearing
d for degree of curvature
The curve number is ALWAYS column 1.
Curve data is written to 'default_filename' if annotation length exceeds
curve length and MISC1 in the feature table is set appropriately.
'Default_order' controls column order and content where curve annotation
exceeds the curve length.
Possible columns are radius,delta,length and tangent corresponding to ICS.PAR
and MISC1 in the feature table.
Note that MISC1 field of the feature table controls the content of curve
annotation and handling of strings that exceed curve length size.
```

```
$CURVE_TABLE
        next_curve =    43              ! next available number
        filename = 'usr_dd_ics:curve.tbl'
        order = 'dDLTtCc'
        default_filename = 'usr_dd_ics:defcurve.tbl'
        default_order = 'DRLT'
        prefix = 'C'                    ! curve lable prefix
        just = 'R'                      ! 'C' (centre) or 'R' (right) justification
        spacing = 2                     ! # of spaces between columns
        curve = 'No.'                   ! curve number
        delta = 'Delta'                 ! delta angle column
        radius = 'Radius'               ! radius column
        length = 'Length'               ! arc column
        tangent = 'Tangent'             ! tangent name
        tangent_bearing = 'Tangent Brg'         ! tangent bearing
        chord = 'Chord'                 ! chord name
        chord_bearing = 'Chord Brg'             ! chord bearing
        degree = 'Degree'               ! degree of curvature
        number_width = 5                ! curve number column fieldwidth
        dist_width = 10                 ! fieldwidth for ALL distance columns
        ang_width = 15                  ! fieldwidth for ALL angle columns
$end
```

PARAMETERS TO DEFINE LINE TABLE FORMAT.
ICS updates next line number after writing line data or default line data.
Maximum line number is 9999.
Table data is written to 'filename' when the LINE TABLE lock is ON.
Specify 'order' to control column order and contents. Where:
D for distance
B for bearing or direction
The line number is ALWAYS column 1.
Line data is written to 'default_filename' if annotation length exceeds line length and MISC1 in the feature table is set appropriately.

```
$LINE_TABLE
        next_line =    51               ! next available number
        filename = 'usr_dd_ics:line.tbl'
        order = 'BD'
        default_filename = 'usr_dd_ics:defline.tbl'
        prefix = 'T'                    ! line lable prefix
        just = 'R'                      ! 'C' (centre) or 'R' (right) justification
        spacing = 2                     ! # of spaces between columns
        line = 'No.'                    ! line number header
        bearing = 'Bearing'             ! bearing column header
        distance = 'Distance'           ! distance column header
        number_width = 5                ! line number column fieldwidth
        dist_width = 10                 ! fieldwidth for distance column
        ang_width = 15                  ! fieldwidth for bearing column
$end
```

PARAMETERS TO CONTROL HORIZONTAL LABEL ALIGNMENT.
In addition to the parameters provided in the command string of LABEL ALIGNMENT, these variables control horizontal alignment annotation

```
$LABEL_ALIGN
        label_reading = 'a'          ! 'A' (ahead) or 'R' (right) reading
        major_tic_x = 200            ! major tick mark interval
        ma_mi_scale = 2              ! scale factor between major and minor tick
        tic_side = 'R'               ! ticks are on 'R' (right) or 'L' (left) of
                                     align
        label_side = 1               ! 1: same as tic    2: opposite to arc
                                     direction
        station_orient = 'PA'        ! 'PA' parallel to align, 'PE' perpendicular
                                     to
        full_sta_format = 1          ! 0 = without +00,  1 = with +00
        with_leader     = 'N'        ! N = No, without leader, Y = Yes, with
                                     leader
$end
```

ICS Databases 4

# 4. ICS DATABASES

Each ICS database contains three sections:

- Header area
- Figure area
- Coordinate area.

The header area contains information ICS needs to use the database, such as the job number, units, and the lengths of the figure and coordinate areas. The figure area contains figure information and figure ID's for all figures, and the coordinate area contains the coordinates and point ID's for all points. Figure 4-1 shows the database structure.

If the ICS database is not large enough to store the necessary points and figures, ICS automatically increases the size according to the DMRS extend setting. (This database expansion feature of ICS depends on the availability of contiguous disk space.)

Note: Extension of the database negatively impacts execution time.

You can move databases created with version 8.8.3 and later using the simple DCL COPY command. When creating a database, if you do not specify a device/directory, the database is device/directory independent, and you can move the database without recompiling it. *Care must be taken to ensure that certain files remain contiguous and that directory/file protections do not change.*

You can also move a database that is device/directory dependent, but you must recompile that database after you move it. Refer to the *DMRS Data Definition Language (DDL) User's Guide* for more information.

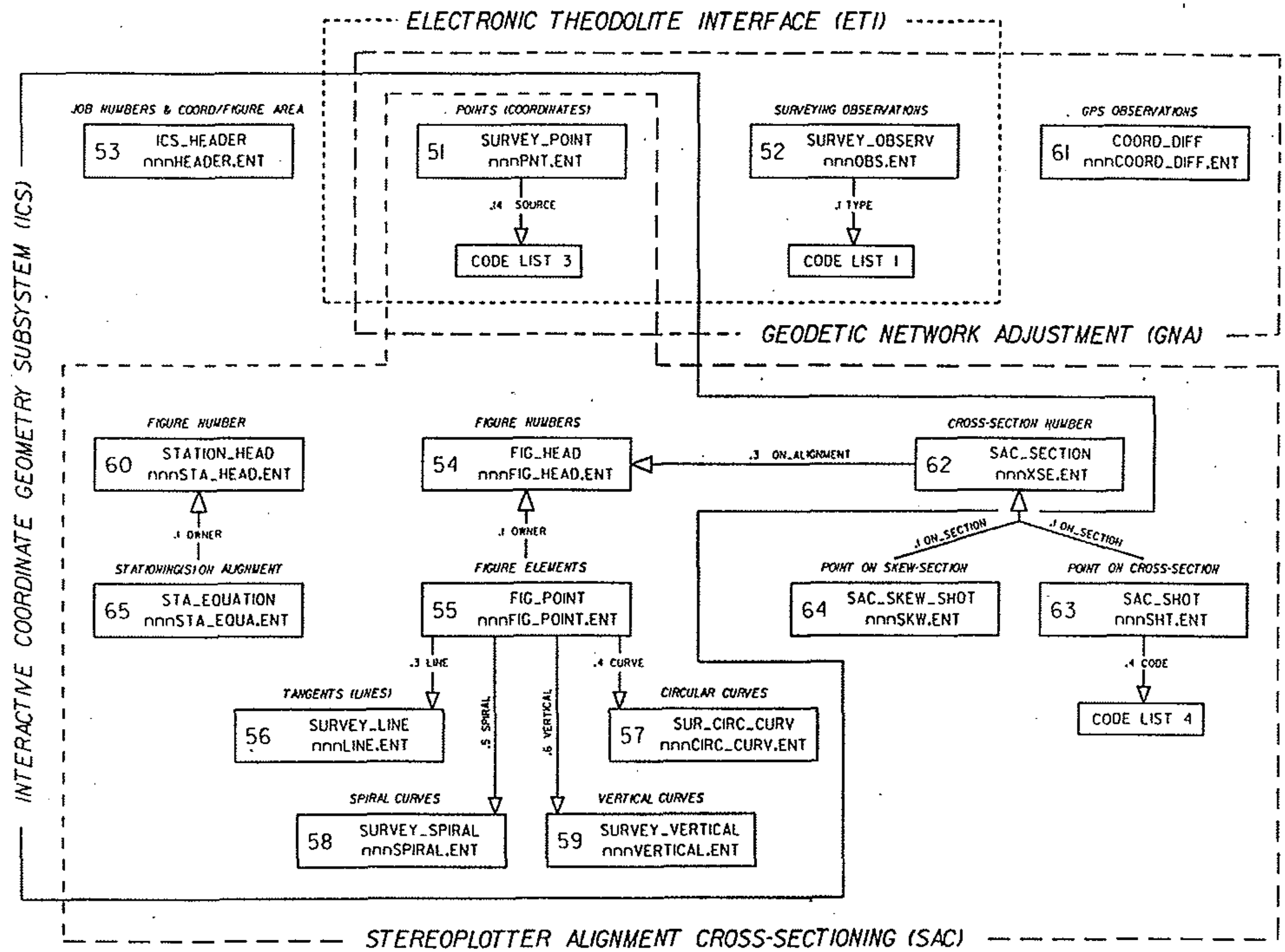



*Figure 4-1. The Database Structure*

## 4.1 Units of Measurement for ICS Databases

ICS recognizes three frequently used systems of measurement:

- SI system (meter)
- English system (foot)
- Chain system (chain).

The relationship among the systems is:

1 survey foot = 12/39.37 meters
= 0.3048 meters

1 chain = 66 survey feet
= 792/39.37 meters
= 20.1168 meters.

## 4.2 Creating a Database

Before you begin a project, you must create a unique database for each ICS job using the command procedure SURVEYDB.COM (SDB) in the ICS product directory. To successfully create a database, you must create it in a numeric directory (for example, [201,15], not [ICS.TAL]). Also, you should be sure that you have enough contiguous disk space to store the new database.

Two methods are available to you for creating a database with the SDB. One method is to respond to the individual SDB prompts for the number of points, observations, coordinate differences, cross-sections, cross-section shots, and skew shots. The numbers you key in are then multiplied by constant factors contained in SDB to generate an appropriate number of available occurrences for each database entity. Unless ETI, GNA, or SAC will be run on the database, a value of one (1) should be keyed in for all of the parameters listed above except for the number of points. This method may or may not create an optimal database for a project.

The other method uses a parameter file, PRO_DD_ICS: SURVEYDB.PAR, and bypasses the constant factors in SDB. To use this method you copy SURVEYDB.PAR to a user directory and edit this file, providing a reasonable number of occurrences for each database entity in the parameter file. This method is more dependable for creating an optimal database for a specific project. A copy of the SURVEYDB.PAR file is shown below on the following page.

| | | |
|---|---|---|
| 1000 | ! Default number of points to allocate; | 0 < x < 99999 |
| 1000 | ! Default number of figures to allocate; | 0 < x < 999999 |
| 2000 | ! Default number of figure elements to allocate; | 0 < x < 999999 |
| 20 | ! Default number of lines; | 0 < x < 100000 |
| 5 | ! Default number of curves; | 0 < x < 100000 |
| 5 | ! Default number of spirals; | 0 < x < 100000 |
| 5 | ! Default number of vertical curves; | 0 < x < 100000 |
| 100 | ! Default number of observations to allocate; | 0 < x < 99999 |
| 10 | ! Default number of coord. diff. to allocate; | 0 < x < 99999 |
| 100 | ! Default number of sections to allocate; | 0 < x < 9999 |
| 300 | ! Default number of shots to allocate; | 0 < x < 999999 |
| 50 | ! Default number of skew_shots to allocate; | 0 < x < 99999 |
| m | ! Default nominal units; | {m, ft, or ch} |

Follow these steps to create a database:

1. At the $ prompt key in

   SDB or @PRO_DD_ICS:SURVEYDB.COM

   *The system displays the Database Screen.* Key in the information as outlined in the following steps.

2. **$_Database filename:**

   Key in the name of the database to create and press <RETURN>. Be sure to key in the entire filename specification (that is, device:[UIC]filename) if the file is to be created on a device and directory other than your directory. The .DBS filename extension is optional.

   For example, key in QS3:[101,215]TALKEETNA.DBS.

3. **$_Number of points or Parameter filename:**

   Key in the approximate number of points to store in the database or the parameter file to be used for creating the database.

   -OR-

   Key in the name of the SURVEYDB.PAR file that you have edited. This file should have been copied from PRO_DD_ICS and edited as described above. Skip the following steps. *The system displays the messages shown in Step 9.*

4. **$_Number of observations:**

   *If you are using ETI and GNA along with ICS,* key in the estimated number of observations, then press <RETURN>.

   -OR-

   *If you are not using ETI and GNA along with ICS,* key in one (1), then press <RETURN>.

5. **$_Number of coordinate differences:**

   *If you are using GNA along with ICS and intend to process coordinate difference observations,* key in the number of coordinate differences, then press <RETURN>. Otherwise, key in one (1) and press <RETURN>.

6. **$_Nominal units (2 character max):**

   Key in one of the following abbreviations for the units to be used in ICS computations and output.

   | | |
   |---|---|
   | Feet | ', f, F, ft, fT, Ft, or FT |
   | Meters | m or M |
   | Kilometers | km, Km, or KM |
   | Chains | ch or CH |

7. **$_Number of x-sections**

   If you are using SAC, key in the appropriate number of x-sections. Otherwise, key in a one (1).

8. **$_Number of shots**

   If you are using SAC, key in the appropriate number of shots. Otherwise, key in a one (1).

9. **$_Number of skew-shots**

   If you are using SAC, key in the appropriate number of shots. Otherwise, key in a one (1).

*The screen displays the following messages if the database is successfully completed:*

*WRITING DATA DEFINITION FILE.. COMPILING DATA DEFINITION FILE ..DDL STOP .. INSERTING MAXIMUM FIGURE TABLE SIZE .. SUCCESSFUL COMPLETION.*

*If this procedure does not successfully create the database, the screen displays the following messages:*

*ERROR CREATING DATABASE .. NO DATABASE GENERATED .. ERROR INSERTING MAX FIGURE TABLE SIZE.*

To find out why the database did not compile, look at messages within the database file with the .LST extension (for example, TALKEETNA.LST). Refer to the *DMRS Data Definition Language User's Guide*, Volume II of the Complete DMRS Document Set, for further information on handling database compilation errors.

## 4.3 Upgrading 8.8.2 Databases: SDB_82to83.COM

ICS databases created in an earlier version of ICS than version 8.8.3 must be upgraded using the command procedure SDB__82TO83.COM. The prompting sequence in this command procedure is similar to the one in SDB. For a detailed explanation of the ICS database Upgrade Utility, see Appendix C.

How to Run ICS 5

# 5. HOW TO RUN ICS

This section provides a step-by-step procedure for running ICS. Explanations of specific ICS commands are given in Sections 6 through 14. Details about ICS operating characteristics are given in Section 2, and details about ICS files are in Section 3. Before running ICS, you must create a database. The procedure for creating a database is given in Section 4.

Before running ICS you must:

- create a design file (if graphics are required)
- create a database

You might also want to:

- change the audit trail filename (optional)
- create/edit FEATURES.TBL
- review/edit ICS.PAR
- review/edit ICS_ANNOTATE.PAR

If you copy the feature table and/or the parameters files to your user directory and edit them, you must make the USR_DD_ICS assignment as described in Section 3.

IGDS and ICS Menus

During initialization of ICS, the IGDS command menu cell is replaced with an ICS command menu cell called IGDMEN. In graphics mode, you can use the IGDS menu while you are using ICS by simply selecting commands from the IGDS menu. After using a command from the IGDS menu, you select the RETURN command from the ICS menu to continue working in ICS.

In the event that the IGDS paper menu is physically moved on the digitizing surface during an ICS session, you must re-attach the IGDMEN cell to properly operate ICS. Key in the following:

```
AM=IGDMEN,CM
```

There are now two ICS application matrix menu cells, ICS and ICSA. The ICS menu contains icons and command names, while ICSA contains only the command names. The menus can be attached as paper menus using the key-in:

AM=ICS,M*

-OR-

AM=ICSA,M*

where * is 1, 2, or 3. They can also be attached as screen menus in any view other than the tutorial view using the key-in:

AM=ICS,S*

-OR-

AM=ICSA,S*

Note: ICS does not support screen menus on LSI terminals. Also, due to display limitations, ICS screen menus are truncated on Unix-based workstations.

The ICS and IGDS menus are implemented on the InterPro 32 and InterPro 32C workstations as a combination of control strip and hierarchical pop-up/pull-down menus. Both paper and pop-up/pull-down menus can be used on InterAct 32 and InterAct 32C workstations. Keying in one of the following commands will attach all pop-up/pull-down menus:

AM=ICS/ICS_CS,CS
AM=ICS/ICS_HMENU,H1
AM=ICS/ICS_HMENU,H2
AM=ICS/ICS_HMENU,H3

Initiating ICS with the Delivered User Command: ICS.UCM

After you enter the Intergraph graphics environment, you can use the user command ICS.UCM to initiate ICS processing. This user command is located in the PRO_DD_ICS product directory, and you can copy it to your working directory. This user command:

- attaches the ICS cell library to your design file
- initiates ICS by attaching one of the ICS menus.

If you are on an InterAct, the system prompts you to select either a screen or table menu option. If you are on an InterAct 32 or InterPro 32, ICS attaches the control strip and hierarchical menu.

By editing the file ICS.UCM, you can tailor the user command to control:

- the menu version type: graphics or alpha
- the menu number: M1, M2, M3, S1, S2, or S3
- the command menu name that ICS is to re-attach upon exiting on an InterAct
- the control strip and hierarchical menu that ICS is to re-attach upon exiting on an InterAct 32 or InterPro 32.

Note: **If you attach ICS with menu number 3, the ICS table or screen menu will not be detached when you exit ICS.** This will result in automatic initialization of ICS at that graphics station. For this reason, you should only use menu numbers M3 or S3 if you require this feature. If you get ICS into this automatic initialization cycle, key in

AM= ,M3

-AND-

AM= ,S3

Instructions for changing ICS.UCM are contained within the file itself. Use the DCL command

TYPE/PAGE ICS.UCM

to review these instructions.

## 5.1 Running ICS in the Alphanumeric Environment

Follow these steps for running ICS in the alphanumeric environment:

1. At the $ prompt, key in

   ICS

2. **Please enter design filename>**

   Key in the name of the design file in which to place the graphics.

   -OR-

   If you do not want to place graphics, press <RETURN> to skip entering the name of the design file.

3. **Please enter audit trial filename>**

   Key in the name of the audit trail file. If unspecified, the extension defaults to .ATR.

   -OR-

   Press <RETURN> to accept the default filename. The default filename is the design filename if you have one or the first four letters of your username if you do not. The default extension is .ATR.

4. ICS>

   Key in the START OF JOB command to run ICS interactively. You can now key any ICS commands. If you want to generate a report, key in the OUTPUT FILE command first.

   -OR-

   Key in the INPUT FILE command to run ICS in batch mode.

   Note: In order to run ICS in batch mode, you must have previously created the input file (see Section 7.1.2).

## 5.2 Running ICS in the Graphics Environment

1. To get into graphics, at the $ prompt, key in

   ICE

   -OR-

   @PRO_DD_ENVR:ICE

2. GRAPHICS>

   Key in the design filename.

3. Attach the cell file and ICS menu using the user command (see description in Section 5). Key in

   UC=PRO_DD_ICS:ICS.UCM

   If you have copied this file to your user directory and edited it, key in UC=*filename* where *filename* is the name of the user command filename.

   Note: If you do not use this user command you must attach the cell file using the key-in

   RC=ICS:[15,50]ICS.CEL

   You must also attach the menu using one of the key-ins described in Section 5.

**Note: Initializing ICS via the AM=ICS/ICS_CS,CS requires the seletion of an ICS menu command (typically Alpha Scroll) to allow initialization to continue.**

4. If the IGDS menu is not attached, the system displays the message:

   IGDS MENU NOT ATTACHED
   IDENTIFY ORIGIN OF IGDS MENU

   Identify the origin of the IGDS menu by pressing <D> on the lower lefthand corner of the menu.

5. If the database was not previously attached, the system displays the message:

   ENTER DATABASE NAME

   Key in the database name.

**Note: Make no further entries unless prompted, until the following message appears in the status field:**

**Select Option**

# CONTROL Commands

6

# 6. CONTROL COMMANDS

The CONTROL menu block includes commands that allow you to get in and out of ICS. See also the alpha END OF JOB and START OF JOB commands in Section 14. CONTROL commands include the following:

| *Command* | *For* |
|---|---|
| NEW JOB | setting up ICS for a new job |
| END OF RUN | closing the ICS database, completing any plotting, and terminating ICS after the last job in a run |
| RETURN TO ICS | returning to ICS from the IGDS menu |

## 6.1 NEW JOB

The NEW JOB command is the same as the START OF JOB command in alpha. In alpha the START OF JOB command must be the first command in any ICS job (with the exception of the optional OUTPUT FILE and INPUT FILE commands). The START OF JOB command instructs ICS to reset all indicators and to prepare for a new job. The command also requires you to specify a job number *j1 j2* and an ICS database *fn.dbs*, and to place headings at the beginning of each page. One useful technique is to use a job number equivalent to your UIC; that is, use [group,member] to assign *j1 j2*.

When running ICS in graphics, NEW JOB is performed automatically during initialization. Upon receiving the message **Select Option**, you can select any ICS command. Select the NEW JOB option from the matrix menu if you want to change jobs after initialization.

Operator Sequence

1. Select NEW JOB or key in START OF JOB.

   *The system responds:*

   START OF JOB j1 j2 dev:[UIC]fn.dbs *heading

2. Identify the following:

   | | |
   |---|---|
   | j1 j2 | the job number (two integers). |
   | dev:[UIC] | the device and user identification code where the database is located. |
   | fn.dbs | the filename of the database. |
   | *heading | the comments to be printed at the top of each page of output. In the ICS environment, ICS uses the text between the asterisk and column 67 as the comments. In the graphics environment, only comments through column 40 are used as the heading. |

## 6.2 END OF RUN

The END OF RUN command is identical to the alpha END OF JOB command except that an END OF RUN command terminates ICS. This command should be the last command you use in the last job in a run. The END OF RUN command closes the ICS database, completes any plotting, sends the message ICS PROCESSING COMPLETE, and terminates ICS.

Note: In alpha you must use the alpha END OF JOB command between jobs in single run. See Section 14.3 for more details.

Operator Sequence

1. Select or key in END OF RUN.

   *ICS closes the database, completes any plotting, and sends the message: ICS PROCESSING COMPLETE.*

   Note: You have not completely exited ICS until all the message fields have been cleared.

## 6.3 RETURN TO ICS (Graphics Only)

The RETURN TO ICS command allows you to return to ICS from the IGDS menu. After you have selected an IGDS command, if you want to continue in ICS, you select this command.

ENVIRONMENT Commands

# 7. ENVIRONMENT COMMANDS

The ENVIRONMENT menu block includes commands for defining input and output files, selecting modes, and for toggling settings on and off. The ENVIRONMENT menu block is divided into three parts:

- FILES Commands
- ON/OFF Commands
- SETTINGS Commands.

## 7.1 FILES Commands

Use the FILES commands to define input and output files and to change feature tables. FILES commands include the following:

| *Command* | *For* |
|---|---|
| OUTPUT FILE | defining an output file where ICS will record its output |
| INPUT FILE | placing ICS in batch mode by instructing it to retrieve commands from an input file |
| NEW FEATURE TABLE | defining a new feature table while you are in an active job. |

### 7.1.1 OUTPUT FILE

The OUTPUT FILE command records ICS output in a specified output file. This is in addition to the usual display of output on the terminal screen and placement of data in the audit trail file. ICS formats this output file for printing on a line printer. If necessary, you can change the format by editing the ICS.PAR file. Be sure to specify the output file before specifying an input file, unless the input file contains an output file command.

If you set the variable *mode* to 1, this command lists all the input records in the output file as well as displaying them on the screen. If you set *mode* to 0, the command does not list the input records in the output file. If you set *mode* to -1, the command closes the output file.

Operator Sequence

1. Select or key in OUTPUT FILE.

   *The system responds:*

   OUTPUT FILE mode *dev:[UIC]output filename.ext

2. Identify the following:

| | |
|---|---|
| mode | 0 to list all input records in the output file; 1 to not list input records; -1 to close the output file. |
| *dev:[UIC] | the device and user identification code where the file is located (in alpha mode, the * is required). |
| filename.ext | the filename. |

### 7.1.2 INPUT FILE

The INPUT FILE command places ICS in the batch mode by instructing ICS to retrieve commands and data from a pre-existing file. Once you select this command and input the appropriate information, ICS reads and executes the input file immediately. Another option you have is to process the input file for syntax only, and you can specify this option using the parameter *syntax_only*. If you process the input file for syntax only, the audit trial file will be commented appropriately, and no commands will be actually executed except for control commands.

All commands and input data in the input file are recorded in the audit trail file. To place output in an output file when using an input file, you must specify the output file before specifying the input file. If you set the variable *mode* to 1, this command lists all the input records in the output file (if specified) as well as displaying them on the screen. If you set *mode* to 0, the command does not list the input records in the output file or on the screen.

ICS processes the information in the input file until one of the following events occurs:

- A prompt is displayed on the terminal. You must respond before processing continues.
- ICS reaches the end of the input file. ICS restores the interactive mode and prompts you for further input.
- ICS encounters an END OF RUN command.
- You press the <ESC> while you are in alpha.
- You select a pull-down menu from the control strip on a Unix workstation. (This is useful for pausing during input file execution.)

Operator Sequence

1. Select or key in INPUT FILE.

   *The system responds:*

   INPUT FILE mode /syntax_only *dev:[UIC]filename.ext

2. Identify the following:

| | |
|---|---|
| mode | 0 to list input records; 1 to not list input records. |
| syntax_only | 1 to process for syntax only. |
| *dev:[UIC] | the device and user identification code where the file is located (in alpha mode, the * is required). |
| filename.ext | the filename. |

### 7.1.3 NEW FEATURE TABLE

The NEW FEATURE TABLE command allows you to change feature tables while in an active job. This provides greater flexibility in defining features by allowing you to group sets of features in different tables and then selectively call the separate tables.

By default, the file USR_DD_ICS:FEATURES.TBL is attached when you enter a design file. This default file must have the extension .TBL for the software to locate it. Other feature tables, however, may have any valid IGDS name, extension, and directory specification. In addition, logical names and subdirectories are acceptable. For example, USR_DD_ICS:FEATURES.BB and ZG0:[110020.USR]BOGUS.FILE are valid alternate feature table file specifications.

To change feature tables in graphics, simply select the NEW FEATURE TABLE command from the menu and respond to the prompt with the desired file specification. To change feature tables in alpha, key in the command:

NEW FEATURE TABLE 0*new_feature_tablename

## 7.2 ON/OFF Commands

Use the ON/OFF commands for toggling settings ON and OFF. When you select an ON/OFF command, if the setting is OFF, it is turned ON; if the setting is ON, it is turned OFF. ON/OFF commands include the following:

| *Command* | *For* |
|---|---|
| FENCE | turning fence mode ON and OFF |
| QUIET MODE | turning quiet mode ON and OFF |
| ALPHA SCROLL | turning alpha scroll ON and OFF |
| ACTIVE OFFSET | turning the active offset ON and OFF |
| ACTIVE FEATURE | switching between the active feature and database information for placing points |
| AUTO PLOT | turning automatic plotting ON and OFF. |

### 7.2.1 FENCE

The FENCE ON/OFF command toggles the active fence to ON or OFF. If a fence is currently active, when you select this command, ICS erases the existing fence and disables the fence for point and figure searches. If a fence is not currently active, when you select this command, ICS redisplays a disabled fence and enables fence locates.

Note: **In 3D design files, only fences placed in top or front views are operated upon.**

### 7.2.2 QUIET MODE (Graphics Only)

The QUIET MODE command allows you to toggle the prompting for new point and figure ID's ON and OFF. This command affects any command that requires assignment of new point or figure ID's, such as STORE, POINT LOCATION, INTERSECT, DIVIDE, HORIZONTAL ALIGNMENT, and VERTICAL ALIGNMENT commands.

When you toggle QUIET MODE to ON, ICS does not prompt you for new point and figure ID's (*n* and *nfg*) within a command. QUIET MODE ON has the same affect as keying in a point or figure number of 0; ICS searches the database for the next available point or figure ID and makes the assignment automatically.

QUIET MODE OFF restores prompting for point and figure ID's within all commands. The default setting is QUIET MODE OFF.

### 7.2.3 ALPHA SCROLL

The ALPHA SCROLL command allows you to toggle between displaying and not displaying output in the tutorial view while running ICS. You should disable ALPHA SCROLL before invoking IGDS tutorials.

Note: If you toggle between full screen view and quad view without disabling the alpha scroll tutorial, the tutorial screen will not be refreshed until you select subsequent ICS commands that spool new output to the tutorial.

### 7.2.4 ACTIVE OFFSET

The ACTIVE OFFSET command allows you to toggle between using active offsets or not using active offsets. You set active offsets using the SET ACTIVE OFFSET command (see Section 7.3.1). ACTIVE OFFSET only affects those ICS commands that use offsets as parameters.

When you enter a design file, active offsets are not defined. Therefore, any command that *requires* offsets (such as PLOT XSECTIONS) prompts you to set active offsets first. Those commands that *accept* offsets but do not require them (such as LOCATE LINE) prompt you for individual offsets within the command line.

If active offsets are set and activated, subsequent ICS commands that use offset parameters automatically use the active offsets and do not prompt for them. If active offsets are defined and then turned off, subsequent commands that accept or require offsets prompt for individual offsets within the command line.

### 7.2.5 ACTIVE FEATURE

The ACTIVE FEATURE command allows you to toggle between having an active feature and not having an active feature for point annotation operations. To use the feature previously set by the AUTO PLOT command for point annotation, you toggle the active feature ON. In order to annotate a point with its feature stored in the database, you toggle the active feature to OFF.

ACTIVE FEATURE affects only the ANNOTATE COORDINATE commands. STORE commands always use the active feature unless overridden by a key-in. With the exception of POINTS, PLOT commands always use the active feature unless overridden by a key-in. PLOT POINTS always uses the feature stored in the database with the point information unless overridden by key-in.

### 7.2.6 AUTO PLOT

The AUTO PLOT command allows you to toggle between automatically plotting or not plotting all points and figures as they are stored or created. The feature ICS uses for automatically placing points and their ID's is the active feature you define using the AUTO PLOT SETTING command. By default, AUTO PLOT is on and the active feature is the feature named *DEFAULT*.

Note: AUTO PLOT sets a default plotting height for elements to be placed in a 3D graphics file.

You can use AUTO PLOT in the alpha environment even though you do not have a design file attached to ICS. This allows a point to be stored in the database with a feature attached to it. If you have a design file attached, the points and figures are plotted with that feature.

## 7.3 SETTINGS Commands

Use the SETTINGS commands to define active parameters. Some of the SETTINGS commands are toggle commands, and operate similar to the ON/OFF commands. The SETTINGS commands include the following:

| *Command* | *For* |
|---|---|
| ACTIVE OFFSET | defining values for the active offsets |
| SET PARAMETER | establishing the form of angular output and scaling ICS positions from a local survey coordinate system to a mapping grid |
| SET HEIGHT | establishing various instrument and target heights within ICS to use for 3D (vertical) calculations |
| 2D/3D MODE | toggling between 2D and 3D input modes |
| ICS.PAR | interactively editing the parameters while you are working in a design file |
| ACTIVE JUSTIFICATION | toggling the ACTIVE JUSTIFICATION ON and OFF for use with the ANNOTATE commands |
| SET ORIENTATION | setting the orientation for symbol placement along an alignment |
| TRAVERSE/SIDESHOT | toggling between traverse mode and sideshot mode |
| FENCE CONTENT | toggling the fence content mode between points and figures |
| REVIEW | reviewing current settings and parameters that affect annotation |
| SET AUTO PLOT | defining the feature that ICS will use for automatic plotting and a default plotting height for elements to be placed in a 3D graphics file |
| INPUT MODES | defining active units for curve mode, height mode, and unit mode. |

## 7.3.1 ACTIVE OFFSET

The ACTIVE OFFSET command in the SETTINGS menu block allows you to define values for the active offsets. When setting offets, *off1* refers to a right offset and *off2* refers to a left offset. ICS commands that accept only one offset value use active offset *off1* and ignore *off2*. Also, in the FIGURE INTERSECT commands, active offset *off1* refers to *offg1*, and *off2* refers to *offg2*.

When you select ACTIVE OFFSET ON, the current active offsets are displayed. Press <D> to accept the value, or key in a new offset.

Note: Active offsets may not be saved between design sessions.

Use the ACTIVE OFFSET ON/OFF command to activate or deactivate the active offsets.

### 7.3.2 SET PARAMETER

The SET PARAMETER command establishes the angular output type and the translations and scale factor for a local survey coordinate system to a mapping grid conversion. SET PARAMETER adds false Easting/Northing values, if specified, to any coordinates stored with subsequent STORE commands. These shifted coordinate values place graphic elements in the design file, but are not stored in the database.

Note: SET PARAMETERS multiplies the grid factor first and then adds false Northing and Easting coordinates *falc1* and *falc2*, respectively.

The grid factor is used for plot commands *only*.

Operator Sequence

1. Select or key in SET PARAMETER.

   *The system responds:*

   SET PARAMETER dirmode /falc1 /falc2 /grid

2. Identify the following:

dirmode — the angular output mode. 0 to set angular output to azimuths; 1 to set angular output to bearings. The default is azimuths. The *dirmode* overrides the angle readout formats specified in the design file TCB and the parameter file ICS.PAR.

falc1, falc2 — false Eastings and Northings to be added to all ICS coordinate values when placing graphics. When specified with a grid factor, these values allow ICS positions to be scaled from a local survey coordinate system to a mapping grid when graphics are placed.

grid — a mean grid factor for the survey. It is usually computed as the product of the map projection scale factor (at the center of the mapping area determined from the appropriate projection table) and an elevation factor computed from the formula

$$E.F. = R_e / (R_e + h)$$

where

$R_e$ — equals the approximate radius of the earth, 6,367,000 meters, or 20,906,000 feet.

h — equals the mean elevation of the survey.

### 7.3.3 SET HEIGHT

The SET HEIGHT command establishes various instrument and target heights within ICS to use for 3D (vertical) calculations. You can perform these calculations when using POINT LOCATION and INTERSECT commands. SET HEIGHT is useful whenever the backsight or foresight target heights are changed, or whenever the instrument height is changed.

In general, the POINT LOCATION commands use only the heights of the first instrument and targets. There are two exceptions to this: the LOCATE ANGLE and LOCATE DEFLECTION commands. These two commands can use separate horizontal and vertical backsight target heights, provided the two heights have been previously set with SET HEIGHT. It is important to note that the proper method to input separate horizontal and vertical backsight target heights for use by LOCATE ANGLE or LOCATE DEFLECTION command is:

SET HEIGHT htFS1 /hi1 /htBS1 0 0 htVBS1

In this case, the values for *htFS2* and *hi2* are zero. The software response indicates that *htFS1, hi1, htBS1,* and *htBS2* have been defined. In reality the values keyed in are treated as *htFS1, hi1, htBS1,* and *htVBS1* if LOCATE ANGLE or LOCATE DEFLECTION are implemented.

When using SET HEIGHT in conjunction with INTERSECT commands, remember that commands that intersect one or more directions use active instrument and target heights. For example, DIRECTION INTERSECT uses the heights of the first and second instrument and targets, while POINTS DIRECTION INTERSECT uses only the heights of the first instrument and targets.

Operator Sequence

1. Select or key in SET HEIGHT.

   *The system responds:*

   SET HEIGHT htFS1 /hi1 /htBS1 /htFS2 /hi2 /htBS2

2. Key in the following:

| | |
|---|---|
| htFS1 | the height of foresight target for instrument 1. |
| hi1 | the height of instrument 1. |
| htBS1 | the height of backsight target for instrument 1, or the height of horizontal backsight target for instrument 1. |
| htFS2 | the height of foresight target for instrument 2. |
| hi2 | the height of instrument 2. |
| htBS2 | the height of backsight target for instrument 2, or the height of vertical backsight target for instrument 1. |

### 7.3.4 2D/3D MODE

The 2D/3D MODE command allows you to toggle between 2D and 3D input modes. The 2D and 3D input mode parameter affects the POINT LOCATION and STORE commands.

Essentially, 2D mode suppresses elevation prompting in STORE commands and vertical angle prompting in POINT LOCATION commands. When 2D mode is activated, elevations of newly defined points default to the active height (as defined by AUTO PLOT), and vertical angles used for locating points default to 90 degrees from zenith. Elevation and vertical angle defaulting can be overridden in 2D mode with key-ins.

3D mode enables prompting for elevations of points to be defined and vertical angles for points to be located. In a 2D design file, if 3D mode is on, the coordinate z value defaults to 0.

Certain exceptions exist. For example, in the case of a 2D design file in 2D mode, no elevation is assigned. See Tables 7-1 and 7-2 for possible input mode/file dimension combinations and corresponding results.

Note: Certain DIVIDE, INTERSECT, and ALIGNMENT commands do not assign elevations to new points. To insert elevation coordinates into the database, use the DEFINE Z and STORE commands.

*Table 7-1. Elevation Input Rules in Graphics Store Commands* (*)

| | 2D MODE | 3D MODE |
|---|---|---|
| 2D FILE | elevation is not defined unless keyed in | elevation defined by AUTO PLOT unless keyed in |
| 3D FILE | elevation defined by AUTO PLOT unless keyed in | elevation must be keyed in (**) |

* *Elevation* in this table refers to elevation coordinate stored in the database. Plotting heights are *always* undefined in 2D files and are *always* determined by AUTO PLOT in 3D files.

** 2D features, as defined in the feature table, are always drawn at the height set by AUTO PLOT when in a 3D file. Keyed-in elevation coordinates are stored in the database regardless of whether the feature is 2D or 3D. 3D features are always drawn in 3D files at the height defined in the database.

*Table 7-2. Elevation Input Rules in Graphics Point Location Commands* (*)

| | 2D MODE | 3D MODE |
|---|---|---|
| 2D FILE | elevation is not defined | elevation defined by va = 90-0-0 unless keyed in |
| 3D FILE | elevation defined by va = 90-0-0 unless keyed in | elevation defined by vertical angle (**) |

* *Elevation* in this table refers to elevation coordinate stored in the database. Plotting heights are *always* undefined in 2D files and are *always* determined by AUTO PLOT in 3D files.

** 2D features, as defined in the feature table, are always drawn at the height set by AUTO PLOT when in a 3D file. Keyed-in elevation coordinates are stored in the database regardless of whether the feature is 2D or 3D. 3D features are always drawn in 3D files at the height defined in the database.

### 7.3.5 ICS.PAR (Graphics Only)

The ICS.PAR command displays a tutorial that allows you to access the parameter file ICS.PAR and interactively edit the parameters while you are working in a design file. Changes you make to the parameter file through the ICS.PAR command are maintained between design sessions.

The logical name USR_DD_ICS directs the ICS software to the correct directory for parameter files and feature tables. If USR_DD_ICS is assigned to any directory other than the product directories PRO_DD_ICS and WRK_DD_ICS, the ICS.PAR command creates a higher version of ICS.PAR in that directory.

ICS prevents modification of PRO_DD_ICS:ICS.PAR by ordinary users. If USR_DD_ICS is not assigned, or if USR_DD_ICS is assigned to PRO_DD_ICS or WRK_DD_ICS, then the ICS.PAR command writes an ICS.PAR file in the user's current default directory. In such a case, the updated parameters are used for the remainder of the current design session but are not reactivated upon entering a new design session, unless USR_DD_ICS is assigned to the current default directory containing the updated ICS.PAR.

The tutorial displayed by the ICS.PAR command is shown in Figure 7-1. After this tutorial is displayed, you can change a parameter by following the steps outlined below.

Note: You should disable the ICS.PAR tutorial before invoking IGDS tutorials.

Operator Sequence

1. Select the ICS.PAR command.

   *The system displays the ICS.PAR tutorial.*

2. To change a parameter, place the cursor over the value and press <D>. Then key in the new value and terminate the entry with a <RETURN>.

3. After you have finished making changes, select UPDATE PARAMETER FILE.

   *The system responds:* USR_DD_ICS:ICS.PAR is edited.

   Note: If at any time it is necessary to quit without updating ICS.PAR, simply select EXIT, and no changes will be made.

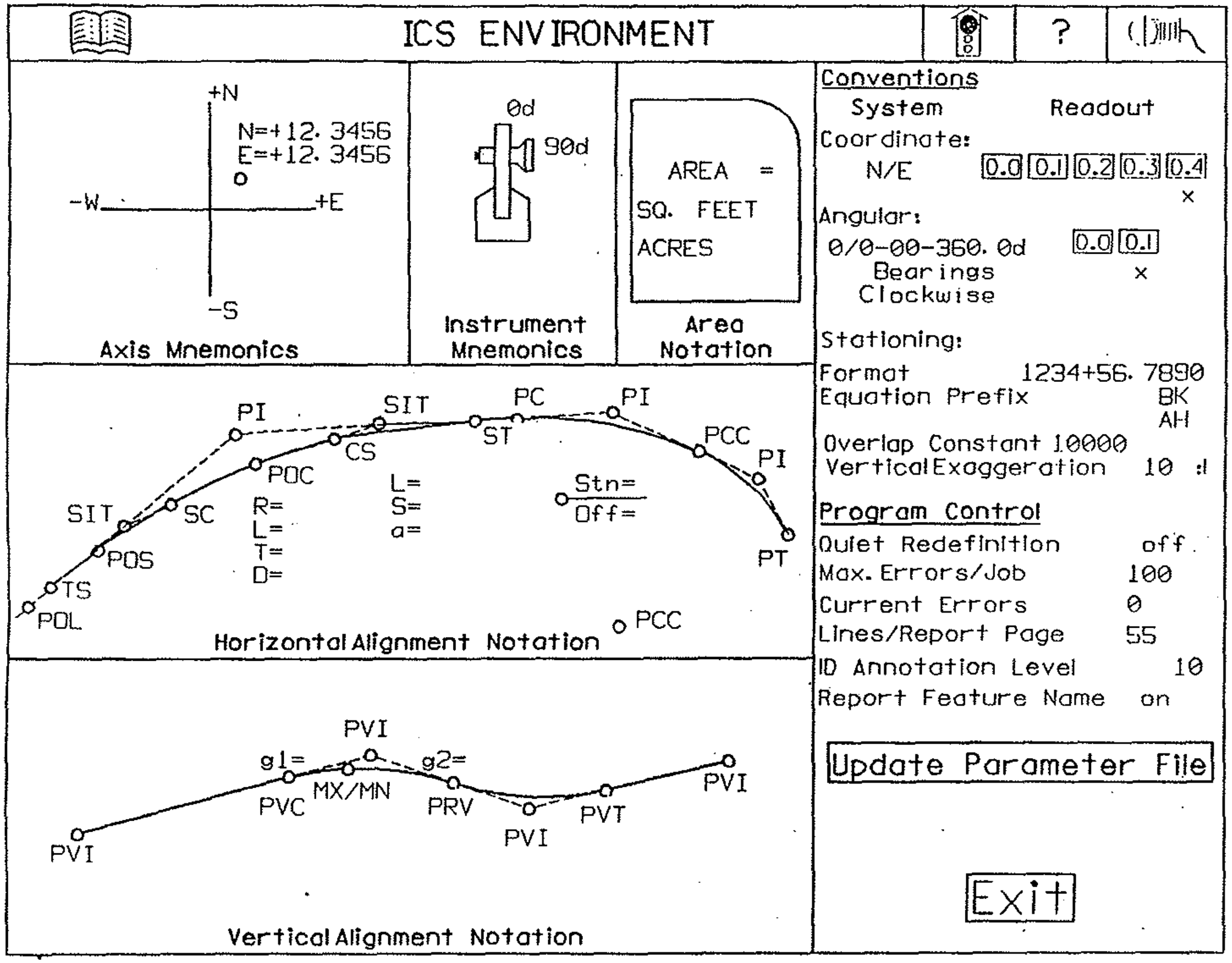



*Figure 7-1. The ICS.PAR Tutorial*

### 7.3.6 ACTIVE JUSTIFICATION

The ACTIVE JUSTIFICATION command allows the current text justification set by IGDS TEXT JUSTIFICATION to override the current feature justification or the justification stored in the database with a point. ACTIVE JUSTIFICATION is used with ANNOTATE commands, the PLOT POINTS command, and the STORE command. The justification setting from the IGDS menu is saved between design sessions.

To change the active justification, toggle ACTIVE JUSTIFICATION to ON and select the new justification setting from the IGDS command menu block TEXT JUSTIFICATION. To annotate points or label lines with the justification of the current feature (as defined by AUTO PLOT), toggle JUSTIFICATION to OFF and ACTIVE FEATURE to ON. To annotate or label with the justification of the feature as stored in the database, toggle both JUSTIFICATION and FEATURE OFF.

By default, ACTIVE JUSTIFICATION is OFF each time a design file is entered. See Tables 7-3 and 7-4 for examples of possible combinations of justification settings. See Table 7-5 for line labeling rules in graphics using feature tables.

*Table 7-3. Annotation Justification Locks in Graphics (*)*

| | FEATURE LOCK ON | FEATURE LOCK OFF |
|---|---|---|
| JUSTIF. LOCK ON | annotation justification defined by IGDS menu | annotation justification defined by IGDS menu |
| JUSTIF. LOCK OFF | annotation justification defined by active feature (**) | annotation justification defined by feature in database |

* Coordinate and elevation annotation only: placement of point ID's is defined by SYMBOL JUSTIFICATION in the feature table. Point symbols are *always* placed with center-center justification.

** Active feature as defined by SET AUTO PLOT; this is independent of AUTO PLOT ON/OFF.

*Table 7-4. Justification Codes for ICS Feature Tables*

| | | |
|---|---|---|
| LT=0 | CT=6 | RT=12 |
| LC=1 | CC=7 | RC=13 |
| LB=2 | CB=8 | RB=14 |

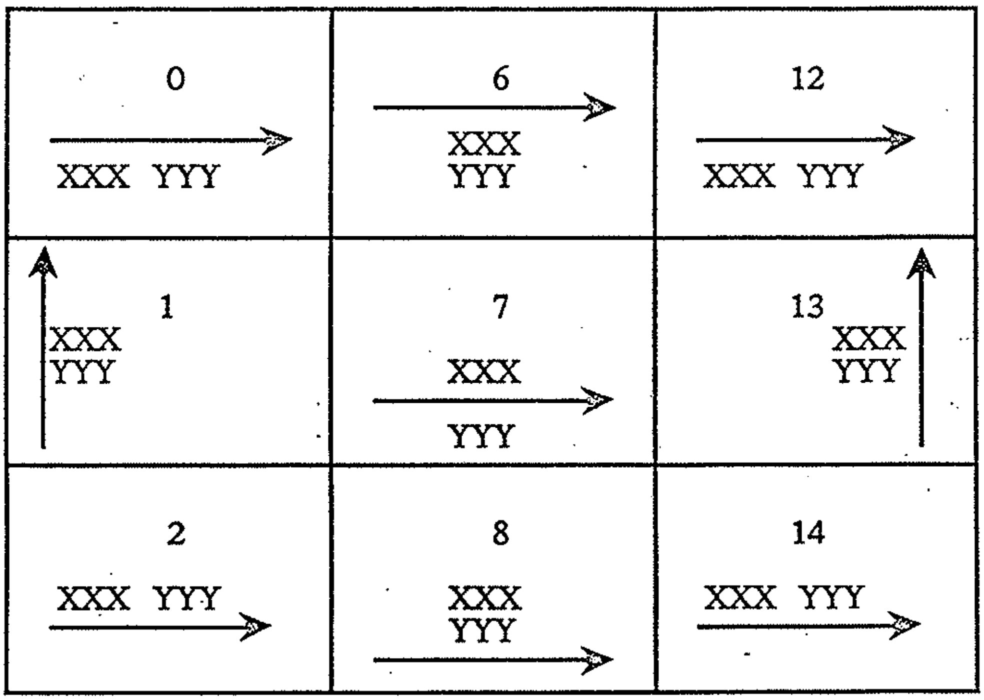
Note: The two-letter acronyms are IGDS justification codes; the numbers are ICS feature table codes. The periods in the table indicate placement of the coordinate locations.

Point symbols are *always* placed with justification 7.

SYMBOL JUSTIFICATION defined in the feature table determines the point ID location--the symbol is placed at the correct location. However, the justification refers to the placement of the symbol relative to the point. For example, 8 indicates that the symbol will be centered below the point.

Coordinate and elevation annotation are placed according to the ANNOTATION TEXT JUSTIFICATION defined in the feature table.

*Table 7-5. Line Labeling Rules in Graphics Using Feature Table Justification Codes*

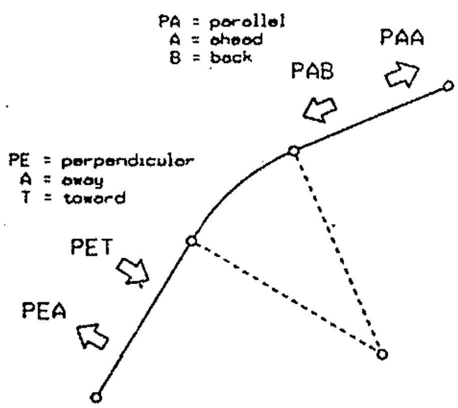



Note: For curve labeling, assuming RLTD order in the feature table, annotation is placed according to the following:

- 6, 7, 8: RL stacked above line; TD stacked below line
- 0, 2, 12, 14: RL side-by-side above line, TD side-by-side below line
- 11, 13: RLTD stacked at mid-chord

### 7.3.7 SET ORIENTATION

The SET ORIENTATION command allows you to set the orientation for symbol placement along an alignment. You can place symbols parallel or perpendicular to a figure, with the symbol facing toward or away from the figure as shown below.




In order to use this command, you must have an active alignment defined using the SET ALIGNMENT command. If there is no active alignment, then all symbols are plotted parallel to the x,y axis of the design plain.

<u>Operator Sequence</u>

1. Select or key in SET ORIENTATION.

   *The system responds:*

   SET ORIENTATION clear or set /mode

2. Identify the following:

| | | |
|---|---|---|
| mode | PAA | to have the symbol placed parallel to the active alignment and facing forward along the alignment. |
| | PAB | to have the symbol placed parallel to the active alignment and facing backward along the alignment. |
| | PET | to have the symbol placed perpendicular to the active alignment and facing toward the alignment. |
| | PEA | Perpendicular to the active figure facing away from the alignment. |
| clear or set | 0 for clear; 1 for set. You must specify this field. | |

### 7.3.8 TRAVERSE/SIDESHOT

The TRAVERSE/SIDESHOT command allows you to toggle between traverse mode and sideshot mode. While in traverse mode, ICS assumes that you are inputting a traverse and does not prompt for the occupied and backsight points. With the traverse mode OFF, ICS assumes that you are inputting sideshots (radial survey data).

Note: Traverse *cannot* be more than 200 points.

Traverse mode can be used with LOCATE ANGLE, LOCATE DIRECTION, LOCATE DEFLECTION, LOCATE LINE, and EXTEND ARC commands. Traverse mode operates in the following manner with each of these point location commands:

<u>LOCATE LINE</u>

With LOCATE LINE, ICS locates points on a line defined by *p1* and *p2*. The first point located in this mode is located from *p1* at the input distance *dis*. This new point is then used as *p1* for the location of the next point on the line.

<u>LOCATE DIRECTION, LOCATE ANGLE, AND LOCATE DEFLECTION</u>

With LOCATE DIRECTION, LOCATE ANGLE, and LOCATE DEFLECTION, ICS locates each new foresight point. Then ICS automatically assigns the new foresight to become the occupied point, using the previous occupied point as the backsight point.

<u>EXTEND ARC</u>

With EXTEND ARC, ICS locates points along an arc defined by *pPC* and *pCC*. The point is located on the arc at an arc distance from the *pPC*. The newly located point is then used as the *pPC* for location of the next point on the arc.

### 7.3.9 FENCE CONTENT

The FENCE CONTENT command allows you to toggle between points and figures for the fence content. This permits you to place a fence and define whether points or figures will be located for input to the description *des* field.

### 7.3.10 REVIEW

The REVIEW command displays a tutorial that allows you to view the current values of parameters, modes, and settings from ICS.PAR, ICS_ANNOTATE.PAR, the feature table, and the ENVIRONMENT menu block.

**REVIEW ANNOTATION AND SETTINGS**

| COORDINATES | LINES |
|---|---|
| *STORE*<br>Right center by active just<br>No label<br>Pt ID: Level 12 centered above | Right and perpend. by active just<br>Distance and direction |
| *PLOT*<br>Right center by active just<br>Content determined by db feature<br>Pt ID: determined by db feature | **CURVES**<br>Perpend. stack by active just<br>Curve label order: RLTD<br>Curve table active<br><br>Rotation and scaling disabled |
| *ANNOTATE*<br>Right center by active just<br>Northing & Easting<br>(BY ELEVATION LOCK) | **ALIGNMENTS**<br>SYMBOLS: Parallel ahead<br>Offsets only will be labeled |
| DEFINE CURVE BY command default<br>DEFINE HEIGHT BY vertical angle<br>UNITS CONVERSION None<br><br>TRAVERSE MODE Off<br>HEIGHT MODE 3D<br>QUIET MODE Off<br><br>FENCE CONTENTS Points | ACTIVE OFFSET Off<br>OFFSET 1: 8.8<br>OFFSET 2: 8.8<br><br>AUTO PLOT On<br>JUSTIFICATION LOCK On<br>ACTIVE FEATURE LOCK Off<br><br>ACTIVE FEATURE:<br>DEFAULT [1] |

*Figure 7-2. The REVIEW Tutorial*

### 7.3.11 SET AUTO PLOT

The SET AUTO PLOT command defines the feature ICS will use for automatic plotting and a default plotting height for elements to be placed in a 3D graphics file. The AUTO PLOT ON/OFF command toggles between automatic plotting and no automatic plotting (see Section 7.2.6). By default, AUTO PLOT is ON and the active feature is the feature named *DEFAULT*.

You can use AUTO PLOT in the alpha environment even though you do not have a design file attached to ICS. This allows a point to be stored in the database with a feature attached to it.

If you specify a feature number greater than zero, AUTO PLOT searches the feature table and, upon finding the feature number, sets it as the active feature. If you key in zero as the feature number followed by an exclamation point (!) and feature name, AUTO PLOT references the feature name following the exclamation point as the desired feature. ICS uses the exclamation point to indicate a feature name. In graphics, you do not have to skip the feature number with a zero, nor do you have to key in the exclamation point; keying in a feature name by itself is sufficient.

Note: If you want to set DEFAULT as the active feature, key in -1 for the feature number.

In addition to the feature number or name, you can assign a value to the default plotting height (*z*) for placement of graphics information.

Note: The AUTO PLOT feature is not changed by use of a feature in a PLOT or ANNOTATE command. The AUTO PLOT feature is only changed by AUTO PLOT.

Operator Sequence

1. Select or key in AUTO PLOT.

   *The system responds:*

   AUTO PLOT fnum /z /!fname (alpha)

   -OR-

   AUTO PLOT feature number or feature name (graphics)

2. Identify the following:

   fnum the feature number or 0.

   z active height (it can be omitted as in A P 0 !X; if not entered, it will be skipped.)

   !fname the feature name.

### 7.3.12 INPUT MODES

The INPUT MODES command allows you to define active units for curve mode, height mode, and unit mode. By default, ICS commands expect a certain unit for their parameters; however, some of these commands allow you to input another unit if you prefix/suffix the value. For example, ALIGNMENT commands that require a degree of curvature to define a curve accept a radius if you prefix the radius value with an *R*. The INPUT MODES command allows you to specify an input mode so that prefixing fields is *not* necessary.

In alpha mode, all the settings are combined in the command INPUT MODES. To skip a field, you key in a comma (,). In graphics, you can select separate menu commands for each mode setting.

Note: INPUT MODES does support batch input, so that you do not have to continually prefix/suffix the fields.

The precedence rules for command line inputs and modes set by the INPUT MODES command are:

1. A command line prefix or suffix overrides *both* a mode setting and the default unit a command expects.

2. A mode setting *only* overrides the default unit a command expects. It does *not* override a command line prefix or suffix.

The command consists of the following fields:

INPUT MODE action /curve mode /height mode /unit mode

<u>Action</u>

The *action* variable has the following possible values:

| | |
|---|---|
| 0 | clear: resets all modes to default values |
| 1 | set: set one or more modes |

<u>Curve Mode</u>

The *curve mode* variable has the following possible values (the default is to define the curve according to the command line specifications):

| | |
|---|---|
| d,D | define curve by degree of curvature |
| r,R | define curve by radius |

Height Mode

The *height mode* variable has the following possible values (the default is to define heights by vertical angle):

| | |
|---|---|
| a,A | define heights by vertical (zenith) angle |
| e,E | define heights by direct elevation value |
| r,R | define heights with rod readings |

Unit Mode

The *unit mode* variable has the following possible values (the default is no conversion):

| | |
|---|---|
| f,F | convert from feet to database units |
| m,M | convert from meters to database units |
| c,C | convert from chains to database units |

Examples of Command Lines

| | |
|---|---|
| INPUT MODE 1 , E | define heights by elevation |
| INPUT MODE 1 r , m | define curves by radius; no effect on current height definition; convert meters to database units |
| INPUT MODE 0 | reset all modes to their defaults |

For example, the following command lines will be interpreted as follows:

```
INPUT MODE 1 R E
SIMPLE SPIRAL 100 13 14 20 21 400 381.97 84-00-00 -1
LOCATE DIR 1 2 30.0 100.0 260.67
```

In this case the SIMPLE SPIRAL command interprets the data item 381.97 as a radius rather than the standard degree of curvature for this item. The LOCATE DIRECTION command interprets the data item 260.67 as an elevation rather than the standard vertical angle.

However, the commands

```
SIMPLE SPIRAL 100 13 14 20 21 400 d12-00-00 84-00-00 -1
LOCATE DIR 1 2 30.0 100.0 d87-20-20
```

would result in a degree of curvature and vertical angle interpretation, respectively. The prefix in each of these commands overrides the mode settings.

**Note: Using a command line prefix to specify interpretation according to the standard default command is NOT recommended when input modes are set. The preferred procedure is to clear input modes prior to selecting the command.**

Operator Sequence for Alpha

1. Key in INPUT MODES.

   *The system responds:*

   INPUT MODES action /curve mode /height mode /unit mode

2. Key in values for the fields according to the specifications shown above. Remember to skip fields with a comma (,).

POINTS/COORDINATES Commands 6

# 8. POINTS/COORDINATES COMMANDS

The POINTS/COORDINATES menu block includes commands for storing and deleting points and coordinates. It also includes commands for computing and storing coordinates for new points relative to previously defined geometric descriptions. The POINTS/COORDINATES menu block is divided into four parts:

- GENERAL Commands
- LOCATE Commands
- DIVIDE Commands
- INTERSECT Commands.

## 8.1 GENERAL Commands

Use the GENERAL commands to store, delete, copy, and move points. Also, use these commands to redefine coordinates or elevations of points. GENERAL POINTS/COORDINATES commands include the following:

| *Command* | *For* |
|---|---|
| STORE | storing points with known coordinates and redefining coordinates |
| DELETE COORDINATES | deleting points from the database |
| COPY POINT | copying the coordinates of a known point to another point ID |
| MOVE POINTS | moving points |
| DEFINE Z | graphically redefining the elevation of a point |

### 8.1.1 STORE

The STORE command assigns a point ID of *n* to the coordinates *c1*, *c2*, and *z* (optional). You can assign up to 2,147,483,647 point ID's to be stored in each ICS database. The point ID must be a positive number in the range from 1 to 2,147,483,647.

In graphics, the STORE command annotates points according to specifications of the active feature if the ACTIVE FEATURE LOCK is ON. If the ACTIVE FEATURE LOCK is OFF, this command annotates points according to specifications of the default feature. The areas of the feature table that affect the way point annotation is placed are: the symbol area, annotation area, Flag 2, Misc 1, and 2nd feature number (see Section 12.1.1 for more details). Also, if the ACTIVE JUSTIFICATION LOCK is on, the justification from the IGDS menu overrides the justification of the active or default feature (see Section 7.3.6).

Although you will probably want to use the COPY POINT and MOVE POINTS commands, you can use STORE to copy the coordinates of a known point to another point ID and to alter the coordinates of a known point. To copy a point from a known point, input the command with the new point ID, 0 for *c1*, 0 for *c2*, 0 for *z* and finally, the ID of the known point from which *c1*, *c2*, and *z* are to be assigned. It is often advisable to copy the coordinates of points to other points before using commands that alter these values. For instance, you might want to copy the coordinates and elevations of points to other points before using an ADJUST/TRANSFORM command.

To alter either a coordinate or the elevation of a known point, you input the following:

- point_ID c1 (new value) 0 0 -1 (to alter c1)
- point_ID 0 c2 (new value) 0 -2 (to alter c2)
- point_ID 0 0 z (new value) -3 (to alter z)

By default, ICS issues a warning if you attempt to use a point ID that already exists. Then you can choose to redefine the point or not. *If you do not want to be warned, you can change a parameter in ICS.PAR (see Section 3.6).*

Operator Sequence

1. Select or key in STORE.

   *The system responds:*

   STORE n c1 c2 /z /p

2. Identify the following:

| | |
|---|---|
| n | the ID of the point to be defined or the ID of the known point to be altered. |
| c1 c2 | the coordinates of the point. The coordinate sequence is set up in the ICS.PAR parameter file (Northing, Easting or x,y). |
| p | the ID of a previously defined point or the field number (*-cno*). If *p* is positive, ICS interprets it as the ID of a previously defined point, and *n* takes its values. If *p* is negative, ICS interprets it as *-cno,* and it will change the coordinate or elevation. Possible values for *-cno* include: -1, -2, and -3. |
| z | the elevation. |

### 8.1.2 DELETE COORDINATES

The DELETE COORDINATES command deletes points. You can use this command in two ways:

- Identifying the point with a key-in to delete database information only
- Identifying the point graphically (with a data point or fence) to delete database information and the graphics associated with the point.

Identifying the Point with a Key-in

Using the key-in method, you key in the points to be deleted in the description *des*. You can enter a list of points (enclosed in parentheses), a figure (no parentheses), or a list of figures (enclosed in brackets). If you enter a figure, all the points in the figure are deleted.

Note: If you later decide you want to delete the graphics that are left, you can do that using any IGDS DELETE command. Because of database linkages, be sure to have the IGDS DELETE ENTITY command set to OFF.

Identifying the Point Graphically

To delete graphics associated with a point, as well as the database information, you must identify the point graphically. If you press <D> on an ICS point symbol and accept with another <D>, that point is deleted from the database, and the point symbol and the point ID are deleted as well.

Note: Other annotation associated with the point is not deleted.

You can also use an IGDS fence to delete several points at a time. (The FENCE toggle command must be ON, and the FENCE CONTENT toggle must be set to points.) After placing a fence in graphics, select DELETE COORDINATES from ICS, and all points within the fence are deleted. The point symbols and point ID's are deleted also, but no other annotation is deleted. Unless the fence overlap lock is ON, ICS will not delete point ID's external to the fence. If the inside lock is ON, be sure that all the point symbols and ID's are completely inside the fence. You will have a chance to accept or reject the identified points listed in the status fields.

Operator Sequence

1. Select or key in DELETE COORDINATES.

   *The system responds:*

   DELETE COORDINATES des

2. Identify the following:

   des — description of the points to be deleted.

### 8.1.3 COPY POINT

The COPY POINT command copies the coordinates and elevation of a known point to another point ID. It is often advisable to copy the coordinates of points to other points before using commands that alter these values. For instance, you might want to copy the coordinates and elevations of points to other points before using an ADJUST/TRANSFORM command.

Note: The graphics will stack if AUTO PLOT is ON.

To copy a list of points, you can key in a list of points in parentheses. You can also identify a list of points graphically, by placing a fence around the points you want to copy before you select the command.

If you want to translate the coordinates of the new points, you can use the optional parameters *dx, dy, dz.* These parameters will increment or decrement the new coordinate values. For example, the command

COPY POINT (100-110) 200 100.0 100.0 10.0

creates points 200 through 210 with coordinate values incremented 100.0, 100.0, and 10.0 from the coordinates of points 100 through 110.

<u>Operator Sequence</u>

1. Select or key in COPY POINT.

   *The system responds:*

   COPY POINT des /n /dx /dy /dz

2. Identify the following:

   des — description of the points to be copied.

   n — the ID of the point to be defined.

   dx — translation in x coordinate for the new point (database EASTING coordinate).

   dy — translation in y coordinate for the new point (database NORTHING coordinate).

   dz — translation in z coordinate for the new point (database ELEVATION coordinate).

## 8.1.4 MOVE POINTS

The MOVE POINTS command moves existing points to new locations and updates the database. You indicate the translations for each coordinate using the parameters *dx*, *dy*, and *dz*. Moving points in the alpha environment is different than moving points in the graphics environment. The sections below describe how you can move points in each of these environments.

Moving Points in the Alpha Environment

In alpha you can move more than one point by keying in a list of points in parentheses. In alpha mode MOVE POINTS does *not* move the graphics point symbol or associated ID. If AUTO PLOT is ON, then this command will plot the points with the new ID's at their new locations. However, if you want existing graphics deleted, you will have to use the IGDS DELETE command.

Moving Points in the Graphics Environment

In graphics you can give precision inputs in the form of a DX= key-in. In a 3D design file you can perform both top and front view moves. The key-ins are interpreted differently according to the view. The key-in **DX=100,200** for a front view would be interpreted as:

delta x = 100
delta y = 0
delta z = 200

The key-in **DX=100,200,25** for a top view would be interpreted as:

delta x = 100
delta y = 200
delta z = 25

In graphics you can use the MOVE POINTS command to perform a fence move or a single point move. If a fence is active, the command assumes you are performing a fence move.

Single Point Move

The single point move resembles the IGDS MOVE ELEMENT command. You press <D> on the point you want to move. Then you indicate the new location by pressing <D> or by using a precision key-in. ICS moves both the ICS point symbol and its associated ID, and updates the database. Then ICS prompts you for the next point you want moved.

Fence Move

The fence move resembles the IGDS fence move command. You press <D> to accept the fence contents and to define the from point. Then you indicate the new location (the to point) by pressing <D> again or by using a precision key-in. ICS then moves all ICS points and their associated ID annotation and updates the database. (Remember to update your screen to see the graphic results.)

Additionally, the fence move can move *all* graphic items within the fence. This option is controlled by the ICS FENCE CONTENTS setting. If FENCE CONTENTS is set to POINTS, ICS moves *only* ICS points and their ID annotation. If it is set to FIGURES, ICS moves all graphics within the fence.

Operator Sequence

1. Select or key in MOVE POINTS.

   *The system responds:*

   MOVE POINTS des dx dy dz

2. Identify the following:

   des description of the points to be moved.

   dx translation in x coordinate (database EASTING coordinate).

   dy translation in y coordinate (database NORTHING coordinate).

   dz translation in z coordinate (database ELEVATION coordinate).

**Note: MOVE POINTS always interprets dx, dy, and dz as translations in the database EASTING, NORTHING, and ELEVATION values. The coordinate order specified in ICS.PAR has no affect.**

### 8.1.5 DEFINE Z (Graphics Only)

DEFINE Z redefines the elevation of any point. Both the database and design file are updated with the new elevation. You must identify the point by pressing <D>; ICS will not accept a key-in for the point.

Note: DEFINE Z uses the STORE command to change the elevation of the point. Although the command is DEFINE Z in graphics, the report file and audit trail file will show the use of STORE to change *z*.

Operator Sequence

1. Select DEFINE Z.

   *The system responds:*

   DEFINE Z p z

2. Identify the following:

   p — the ID of the point for which you are defining the elevation; you must identify it by pressing <D>.

   z — new elevation.

## 8.2 LOCATE Commands

Use the LOCATE commands to compute and store coordinate values in a database. With the exception of TANGENT and PARALLEL LINE commands, ICS computes only one point for each command execution. Most of the LOCATE commands compute points relative to lines and/or arcs. The lines and arcs do not have to exist; only the parameters used to define them such as the points, direction, radius, etc. must be known.

Offsetting a Point

Some LOCATE commands allow you to specify a positive or negative offset. A positive offset is to the right in a forward direction; a negative offset is to the left. For example, if you defined a figure from ***p1*** to ***p2***, a positive offset would be to the right of the line, looking from ***p1*** to ***p2***, and a negative offset would be to the left. If you defined a figure from ***p2*** to ***p1***, a positive offset would be to the right of the line, looking from ***p2*** to ***p1***.

Computing Elevations of Points

To compute and store elevations, use the following POINT LOCATION commands: LOCATE LINE, LOCATE DIRECTION, LOCATE ANGLE, and LOCATE DEFLECTION commands. These commands accept a vertical angle, an elevation value, or a rod reading to define the z coordinate. The Earth's curvature is accounted for when using vertical angles.

By default, these LOCATE commands accept vertical angles for defining the z coordinate. If you want to enter an elevation or rod reading instead, you have two choices. One option is to change the default using the INPUT MODES command in the ENVIRONMENT menu block (see Section 7.3.12). This command allows you to change the default for all LOCATE commands to either elevation or rod reading.

The other option you have is to indicate which option you are using by prefixing the value. You prefix an elevation value with an upper or lowercase *E* and a rod reading with an upper or lowercase *L*. See Section 2.3.4 for more details.

Note: If you set the default to elevation or rod reading and want to input a vertical angle, prefix the vertical angle value with an upper or lowercase *A*.

To compute elevations ($z$) using a vertical angle or a rod reading, you must first establish an instrument height (H.I.). You can set the instrument height with the SET HEIGHT command, which is in the ENVIRONMENT menu block (see Section 7.3.3).

Note: Do not confuse the SET HEIGHT command with the LEVEL RUN command. The SET HEIGHT command sets a given instrument height and target heights, while the LEVEL RUN command calculates the instrument elevation and point elevations.

You have two options when using a vertical angle to compute elevations. If you set the vertical reference direction in the ICS.PAR parameter file to horizontal, you can enter angles measured from the horizon. If you set it to zenith, you can enter angles measured either from the zenith or the nadir.

Using Traverse Mode

LOCATE LINE, LOCATE DIRECTION, LOCATE ANGLE, and LOCATE DEFLECTION can be used in conjunction with the TRAVERSE/SIDESHOT environment setting (see Section 7.3.8). If you turn traverse mode on, when you use any of these commands repeatedly, the new point located becomes the known point for the next time you use the command. This allows you to input *traverse* measurements. With sideshot mode on, the known point (and the backsight point, if used) remains the same for consecutive uses of the commands. This allows you to rapidly input *sideshot* measurements.

LOCATE commands include the following:

| *Command* | *For* |
|---|---|
| LOCATE LINE | locating a new point by distance along a line defined by two points |
| LOCATE DIRECTION | locating a new point by distance and direction from a known point |
| LOCATE ANGLE | locating a new foresight point by angle and distance |
| LOCATE DEFLECTION | locating a new foresight point by deflection angle and distance |
| TANGENT OFFSET | locating a new point at the intersection of a line defined by two points and a perpendicular offset from a third point |
| PARALLEL LINE | locating two new points defining a line offset parallel to a line defined by two known points |
| EXTEND ARC | locating a new point at a given arc distance from a known point |
| ANGLE RESECTION | locating a new point by angular resection (three known points used) |
| TANGENT | locating the points of tangency for lines tangent to two circles |
| LEVEL RUN | calculating the instrument elevation and point elevations |

### 8.2.1 LOCATE LINE

The LOCATE LINE command locates point *n* at a horizontal distance *dis* from point *p1*, in a direction toward point *p2*. A negative distance reverses the above direction. You can enter a slope distance instead of a horizontal distance, if you follow the distance by the vertical angle *va*. You also have the option to specify an offset *off*.

If you know the elevation of point *p1* and specify the vertical angle, ICS will use the heights of the active instrument and foresight target from SET HEIGHT to compute the elevation of point *n*. You can specify an optional offset. If you specify *Tdis*, ICS will scale the distance *dis* by the ratio distance from *p1* to *p2*, divided by *Tdis* (use this for prorating a database distance given the corresponding measured or record distance).

<u>Setup</u>

If you intend to use elevations, use the SET HEIGHT command to establish the instrument and target heights.

<u>Operator Sequence</u>

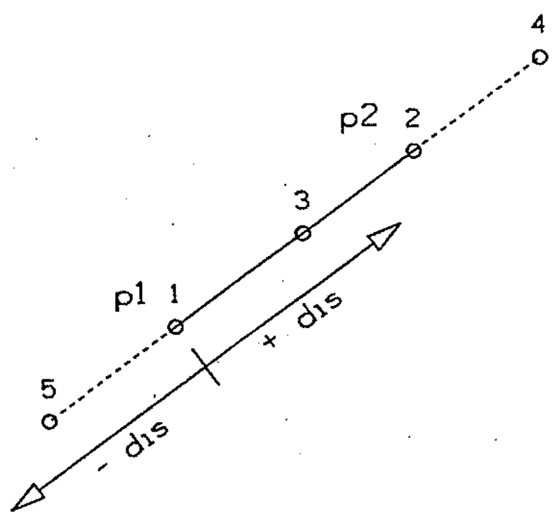
1. Select or key in LOCATE LINE.

   *The system responds:*

   LOCATE LINE p1 p2 n dis /va /off /Tdis

2. Identify the following:

| | |
|---|---|
| p1 | the known point from which the distance is measured. |
| p2 | the known point that indicates the direction of the line. |
| n | the ID of the point to be defined. |
| dis | the horizontal/slope distance to point *n*. It can be expressed using the D delimiter. |
| va | vertical angle. Skip this field by keying in 90. if you are using zeniths. |
| off | offset. Positive to the right (from *p1* to *p2*); negative to the left. |
| Tdis | measured terrain distance between *p1* and *p2*. |




REPORT FILE LISTING:

```
LOCATE LINE  p1 p2 n dis /va /off /Mdis
   1 2 3 75
         3     2736.1818     1228.3354                DEFAULT
LOCATE LINE  p1 p2 n dis /va /off /Mdis
   1 2 4 (D1 2+75)
         4     2817.1368     1343.7004                DEFAULT
LOCATE LINE  p1 p2 n dis /va /off /Mdis
   1 2 5 -75
         5     2650.0200     1105.5504                DEFAULT
```

*Figure 8-1. LOCATE LINE*

## 8.2.2 LOCATE DIRECTION

LOCATE DIRECTION locates point *n* from point *p* at a direction *dir* and horizontal distance *dis*. You can enter a slope distance instead of a horizontal distance, if the distance is followed by the vertical angle *va*. You also have the option to specify an offset *off*.

If the elevation of point *p* is known and you specify the vertical angle, ICS computes the elevation of point *n* using both active instrument height and foresight target height. You set the active instrument height using the SET HEIGHT command. ICS computes the foresight elevation using the vertical angle and slope distance *dis* to point *n*.

Setup

If you intend to use elevations, use the SET HEIGHT command to establish the instrument and target heights.

Operator Sequence

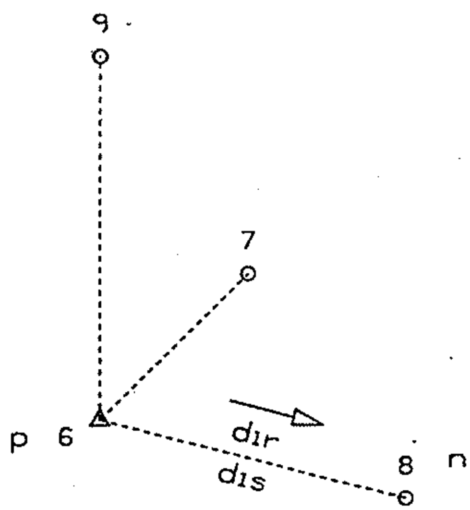
1. Select or key in LOCATE DIRECTION.

   *The system responds:*

   LOCATE DIRECTION p n dir dis /va /off

2. Identify the following:

| | |
|---|---|
| p | the ID of the point from which the measurements are taken. |
| n | the ID of the point to be defined. |
| dir | direction of the line. It can be expressed as an azimuth or bearing, or can be calculated using the A delimiter. |
| dis | the horizontal/slope distance. It can be expressed using the D delimiter. |
| va | vertical angle. Skip this field by keying in 90. if you are using zeniths. |
| off | offset. Positive to the right; negative to the left. |




REPORT FILE LISTING:

```
STORE  n c1 c2 /z /p  or  p c1 c2 /z /-cno
   6 2702.6599 1711.2489 0.0000
         6      2702.6599     1711.2489        0.0000 DEFAULT
LOCATE DIRECTION  p n dir dis /va /off
   6 7 45. 100
         7      2773.3706     1781.9596               DEFAULT
LOCATE DIRECTION  p n dir dis /va /off
   6 8 S75-26-36.6E 150
         8      2664.9597     1856.4339               DEFAULT
LOCATE DIRECTION  p n dir dis /va /off
   6 9 (A6 7 -45.) 175
         9      2877.6599     1711.2489               DEFAULT
```

*Figure 8-2. LOCATE DIRECTION*

### 8.2.3 LOCATE ANGLE

LOCATE ANGLE locates the point *nFS* at a horizontal distance *dis* and angle *ang* from point *pOC* with a backsight to point *pBS*. The angle can be positive for clockwise (to the right) or negative for counterclockwise (to the left). You can enter a slope distance instead of a horizontal distance if you follow the distance by the vertical angle *vaFS*. You also have the option to specify an offset *off*.

If the elevation of *pOC* is known, ICS uses the height of the active instrument and the foresight target height (elevation) to compute the elevation of point *nFS*. The foresight elevation is computed using the vertical angle and slope distance. If a vertical angle is measured to the backsight, and the elevation of *pBS* is known, then LOCATE ANGLE either computes the elevation of *pOC* (if it is unknown) or updates the current value.

Additionally, you have the option to use the parameters *vaBS* and *pVB*. The point of vertical backsight *pVB* indicates where the vertical angle of the backsight *vaBS* measurement is taken. If *pVB* is the same as the horizontal backsight point *pBS*, you key in the same point ID for both data fields. The *pVB* should be defined in x, y, and z.

Setup

If you intend to use elevations, use the SET HEIGHT command to establish the instrument and target heights.

Operator Sequence

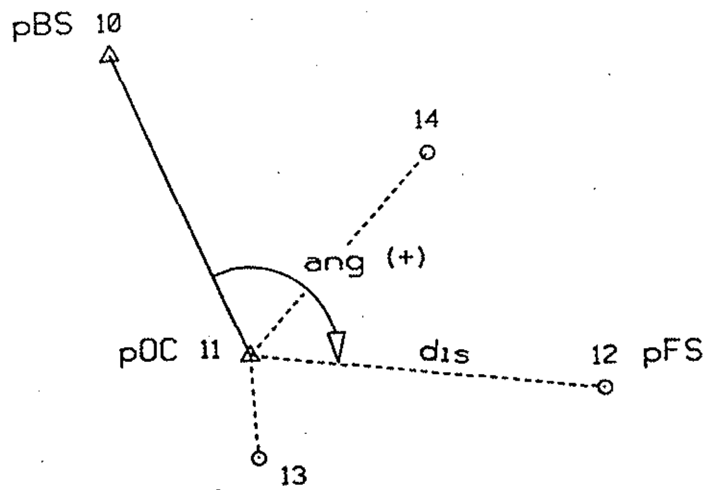
1. Select or key in LOCATE ANGLE.

   *The system responds:*

   LOCATE ANGLE pBS pOC nFS ang dis /vaFS /off /vaBS pVB

2. Identify the following:

| | |
|---|---|
| pBS | the known point used as the backsight. |
| pOC | the known point from which the measurements are taken. |
| nFS | the ID of the point to be defined. |
| ang | the horizontal angle. Positive to the right; negative to the left. It can be expressed in the standard angle format, or can be calculated using the G delimiter. |
| dis | the horizontal/slope distance to point *nFS*. It can be expressed using the D delimiter. |
| vaFS | the vertical angle to the foresight point. Skip this field by keying in 90. if you are using zeniths. |
| off | offset. Positive to the right; negative to the left. |
| vaBS | the vertical angle to the backsight point. Skip this field by keying in 90. if you are using zeniths. |
| pVB | the known point of vertical backsight, where the vertical angle to the backsight *vaBS* is measured. |




REPORT FILE LISTING:

```
LOCATE ANGLE  pBS pOC nFS ang dis /vaFS /off /vaBS pVB
   10 11 12 120. 170
        12      2225.6764      1371.2959               DEFAULT
LOCATE ANGLE  pBS pOC nFS ang dis /vaFS /off /vaBS pVB
   10 11 13 -160. 50
        13      2194.9562      1205.4519               DEFAULT
LOCATE ANGLE  pBS pOC nFS ang dis /vaFS /off /vaBS pVB
   10 11 14 (55.+10-30-30) 130
        14      2341.4885      1289.3492               DEFAULT
```

*Figure 8-3. LOCATE ANGLE*

## 8.2.4 LOCATE DEFLECTION

The LOCATE DEFLECTION command locates *nFS* at distance *dis* and deflection angle *defl* from the extension of the line from *pBS* through *pOC*. The deflection can be positive for clockwise (to the right) or negative for counterclockwise (to the left). You can enter a slope distance instead of a horizontal distance if you follow the distance by the vertical angle *vaFS*. You also have the option to specify an offset *off*.

If the elevation of *pOC* is known, ICS uses the height of the active instrument and the foresight target height (elevation) to compute the elevation of point *nFS*. The foresight elevation is computed using the vertical angle and slope distance. If a vertical angle is measured to the backsight, and the elevation of *pBS* is known, then LOCATE DEFLECTION will compute the elevation of *pOC* (if it is unknown) or update the current value if present in the database.

Additionally, you have the option to use the parameters *vaBS* and *pVB*. The point of vertical backsight *pVB* indicates where the vertical angle of the backsight *vaBS* measurement is taken. If *pVB* is the same as the horizontal backsight point *pBS*, you key in the same point ID for both data fields. The *pVB* should be defined in x, y, and z.

Setup

If you intend to use elevations, use the SET HEIGHT command to establish the instrument and target heights.

Operator Sequence

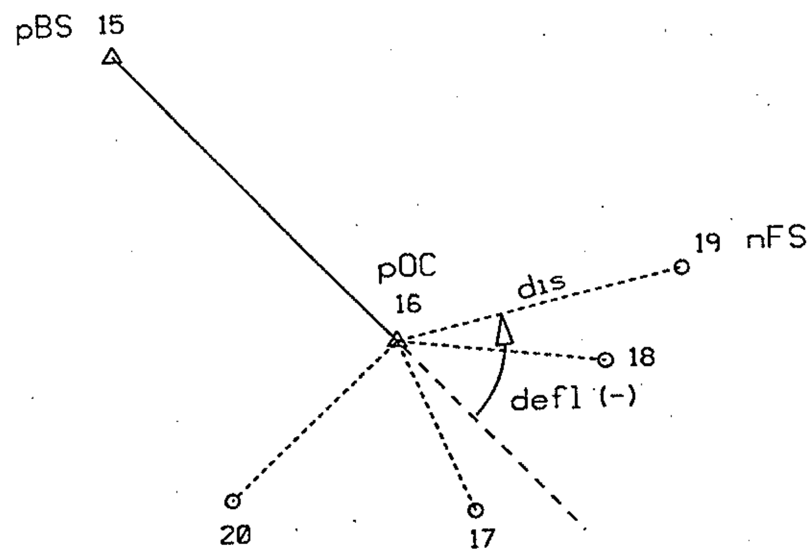
1. Select or key in LOCATE DEFLECTION.

   *The system responds:*

   LOCATE DEFLECTION pBS pOC nFS defl dis /vaFS /off /vaBS pVB

2. Identify the following:

| | |
|---|---|
| pBS | the known backsight point on the line. |
| pOC | the known occupied point on the line. |
| nFS | the ID of the point to be defined. |
| defl | the deflection angle. Positive to the right; negative to the left. It can be expressed in the standard angle format, or can be calculated using the G delimiter. |
| dis | the horizontal/slope distance to point *nFS*. It can be expressed using the D delimiter. |
| vaFS | the vertical angle to the foresight point. Skip this field by keying in 90. if you are using zeniths. |
| off | offset. Positive to the right; negative to the left. |
| vaBS | the vertical angle to the backsight point. Skip this field by keying in 90. if you are using zeniths. |
| pVB | the known point of vertical backsight, where the vertical angle to the backsight *vaBS* is measured. |




REPORT FILE LISTING:

```
LOCATE DEFLECTION  pBS pOC nFS defl dis /vaFS /off /vaBS pVB
   15 16 17 +20. 90
        17     2173.9007     1774.3002                DEFAULT
LOCATE DEFLECTION  pBS pOC nFS defl dis /vaFS /off /vaBS pVB
   15 16 18 -40. 100
        18     2244.6994     1838.2341                DEFAULT
LOCATE DEFLECTION  pBS pOC nFS defl dis /vaFS /off /vaBS pVB
   15 16 19 -G18 16 17 140.
        19     2288.4607     1875.2955                DEFAULT
LOCATE DEFLECTION  pBS pOC nFS defl dis /vaFS /off /vaBS pVB
   15 16 20 90. 110
        20     2181.4731     1658.6686                DEFAULT
```

*Figure 8-4. LOCATE DEFLECTION*

## 8.2.5 TANGENT OFFSET

The TANGENT OFFSET command finds the points of intersection between the line connecting points *p1* and *p2*, and the perpendicular offsets to points in the description *des*. The intersection points are stored in the database as point numbers *n*, *n*+1, *n*+2, etc.

Operator Sequence

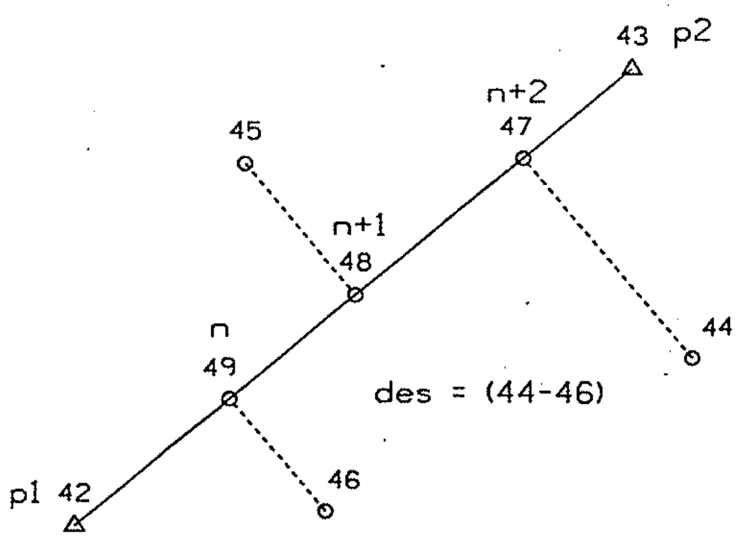
1. Select or key in TANGENT OFFSET.

   *The system responds:*

   TANGENT OFFSET n des p1 p2

2. Identify the following:

| | |
|---|---|
| n | the ID of the first intersection point to be defined. |
| des | description of the set of offset points to be projected perpendicular onto the line defined by *p1* and *p2*. |
| p1 | a known point on the line. |
| p2 | another known point on the line. |




```
TANGENT OFFSET  n des p1 p2
    0 (44-46) 42 43
         47      2823.2561      3309.1878              DEFAULT
    From         42 to          47        276.4319FT
    From         47 to          44        125.5878FT right
          48      2756.7665      3230.3689              DEFAULT
    From         42 to          48        173.3141FT
    From         48 to          45         82.4514FT left
          49      2706.2475      3170.4821              DEFAULT
    From         42 to          49         94.9649FT
    From         49 to          46         70.9545FT right
```

*Figure 8-5. TANGENT OFFSET*

### 8.2.6 PARALLEL LINE

The PARALLEL LINE command computes the end points of a line that is offset parallel a distance *off* from a line previously defined by points *p1* and *p2*. You have the option to specify *dz* to make the z value of each new point equal to the sum of the z of the original point and *dz*. If the z value of the original point is undefined, *dz* has no effect on the z value.

<u>Operator Sequence</u>

1. Select or key in PARALLEL LINE.

   *The system responds:*

   PARALLEL LINE p1 p2 off /n1 n2 /dz

2. Identify the following:

   p1 — the known starting point of the line.

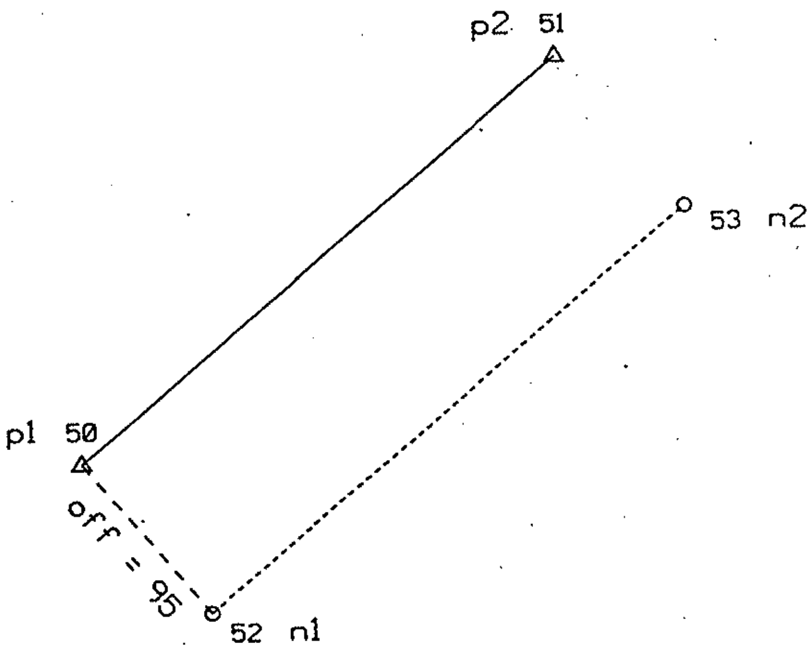
   p2 — the known end point of the line.

   off — the offset for the new line. Positive to the right (from *p1* to *p2*); negative to the left.

   n1 — the ID of the starting point to be defined.

   n2 — the ID of the end point to be defined.

   dz — the value to be added to the original z value.




REPORT FILE LISTING:

```
PARALLEL LINE  p1 p2 off /n1 /n2 /deltaz
    50 51 95
         52      2643.4715      3643.7894               DEFAULT
         53      2837.3396      3872.1713               DEFAULT
```

*Figure 8-6. PARALLEL LINE*

---

### 8.2.7 EXTEND ARC

The EXTEND ARC command locates the coordinates of point *nPT*. Point *nPT* is located at an arc distance of *arc* measured from *pPC* with the radius of the arc equal to the distance from *pCC* to *pPC*. A positive value for the distance locates *nPT* in a clockwise direction; a negative value locates *nPT* in a counterclockwise direction.

Operator Sequence

1. Select or key in EXTEND ARC.

   *The system responds:*

   EXTEND ARC pPC pCC nPT arc

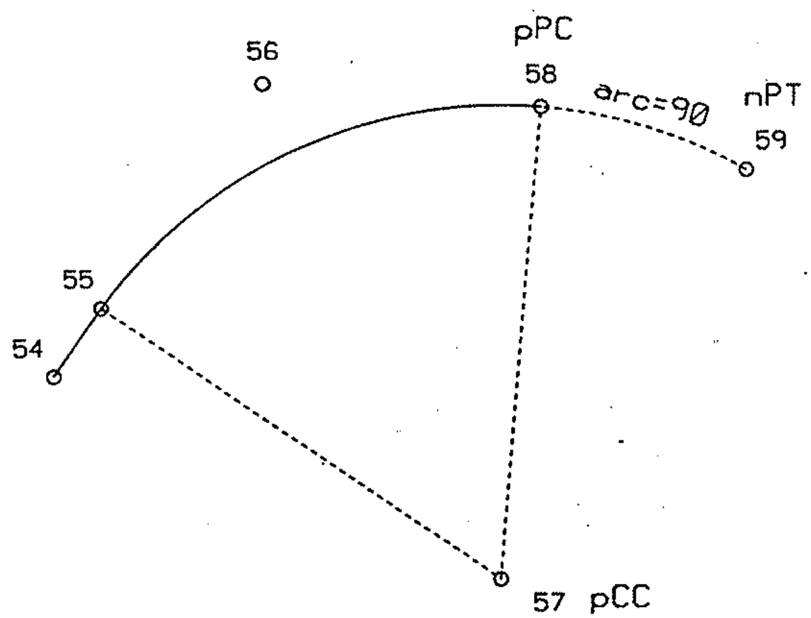
2. Identify the following:

   pPC the known point from which the measurements are taken.

   pCC the known point that is the center of the arc.

   nPT the ID of the point to be defined.

   arc the distance along the arc from *pPC* to *nPT*. Positive for clockwise (to the right); negative for counterclockwise (to the left).




REPORT FILE LISTING:

```
EXTEND ARC    pPC pCC nPT arc
    58 57 59 90.
         59      2334.8393      3371.7651                   DEFAULT
```

*Figure 8-7. EXTEND ARC*

### 8.2.8 ANGLE RESECTION

The ANGLE RESECTION command is used to locate an occupied position based on three known points and the angles between them. The command locates a new point *n*, given three known points (*p1, p2, p3*), the angle *ang1* measured from the line connecting *n* and *p1* to the line connecting *n* and *p2*, and the angle *ang2* measured from the line connecting *n* and *p2* to the line connecting *n* and *p3*. See Figure 8-8.

For fairly accurate resection solutions, both angles should be between 60 and 120 degrees. An optimal solution is found if the angles equal 120 degrees. ICS computes solutions for all cases except where all four points (*p1, p2, p3, n*) fall on the edge of the same circle. If the new point *n* is not inside the triangle whose vertices are *p1, p2, and p3* [(*ang1* + *ang2*) < 180 deg], then the known points (*p1, p2, p3*) should be listed in a clockwise order with respect to the new point *n*.

Operator Sequence

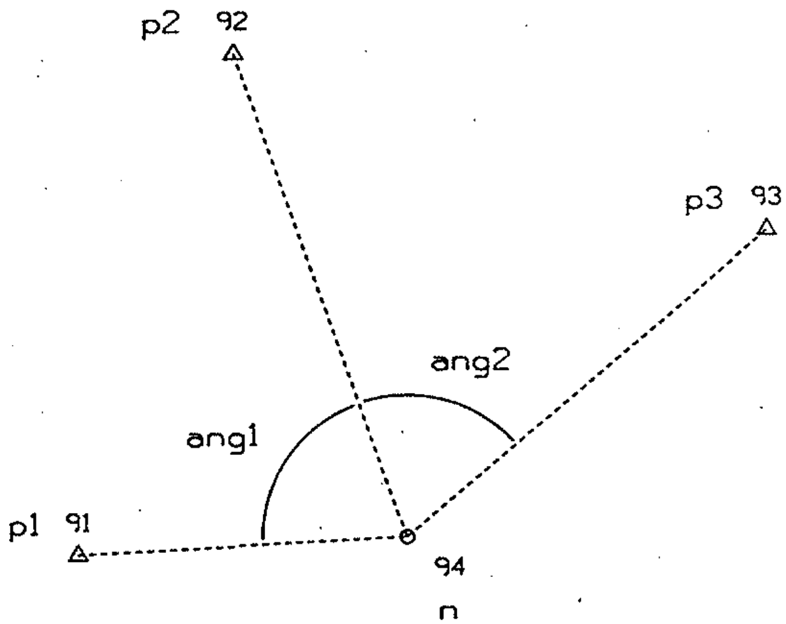
1. Select or key in ANGLE RESECTION.

   *The system responds:*

   ANGLE RESECTION p1 p2 p3 ang1 ang2 /n

2. Identify the following:

| | |
|---|---|
| p1 | the first known point. |
| p2 | the second known point. |
| p3 | the third known point. |
| ang1 | the angle measured between the line connecting *n* and *p1* and the line connecting *n* and *p2*. It can be expressed in the standard angle format, or can be calculated using the G delimiter. |
| ang2 | the angle measured between the line connecting *n* and *p2* and the line connecting *n* and *p3*. It can be expressed in the standard angle format, or can be calculated using the G delimiter. |
| n | the ID of the point to be defined by the angle resection. |




REPORT FILE LISTING:

```
ANGLE RESECTION  p1 p2 p3 ang1 ang2 /n
    91 92 93 73-38-40 68-43-10
         94      2197.1571      3773.1552                    DEFAULT
```

*Figure 8-8. ANGLE RESECTION*

### 8.2.9 TANGENT

The TANGENT command locates the tangent to two circles. This command finds the point of tangency *n1* on a circle with center *pCC1* and radius *r1*, and the point of tangency *n2* on a circle with center *pCC2* and radius *r2*. You can use one of two methods to locate the tangent with this command.

<u>Method 1: Using the Optional Parameters</u>

If you invoke the optional *sign* and *ext* variables, you must enter the larger circle first. If the tangent leaves the first (or larger) circle in a clockwise direction, as shown in Figure 8-9, the value of *sign* should be 1; otherwise, it should be -1. If the tangent is external, the value of *ext* should be 1. For a cross or internal tangent, *ext* should be -1.

<u>Method 2: Not Using the Optional Parameters</u>

If you do not invoke the *ext* and *sign* variables, you can enter the arcs in any order. In this case you describe the desired tangent by looking from the center of the first circle you enter toward the center of the second circle you enter. If the point of tangency is on the left, you key in the radius of the circle as a negative value. Otherwise, key in the radius of the circle as a positive value.

<u>Operator Sequence</u>

1. Select or key in TANGENT.

   *The system responds:*

   TANGENT n1 pCC1 r1 n2 pCC2 r2 /sign /ext

2. Identify the following:

   n1 the ID of the point that is to be defined as the point of tangency on the first circle.

   Note: For Method 1 the first circle must be the larger circle.

pCC1 the known point that is the center of the first circle.

r1 the radius of the first circle. It can be expressed using the D delimiter.

n2 the ID of the point that is to be defined as the point of tangency on the second circle.

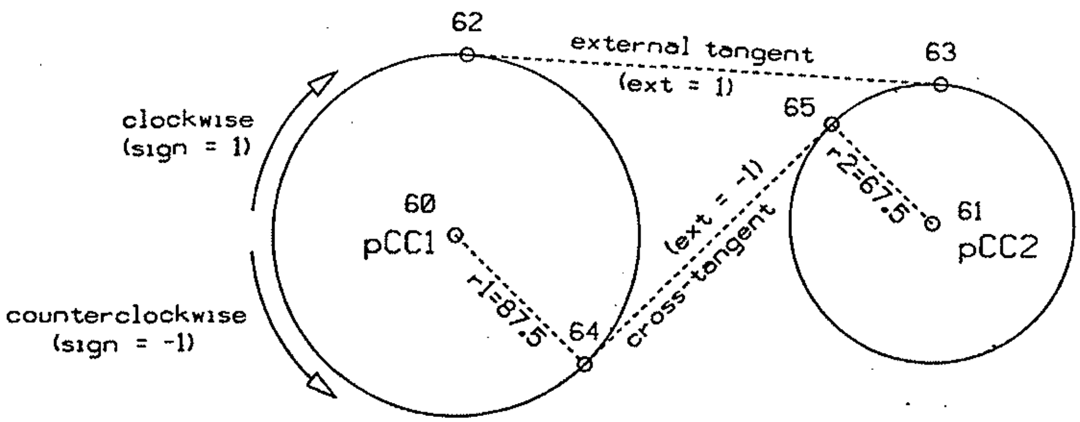
Note: For Method 1 the second circle must be the smaller circle.

pCC2 the known point that is the center of the second circle.

r2 the radius of the second circle. It can be expressed using the D delimiter.

sign *for Method 1 only*: the direction the tangent leaves the large circle. 1 for clockwise (to the right); -1 for counterclockwise (to the left).

ext *for Method 1 only*: 1 for external tangent; -1 for cross or internal tangent.




REPORT FILE LISTING:

```
TANGENT  n1 pCC1 r1 n2 pCC2 r2 /dir /ext
   0 60 87.5 0 61 67.5 1 1
         62      2868.6793      4149.5994              DEFAULT
         63      2855.3235      4374.7038              DEFAULT
  From           62 to          63   93-23-43.7     225.5003FT
TANGENT  n1 pCC1 r1 n2 pCC2 r2 /dir /ext
   0 60 87.5 0 61 67.5 -1 -1
         64      2719.3346      4206.1622              DEFAULT
         65      2835.7692      4323.0740              DEFAULT
  From           64 to          65   45- 7- 1.9     165.0012FT
```

*Figure 8-9. TANGENT*

### 8.2.10 LEVEL RUN

The LEVEL RUN command calculates the instrument elevation and foresight (or target) point elevations. The instrument elevation is the sum of the backsight elevation and the rod shot of the backsight, and it is useful for calculating further sideshots. The elevation of the foresight is the instrument elevation minus the rod shot of the foresight.

The instrument elevation is needed for the LOCATE FROM ALIGNMENT command. Therefore, if you need to determine the instrument height, run the LEVEL RUN command before LOCATE FROM ALIGNMENT. (You can enter the elevation directly with the LOCATE FROM ALIGNMENT command.) To calculate the instrument height you must include the rod shot of the backsight *rBS*. The optional parameters *nFS* and *rFS* are not needed if you only want to calculate the instrument height.

To calculate a new foresight point, include the optional parameters *nFS* and *rFS* when you run LEVEL RUN. If foresight point *nFS* is already defined, only the z coordinate changes; the x and y coordinates stay the same. If foresight point *nFS* is previously undefined, the x and y coordinates of the point generated are set equal to (0,0).

<u>Operator Sequence</u>

1. Select or key in LEVEL RUN.

   *The system responds:*

   LEVEL RUN pBS rBS /nFS rFS

2. Identify the following:

   pBS — the known backsight point. The elevation of this point is used to compute the instrument elevation.

   rBS — the rod shot on the backsight.

   nFS — the ID of the new foresight point, or a previously defined point (pFS) in x,y.

   rFS — the rod shot of the foresight.

REPORT FILE LISTING:

```
 STORE 100 0 0 500.013
       100        0.0000         0.0000      500.0130 DEFAULT

 STORE 204 0 0 507.323
       204        0.0000         0.0000      507.3230 DEFAULT
 LEVEL RUN 100 5.68 101 3.91
%LEV-I-INELEV, instrument elevation is         505.6930
%LEV-I-ELEVFS, foresight elevation at point       101 is         501.7830
 LEVEL RUN 101 7.94 102 4.83
%LEV-I-INELEV, instrument elevation is         509.7230
%LEV-I-ELEVFS, foresight elevation at point       102 is         504.8930
 LEVEL RUN 102 11.35 103 4.12
%LEV-I-INELEV, instrument elevation is         516.2430
%LEV-I-ELEVFS, foresight elevation at point       103 is         512.1230
 LEVEL RUN 103 3.22 104 8.10
%LEV-I-INELEV, instrument elevation is         515.3430
%LEV-I-ELEVFS, foresight elevation at point       104 is         507.2430
```

*Figure 8-10. LEVEL RUN*

## 8.3 DIVIDE Commands

Use the DIVIDE commands to establish new coordinate values relative to previously defined descriptions. These commands provide a simple and direct method for establishing multiple points. You can specify numbers to be assigned to the new values in the command line, or you can allow ICS to assign default values. DIVIDE commands include the following:

| *Command* | *For* |
|---|---|
| DIVIDE LINE | dividing a line into a specified number of segments of equal length |
| DIVIDE ARC | dividing an arc into a specified number of segments of equal length |
| DIVIDE FIGURE | dividing a figure into a specified number of segments of equal length |
| SUBDIVIDE SECTION | subdividing a section of land into sixteen parcels according to PLSS specifications |

---

### 8.3.1 DIVIDE LINE

The DIVIDE LINE command divides the straight-line distance between points *p1* and *p2* into a number of segments (*div*) of equal length. The point numbers for the intermediate points begin with *n* and increase by 1, with the last intermediate point given the number *n*+(*div*-2). If you omit *n*, ICS assigns the next available point ID's to the intermediate points.

Operator Sequence

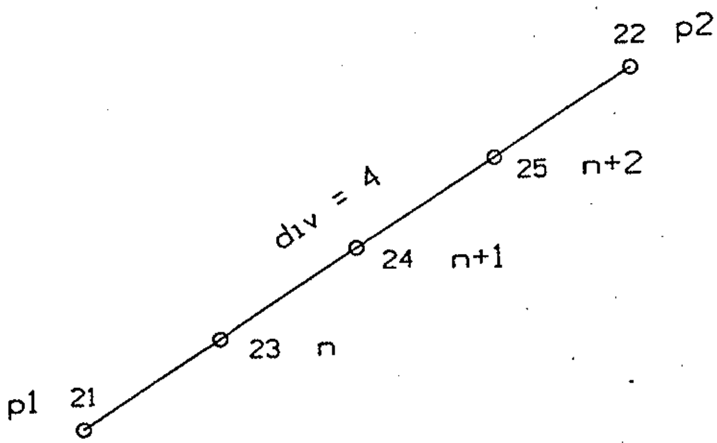
1. Select or key in DIVIDE LINE.

   *The system responds:*

   DIVIDE LINE p1 p2 div /n

2. Identify the following:

   | | |
   |---|---|
   | p1 | the starting point of the line to be divided. |
   | p2 | the end point of the line to be divided. |
   | div | the number of segments the line is to be divided into. |
   | n | the ID of the point to be placed at the end of the first section. |




REPORT FILE LISTING:

```
DIVIDE LINE  p1 p2 div /n
   21 22 4 0
        23      2722.9669      2171.9179                DEFAULT
        24      2765.1679      2237.9500                DEFAULT
        25      2807.3689      2303.9820                DEFAULT
```

*Figure 8-11. DIVIDE LINE*

## 8.3.2 DIVIDE ARC

The DIVIDE ARC command divides a clockwise arc with center *pCC* into a number o equal segmentsf (*div*) between *pPC* and *pPT*. The point ID's for the intermediate points begin with *n* and increase by 1, with the last intermediate point given the number *n*+(*div*-2). If you omit *n*, ICS assigns the next available point ID's to the intermediate points.

<u>Operator Sequence</u>

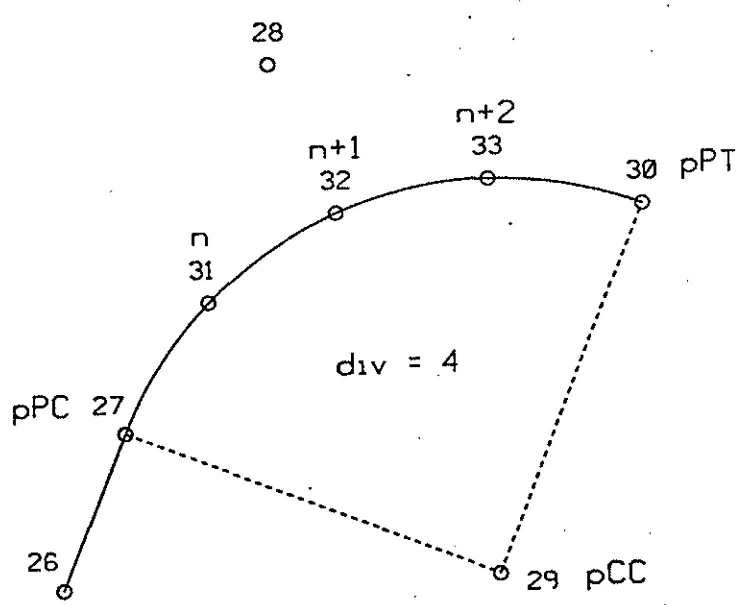
1. Select or key in DIVIDE ARC.

   *The system responds:*

   DIVIDE ARC pPC pPT pCC div /n

2. Identify the following:

| | |
|---|---|
| pPC | the arc's first known point (in a clockwise direction). |
| pPT | the arc's second known point (in a clockwise direction). |
| pCC | the known point that is the center of the arc. |
| div | the number of segments the arc is to be divided into. |
| n | the ID of the first intermediate point to be defined. |




REPORT FILE LISTING:

```
DIVIDE ARC  pPC pPT pCC div /n
    27 30 29 4 0
         31    2787.3635    2649.2876              DEFAULT
         32    2829.7954    2710.5462              DEFAULT
         33    2845.5547    2783.3798              DEFAULT
```

*Figure 8-12. DIVIDE ARC*

### 8.3.3 DIVIDE FIGURE

The DIVIDE FIGURE command divides the figure described by *des* into a number of equal segments (*div*). The point numbers for the intermediate points begin with *n* and increase by 1, with the last intermediate point given the number *n*+(*div*-2). If you omit *n*, ICS assigns the next available point ID's to the intermediate points.

Note: Defining the figure is not necessary.

<u>Operator Sequence</u>

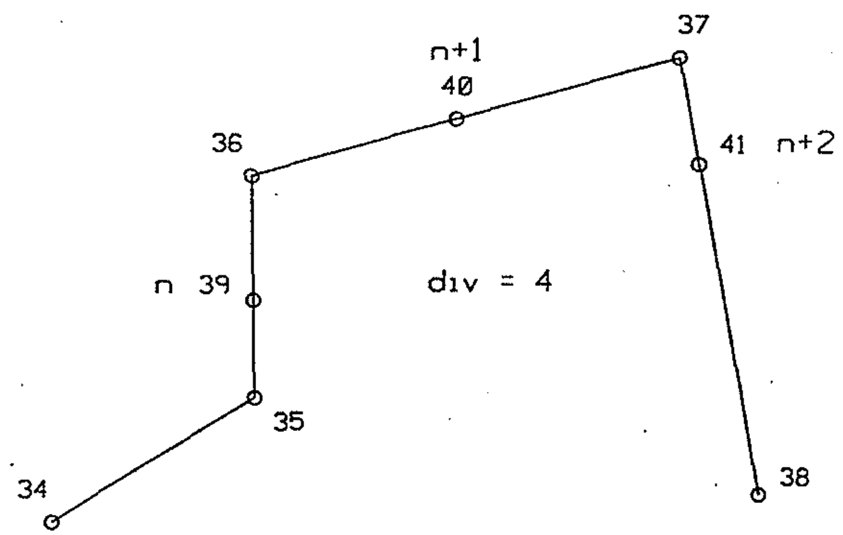
1. Select or key in DIVIDE FIGURE.

   *The system responds:*

   DIVIDE FIGURE des div /n

2. Identify the following:

   | | |
   |---|---|
   | des | the figure ID or a list of points to describe the figure to be divided. |
   | div | the number of segments to divide the figure into. |
   | n | the ID of the first intermediate point to be defined. |




REPORT FILE LISTING:

```
 DIVIDE FIGURE  des div /n
    1 4 0
      Point              Coordinates
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     Points on line          35          36
        39      2261.3043      2153.2290
     Points on line          36          37
        40      2349.0459      2249.9356
     Points on line          37          38
        41      2327.2610      2364.5213
```

*Figure 8-13. DIVIDE FIGURE*

## 8.3.4 SUBDIVIDE SECTION

The SUBDIVIDE SECTION command subdivides a section of land into 16 parcels according to requirements of the Public Land Survey System (PLSS). For legal requirements see the *Manual of Instructions for the Survey of Public Lands of the United States* published by the U.S. Bureau of Land Management (1973, Technical Bulletin No. 6, Government Printing Office, Washington, D.C.).

You enter the section number *nSec* (a value from 1 to 36) and a list of four section corner point numbers in any order in the description *des1*. The point numbers for the new points generated begin with *n* and increase by 1 with the last point generated given the number *n*+(16-2). If you omit *n*, the new points are assigned the next available point numbers in the database, incremented by 1.

Additionally, you have the option to include a list of quarter section points in any order in the description *des2*. You can enter any number of points up to four; quarter section points that you do not enter will be automatically computed.

ICS computes the points dividing the section into quarter-quarter section parcels as described below.

```
NW----2----N----4----NE
|          |          |
|          |          |
6    7     8    9    10
|          |          |
|          |          |
W----12----C----13----E
|          |          |
|          |          |
16   17    18   19   20
|          |          |
|          |          |
[illegible]           SE
```

NW, NE, SW, and SE are the section corners, which you enter in *des1*. N, E, S, and W are the quarter section points, which you can optionally enter in *des2*; otherwise, these points are computed. The other points are computed as follows:

- Points 6, 8, and 10
  Sections 1-6 set 20 chains North of W, C, and E.
  Sections 7-36 set equidistant on N-S 1/4 section line.

- Points 16, 18, 20, 4, 14, and 24
  Sections 1-36 set equidistant on N-S 1/4 section line.

- Points 2, 12, and 22
  Sections 6, 7, 18, 19, 30, and 31 set 20 chains West of points N, C, and S.
  Sections 1-5, 8-17, 20-29, and 32-36 set equidistant on E-W 1/4 section line.

- Points 7, 9, 17, and 19
  Sections 1-36 set by intersection of lines joining opposite quarter-quarter points.

Operator Sequence

1. Select or key in SUBDIVIDE SECTION.

   *The system responds:*

   SUBDIVIDE SECTION nSec des1 /n /des2

2. Identify the following:

   nSec — the section number in the township (in the range from 1 to 36).

   des1 — description of the four section corner point ID's listed in any order. *All four points are required.*

   n — the point ID for the first point generated.

   des2 — optional description of quarter section points. You can list any number of them up to four. Quarter sections points not entered will be computed.

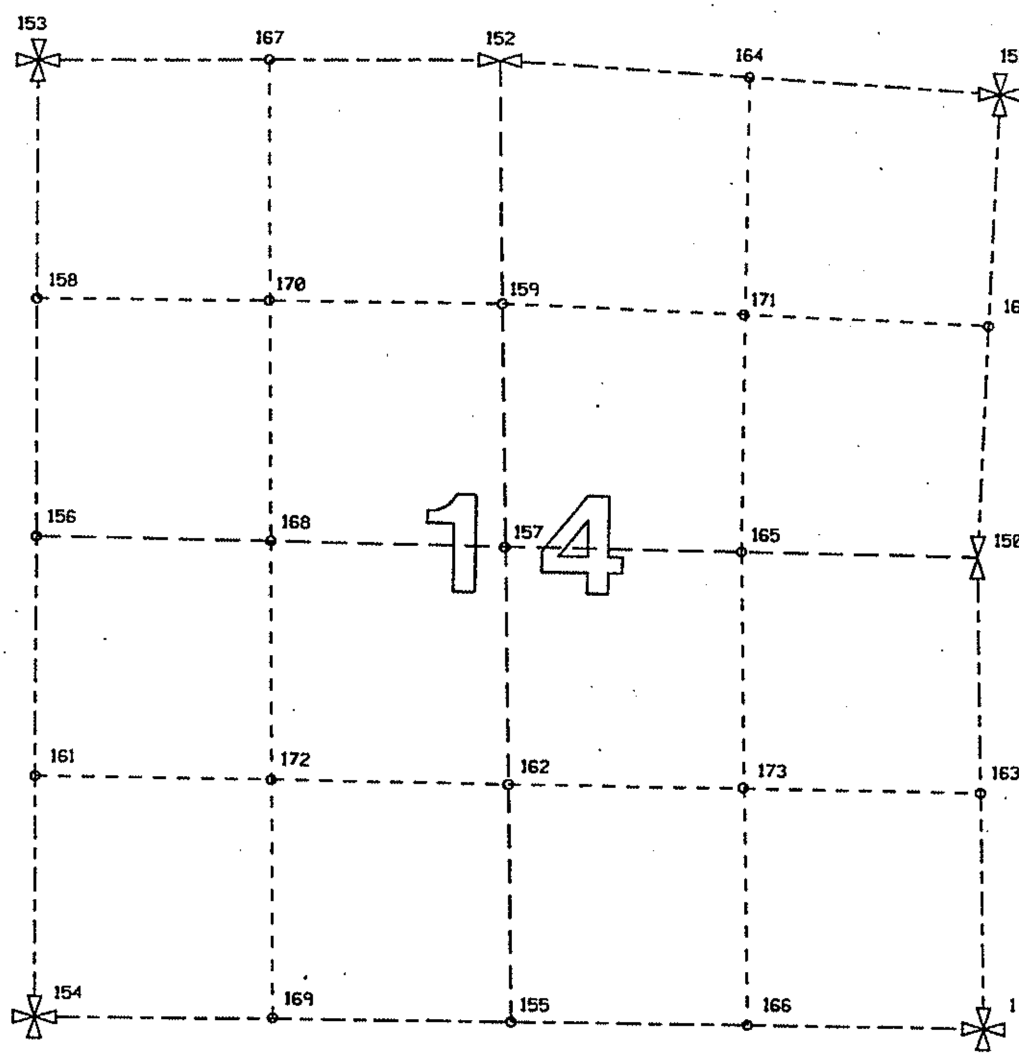



*Figure 8-14. SUBDIVIDE SECTION*

REPORT FILE LISTING:

```
SUBDIVIDE SECTION  nsect des1 /n /des2
   14 (1 151 153 154) 0 (150 152)
      153 NW CORNER        16973.6769      925.7149        0.0000
      167                  16996.8154     2182.9149
      152 N 1/4 CORNER     17019.9539     3440.1149        0.0000
      164                  16950.5379     4813.0079
      151 NE CORNER        16881.1219     6185.9009        0.0000
      158                  15655.7304      944.9832
      170                  15664.1905     2211.1258
      159                  15672.6506     3477.2683
      171                  15634.7890     4810.3741
      160                  15596.9274     6143.4799
      156 W 1/4 CORNER     14337.7839      964.2514
      168                  14331.5656     2239.3366
      157 CENTER           14325.3473     3514.4218
      165                  14319.0401     4807.7403
      150 E 1/4 CORNER     14312.7329     6101.0589        0.0000
      161                  13017.9250      983.5477
      172                  13016.7440     2267.0441
      162                  13015.5631     3550.5406
      173                  13014.3378     4843.1537
      163                  13013.1124     6135.7669
      154 SW CORNER        11698.0660     1002.8439        0.0000
      169                  11701.9225     2294.7516
      155 S 1/4 CORNER     11705.7790     3586.6594
      166                  11709.6354     4878.5672
        1 SE CORNER        11713.4919     6170.4749        0.0000
```

*Figure 8-14. SUBDIVIDE SECTION (Continued)*

## 8.4 INTERSECT Commands

The INTERSECT commands compute the intersection of lines, arcs (curves), figures, and spirals. Lines and arcs do *not* have to already exist to perform an intersection; only the parameters used to define them, such as the points, direction, radius, etc., must be known. Figures and spirals *must* be defined in the database to perform an intersection.

Using the Point *pID* for Selecting an Intersection Point

Whenever you execute an INTERSECT command, ICS calculates a single point of intersection. If more than one intersection is possible, ICS computes all possible intersections and the distance from each possible intersection to a point *pID* you specify. If *pID* is positive, ICS defines the closest intersection point to *pID* as the intersection; if *pID* is negative, ICS defines the farthest intersection point to *pID* as the intersection.

Offsetting the Point of Intersection

You can also specify a positive or negative offset(s) for the point of intersection. A positive offset is to the right in a forward direction; a negative offset is to the left. For example, if you defined a figure from *p1* to *p2*, a positive offset would be to the right of the line, looking from *p1* to *p2*, and a negative offset would be to the left. If you defined a figure from *p2* to *p1*, a positive offset would be to the right of the line, looking from *p2* to *p1*.

When a figure is offset, the lines making up the figure remain the same length except where no intersection would occur without an extension. In that case the appropriate line is extended until an intersection occurs. Likewise, when intersecting lines, a line is extended to create an intersection, and when intersecting an arc, an arc is extended.

Computing Elevations of Points

Some of the INTERSECT commands allow you the option to define elevations of points. These commands accept a vertical angle, an elevation value, or a rod reading to define the z coordinate. The Earth's curvature is accounted for when using vertical angles.

By default, these INTERSECT commands accept vertical angles for defining the z coordinate. If you want to enter an elevation or rod reading instead, you have two choices. One option is to change the default using the INPUT MODES command in the ENVIRONMENT menu block (see Section 7.3.12). This command allows you to change the default for INTERSECT commands to either elevation or rod reading.

The other option you have is to indicate which option you are using by prefixing the value. You prefix an elevation value with an upper or lowercase *E* and a rod reading with an upper or lowercase *L*. See Section 2.3.4 for more details.

Note: If you set the default to elevation or rod reading and want to input a vertical angle, prefix the vertical angle value with an upper or lowercase *A*.

To compute elevations (*z*) using a vertical angle or a rod reading, you must establish an instrument height (H.I.) before first. You can set the instrument height with the SET HEIGHT command, which is in the ENVIRONMENT menu block (see Section 7.3.3).

Note: Do not confuse the SET HEIGHT command with the LEVEL RUN command. The SET HEIGHT command sets a given instrument height and target heights, while the LEVEL RUN command calculates the instrument elevation and point elevations.

You have two options when using a vertical angle to compute elevations. If you set the vertical reference direction in the ICS.PAR parameter file to horizontal, you can enter angles measured from the horizon. If you set it to zenith, you can enter angles measured either from the zenith or the nadir.

Intersecting Spirals with Other Geometries

ICS contains three spiral intersection commands: LINE SPIRAL, CURVE SPIRAL, and SPIRAL SPIRAL. These commands use the most recently created spiral. If you have stored spirals within a figure, it is usually preferable to use the comparable figure intersection command.

Although ICS allows compound (combining) spirals to be stored within figures, the intersection of compound spirals with other geometries is not currently supported. This limitation can be overcome by recognizing that a compound spiral is a portion of a longer simple spiral. This simple spiral shares a common tangent and CS with the compound spiral and has a length $l_s$ of

$$l_s = (l_{cs} * dc2)/(dc2\text{-}dc1)$$

where *dc1* and *dc2* (*dc2* > *dc1*) are degrees of curvature at the endpoints of the compound spiral, and $l_{cs}$ is the length of the compound spiral.

Thus, intersections with compound spirals can be computed by creating a temporary figure having a SIMPLE SPIRAL OUT of length $l_s$ from the CS at the end of the short tangent of the compound spiral. You then intersect this figure with the other geometry.

INTERSECT commands include the following:

| *Command* | *For* |
|---|---|
| POINTS INTERSECT | locating the intersection of two lines, each defined by two points |
| DIRECTION INTERSECT | locating the intersection of two lines, each defined by a point and direction |
| POINTS DIRECTION INTERSECT | locating the intersection of two lines, one defined by two points and the other defined by a point and direction |
| ARC LINE POINTS | locating the intersection of an arc and a line that is defined by two points |
| ARC LINE DIRECTION | locating the intersection of an arc and a line that is defined by a point and direction |
| ARC ARC INTERSECT | locating the intersection of two arcs |
| FIGURE LINE INTERSECT | locating the intersection of a figure or list of points and a line defined by a point and a direction |
| FIGURE ARC INTERSECT | locating the intersection of a figure or list of points and an arc |
| FIGURE FIGURE INTERSECT | locating the intersection of two figures or lists of points |
| LINE SPIRAL | locating the intersection of a line and a spiral |
| CURVE SPIRAL | locating the intersection of an arc and a spiral |
| SPIRAL SPIRAL | locating the intersection of two spirals |

### 8.4.1 POINTS INTERSECT

The POINTS INTERSECT command locates the coordinates of point *n* defined by the intersection of the line through *p1* and *p2* with the line through *p3* and *p4*. If the angle is less than 6 degrees, ICS issues a warning, but it does perform the intersection. You can offset the point of intersection by distance *off1* from the first line and/or distance *off2* from the second line.

Operator Sequence

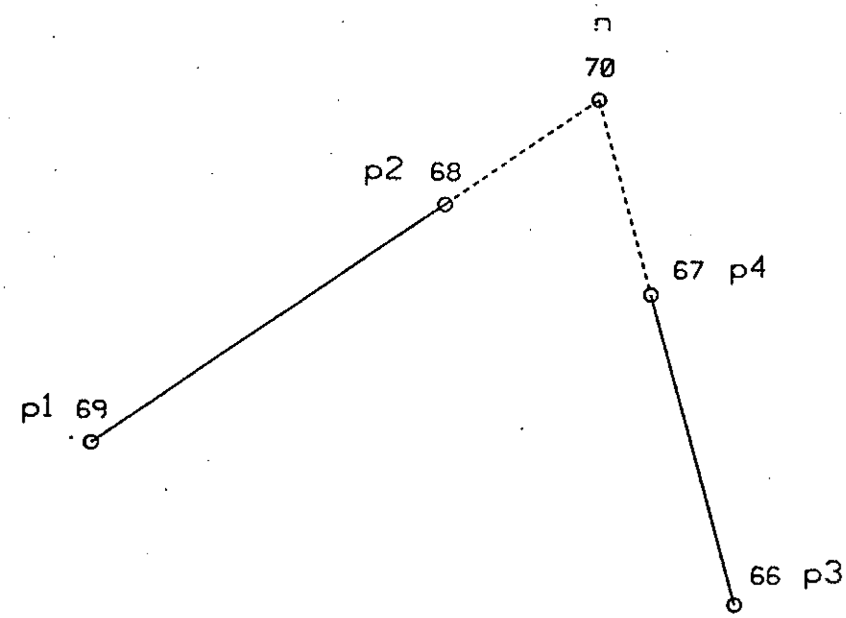
1. Select or key in POINTS INTERSECT.

   *The system responds:*

   POINTS INTERSECT n p1 p2 p3 p4 /off1 /off2

2. Identify the following:

   | | |
   |---|---|
   | n | the ID of the point to be defined by the intersection. |
   | p1 | a known point on the first line. |
   | p2 | another known point on the first line. |
   | p3 | a known point on the second line. |
   | p4 | another known point on the second line. |
   | off1 | the distance to offset the point of intersection from the first line. Positive to the right (from *p1* to *p2*); negative to the left. |
   | off2 | the distance to offset the point of intersection from the second line. Positive to the right (from *p3* to *p4*); negative to the left. |




REPORT FILE LISTING:

```
POINTS INTERSECT  n p1 p2 p3 p4 /off1 /off2
   0 69 68 66 67
        70      2876.4053      5318.5477               DEFAULT
```

*Figure 8-15. POINTS INTERSECT*

### 8.4.2 DIRECTION INTERSECT

The DIRECTION INTERSECT command locates the coordinates of point *n* defined by the intersection of two lines: one through point *p1* at direction *dir1* and the other line through point *p2* at direction *dir2*. If the angle is less than 6 degrees, ICS issues a warning, but it does perform the intersection. You can offset the point of intersection by distance *off1* from the first line or *off2* from the second line.

If you specify *va1* and *va2*, and define elevations of points *p1* and *p2*, ICS will first compute the intersection of point *n* in the xy plane. ICS will then use the vertical angles to compute the mean elevation of point *n*, using the heights of both active instruments 1 and 2, and foresight targets for both points *p1* and *p2*.

<u>Setup</u>

If you intend to use elevations, use the SET HEIGHT command to establish the instrument and target heights.

<u>Operator Sequence</u>

1. Select or key in DIRECTION INTERSECT.

   *The system responds:*

   DIRECTION INTERSECT n p1 dir1 p2 dir2 /off1 /off2 /va1 va2

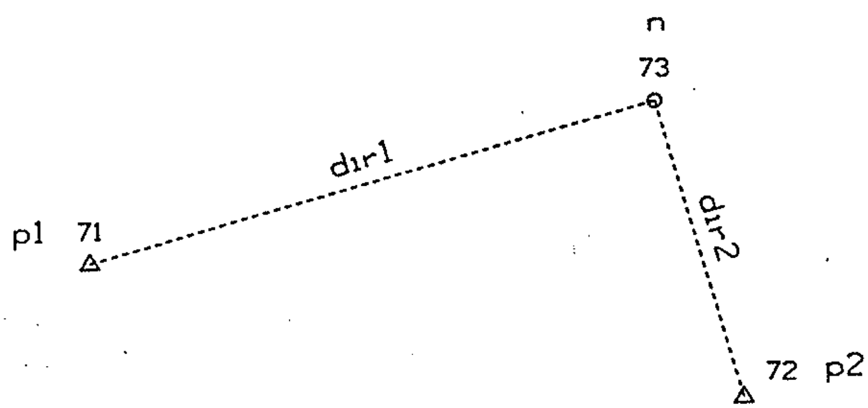
2. Identify the following:

| | |
|---|---|
| n | the ID of the point to be defined by the intersection. |
| p1 | a known point on the first line. |
| dir1 | the direction of the first line. It can be expressed as an azimuth or bearing, or can be calculated using the A delimiter. |
| p2 | a known point on the second line. |
| dir2 | the direction of the second line. It can be expressed as an azimuth or bearing, or can be calculated using the A delimiter. |
| off1 | the distance to offset the point of intersection from the first line. Positive to the right; negative to the left. |
| off2 | the distance to offset the point of intersection from the second line. Positive to the right; negative to the left. |

va1 the vertical angle from *p1* to point *n*. Skip this field by keying in 90. if you are using zeniths.

va2 the vertical angle from *p2* to point *n*. Skip this field by keying in 90. if you are using zeniths.

---




REPORT FILE LISTING:

```
DIRECTION INTERSECT  n p1 dir1 p2 dir2 /off1 /off2 /va1 va2
   0 71 74-20-15 72 N15-36-40W
        73      2303.7440      5340.9799                DEFAULT
```

*Figure 8-16. DIRECTION INTERSECT*

---

### 8.4.3 POINTS DIRECTION INTERSECT

The POINTS DIRECTION INTERSECT command locates the coordinates of point *n* defined by the intersection of a line through *p1* and *p2* with a line from *p3* at direction *dir*. If the angle is less than 6 degrees, ICS issues a warning, but it does perform the intersection. You can offset the point of intersection by distance *off1* from the first line and/or *off2* from the second line.

You have the option to specify the vertical angle *va* to compute the elevation of point *n*. ICS computes this elevation using the heights of the active instrument and foresight target for instrument 1 at point *p3*.

Setup

If you intend to use elevations, use the SET HEIGHT command to establish the instrument and target heights.

Operator Sequence

1. Select or key in POINTS DIRECTION INTERSECT.

   *The system responds:*

   POINTS DIRECTION INTERSECT n p1 p2 p3 dir /off1 /off2 /va

2. Identify the following:

   n — the ID of the point to be defined by the intersection.

   p1 — a known point on the first line.

   p2 — another known point on the first line.

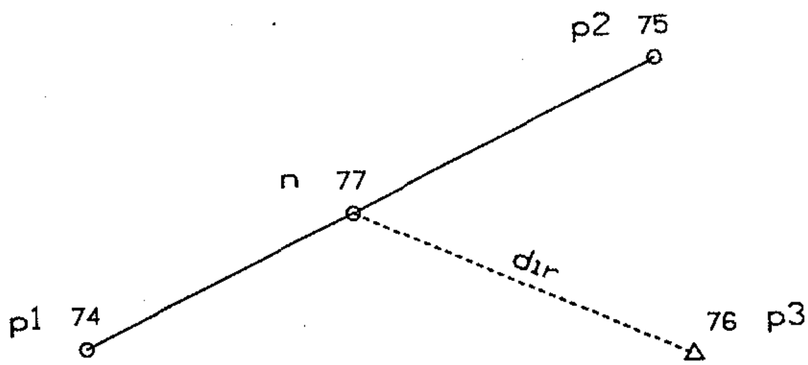
   p3 — a known point on the second line.

   dir — the direction of the second line. It can be expressed as an azimuth or bearing, or can be calculated using the A delimiter.

   off1 — the distance to offset the point of intersection from the first line. Positive to the right (from *p1* to *p2*); negative to the left.

   off2 — the distance to offset the point of intersection from the second line. Positive to the right; negative to the left.

   va — the vertical angle from *p3* to point *n*. Skip this field by keying in 90. if you are using zeniths.




REPORT FILE LISTING:

```
POINTS DIRECTION INTERSECT  n p1 p2 p3 dir /off1 /off2 /va
   0 74 75 76 -65-20-30
        77      2759.3310      5715.9785                DEFAULT
```

*Figure 8-17. POINTS DIRECTION INTERSECT*

## 8.4.4 ARC LINE POINTS

The ARC LINE POINTS command locates the coordinates of point *n* defined by the intersection of a circle centered at point *pCC* with radius *r*, and a line through *p1* and *p2*. You can offset the point of intersection from the line by the distance *off*.

Operator Sequence

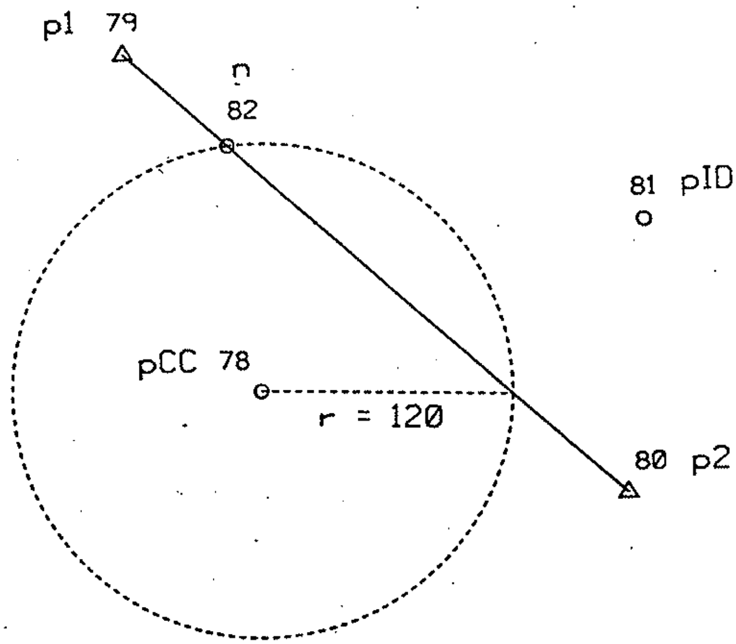
1. Select or key in ARC LINE POINTS.

   *The system responds:*

   ARC LINE POINTS n pCC r p1 p2 pID /off

2. Identify the following:

| | |
|---|---|
| n | the ID of the point to be defined by the intersection. |
| pCC | the known point that is the center of the arc. |
| r | the radius of the arc. It can be expressed using the D delimiter. |
| p1 | a known point on the line. |
| p2 | another known point on the line. |
| pID | a known point that indicates the intersection to be selected. Positive near to *pID*; negative far from *pID*. |
| off | the distance to offset the point of intersection from the line. Positive to the right; negative to the left. |




REPORT FILE LISTING:

```
ARC LINE POINTS  n pCC r p1 p2 pID /off
   0 78 120.0 79 80 -81
        82     2856.3894     6206.1082              DEFAULT
```

*Figure 8-18. ARC LINE POINTS*

---

### 8.4.5 ARC LINE DIRECTION

The ARC LINE DIRECTION command locates the coordinates of point *n* defined by the intersection of a circle centered at pCC with radius *r* and a line through point *p* at a direction *dir*. You can offset the point of intersection from the line by the distance *off*.

Operator Sequence

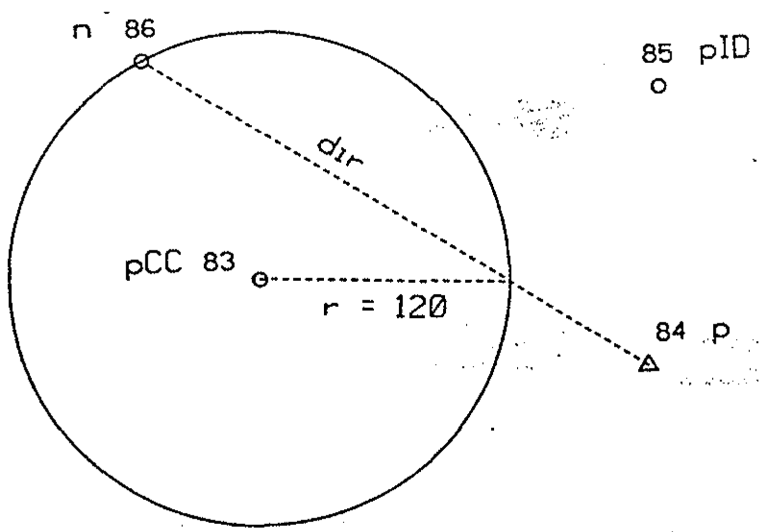
1. Select or key in ARC LINE DIRECTION.

   *The system responds:*

   ARC LINE DIRECTION n pCC r p dir pID /off

2. Identify the following:

| | |
|---|---|
| n | the ID of the point to be defined by the intersection. |
| pCC | the known point that is the center of the arc. |
| r | the radius of the arc. It can be expressed using the D delimiter. |
| p | a known point on the line. |
| dir | the direction of the line. It can be expressed as an azimuth or bearing, or can be calculated using the A delimiter. |
| pID | a known point that indicates the intersection to be selected. Positive near to *pID*; negative far from *pID*. |
| off | the distance to offset the point of intersection from the line. Positive to the right; negative to the left. |




REPORT FILE LISTING:

```
ARC LINE DIRECTION  n pCC r p dir pID /off
   0 83 120.0 84 N56-30-27W -85
        86      2365.0413      6161.2429                DEFAULT
```

*Figure 8-19. ARC LINE DIRECTION*

### 8.4.6 ARC ARC INTERSECT

The ARC ARC INTERSECT command locates the coordinates of point *n* defined by the intersection of the arc centered at *pCC1* with radius *r1* and the arc centered at *pCC2* with radius *r2*. You can specify the arcs in any order.

Operator Sequence

1. Select or key in ARC ARC INTERSECT.

   *The system responds:*

   ARC ARC INTERSECT n pCC1 r1 pCC2 r2 pID

2. Identify the following:

   n — the ID of the point to be defined by the intersection.

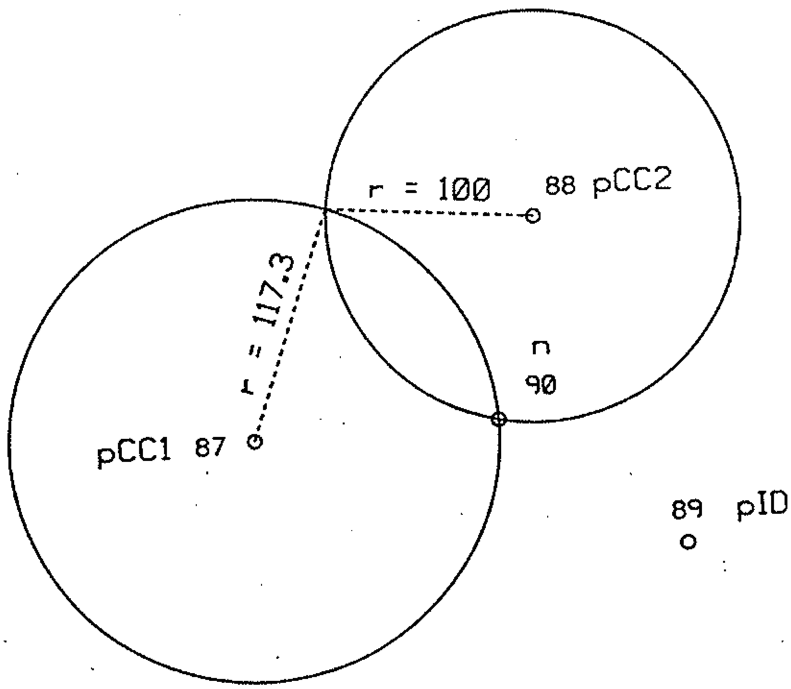
   pCC1 — the known point that is the center of the first arc.

   r1 — the radius of the first arc. It can be expressed using the D delimiter.

   pCC2 — the known point that is the center of the second arc.

   r2 — the radius of the second arc. It can be expressed using the D delimiter.

   pID — a known point that indicates the intersection to be selected. Positive near to *pID*; negative far from *pID*.




REPORT FILE LISTING:

```
ARC ARC INTERSECT  n pCC1 r1 pCC2 r2 pID
   0 87 117.3 88 100 89
        90      2744.3537      6796.4927                DEFAULT
```

*Figure 8-20. ARC ARC INTERSECT*

### 8.4.7 FIGURE LINE INTERSECT

The FIGURE LINE INTERSECT command locates the coordinates of point *n* defined by the intersection of figure *fg* with a line through point *p* at direction *dir*. You have the option to offset the point of intersection from the figure by distance *offg* and offset the point of intersection from the line by distance *off*.

Note: FIGURE LINE INTERSECT extends lines to infinity when no intersection would result otherwise. If you are not careful, unexpected intersections may result.

Operator Sequence

1. Select or key in FIGURE LINE INTERSECT.

   *The system responds:*

   FIGURE LINE INTERSECT n fg p dir pID /offg /off

2. Identify the following:

   n — the ID of the point to be defined by the intersection.

   fg — the figure ID. Note: You can also enter a *des* here.

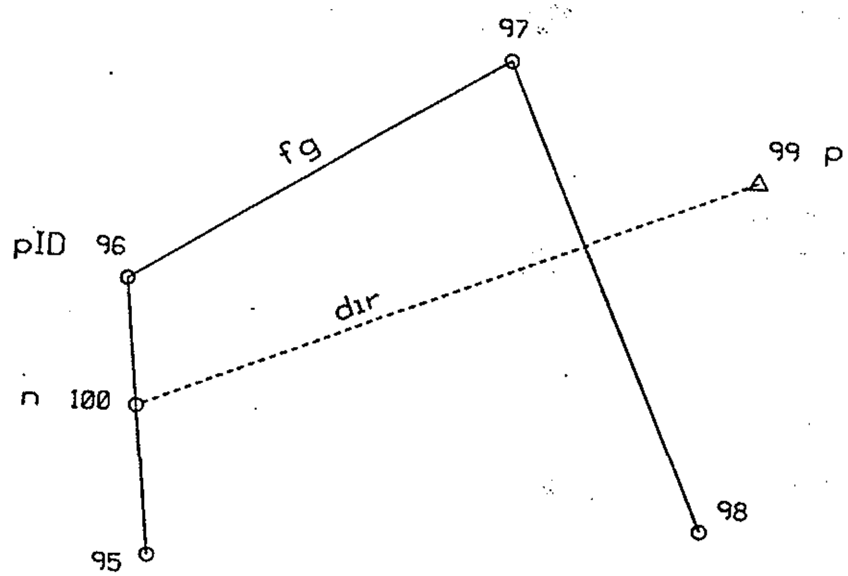
   p — the known point on the line.

   dir — the direction of the line. It can be expressed as an azimuth or bearing, or can be calculated using the A delimiter.

   pID — a known point that indicates the intersection to be selected. Positive near to *pID*; negative far from *pID*.

   offg — the distance to offset the point of intersection from the figure. Positive to the right; negative to the left.

   off — the distance to offset the point of intersection from the line. Positive to the right; negative to the left.




REPORT FILE LISTING:

```
FIGURE LINE INTERSECT  n fg p dir pID /offg /off
   0 2 99 S67W 96
   Intersection is at line           95          96
       100      2702.5531      7122.3297                DEFAULT
```

*Figure 8-21. FIGURE LINE INTERSECT*

### 8.4.8 FIGURE ARC INTERSECT

The FIGURE ARC INTERSECT command locates the coordinates of point *n* defined by the intersection of figure *fg* and a circle centered at *pCC* with radius *r*. You have the option to offset the point of intersection from the figure by distance *offg*.

Note: FIGURE ARC INTERSECT extends lines to infinity when no intersection would result otherwise. If you are not careful, unexpected intersections may result.

<u>Operator Sequence</u>

1. Select or key in FIGURE ARC INTERSECT.

   *The system responds:*

   FIGURE ARC INTERSECT n fg pCC r pID /offg

2. Identify the following:

   n — the ID of the point to be defined by the intersection.

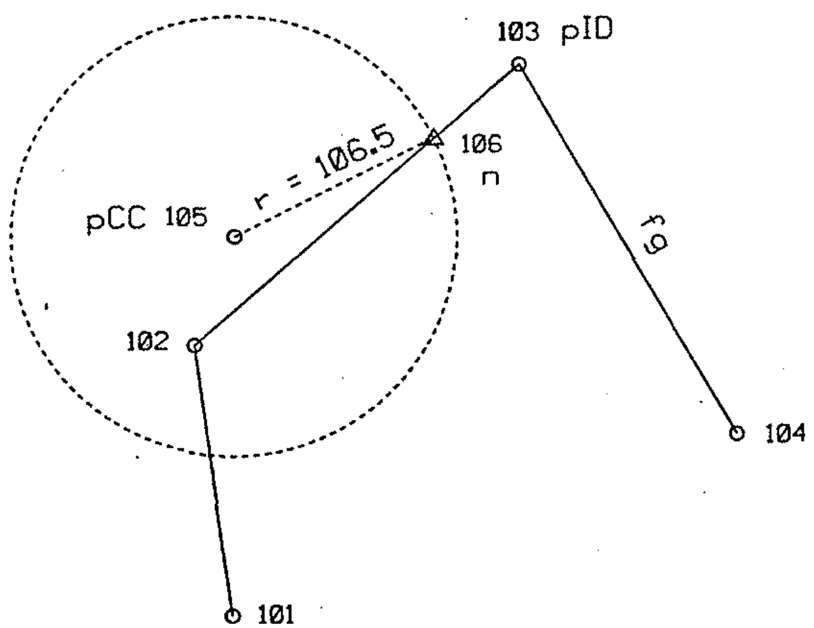
   fg — the figure ID. Note: You can also enter a *des* here.

   pCC — the known point that is the center of the arc.

   r — the radius of the arc. It can be expressed using the D delimiter.

   pID — a known point that indicates the intersection to be selected. Positive near to *pID*; negative far from *pID*.

   offg — the distance to offset the point of intersection from the figure. Positive to the right; negative to the left.




REPORT FILE LISTING:

```
FIGURE ARC INTERSECT  n fg pCC r pID /offg
   0 3 105 106.5 103
   Intersection is at line         102         103
       106     2874.5028     7774.8630                DEFAULT
```

*Figure 8-22. FIGURE ARC INTERSECT*

### 8.4.9 FIGURE FIGURE INTERSECT

The FIGURE FIGURE INTERSECT command locates the coordinates of point *n* defined by the intersection of figure *fg1* and figure *fg2*. You have the option to offset the point of intersection from the figure *fg1* by distance *offg1* and to offset the point of intersection from the figure *fg2* by distance *offg2*.

Note: FIGURE FIGURE INTERSECT extends lines to infinity when no intersection would result otherwise. If you are not careful, unexpected intersections may result.

Operator Sequence

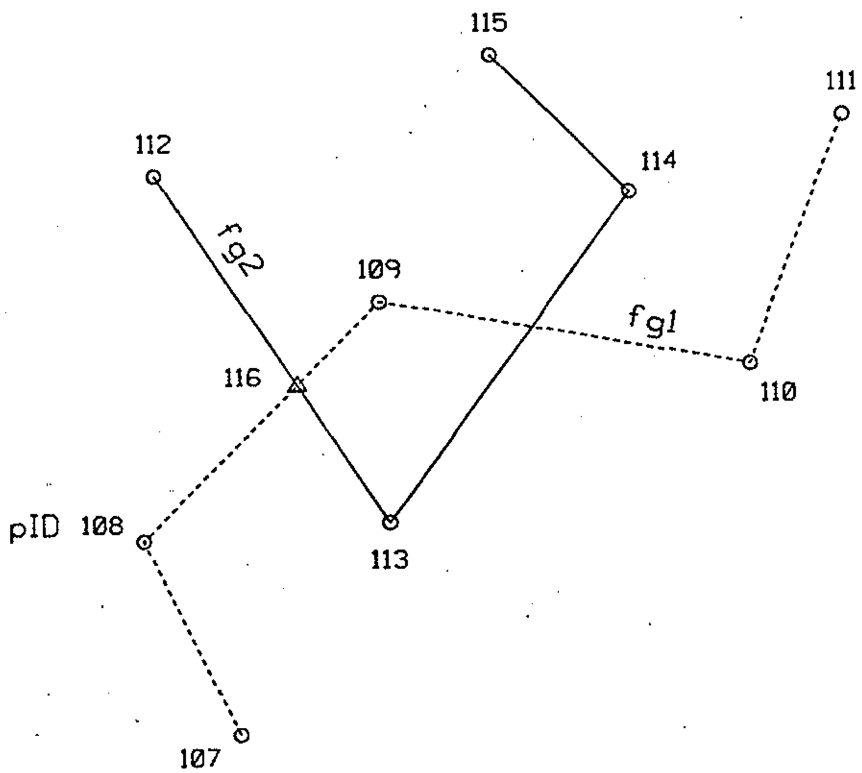
1. Select or key in FIGURE FIGURE INTERSECT.

   *The system responds:*

   FIGURE FIGURE INTERSECT n fg1 fg2 pID /offg1 /offg2

2. Identify the following:

| | |
|---|---|
| n | the ID of the point to be defined by the intersection. |
| fg1 | the first figure ID. Note: You can also enter a *des* here. |
| fg2 | the second figure ID. Note: You can also enter a *des* here. |
| pID | a known point that indicates the intersection to be selected. Positive near to *pID*; negative far from *pID*. |
| offg1 | the distance to offset the point of intersection from the first figure. Positive to the right; negative to the left. |
| offg2 | the distance to offset the point of intersection from the second figure. Positive to the right; negative to the left. |




REPORT FILE LISTING:

```
FIGURE FIGURE INTERSECT  n fg1 fg2 pID /offg1 /offg2
  0 4 5 108
 Intersection at line        108       109 and line        112        113
     116     2285.3188     7170.7428              DEFAULT
```

*Figure 8-23. FIGURE FIGURE INTERSECT*

### 8.4.10 LINE SPIRAL

The LINE SPIRAL command locates the coordinates of point *n* defined as the intersection of the line through *p1* and *p2* and the most recently stored simple spiral. (Spirals have to be defined using the SIMPLE SPIRAL or the SPIRAL LENGTH command). See Section 8.4 for information on intersecting compound spirals.

Note: You might want to store the spiral in a figure and use the FIGURE LINE INTERSECT command.

Operator Sequence

1. Select or key in LINE SPIRAL.

   *The system responds:*

   LINE SPIRAL n p1 p2 pID

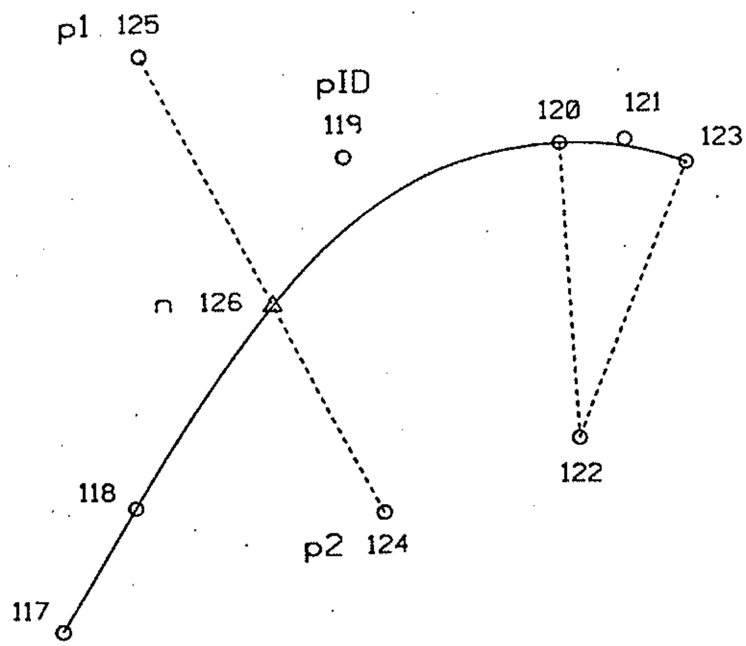
2. Identify the following:

   n — the ID of the point to be defined by the intersection.

   p1 — a known point on the line.

   p2 — another known point on the line.

   pID — a known point that indicates the intersection to be selected. Positive near to *pID*; negative far from *pID*.




REPORT FILE LISTING:

```
 LINE SPIRAL  n p1 p2 pID
   0 125 124 119
       126      2759.4422      8207.9523              DEFAULT
   Arc length from point        118 to point         126 is        118.0510FT
```

*Figure 8-24. LINE SPIRAL*

### 8.4.11 CURVE SPIRAL

The CURVE SPIRAL command locates the coordinates of point *n* defined as the intersection of the most recently stored simple spiral and the circle centered at *pCC* with radius *r*. (Spirals have to be defined using the SIMPLE SPIRAL or the SPIRAL LENGTH command). See Section 8.4 for information on intersecting compound spirals.

Note: You might want to store the spiral in a figure and use the FIGURE ARC INTERSECT command.

Operator Sequence

1. Select or key in CURVE SPIRAL.

   *The system responds:*

   CURVE SPIRAL n r pCC pID

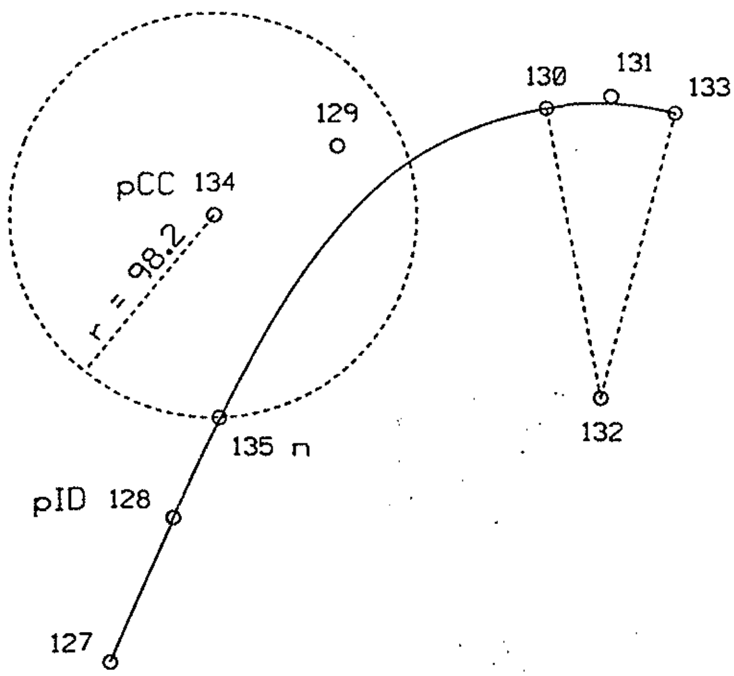
2. Identify the following:

   n — the ID of the point to be defined by the intersection.

   r — the radius of the circular curve. It can be expressed using the D delimiter.

   pCC — the known point that is the center of the arc.

   pID — a known point that indicates the intersection to be selected. Positive near to *pID*; negative far from *pID*.




REPORT FILE LISTING:

```
CURVE SPIRAL  n r pCC pID
   0 98.2 134 128
       135     2710.1201     8699.0132              DEFAULT
          Arc length from        128 to        135 is        53.3557FT
```

*Figure 8-25. CURVE SPIRAL*

### 8.4.12 SPIRAL SPIRAL

The SPIRAL SPIRAL command locates the intersection point *n* of two simple spirals. The first spiral is the one you have most recently stored using the SIMPLE SPIRAL or SPIRAL LENGTH command. The second spiral is defined by the known point of transition between tangent and spiral *pTS*, a direction *dirBT*, the length of the spiral *ls*, and the radius *r* of the curve for the second spiral. See Section 8.4 for information on intersecting compound spirals.

Note: You might want to store the spirals in figures and use the FIGURE FIGURE INTERSECT command.

<u>Operator Sequence</u>

1. Select or key in SPIRAL SPIRAL.

   *The system responds:*

   SPIRAL SPIRAL n pTS dirBT ls r

2. Identify the following:

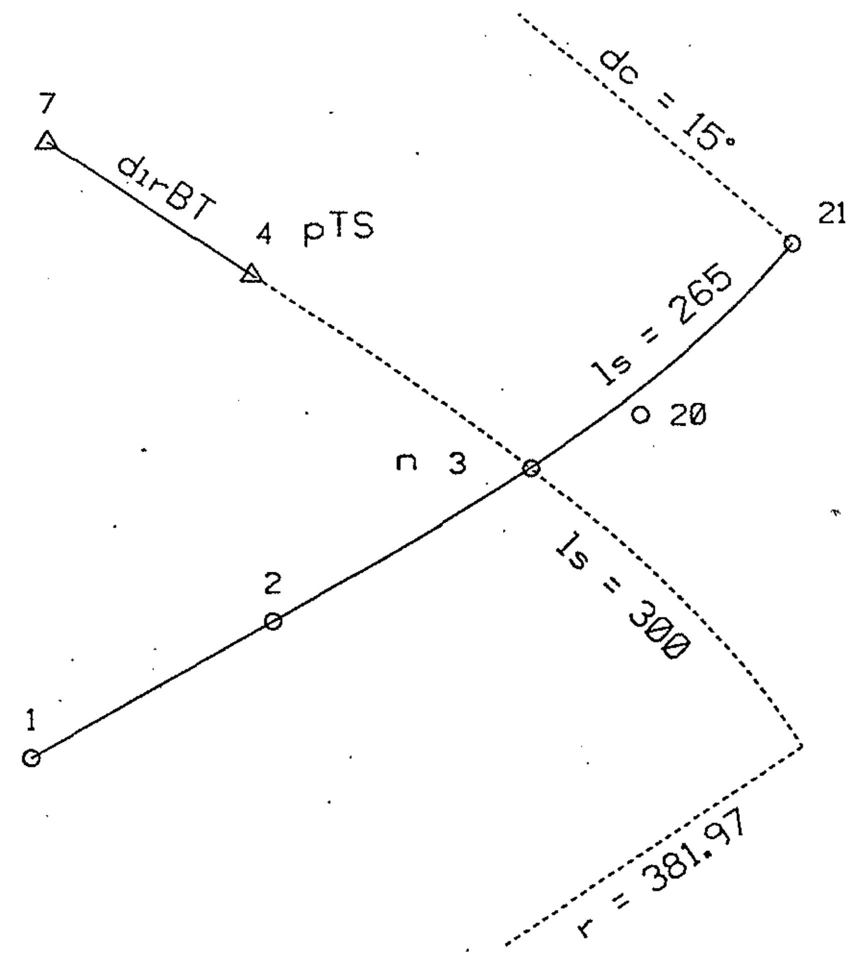
   n — the ID of the point to be defined by the intersection.

   pTS — the point of transition from tangent to the second spiral.

   dirBT — the direction of back tangent for the second spiral. It can be expressed as an azimuth or bearing, or can be calculated using the A delimiter.

   ls — the arc length of the second spiral (from *pTS* to *pSC*).

   r — the radius of the circular curve of the second spiral. Positive for clockwise spiral (to the right); negative for counterclockwise spiral (to the left). It can be expressed using the D delimiter.




REPORT FILE LISTING:

```
SIMPLE SPIRAL  cid pBT pTS nSIT nSC ls dc or r sign
   100 1 2 20 21 265 15. -1
SPIRAL SPIRAL  n pTS dirBT ls r
   0 4 A7 4 300 381.97
         3      2230.54       8802.72              DEFAULT
   Distance from          2 to intersection is       122.68FT
   Distance from          4 to intersection is       139.07FT
```

*Figure 8-26. SPIRAL SPIRAL*

FIGURES/ALIGNMENTS Commands

9

# 9. FIGURES/ALIGNMENTS COMMANDS

The FIGURES/ALIGNMENTS menu block includes commands for storing and deleting figures, establishing horizontal and vertical alignments, and computing points on or offsets to and from figures that have been stored as alignments. The FIGURES/ALIGNMENTS menu block is divided into three parts:

- GENERAL Commands
- HORIZONTAL ALIGNMENT Commands
- VERTICAL ALIGNMENT Commands.

## 9.1 GENERAL Commands

Use the GENERAL commands to store and delete figures, insert curves and spirals, assign points parallel to an existing figure, and clear the active alignment. GENERAL FIGURES/ALIGNMENTS commands include the following:

| *Command* | *For* |
|---|---|
| STORE FIGURE | associating a group of points (optionally with curves, spirals, and/or stationing information included) and storing the information in the database |
| INSERT CURVE | inserting a curve in a figure in graphics only |
| INSERT SPIRAL | inserting a spiral in a figure in graphics only |
| DELETE FIGURE | deleting a figure (its ID, grouping of points, and stationing) from the database |
| PARALLEL FIGURE | assigning points parallel to an existing figure |
| CLEAR ALIGNMENT | clearing the current (active) horizontal and associated vertical alignments |

### 9.1.1 STORE FIGURE

The STORE FIGURE command assigns a figure ID *nfg* to a grouping of point numbers used in the figure description *des*. The description can include circular curve and spiral information. After you assign the ID, you can use it to identify the figure. In this way you can manipulate an entire group of points by specifying a figure ID. Additionally, you can use the optional parameters *pAL* and *sAL* to place stationing information in the database and activate an alignment while storing the figure. STORE FIGURE plots new figures as linework when executed with AUTO PLOT on.

You can assign up to 2,147,483,647 figure ID's to be stored in each ICS database. The figure ID must be a positive number in the range from 1 to 2,147,483,647. If you assign 0 to *nfg*, ICS automatically assigns the next available figure number to the figure.

By default, ICS issues a warning if you attempt to use a figure ID that already exists. Then you can choose to redefine the figure or not. *If you do not want to be warned, you can change a parameter in ICS.PAR (see Section 3.6).*

ICS stores point ID's (not their coordinates) in the figure area of the ICS database. The point numbers, in turn, refer to coordinates in the coordinate area of the same ICS database. Based on this logic, if you delete a figure, you are merely deleting the association of points, not the points themselves.

How to Key in the Description *des*

You can use the description *des* in various ways (see Section 2.3.1 also). When listing point numbers as input in *des*, you can use a hyphen to indicate a range of point numbers. For a lengthy list of points, you can continue the description on subsequent lines by pressing <CR> without closing the parentheses.

If you enter another figure number in *des* instead of a list of points, then the figure is stored as a duplicate of the original figure. For example, the command

STORE FIGURE 60 10

creates figure 60, which is identical to figure 10.

You can also indicate curves and spirals in the description. To indicate that two points are connected by a curve rather than a straight line, you separate the two points in the data field by three additional items of information: the letter C which indicates a curve, the point number of the center of curvature, and the letter L or R to indicate left or right. For example, the command

STORE FIGURE 12 (63 C4R 2)

creates figure 12, which is a curve to the right from point 63 to point 2. The curve is centered at point 4.

To indicate that two points are connected by a spiral, input the point ID of the point of intersection of the tangents with an S before it. The input for storing a spiral in a figure is either (pTS pSIT pSC) or (pCS pSIT pST). For example, the command

STORE FIGURE 8 (5 S6 4 C12L 15)

creates figure 8, which has a spiral from point 5 to point 4 with the intersection of the tangents at point 6.

How to Graphically Produce the Description *des*

After entering a *nfg*, you have the option to graphically enter the *des*. To do so, follow the prompts in the error/input field of your screen. When the prompt reads **Identify point/Reset to end** you can begin data-pointing the appropriate points. After each point highlights, the prompt ACCEPT/REJECT appears in the error/input field, and the point ID number is displayed in the status field. If you have selected the point you wanted, you press <D>; otherwise, you press <R>.

You can speed up the process by accepting a highlighted point with a data point on the next point you want in the description. When you have identified all the points in *des*, move away from the points and press <D> to accept the last point. Do not accept on top of the last point, or it will be repeated in the description, such as (1 2 3 3). After accepting the last point, you complete the description by pressing <R>, which closes the description with a close parenthesis. Finally, press <D> to execute the command, or <R> to abort the command.

While you are graphically inserting points into *des*, you might want to insert a curve or spiral. You can do that using the INSERT CURVE and INSERT SPIRAL commands, or by keying in the curve and spiral portions of the *des* using the C, R, L, and S characters.

Note: Refer to the INSERT CURVE and INSERT SPIRAL commands to see how to graphically store figures with curves and spirals (see Sections 9.1.2 and 9.1.3).

Operator Sequence

1. Select or key in STORE FIGURE.

   *The system responds:*

   STORE FIGURE nfg des /pAL sAL (alpha)

   -OR-

   STORE FIGURE nfg des (graphics)

2. Identify the following.

   nfg — the ID of the figure to be stored. If you want to change an existing figure description, enter that figure ID instead of a new figure ID.

   des — description of the points to store in *nfg*.

   pAL — any position in the figure for which alignment stationing is to be defined.

   sAL — the station of the point *pAL*. Do not include the +; for example, 10+65.3 should be keyed in as 1065.3. Defining *pAL* and *sAL* activates the alignment.

## 9.1.2 INSERT CURVE (Graphics Only)

Use the INSERT CURVE command with the STORE FIGURE command to insert a curve in the figure you are storing. You identify the starting point of the curve before you select INSERT CURVE. The system then prompts you to identify the curve center point *pCC* and the curve end point.

ICS automatically calculates the curve as the shortest arc from the *PC* to the *PT*. If you wish to place a curve that is the longer of the two possible arcs, you must key in C, then the *pCC* point number, and the direction (L or R) for the curve. You can do this instead of identifying the *pCC* with a data point. See Section 9.1.1 for more details.

Note: This command builds curve information for the ICS STORE FIGURE command and submits it to ICS for processing. The results of this command appear only in the description of a figure.

Setup

Before selecting this command you must select the STORE FIGURE command and identify the starting point of the curve.

Operator Sequence

1. Select INSERT CURVE.

2. **Identify center of arc**

   Place the cursor on the pCC and press <D>.

   *The system highlights the point you select.*

3. **Identify center of arc RJCT/ACCPT**

   Press <D> to accept the highlighted point. If you place a data point on the end point of the arc, you can skip Step 4 and go to Step 5.

   Note: If there is not a tangent going into the curve, the system prompts you to press <D> if the curve is to the right or press <R> if the curve is to the left.

   -OR-

   Press <R> to reject the highlighted point. Return to Step 2.

4. **Identify arc end point**

   Place the cursor on the pPT and press <D>.

   *The system highlights the point you select.*

5. **Identify arc end point RJCT/ACCPT**

   Press <D> to accept the highlighted point.

   *The system returns to the STORE FIGURE command prompts.*

   -OR-

   Press <R> to reject the highlighted point. Return to Step 4.

### 9.1.3 INSERT SPIRAL (Graphics Only)

Use the INSERT SPIRAL command with the STORE FIGURE command to insert a spiral in the figure. You identify the first point of the spiral (TS or CS) before selecting INSERT SPIRAL. When you select INSERT SPIRAL, the system prompts you to identify the tangent intersection point *pSIT*. A figure containing a spiral must also, at a minimum, contain the curve(s) associated with it.

Note: This command builds spiral information for the ICS STORE FIGURE command and submits it to ICS for processing. The results of this command appear only in the description of a figure.

Setup

Before selecting this command you must select the STORE FIGURE command and identify the starting point of the spiral.

Operator Sequence

1. Select INSERT SPIRAL.

2. **Identify intersection point**

   Place the cursor on the pSIT and press <D>.

   *The system highlights the point you select.*

3. **Identify intersection point RJCT/ACCPT**

   Press <D> to accept the highlighted point. If you place a data point on the curve tangent point, you can skip Step 4 and go to Step 5.

   -OR-

   Press <R> to reject the highlighted point. Return to Step 2.

4. **Identify curve (tangent) point**

   Place the cursor on the pSC or pST and press <D>.

   *The system highlights the point you select.*

5. **Identify curve (tangent) point RJCT/ACCPT**

   Press <D> to accept the highlighted point.

   *The system returns to the STORE FIGURE command prompts.*

   -OR-

   Press <R> to reject the highlighted point. Return to Step 4.

### 9.1.4 DELETE FIGURE

The DELETE FIGURE command deletes the description of a group of points as a figure in the database. This command does not delete coordinates; it only deletes the figure that associates the points. To delete more than one figure, you can put the list of figures to be deleted in brackets (for example: [3 7 12-16] ).

In graphics, you can identify figures with a data point on the figure element (linework), a fence, or a key-in. ICS does not delete graphics if you use a key-in but does remove the database entry. If you use a fence, you need to perform a view update in order to see the results of the delete.

Operator Sequence

1. Select DELETE FIGURE.

   *The system responds:*

   DELETE FIGURE des

2. Identify the following:

   des    description of the figure(s) to be deleted.

Note: The figure ID placed with closed figures is not deleted.

### 9.1.5 PARALLEL FIGURE

The PARALLEL FIGURE command assigns points parallel to a description of points *des*, offset at a specified distance *off*. A positive value for *off* offsets the new points to the right of the original figure looking ahead along the description. (If you do not know how a figure was stored in the database, use the command LIST FIGURES.) If you want to offset to the left of the original figure, enter a negative offset.

The new point numbers begin with *n* and increase by 1. The last point is given the number *n+(k-1)*, where *k* denotes the number of points in the original figure. You also have the option to store the new points as a figure. If you use the optional *nfg* parameter, the new points are assigned the figure ID *nfg*.

Additionally, if you want the new points to have a different elevation than the original points, you can use the optional *dz* parameter. If you specify this parameter, each new point elevation will be the sum of the original point elevation and *dz*.

Note: This command computes offset spirals for graphics display and plotting *only*. ICS *does not recognize* spirals in a parallel figure as being parallel to another spiral.

Operator Sequence

1. Select or key in PARALLEL FIGURE.

   *The system responds:*

   PARALLEL FIGURE des off n /nfg /dz

2. Identify the following:

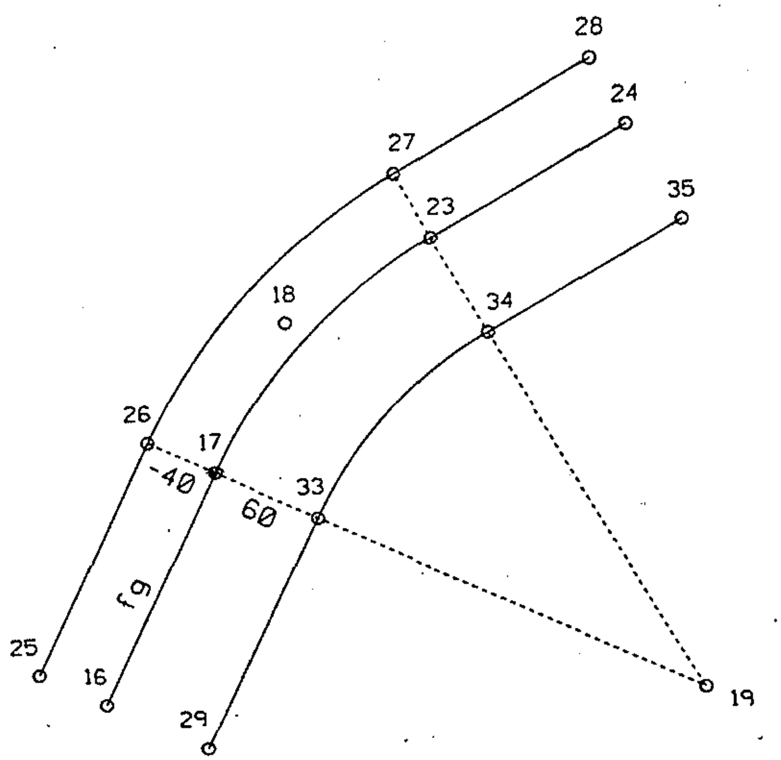
   des — description of the figure.

   off — the distance to offset the parallel points from the original points. Positive to the right; negative to the left.

   n — the ID of the first point in the new figure.

   nfg — the figure ID to be assigned to the parallel figure.

   dz — the translation in the z coordinate. Positive to increase the elevation; negative to decrease.




REPORT FILE LISTING:

```
STORE FIGURE  nfg des /pAL sAL or  fg des /pAL sAL
   3 (16 17 C19R 23 24)
         3 (16 17 C19R 23 24)
     Plotting feature is        DEFAULT                         [   1]
PARALLEL FIGURE  fg off /n /Nfg /dZ
   3 -40 0 40
        40 (25 26 C19R 27 28)
     Plotting feature is        DEFAULT                         [   1]
PARALLEL FIGURE  fg off /n /Nfg /dZ
   3 60 0 50
        50 (29 33 C19R 34 35)
     Plotting feature is        DEFAULT                         [   1]
```

*Figure 9-1.  PARALLEL FIGURE*

### 9.1.6 CLEAR ALIGNMENT

The CLEAR ALIGNMENT command disregards the current active horizontal and vertical alignments. This command is useful when you encounter erroneous data or key in erroneous stationing, such as illegal vertical curve lengths, when running the VERTICAL START and VERTICAL END commands. The SET ALIGNMENT command also clears the old alignment and lets you start the vertical alignment computations again (or activate a different horizontal alignment). See Section 9.2.1 for details on SET ALIGNMENT.

Operator Sequence

1. Select or key in CLEAR ALIGNMENT. (There are no data fields.)

   *ICS disregards the current horizontal and vertical alignments.*

## 9.2 HORIZONTAL ALIGNMENT Commands

HORIZONTAL ALIGNMENT commands allow you to access previously defined point or figure descriptions, and then compute new points, circular curves, and spiral curves and define station equations. With these commands, you can also compute points on or offsets to and from figures that have been stored as alignments.

Creating an Alignment

You can create alignments by using the STORE FIGURE or the SET ALIGNMENT command. When you use the STORE FIGURE command, you have the option to place stationing information in the database and activate an alignment while storing the figure. This alignment then stays active until you activate another alignment using the STORE FIGURE or SET ALIGNMENT command.

If you want to activate an alignment that was previously defined, use SET ALIGNMENT (ACTIVATE in graphics). If you did not place stationing information in the database when you defined a figure, you can use the SET ALIGNMENT command to do that at a later time. The alignment information (des, pAL, and sAL) is stored in the database if you enter this information with the STORE FIGURE command or the SET ALIGNMENT command.

Note: When you specify stations beyond the limits of the stored alignment, ICS commands extend the alignment along tangents at each end.

Using the Degree of Curvature or Radius to Define a Curve

Some of the HORIZONTAL ALIGNMENT commands prompt you to enter a degree of curvature. By default these commands look for a degree of curvature. If you want to give the radius instead, you have two choices. One option is to change the default setting to radius using the INPUT MODES command in the ENVIRONMENT portion of the menu (see Section 7.3.12).

The other option you have is to give a radius value by prefixing the value with an upper or lowercase *r*. See Section 2.3.4 for more details.

Note: If you set the default to radius and want to input a degree of curvature, prefix the degree of curvature value with an upper or lowercase *d*.

By default, the FIT CURVE command looks for a radius. You can change this default to degree of curvature, or you can key in a degree of curvature by prefixing the value with an upper or lowercase *d*.

HORIZONTAL ALIGNMENT commands include the following:

| *Command* | *For* |
|---|---|
| SET ALIGNMENT: ACTIVATE and DEFINE STATIONING | activating a horizontal alignment (and optionally an associated vertical alignment) and defining stationing |
| STATION EQUATION | defining a station equation for the horizontal alignment |
| REVERSE CURVE | finding the points that define a curve, a reverse curve, and optionally a tangent between the two curves that will place a reverse curve transition |
| POINTS ON ALIGNMENT | locating and computing point(s) at a fixed interval along an alignment |
| FROM ALIGNMENT | locating a point at a given station and offset |
| ALIGNMENT OFFSET | locating points on an alignment by extending a perpendicular from a point, a series of points, or a figure to the alignment (the projection of the points are perpendicular to the active alignment) |
| STREETS INTERSECT | calculating the street returns of two streets, a street and a property line, or a street and a right-of-way |
| FIT CURVE: TANGENTS and EDGE | fitting a curve based on tangents or three points on the curve |
| SIMPLE CURVE | calculating a curve based on an the degree of curvature/radius and the central angle |
| SIMPLE SPIRAL: IN and OUT | calculating parameters associated with a simple spiral, given an incoming tangent, spiral length, and degree of curvature/radius of the circular curve. |
| POINTS ON SPIRAL: IN and OUT | locating points at even intervals along a simple spiral |
| COMPOUND SPIRAL | calculating the critical points for a compound (combining) spiral as defined by length and by a degree of curvature/radius at each end |
| FIT ALIGNMENT | calculating a spiral in, followed by a circular curve and then a spiral out, given a set of defined parameters |

### 9.2.1 SET ALIGNMENT: ACTIVATE and DEFINE STATIONING

The graphics options ACTIVATE and DEFINE STATIONING are combined in the alpha SET ALIGNMENT command. The SET ALIGNMENT command(s) activate a horizontal alignment. The difference between the two graphic SET ALIGNMENT commands is that the ACTIVATE command does not alter any stationing that is defined on the alignment, and the DEFINE STATIONING command does. DEFINE STATIONING erases all of the previously defined stationing of the alignment, redefines the stationing on the alignment, and activates the horizontal alignment. The alpha SET ALIGNMENT command gives you the option to just activate an alignment or to activate an alignment and redefine station equations.

Note: If you want to activate an associated vertical alignment as well as a horizontal alignment, you can do this using the optional *Vfg* parameter.

<u>Operator Sequence</u>

1. Select ACTIVATE or DEFINE STATIONING.

   -OR-

   Key in SET ALIGNMENT.

   *The system responds:*

   SET ALIGNMENT Hfg /pAL sAL /Vfg

2. Identify the following:

   Hfg — the horizontal figure ID (alignment).

   pAL — any point on the horizontal alignment. Skip this field with a zero (0).

   sAL — the station of the point *pAL*. Skip this field with a zero (0).

   Vfg — the vertical figure ID (alignment).

## 9.2.2 STATION EQUATION

The STATION EQUATION command defines a station equation for the active horizontal alignment, beginning at a specified point on the alignment, where the previous station equation ends (*Bksta*) and the new station equation begins (*Ahsta*). You have the option to define the station equation for a horizontal alignment that is not active by using the parameter *Hfg*.

The *Ahsta* should normally be greater than *Bksta*. If the *Ahsta* is less than *Bksta*, ICS automatically adds a constant to *Ahsta*. This constant is specified in the ICS.PAR parameter file. You should define station equations in sequence from the beginning of the alignment toward the end of the alignment.

Operator Sequence

1. Select or key in STATION EQUATION.

   *The system responds:*

   STATION EQUATION Bksta Ahsta /Hfg

2. Identify the following:

   Bksta the back station.

   Ahsta the ahead station.

   Hfg the figure ID or group of points (alignment).

REPORT FILE LISTING:

```
STATION EQUATION  bksta ahsta /fg
   520.5269 100
 %EQU-I-STAEQU, Station equation defined.
```

*Figure 9-2. STATION EQUATION*

### 9.2.3 REVERSE CURVE

The REVERSE CURVE command finds the point of intersection of two reverse curves, the tangent point of the second curve, and the location of the center of each curve. You can use the optional parameter *tl* to specify the tangent length between the two curves. If you do not specify *tl*, ICS assumes the tangent length is zero.

Operator Sequence

1. Select or key in REVERSE CURVE.

   *The system responds:*

   REVERSE CURVE p1 dir1 r1 p2 dir2 r2 nCC1 /tl

2. Identify the following:

   p1 the ID of the known point *PC* of the first curve.

   dir1 the direction of the back tangent through *p1*.

   r1 the radius of the first curve.

   p2 the ID of a known point through which the ahead tangent of the second curve passes.

   dir2 the direction of the ahead tangent of the second curve.

   r2 the radius of the second curve.

   nCC1 the ID of the point to be defined as the center of the first curve. ID's for the other generated points are *nCC1*+1, *nCC1*+2, etc.

   tl the tangent length. The default is zero.

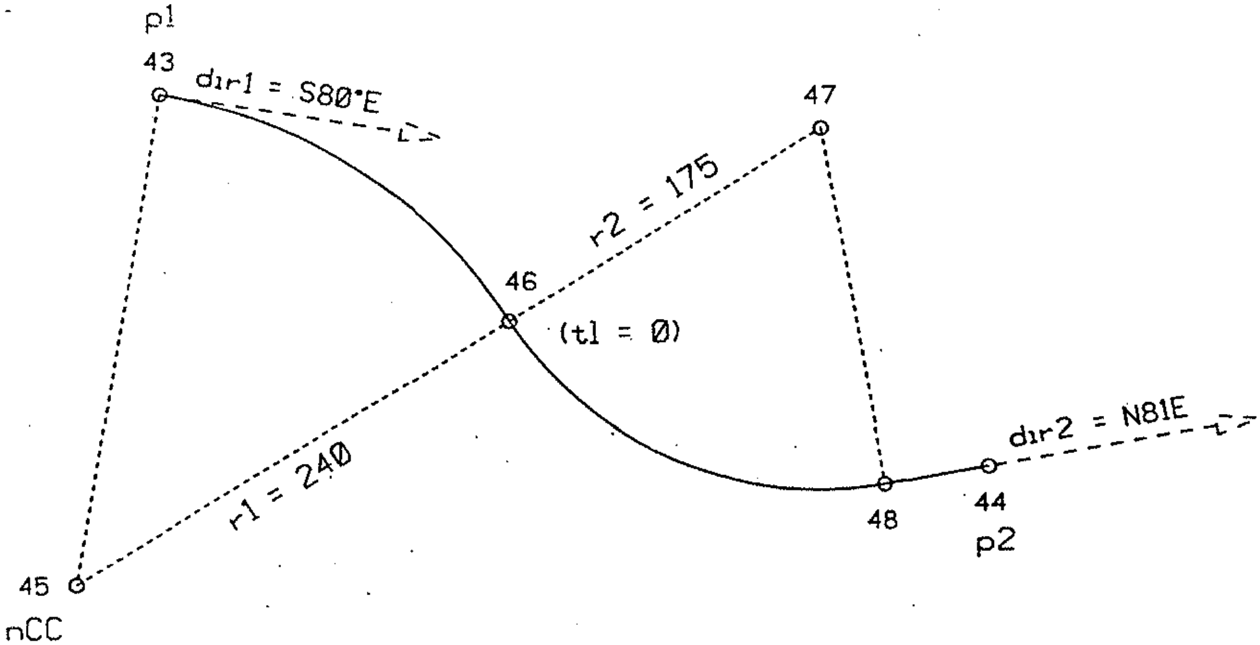



*Figure 9-3. REVERSE CURVE*

REPORT FILE LISTING:

```
REVERSE CURVE  p1 dir1 r1 p2 dir2 r2 nCC1 /tl
   43 S80E 240 44 N81E 175 0 0
Circular curve 1 with fixed beginning point
Center of curve 1:
        45      3159.1350      2094.6913              DEFAULT
        Length          204.7567FT  Degree      23-52-23.7
        Radius          240.0000FT  Delta       48-52-55.4
Point of reverse curvature:
        46      3283.1674      2300.1566              DEFAULT
Circular curve 2 with floating end point
Center of curve 2:
        47      3373.6076      2449.9751              DEFAULT
        Length          207.3340FT  Degree      32-44-25.6
        Radius          175.0000FT  Delta       67-52-55.4
Point of tangency of curve 2:
        48      3200.7621      2477.3511              DEFAULT
```

*Figure 9-3. REVERSE CURVE (Continued)*

### 9.2.4 POINTS ON ALIGNMENT

The POINTS ON ALIGNMENT command computes points at a fixed interval along an alignment. ICS computes points at each position that has a station evenly divisible by interval *x*, and optional offset *off*. The computed points are stored in the database as point numbers *n*, *n*+1, *n*+2, etc.

If you want to have points calculated on only a section of the alignment, you can use the optional parameters *sBEG* and *sEND*. ICS calculates points only on the portion of the alignment that falls between stations *sBEG* and *sEND* if you specify these parameters. You can assign zero to *sBEG* and *sEND* as a wild card to indicate from the beginning, or to the end, respectively.

Setup

Be sure to run SET ALIGNMENT before running POINTS ON ALIGNMENT. See Section 9.2.1 for more information.

Operator Sequence

1. Select or key in POINTS ON ALIGNMENT.

   *The system responds:*

   POINTS ON ALIGNMENT x n /off /sBEG /sEND

2. Identify the following:

   x the distance between the new points.

   n the ID of the first point to be defined.

   off the distance to offset the points from the alignment.

   sBEG the station that is the starting point for placing points.

   sEND the station that is the ending point for placing points.

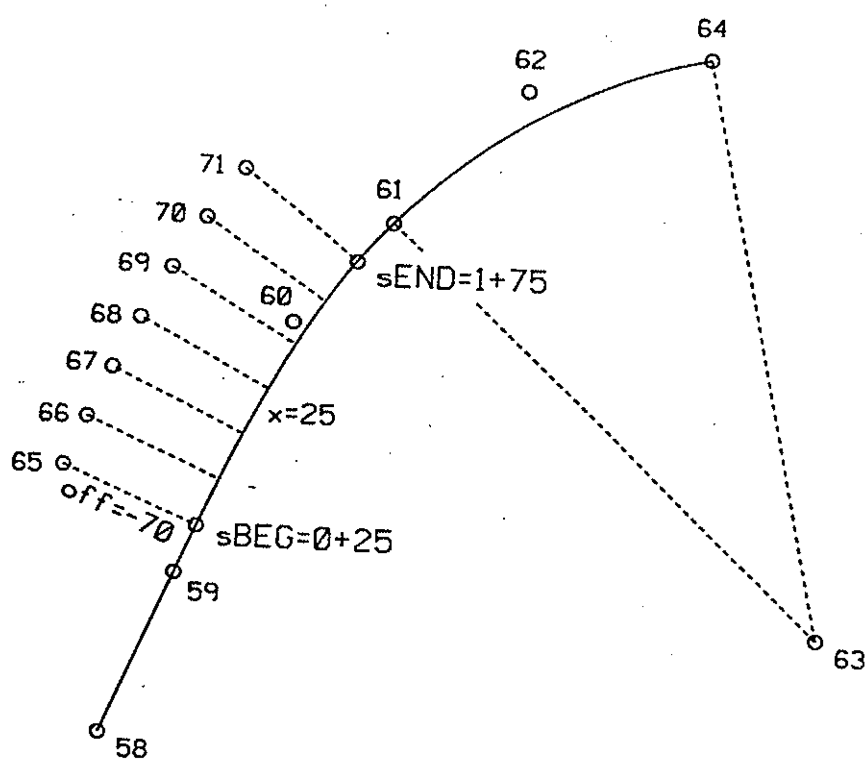



*Figure 9-4.* *POINTS ON ALIGNMENT*

REPORT FILE LISTING:

```
STORE FIGURE  nfg des /pAL sAL or  fg des /pAL sAL
   5 (58 59 S60 61 C63R 64)
         5 (58 59 S60 61 C63R 64)
     Plotting feature is       DEFAULT                         [   1]
%PLS-I-BEGPLO, plot task in progress
%PLS-I-ENDPLO, plot task complete
SET ALIGNMENT  Hfg/pAL sAL /Vfg
   5 59 0
%SAL-I-HORALAS, horizontal alignment assigned successfully to figure       5
POINTS ON ALIGNMENT  x /n /off /sBEG /sEND
   25 0 -70 25 175
   Point              Coordinates                  Station      Offset
--------------------------------------------------------------------------
     Spiral           59          61
         65      3697.2617   5111.3607              0+25      -70.0000
         66      3721.2481   5121.7633              0+50      -70.0000
         67      3745.6949   5133.0061              0+75      -70.0000
         68      3770.4427   5145.3848              1+00      -70.0000
         69      3795.2929   5159.2017              1+25      -70.0000
         70      3819.9975   5174.7597              1+50      -70.0000
         71      3844.2480   5192.3477              1+75      -70.0000
```

*Figure 9-4. POINTS ON ALIGNMENT (Continued)*

## 9.2.5 FROM ALIGNMENT

*In alpha, this command is LOCATE FROM ALIGNMENT.* The FROM ALIGNMENT command locates point *n* at station *sAL* and optional offset *off*. The alignment is described in *des* and its stationing is defined by point *pAL* with station *sAL* in the SET ALIGNMENT command. You have the option to specify an offset *off*.

This command is useful for inputting cross-section data. You can specify the optional parameter *rod* to compute the elevation of *n*. The elevation of *n* is the instrument elevation minus the rod shot *rod*. You define the instrument elevation using the LEVEL RUN command, entering *pBS rBS*.

Setup

Be sure to run SET ALIGNMENT before running FROM ALIGNMENT to activate an alignment. See Section 9.2.1 for more information. Also, if you intend to use elevations, use the LEVEL RUN command to establish the instrument elevation.

Operator Sequence

1. Select or key in FROM ALIGNMENT.

   *The system responds:*

   LOCATE FROM ALIGNMENT n sAL /off /rod

2. Identify the following:

   n — the ID of the point whose coordinates are to be located.

   sAL — the station at point *n*.

   off — the distance to offset point *n* from alignment *des*.

   rod — the rod height that will be used along with current instrument elevation (from LEVEL RUN) to compute *n*.

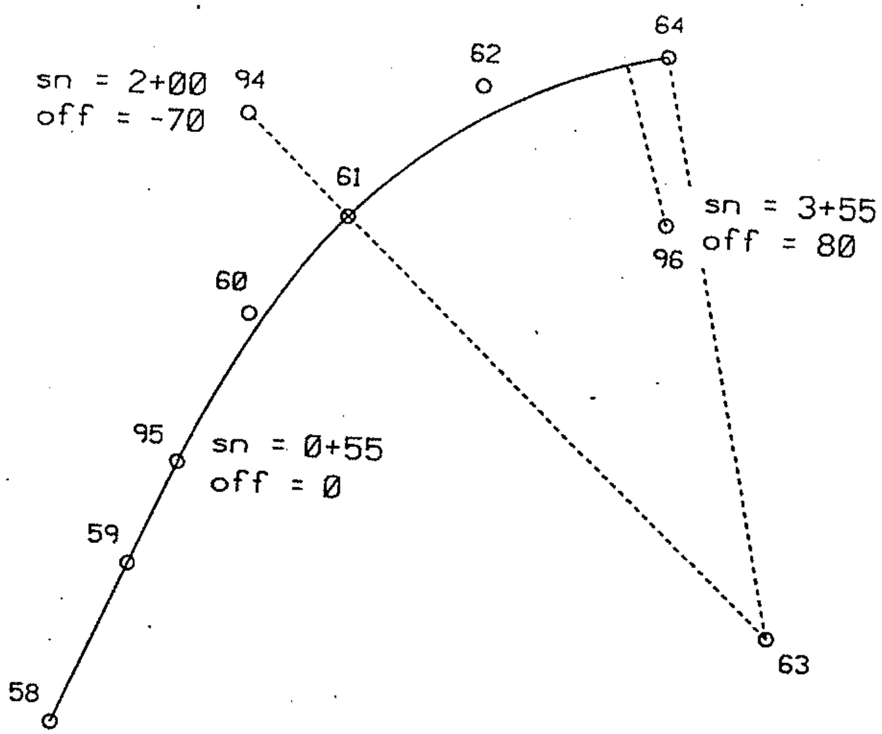



*Figure 9-5. LOCATE FROM ALIGNMENT*

REPORT FILE LISTING:

```
STORE FIGURE  nfg des /pAL sAL or  fg des /pAL sAL
   6 (58 59 S60 61 C63R 64)
         6 (58 59 S60 61 C63R 64)
     Plotting feature is       DEFAULT                          [   1]
%PLS-I-BEGPLO, plot task in progress
%PLS-I-ENDPLO, plot task complete
SET ALIGNMENT  Hfg/pAL sAL /Vfg
   6 59 0
%SAL-I-HORALAS, horizontal alignment assigned successfully to figure          6
LOCATE FROM ALIGNMENT  n sn /off /rod
   94 200 -70
      94       3367.6661       5212.2275         0.0000           2+00     -70.0000
LOCATE FROM ALIGNMENT  n sn /off /rod
   95 55
        95       3197.3808       5187.7703         0.0000           0+55
LOCATE FROM ALIGNMENT  n sn /off /rod
   96 355 +80
      96       3324.0450       5416.2464         0.0000           3+55      80.0000
```

*Figure 9-5. LOCATE FROM ALIGNMENT (Continued)*

### 9.2.6 ALIGNMENT OFFSET

The ALIGNMENT OFFSET command locates points on an alignment by extending a perpendicular from a point, a series of points, or a figure to the active alignment.

Note: ALIGNMENT OFFSET extends lines to infinity when no intersection would result otherwise. If you are not careful, unexpected intersections may result.

Additionally, if the alignment begins or ends with a curve, ICS extends the curve if no intersection would result otherwise. ICS extends curves from the end of the alignment to ensure that the projection of each point is perpendicular to the alignment (or extension of the alignment). See Figure 9-6.

This command calculates the distance along the extended curve from the end of the alignment to the point of perpendicularity. Then the command applies this distance from the end point of the alignment along the tangent to the curve. Thus, the actual extended alignment is along the tangent to the curve (not along the extended curve itself).

<u>Setup</u>

Be sure to run SET ALIGNMENT before running ALIGNMENT OFFSET to activate an alignment. See Section 9.2.1 for more information.

Operator Sequence

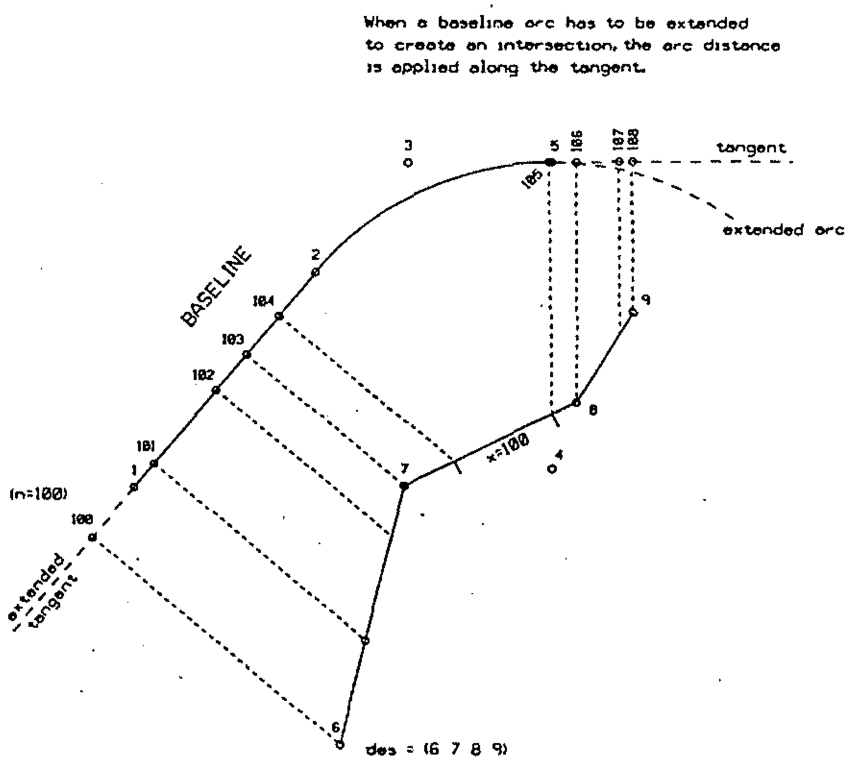
1. Select or key in ALIGNMENT OFFSET.

   *The system responds:*

   ALIGNMENT OFFSET des /x /n /sBEG sEND

2. Identify the following:

| | |
|---|---|
| des | description of the alignment: a figure ID or list of points to project (possibly a second alignment or baseline). |
| x | the interval between points described by *des.* |
| n | the ID of the first point to be defined by the intersection. |
| sBEG | the beginning station on the active alignment. |
| sEND | the ending station on the active alignment. |




REPORT FILE LISTING:

```
STORE FIGURE  nfg des /pAL sAL or  fg des /pAL sAL
   7 (58 59 S60 61 C63R 64)
         7 (58 59 S60 61 C63R 64)
     Plotting feature is       DEFAULT                          [   1]
%PLS-I-BEGPLO, plot task in progress
%PLS-I-ENDPLO, plot task complete
SET ALIGNMENT  Hfg/pAL sAL /Vfg
   7 59 0
%SAL-I-HORALAS, horizontal alignment assigned successfully to figure        7
ALIGNMENT OFFSET  des /x /n /sBEG /sEND
   (65 66 67)
        68      3611.0485      5651.1068              DEFAULT
        69      3730.1421      5703.3574              DEFAULT
        70      3884.4924      5850.9628              DEFAULT
```

*Figure 9-6. ALIGNMENT OFFSET*

### 9.2.7 STREETS INTERSECT

The STREETS INTERSECT command locates the intersection *nINT* and, for a desired radius *r*, street return points (CC, nPC, and nPT) of two streets, a street and a property line, or a street and a right-of-way. The first street is described by center line *fg1* and width *w1*; the second street is described by center line *fg2* and width *w2*.

If more than one intersection point is possible, the command finds the desired intersection point relative to point *pID*. If *pID* is positive, ICS defines the closest intersection point to *pID* as the intersection; if *pID* is negative, ICS defines the farthest intersection point to *pID* as the intersection.

If you assign a positive value to the radius, STREETS INTERSECT calculates centers of curvature (*nLL, nLR, nRL, nRR*), points of curvature (*nLL+1, nLR+1, nRL+1, nRR+1*), and points of tangency (*nLL+2, nLR+2, nRL+2, nRR+2*) for each of the four street returns. (LL indicates left of figure one and left of figure 2; LR indicates left of figure one and right of figure two, etc.)

Note: Because of the numbering scheme, it is important to choose the ID's for *nLL, nLR, nRL,* and *nRR* with care: always leave available point ID's to allow for the PC (*nLL*+1) and PT (*nLL*+2). For example, if you enter 50 for *nLL*, then enter 53 for *nLR*.

Points of curvature are placed on the first street, and points of tangency on the second street. If you assign a value of zero (0) to the radius, ICS does not calculate the returns; ICS calculates only property line intersections (*nLL, nLR, nRL, nRR*) based on the street widths.

You can define the center lines of one or both streets as the street edge by setting the street width to zero (0). If you set a street width to zero, you will want to have ICS calculate returns only in the appropriate sectors. Thus, you should put zeros in the other sectors.

You can assign a value of zero (0) to points *nLL, nLR, nRL,* or *nRR*, but ICS will not calculate returns in the *LL, LR, RL,* or *RR* sector. *It is therefore important to note that keying in a 0 for points nLL, nLR, nRL, and nRR does not instruct ICS to assign the next available point numbers to those particular points.* If you set the variable *nINT* to zero, ICS does not calculate and store the coordinates of the intersection of the center lines.

Note: STREETS INTERSECT extends lines to infinity when no intersection would result otherwise. If you are not careful, unexpected intersections may result.

Operator Sequence

1. Select or key in STREETS INTERSECT.

   *The system responds:*

   STREETS INTERSECT r fg1 w1 fg2 w2 pID nINT nLL nLR nRL nRR

2. Identify the following:

   r — the radius of the street returns.

   fg1 — the ID of the figure containing the center line of the first street. Note: You can also enter a *des* here.

   w1 — the total width of the first street.

   fg2 — the ID of the figure containing the center line of the second street. Note: You can also enter a *des* here.

   w2 — the total width of the second street.

   pID — a known point that indicates the intersection to be selected.

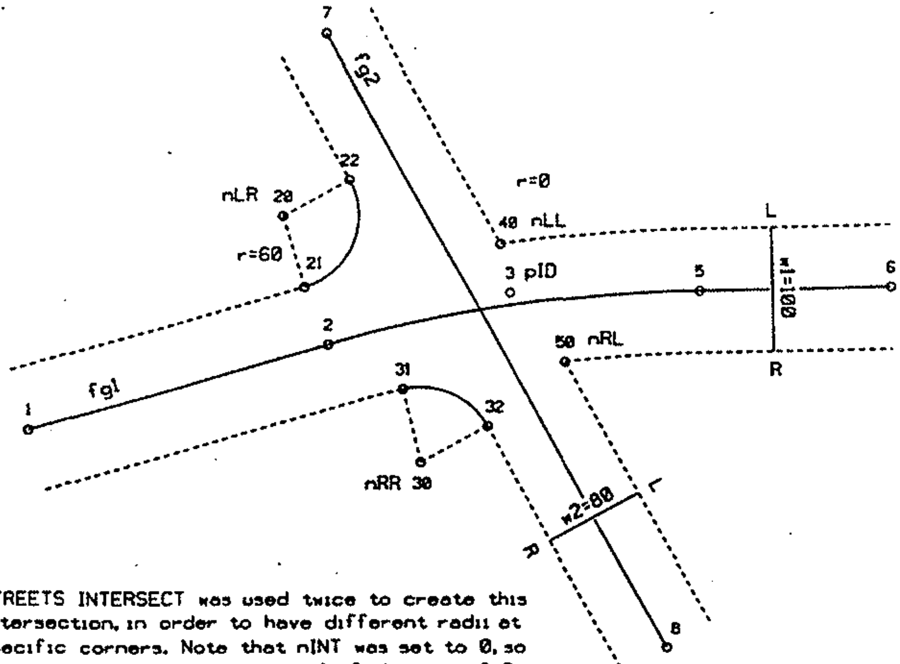
   nINT — the ID of the point to be defined by the intersection of the center lines.

   nLL — the ID of the point that is to be defined by one of the following: the center of the curve of return (if *r* has a value) or the property line (if *r* is zero). This point is located at the intersection of the left side of both streets.

   nLR — the ID of a point that is to be defined by one of the following: the center of curve of return (if *r* has a value), or the property line (if *r* is zero). This point is located at the intersection of the left side of the first street and right side of the second street.

   nRL — the ID of a point which is to be defined by one of the following: the center of curve of return (if *r* has a value), or the property line (if *r* is zero). This point is located at the intersection of the right side of the first street and left side of the second street.

   nRR — the ID of a point which is to be defined by one of the following: the center of curve of return (if *r* has a value), or the property line (if *r* is zero). This point is located at the intersection of the right side of both streets.




STREETS INTERSECT was used twice to create this intersection, in order to have different radii at specific corners. Note that nINT was set to 0, so it was not computed - typical of the use of 0 in this command.

REPORT FILE LISTING:

```
STREETS INTERSECT  r fg1 w1 fg2 w2 pID nINT nLL nLR nRL nRR
   60 1 100 2 80 3 0 0 20 0 30
CL intersection is at  Curve          2        5 and Line          7         8
        LR Intersect  Line           1        2 and Line          7         8
        20      3875.3669      2896.2313            DEFAULT
        21      3817.6882      2912.7591            DEFAULT
        22      3904.1912      2948.8541            DEFAULT
        RR Intersect  Curve          2        5 and Line          7         8
        30      3676.6870      3005.0586            DEFAULT
        31      3735.0271      2991.0434            DEFAULT
        32      3705.5112      3057.6814            DEFAULT
STREETS INTERSECT  r fg1 w1 fg2 w2 pID nINT nLL nLR nRL nRR
   0 1 100 2 80 3 0 40 0 50 0
CL intersection is at  Curve          2        5 and Line          7         8
        LL Intersect  Curve          2        5 and Line          7         8
        40      3851.9693      3068.6739            DEFAULT
        RL Intersect  Curve          2        5 and Line          7         8
        50      3757.9755      3120.1592            DEFAULT
```

*Figure 9-7. STREETS INTERSECT*

### 9.2.8 FIT CURVE: TANGENTS and EDGE

The graphics options FIT CURVE TANGENT and FIT CURVE EDGE are combined in the alpha FIT CURVE command. The FIT CURVES command(s) calculates critical points of a curve with radius *r* tangent to the line connecting *pBT* to *pPI* and the line connecting *pPI* to *pAT*. Point *nPC* is the point of curvature, *nPT* is the point of tangency, and point *nCC* is the center of the curve.
You can use the FIT CURVE TANGENTS command in several different ways:

- To calculate all critical points of the curve (*nPC, nPT*, and *nCC*)
  In this case you must use the optional radius *r* parameter. Also, you identify new point ID's for *nPC, nPT*, and *nCC*.

- To calculate the *nPT* and the *nCC*
  In this case you know the point of curvature, and you input the known point of curvature for both fields *pBT* and *nPC*. You do not use *r*.

- To calculate the *nPC* and the *nCC*
  In this case you know the point of tangency, and you input the known point of tangency for both fields *pAT* and *nPT*. You do not use *r*.

- To calculate the *nCC* only
  In this case you know the point of curvature and point of tangency. You input the known point of curvature for both *pBT* and *nPC* and input the known point of tangency for both *pAT* and *nPT*. You do not use *r*.

The FIT CURVE EDGE command calculates critical points of the curve through the *pPC*, a point on the curve *pPOC* and the *pPT*. It also locates the *nCC* and *nPI*.

Note: You cannot define the *nCC* or *nPI* point ID's in FIT CURVE EDGE. ICS automatically defines these ID's.

Operator Sequence

1. Select or key in FIT CURVE.

   *The system responds:*

   FIT CURVE pBT pPI pAT nPC nCC nPT /r (alpha and graphics TANGENTS)

   -OR-

   FIT CURVE pPC pPOC pPT (graphics EDGE)

2. Identify the following:

TANGENTS:

pBT the known point on the back tangent (can be the PC if known).

pPI the known point of intersection of the tangents.

pAT the known point on the ahead tangent (can be the PT if known).

nPC the ID of the point to be defined as the tangent between the curve and back tangent. This is not always a new point; it can be *pPC*.

nCC the ID of the point to be defined as the center of the curve.

nPT the ID of the point to be defined as the tangent between the curve and ahead tangent. This is not always a new point; it can be *pPT*.

r the radius of the curve; omit the radius if *pBT* equals *nPC* or if *pAT* equals *nPT*. You have the option to give the degree of curvature instead if you prefix the value with a *d*. For a circular arc, $dc=18000/(\pi r)$, where $r$ is the radius of the arc.

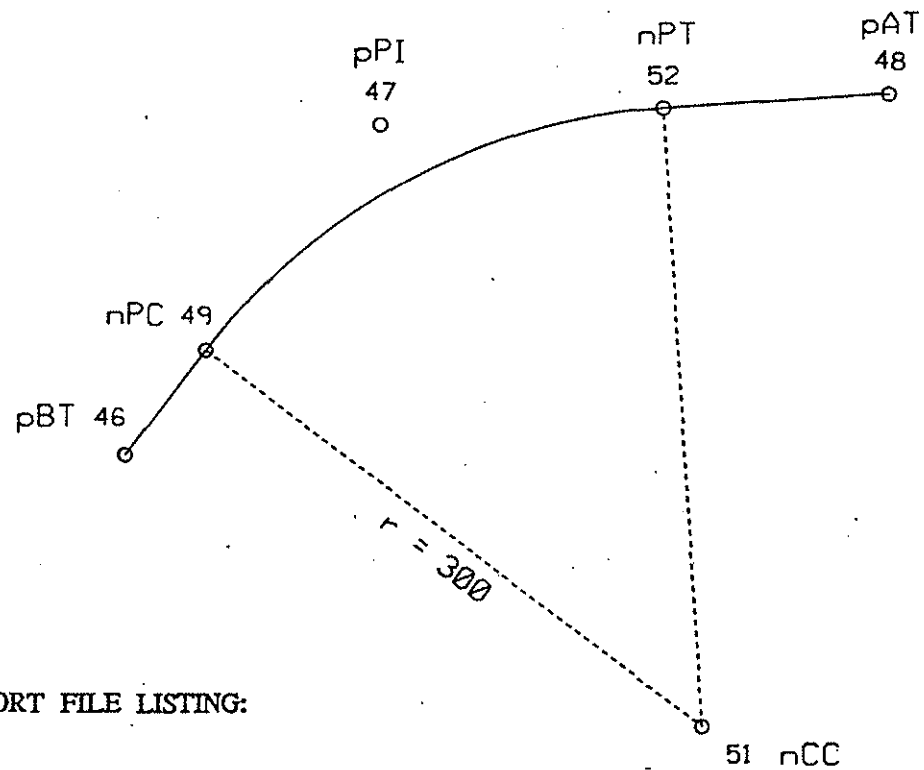
-OR-

EDGE:

pPC the known point of curvature.

pPOC a known point on the curve.

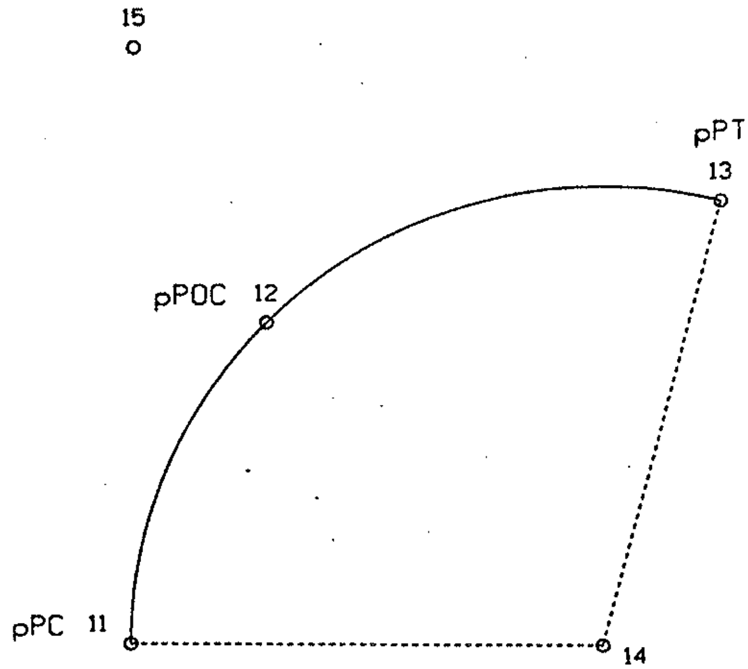
pPT the known point of tangency.




REPORT FILE LISTING:

```
FIT CURVE  pBT pPI pAT nPC nCC nPT /r
   46 47 48 0 0 0 300
 Circular Curve              (Clockwise)
        Length         256.8553FT  Degree       19- 5-54.9
        Radius         300.0000FT  Delta        49- 3-20.7
        Tangent        136.8942FT  Back         36-39-38.3
        External        29.7575FT  Forward      85-42-59.0
        Long Chord     249.0815FT               61-11-18.7
        Mid. Ord.       27.0722FT
 Point of Curvature
        49      3228.3824     3621.3939               DEFAULT
 Intersection of Tangents
        47      3338.1970     3703.1299       0.0000  DEFAULT
 Center of Curvature
        51      3049.2601     3862.0497               DEFAULT
 Point of Tangency
        52      3348.4221     3839.6417               DEFAULT
```

*Figure 9-8. FIT CURVE TANGENTS*




REPORT FILE LISTING:

```
FIT CURVE  pBT pPI pAT nPC nCC nPT /r
   11 12 13
 Circular Curve              (Clockwise)
        Length        451.9749FT  Degree      23- 7-12.1
        Radius        247.8188FT  Delta      104-29-48.2
        Tangent       320.0435FT  Back       180-54-30.4
        External      156.9552FT  Forward    105-24-18.6
        Long Chord    391.8868FT              53- 9-24.5
        Mid. Ord.      96.0943FT
 Point of Curvature
        11      3644.3959     3605.6559       0.0000 DEFAULT
 Intersection of Tangents
        15      3964.3992     3610.7301              DEFAULT
 Center of Curvature
        14      3640.4668     3853.4436              DEFAULT
 Point of Tangency
        13      3879.3819     3919.2749       0.0000 DEFAULT
```

*Figure 9-9. FIT CURVE EDGE*

### 9.2.9 SIMPLE CURVE

The SIMPLE CURVE command calculates the center of the curve *nCC*, given point *pBT* on the back tangent, the point *pPC* where the curve begins, the degree of curvature *dc*, the central angle *delta*, and the central angle direction *sign*. SIMPLE CURVE locates the coordinates of point *nPI*, where the two tangents to the circle intersect, and point *nPT*, where the curve ends.

Operator Sequence

1. Select or key in SIMPLE CURVE.

   *The system responds:*

   SIMPLE CURVE nCC pBT pPC nPI nPT dc delta sign

2. Identify the following:

   nCC the ID of the point to be defined as the center of the circle.

   pBT the known point on the back tangent.

   pPC the known point of transition from tangent to curve.

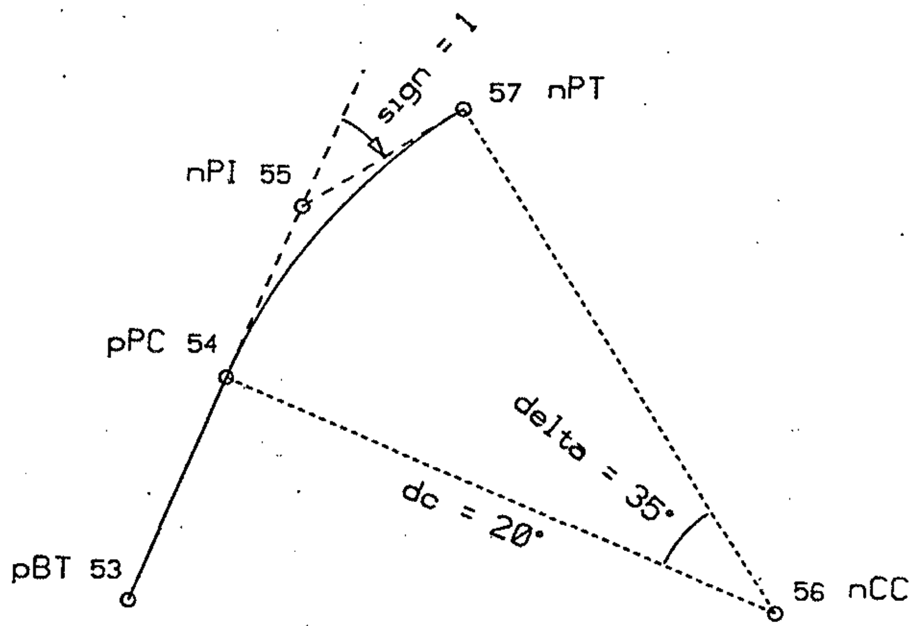
   nPI the ID of the point to be defined by the intersection of the two tangents.

   nPT the ID of the point to be defined as the transition from curve to tangent.

   dc the degree of curvature; for a circular arc, $dc$=18000/($\pi r$), where $r$ is the radius of the arc. You have the option to give the radius instead of the degree of curvature if you prefix the radius value with an $r$.

   delta the central angle of the curve (or deflection angle of the tangents). It can be expressed in the standard angle format, or can be calculated using the G delimiter.

   sign 1 for a clockwise curve (to the right); -1 for a counterclockwise curve (to the left).




REPORT FILE LISTING:

```
SIMPLE CURVE  nCC pBT pPC nPI nPT dc or r delta sign
  0 53 54 0 0 20. 35. 1
 Circular Curve         56   (Clockwise)
        Length          175.0000FT  Degree      20- 0- 0.0
        Radius          286.4789FT  Delta       35- 0- 0.0
        Tangent          90.3264FT  Back        21-48- 4.4
        External         13.9026FT  Forward     56-48- 4.4
        Long Chord      172.2917FT              39-18- 4.4
        Mid. Ord.        13.2591FT
 Point of Curvature
        54      3252.5249      5625.6059         0.0000 DEFAULT
 Intersection of Tangents
        55      3336.3910      5659.1520                DEFAULT
 Center of Curvature
        56      3146.1301      5891.5952                DEFAULT
 Point of Tangency
        57      3385.8488      5734.7350                DEFAULT
```

*Figure 9-10. SIMPLE CURVE*

### 9.2.10 SIMPLE SPIRAL IN and SIMPLE SPIRAL OUT

The graphics options SIMPLE SPIRAL IN and SIMPLE SPIRAL OUT are combined in the alpha SIMPLE SPIRAL command. The SIMPLE SPIRAL command(s) uses a set of known simple spiral parameters to calculate the other parameters associated with the spiral.

The known parameters for this command include: the curve identification number *cid*, a point *pBT* on the back tangent, the point *pTS* of transition from tangent to spiral for spiral in, or the point *pCS* of transition from curve to spiral for spiral out, length *ls* of the spiral, degree of curvature *dc*, and direction *sign*. The sign is taken from *pTS* to *nSC* for spiral in, and from *pCS* to *nST* for spiral out.

Note: *cid* has no significance to this command. To simply fill the field, you can enter any number from 0 to 999.

Operator Sequence

1. Select SIMPLE SPIRAL IN or SIMPLE SPIRAL OUT, or key in SIMPLE SPIRAL.

   *The system responds:*

   SIMPLE SPIRAL (in) cid pBT pTS nSIT nSC ls dc sign

   -OR-

   SIMPLE SPIRAL (out) cid pBT nST nSIT pCS -ls dc sign

2. Identify the following:

   cid — the curve identification number; a value between 0 and 999.

   pBT — a known point on the back tangent.

   pTS — the known point of transition from tangent to spiral (for spiral in).

   nST — the ID for the point to be defined as the transition from spiral to tangent (for spiral out).

   nSC — the ID for the point to be defined as the transition from spiral to circular curve (for spiral in).

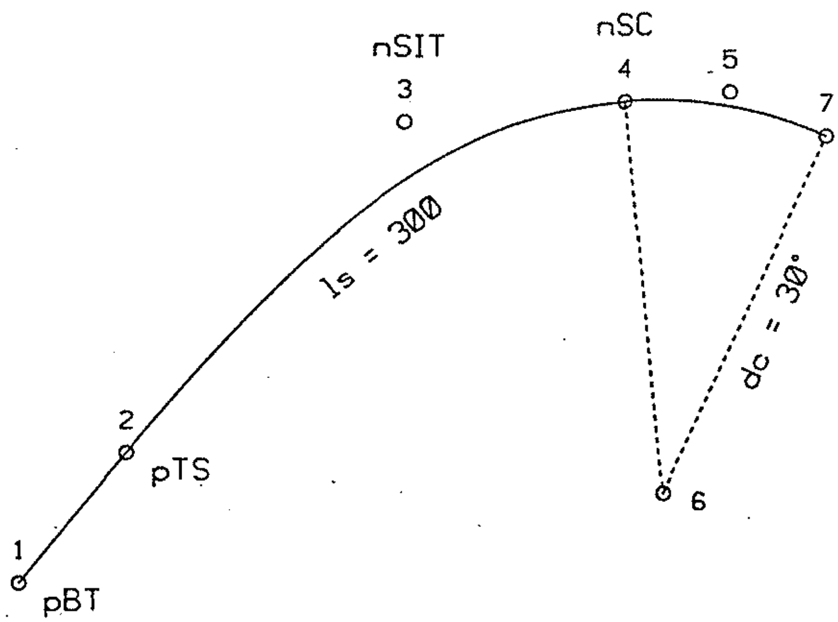
pCS the known point of transition from circular curve to spiral (for spiral out).

nSIT the ID for the point to be defined by the intersection of the spiral tangents.

ls the length of spiral for spiral in, measured along the spiral from *pTS* to *nSC*. -ls for the length of spiral for spiral out, measured along the spiral from *pCS* to *nST* (you must key in the -).

dc the degree of curvature; for a circular arc, $dc$=18000/($\pi r$), where $r$ is the radius of the arc. You have the option to give the radius instead of the degree of curvature if you prefix the radius value with an $r$.

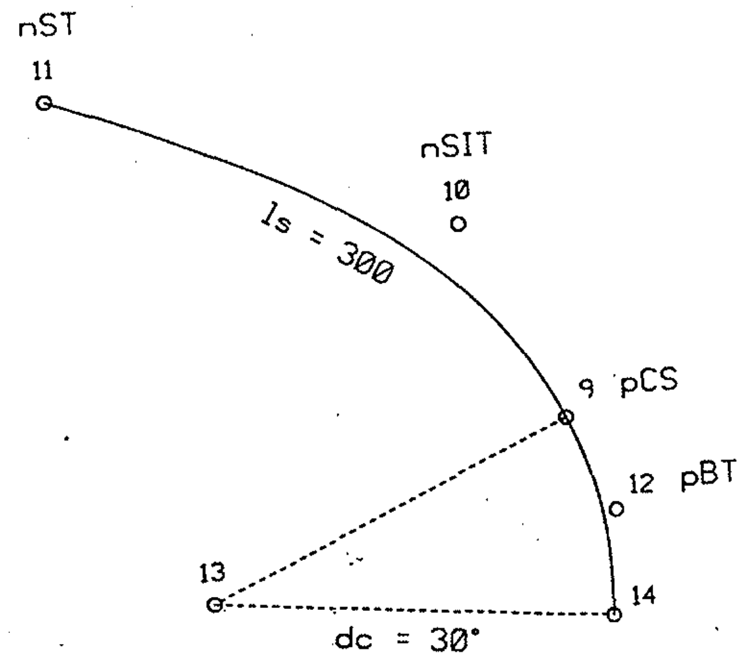
sign 1 for spiral clockwise (to the right); -1 for spiral counterclockwise (to the left).




REPORT FILE LISTING:

```
SIMPLE SPIRAL  cid pBT pTS nSIT nSC ls dc or r sign
    100 1 2 0 0 300 30. 1
  Spiral       100      (Clockwise)
         Length          300.0000FT  Angle       45- 0- 0.0
         Long Tan.       206.8690FT  Back        39-41- 6.5
         Short Tan.      106.2732FT  Forward     84-41- 6.5
         Long Chord      291.8556FT              54-36-20.3
  Tangent to Spiral
          2     3699.3609      1117.4459        0.0000 DEFAULT
  Intersection of Tangents
          3     3858.5600      1249.5459               DEFAULT
  Spiral to Curve
          4     3868.4040      1355.3622               DEFAULT
```

*Figure 9-11. SIMPLE SPIRAL IN*




REPORT FILE LISTING:

```
SIMPLE SPIRAL  cid pBT pTS nSIT nSC ls dc or r sign
   100 8 0 0 9 -300 30. -1
 Spiral        100     (Counterclockwise)
        Length         300.0000FT  Angle       45- 0- 0.0
        Long Tan.      206.8690FT  Back       289-17-22.8
        Short Tan.     106.2732FT  Forward    334-17-22.8
        Long Chord     291.8556FT             304-12-36.5
 Tangent to Spiral
        11     3903.4508     1643.2468              DEFAULT
 Intersection of Tangents
        10     3835.1129     1838.5023              DEFAULT
 Spiral to Curve
         9     3739.3609     1884.6059        0.0000 DEFAULT
```

*Figure 9-12. SIMPLE SPIRAL OUT*

### 9.2.11 POINTS ON SPIRAL IN and POINTS ON SPIRAL OUT

The graphics options POINTS ON SPIRAL IN and POINTS ON SPIRAL OUT are combined in the alpha POINTS ON SPIRAL command. The POINTS ON SPIRAL command(s) locates points along a simple spiral at even stations separated by distance *x*.

A spiral in is defined by its point *pTS* of transition from tangent to spiral, the station of *pTS* (*sTS*), the point *pSC* of transition from the spiral to the curve, and the distance *x*. A spiral out is defined by its point -*pST* of transition from spiral to tangent, the station of *pST* (*sST*), the point *pCS* of transition from curve to spiral, and distance *x*.

Additionally, you can use the optional parameter *off* to offset the points by a distance *off* to the right or to the left of the alignment. If you want to have points calculated on only a section of the alignment, you can use the optional parameters *sBEG* and *sEND*. ICS calculates points only on the portion of the alignment that falls between *sBEG* and *sEND* if you specify these parameters.

Note: If your spiral is part of an alignment, you will probably want to use the POINTS ON ALIGNMENT command instead of POINTS ON SPIRAL.

<u>Operator Sequence</u>

1. Select POINTS ON SPIRAL IN or POINTS ON SPIRAL OUT, or key in POINTS ON SPIRAL.

   *The system responds:*

   POINTS ON SPIRAL (in) pTS sTS pSIT pSC x n /off /sBEG /sEND

   -OR-

   POINTS ON SPIRAL (out) -pST sST pSIT pCS x n /off /sBEG /sEND

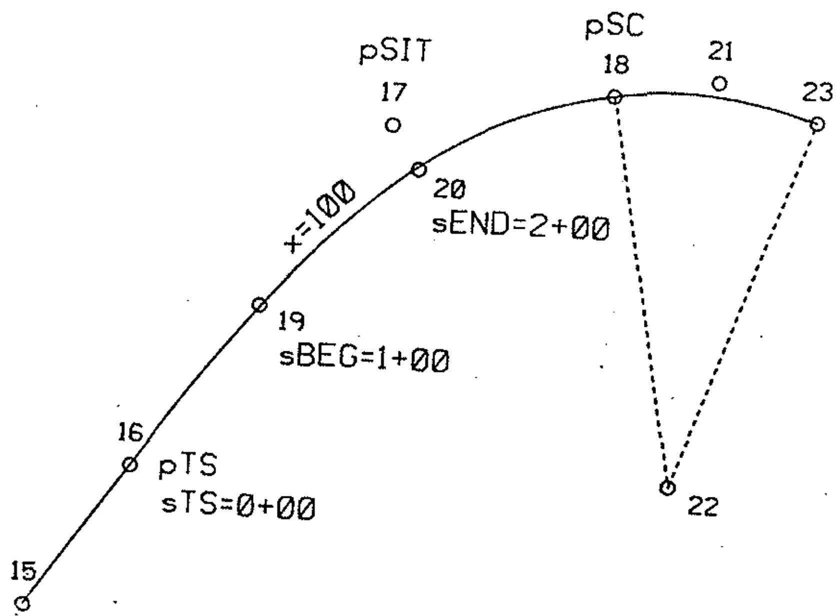
2. Identify the following:

   pTS the known point of transition from tangent to spiral (for spiral in).

   sTS the station number at point *pTS* (for spiral in). Do not include the +; for example, 10+65.3 should be keyed in as 1065.3.

   pSC the known point of transition from spiral to curve (for spiral in).

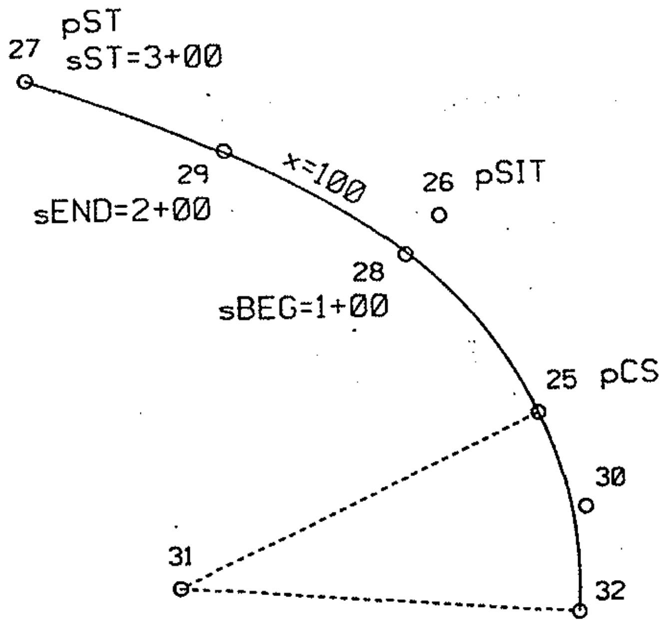
| | |
|---|---|
| pST | the known point of transition from spiral to tangent (for spiral out). |
| sST | the station number at point *pST* (for spiral out). Do not include the +; for example, 10+65.3 should be keyed in as 1065.3. |
| pCS | the known point of transition from curve to spiral (for spiral out). |
| pSIT | the predetermined point of intersection of the tangents. |
| x | the distance between points on the spiral. |
| n | the ID of the first point to be defined. |
| off | the distance to offset the calculated points. Positive to the right; negative to the left. |
| sBEG | the station number that defines the starting point for placing points. |
| sEND | the station number that defines the ending point for placing points. |




REPORT FILE LISTING:

```
 POINTS ON SPIRAL  pTS sTS pSIT pSC x n /off/sBEG sEND
    16 0 17 18 100 0 0 100 200
Tangent
       Point          Coordinates              Station      Arc.Length  Azimuth
                                                  FT            FT
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          19     3284.4632     1184.5313      1+00.0000     100.0000  39-41-39.3
          20     3353.2785     1256.6912      2+00.0000     200.0000  54-41-32.3
```

*Figure 9-13. POINTS ON SPIRAL IN*




REPORT FILE LISTING:

```
 POINTS ON SPIRAL  pTS sTS pSIT pSC x n /off/sBEG sEND
    -27 300 26 25 100 0 0 100 200
Tangent
       Point           Coordinates            Station      Arc Length  Azimuth
                                                FT            FT
 ---------------------------------------------------------------------------
         28     3314.1615     1814.8454      1+00.0000     99.9935 309-41-44.0
         29     3366.0568     1729.7015      2+00.0000    199.9935 294-41-51.0
```

*Figure 9-14. POINTS ON SPIRAL OUT*

### 9.2.12 COMPOUND SPIRAL

The COMPOUND SPIRAL command locates the critical points of a compound (combining) spiral with a known degree of curvature at each end. COMPOUND SPIRAL calculates the point *nSIT* of intersection between the two tangents and the point *nSC* of transition between the compound spiral and second curve. To define the compound spiral, you must specify a curve identification number *cid*, a point *pBT* on the back tangent, the point *pCS* of transition between the first curve and the compound spiral, the length *ls* of the arc, the degree of curvature of each curve (*dc1* and *dc2*), and the direction *sign* of curvature.

Note: *cid* has no significance to this command. To simply fill the field, you can enter any number from 0 to 999.

ICS does not calculate intersections of compound spirals with any other element. If you want to find an intersection point of a compound spiral, you must define a simple spiral of which the compound spiral is a part. (See Section 8.4 for more details.)

Operator Sequence

1. Select or key in COMPOUND SPIRAL.

   *The system responds:*

   COMPOUND SPIRAL cid pBT pCS nSIT nSC ls dc1 dc2 sign

2. Identify the following:

   cid — the curve identification number; a value from 0 to 999.

   pBT — a known point on the back tangent.

   pCS — the known point of transition from the first curve to the spiral.

   nSIT — the ID of the point to be defined by the intersection of the two tangents.

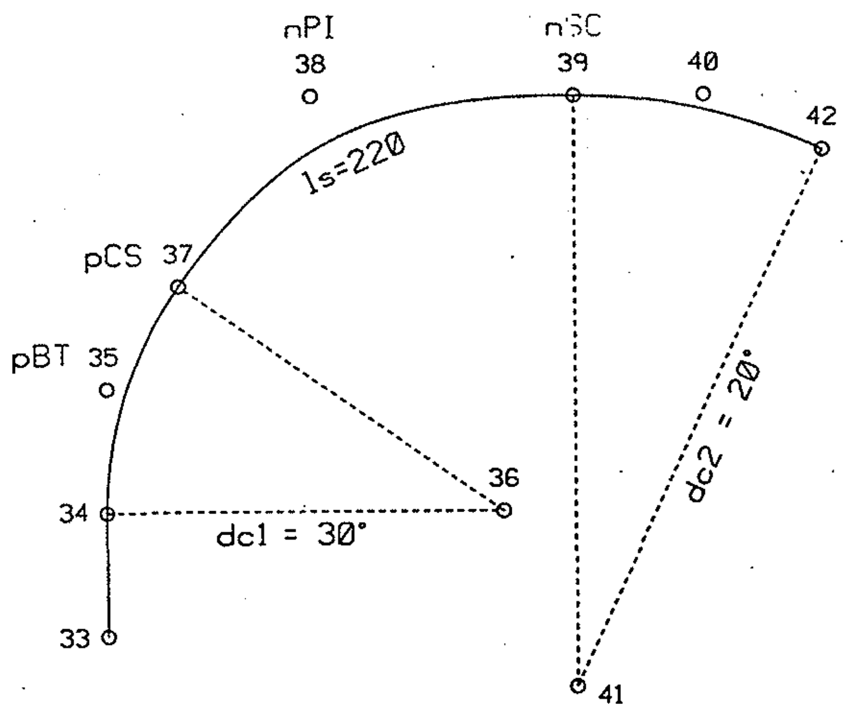
   nSC — the ID of the point to be defined as the transition from the spiral to the second curve.

   ls — the arc length of compound spiral, measured from *pCS* to *nSC*.

dc1 — the degree of curvature of the first circular curve; for a circular arc, $dc$=18000/($\pi r$), where $r$ is the radius of the arc. You have the option to give the radius instead of the degree of curvature if you prefix the radius value with an *r*.

dc2 — the degree of curvature of the second circular curve. You have the option to give the radius instead of the degree of curvature if you prefix the radius value with an *r*.

sign — 1 for a clockwise curve (to the right); -1 for a counterclockwise curve (to the left).




REPORT FILE LISTING:

```
COMPOUND SPIRAL  cld pBT pCS nPI nSC ls dc1 dc2 sign
   100 35 37 0 0 220 30. 20. 1
 Compound Spiral        100      (Clockwise)
        Length          220.0000FT
        Back Tan.       111.7799FT  Back         35- 0- 0.0
        Fwd. Tan.       126.6826FT  Forward      90- 0- 0.0
        Long Chord      211.6307FT               64-21-48.0
 Curve to Spiral
        37      3836.7939     2111.6533            DEFAULT
 Intersection of Tangents
        38      3928.3587     2175.7676            DEFAULT
 Spiral to Curve
        39      3928.3587     2302.4502            DEFAULT
```

*Figure 9-15. COMPOUND SPIRAL*

## 9.2.13 FIT ALIGNMENT

The FIT ALIGNMENT command calculates the critical points for a spiral in, circular curve, and spiral out combination, given: a point *pBT* on the back tangent, the point *pPI* of intersection between the two tangents (the outside tangents—entering the first spiral and exiting the last), the degree of curvature *dc*, the length (*ls1 and ls2*) of the first and second spirals, the total deflection angle *def1*, and the direction *sign*. FIT ALIGNMENT calculates several coordinates, as summarized in the listing of Step 2 below. These coordinates are numbered *nCC, nCC+1, nCC+2, . . . nCC+8.*

Operator Sequence

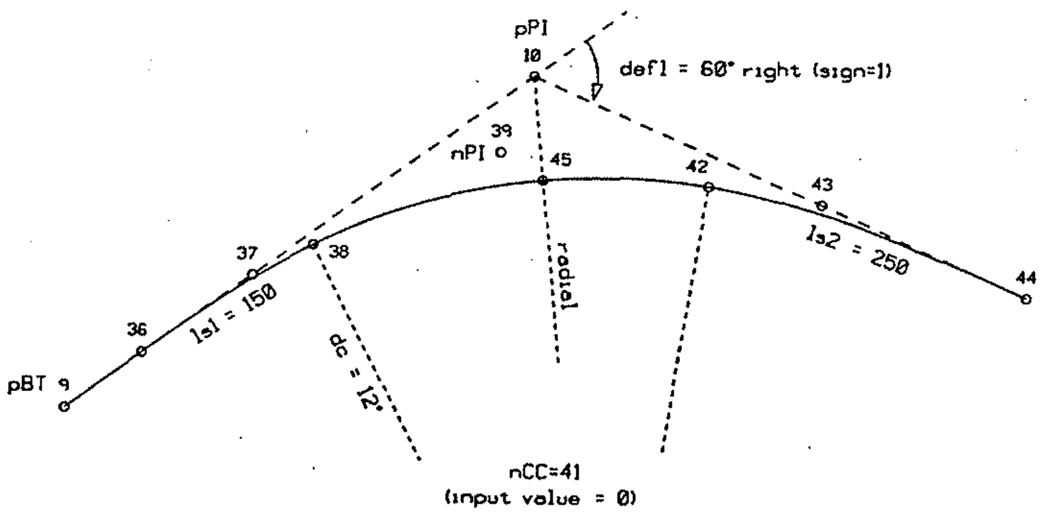
1. Select or key in FIT ALIGNMENT.

   *The system responds:*

   FIT ALIGNMENT nCC pBT pPI dc ls1 ls2 def1 sign

2. Identify the following:

| | |
|---|---|
| nCC | the ID of the point to be defined as the center of curvature. |
| pBT | the known point on the back tangent. |
| pPI | the known point of intersection of the outside tangents. |
| dc | the degree of curvature; for the circular arc, $dc=18000/(\pi r)$, where $r$ is the radius of the arc. You have the option to give the radius instead of the degree of curvature if you prefix the radius value with an *r*. |
| ls1 | the arc length of the first spiral. |
| ls2 | the arc length of the second spiral. |
| def1 | the total deflection angle for the tangents. It can be expressed in the standard angle format, or can be calculated using the G delimiter. |
| sign | 1 for spiral clockwise (to the right); -1 for spiral counterclockwise (to the left). |




REPORT FILE LISTING:

```
FIT ALIGNMENT  nCC pBT pPI dc or r ls1 ls2 defl sign
   0 9 10 12. 150 250 60. 1
 Spiral         0      (Clockwise)
        Length          150.0000FT  Angle        9- 0- 0.0
        Long Tan.       100.1296FT  Back        54- 4-17.7
        Short Tan.       50.1178FT  Forward     63- 4-17.7
        Long Chord      149.8356FT              57- 4-15.4
 Tangent to Spiral
        36     3162.5484     2708.2449                DEFAULT
 Intersection of Tangents
        37     3221.3018     2789.3249                DEFAULT
 Spiral to Curve
        38     3243.9990     2834.0086                DEFAULT
 Circular Curve          41   (Clockwise)
        Length          300.0000FT  Degree      12- 0- 0.0
        Radius          477.4648FT  Delta       36- 0- 0.0
        Tangent         155.1377FT  Back        63- 4-17.7
        External         24.5714FT  Forward     99- 4-17.7
        Long Chord      295.0895FT              81- 4-17.7
        Mid. Ord.        23.3688FT
```

*Figure 9-16. FIT ALIGNMENT*

REPORT FILE LISTING (continued):

```
 Point of Curvature
        38      3243.9990     2834.0086               DEFAULT
 Intersection of Tangents
        39      3314.2574     2972.3252               DEFAULT
 Center of Curvature
        41      2818.3043     3050.2415               DEFAULT
 Point of Tangency
        42      3289.7971     3125.5225               DEFAULT
 Spiral         0      (Clockwise)
        Length          250.0000FT  Angle      15- 0- 0.0
        Long Tan.       167.2690FT  Back      114- 4-17.7
        Short Tan.       83.8812FT  Forward    99- 4-17.7
        Long Chord      249.2393FT            109- 4-28.1
 Tangent to Spiral
        44      3208.3465     3361.0772               DEFAULT
 Intersection of Tangents
        43      3276.5717     3208.3544               DEFAULT
 Spiral to Curve
        42      3289.7971     3125.5225               DEFAULT
        TT Back         355.7527FT  Tot. Delta  60- 0- 0.0
        TT Forward      399.5034FT  Main Back   54- 4-17.7
        Total Len.      700.0000FT  Main Fwd.  114- 4-17.7
   Radial from          10 intersects alignment at        45
   Offset is        78.1488FT
        45      3293.5150     3003.9007               DEFAULT
```

*Figure 9-16. FIT ALIGNMENT (Continued)*

## 9.3 VERTICAL ALIGNMENT Commands

A vertical alignment consists of a series of alternating tangents and parabolic curves. You generate a vertical alignment in ICS using the VERTICAL START and VERTICAL END commands. Once you create the vertical alignment figure, you can assign the vertical alignment using the SET ALIGNMENT command. After assigning a vertical alignment, you can reference it without having to re-enter all of the points of vertical intersection (PVI's) and curve lengths. The only limit on the size of a vertical alignment is the size of the database. VERTICAL ALIGNMENT commands include the following:

| *Command* | *For* |
|---|---|
| SET PROFILE | setting up parameters ICS uses to placing a vertical alignment |
| VERTICAL START/END | creating a vertical alignment |
| CREATE PROFILE | plotting a simple centerline profile to the design file (graphics only) |
| EVEN STATIONS | computing and plotting points along a vertical alignment at even stations |
| ODD STATIONS | computing and plotting a point at a particular station on a vertical alignment |
| CURVE DRAIN | computing and plotting high and low elevations on the vertical alignment |

### 9.3.1 SET PROFILE

The SET PROFILE command sets up the parameters ICS uses to place a vertical alignment. You must use this command before all other VERTICAL ALIGNMENT commands to ensure proper placement of the vertical alignment. The parameters this command sets up include: the vertical exaggeration, the x and y coordinates to be used as the origin of the vertical alignment, the station on the alignment associated with the x-origin coordinate, and the elevation on the alignment associated with the y-origin coordinate.

The vertical exaggeration *vEX* is a ratio of *vEX*:1. For example, if the *vEX* is set to 10, the vertical scale will be multiplied by 10. A value of 1 plots the vertical alignment to scale without vertical exaggeration. Vertical exaggeration is usually necessary since most vertical alignments would appear flat if plotted to scale. The default value of *vEX* is 10, and you can change this default in the ICS.PAR parameter file.

The stationing origin *sORG* is the x coordinate of the vertical line on the design plane where the station *sta* is to be assigned. The vertical origin *zORG* is the y coordinate of the horizontal line on the design plane where the elevation *elev* is to be assigned. All subsequent vertical alignment plots will be relative to this point on the design plane.

Running SET PROFILE in Graphics

When you select the SET PROFILE command in graphics, ICS displays the SET PROFILE tutorial shown in Figure 9-17. This tutorial allows you to tailor the profile sheet. Through the tutorial you can define the following:

- Horizontal and Vertical Axes Extents
  The extents are defined by the beginning station on the active alignment, the ending station on the active alignment, the beginning elevation, and the ending elevation.

- Horizontal Stationing Major and Minor Intervals
  Both station text and major tic marks are placed at major intervals. Minor tic marks are placed at minor intervals.

- Vertical Stationing Major and Minor Intervals
  Both elevation text and major tic marks are placed at major intervals. Minor tic marks are placed at minor intervals.

- Vertical Exaggeration Factor

- Station Label Format
  The format can be full such as 100+00, or short such as 100. The current value is marked with an *X*.

- Station Label Orientation
  The orientation can be parallel or perpendicular to the horizontal axis. The current value is marked with an *X*.

- Number of Vertical Axes
  The axes displayed can be the left side only, or both the left and right sides. The current value is marked with an *X*.

The items that are marked with *X*'s can be set by placing the cursor over the value you want and pressing <D>. The other items require key-ins. You set these values by placing the cursor over the current values, pressing <D>, and keying in the new values.

After you set up the profile sheet, position the cursor over the COMPLETE box on the tutorial, and press <D>. ICS then prompts you to identify the x,y location in the design file for the horizontal (x) and vertical (y) origin of the profile sheet. ICS places the sheet in the design file at the specified location. You can exit the SET PROFILE tutorial at any time prior to placing the profile sheet in the design file by placing the cursor over the QUIT box and pressing <D>.

Note: You must define the active horizontal and vertical alignments using the SET ALIGNMENT ACTIVATE command. SET PROFILE no longer defines the active alignments.

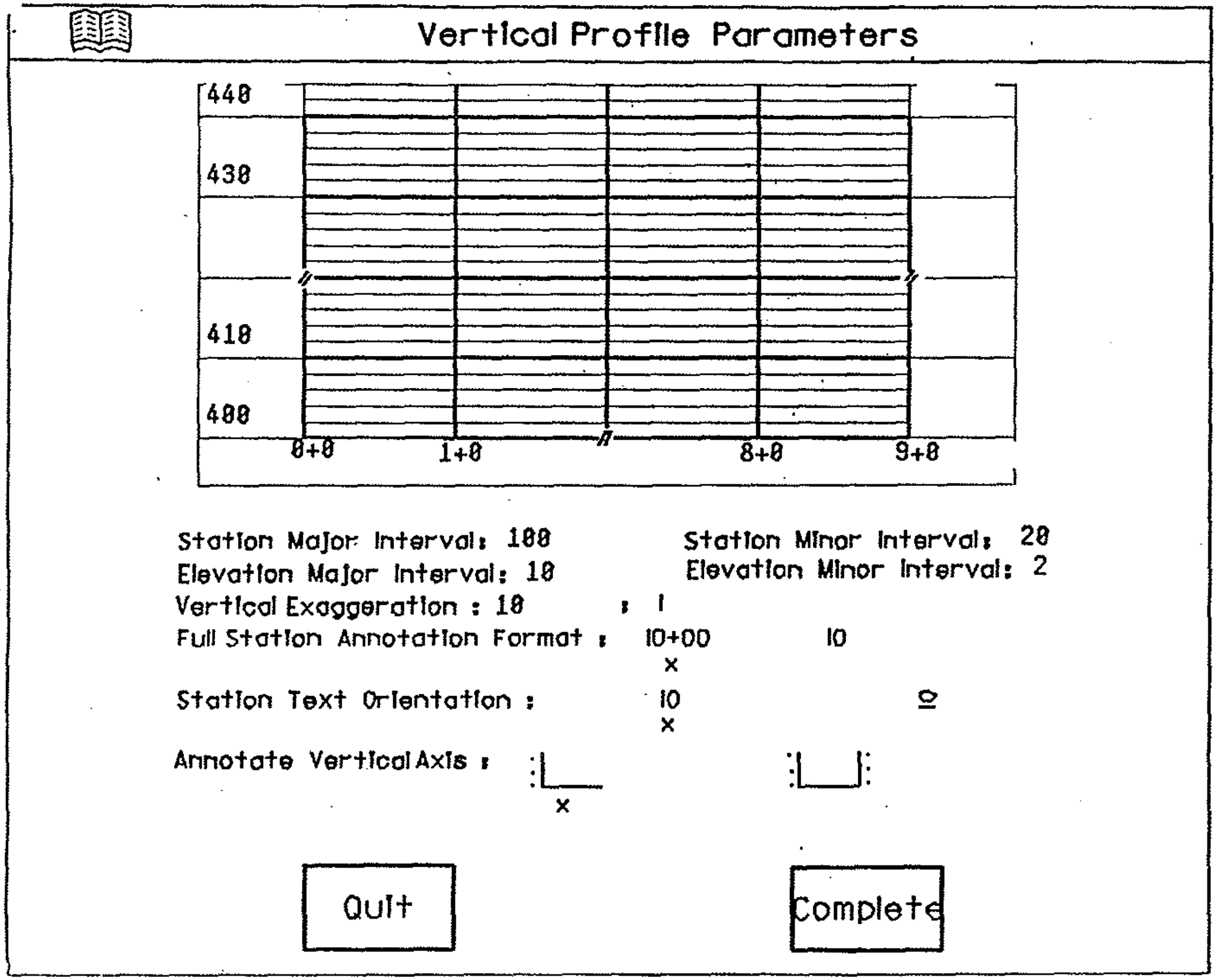



*Figure 9-17. The SET PROFILE Tutorial*

Setup

Remember to run the SET ALIGNMENT command prior to selecting SET PROFILE.

Operator Sequence

1. Select or key in SET PROFILE.

   *In graphics, the system displays the SET PROFILE tutorial shown in Figure 9-17.* Set the parameters as described above, and you are finished. Do not go to Step 2.

   *In alpha, the system responds:*

   SET PROFILE vEX sORG sta zORG elev

2. Identify the following:

   vEX the ratio of vertical exaggeration.

   sORG the stationing origin x coordinate in the design file.

   sta the station of this origin.

   zORG the elevation origin y coordinate in the design file.

   elev the elevation of this origin.

REPORT FILE LISTING:

```
STORE FIGURE  nfg des /pAL sAL or  fg des /pAL sAL
   0 (1 2 C4R 5 S6 7 8 C10R 11)
SET ALIGNMENT  Hfg/pAL sAL /Vfg
   1 1 100
%SAL-I-HORALAS, horizontal alignment assigned successfully to figure          1
SET PROFILE  vEX sORG sta zORG elev
   10 -7160.6899 0 1540.3669 400
%SPR-I-VEXAG, vertical exaggeration is          10.0000: 1
%SPR-I-STAORG, station origin is 0+00            at        -7160.6899
%SPR-I-ELEORG, elevation origin is         400.0000 at         1540.3669
```

Note: See Figure 9-20 for the graphics display.

*Figure 9-18. SET PROFILE*

## 9.3.2 VERTICAL START/END

The alpha commands VERTICAL START and VERTICAL END are combined in the graphics command VERTICAL START/END. When you select VERTICAL START/END in graphics, the VERTICAL START prompt appears. When you press <R> to end VERTICAL START, the VERTICAL END prompt appears. Both the VERTICAL START and VERTICAL END commands prompt you for the points *nPVI* of vertical intersection, the stations *sAL* of those points, and the elevations *elev* of those points.

In graphics, use the VERTICAL START command to define all the vertical points of intersection. In alpha, use the VERTICAL END command to define the last point of vertical intersection. Both graphics and alpha use VERTICAL END to assign an ID to the vertical figure. Figure 9-19 shows the combined output from VERTICAL START and VERTICAL END.

Note: The system does not display any output until after the VERTICAL END command completes processing.

VERTICAL START

The VERTICAL START command is the first command you use to create vertical alignments. VERTICAL START enters all of the points of vertical intersection (PVI), except for the last point. The VERTICAL END command enters the last point on a vertical alignment.

When you run VERTICAL START, you are prompted for the PVI ID number, the station of the PVI, and the elevation of the PVI. You must omit the vertical curve length *Vcl1* for the first PVI. VERTICAL START accepts unequal tangent lengths for subsequent vertical curve processing. If you specify *Vcl2*, ICS considers *Vcl1* to be the first vertical tangent and *Vcl2* to be the second vertical tangent. The entire vertical curve length is then equal to *Vcl1* + *Vcl2*. If you do not specify *Vcl2*, then *Vcl1* represents the total vertical curve length, and the tangents have equal lengths.

In alpha and graphics you can use key-ins to identify the vertical points of intersection. In graphics you can also use data points. Valid key-ins for defining points of vertical intersection include:

- station and elevation
- station and grade %
- distance and grade %

**Note: If you use grades, you must *suffix* the grade key-in with a percent symbol (%). If you use a distance, you must *prefix* the distance with an upper or lowercase *D*.**

For example, the following key-ins are valid:

12500 110.0 * next PVI is station 125+00, elevation is 110.0

12500 2% * next PVI is station 125+00, elevation is computed as 2% grade from last PVI

D200 -5% * next PVI is a distance 200 from last PVI, elevation is computed as -5% from the previous PVI

As you enter PVI's, ICS prompts you for the PVI ID and the vertical curve length. You key in zero (0) or press <D> for the next available PVI. Remember, you can omit keying in the vertical curve length only for the first PVI. VERTICAL START computes and stores the results in the database.

Note: All stationing must be increasing and the *PVI*'s must be entered in increasing order.

Beware of overlapping vertical curves. Parabolic curves used in vertical curve commands must not overlap. Compound curves can be stored in the vertical alignment by entering zero (0) as the *Vcl1* for the second curve.

Operator Sequence

1. Select VERTICAL START/END or key in VERTICAL START.

   *The system responds:*

   VERTICAL START nPVI sAL elev /Vcl1 /Vcl2

3. Identify the following:

   nPVI the ID of the point to be defined as the *PVI*.

   sAL the station at *nPVI*.

   elev the elevation of *nPVI*.

   Vcl1 the first vertical curve length.

   Vcl2 the second vertical curve length, for unequal tangents.

VERTICAL END

When you select VERTICAL START/END in graphics, the VERTICAL START prompt appears. When you press <R> to end VERTICAL START, the VERTICAL END prompt appears. The VERTICAL END command enters the final *PVI* and figure ID *Vfg* for a vertical alignment.

The figure ID must be a positive number in the range from 0 to 2,147,483,647. When you assign 0 to *Vfg*, ICS automatically assigns the next available figure number to the figure. When you assign a valid non-zero value to *Vfg*, ICS stores a figure with the number you assign.

The figure ID you enter in the VERTICAL END command can be used in the SET ALIGNMENT command in future ICS runs. In graphics you can enter the last PVI by placing a data point.

You can place other commands between the VERTICAL START and VERTICAL END commands, but there must not be any other vertical alignment commands. ICS plots the vertical alignment after you run VERTICAL END.

Operator Sequence

1. In alpha, key in VERTICAL END. In graphics, select VERTICAL START/END, and press <R> to end VERTICAL START.

   Note: In graphics, the last *nPVI, sAL,* and ***elev*** values defined with VERTICAL START are used by the VERTICAL END command. You will not be prompted for these values.

   *The system responds:*

   VERTICAL END nPVI sAL elev Vfg /n

3. Identify the following:

   nPVI the ID of the point to be defined as the last *PVI*.

   sAL the station of the last *PVI*.

   elev the elevation of the last *PVI*.

   Vfg the ID of the figure to be defined by the vertical alignment generated.

   n the ID of the point to be defined.

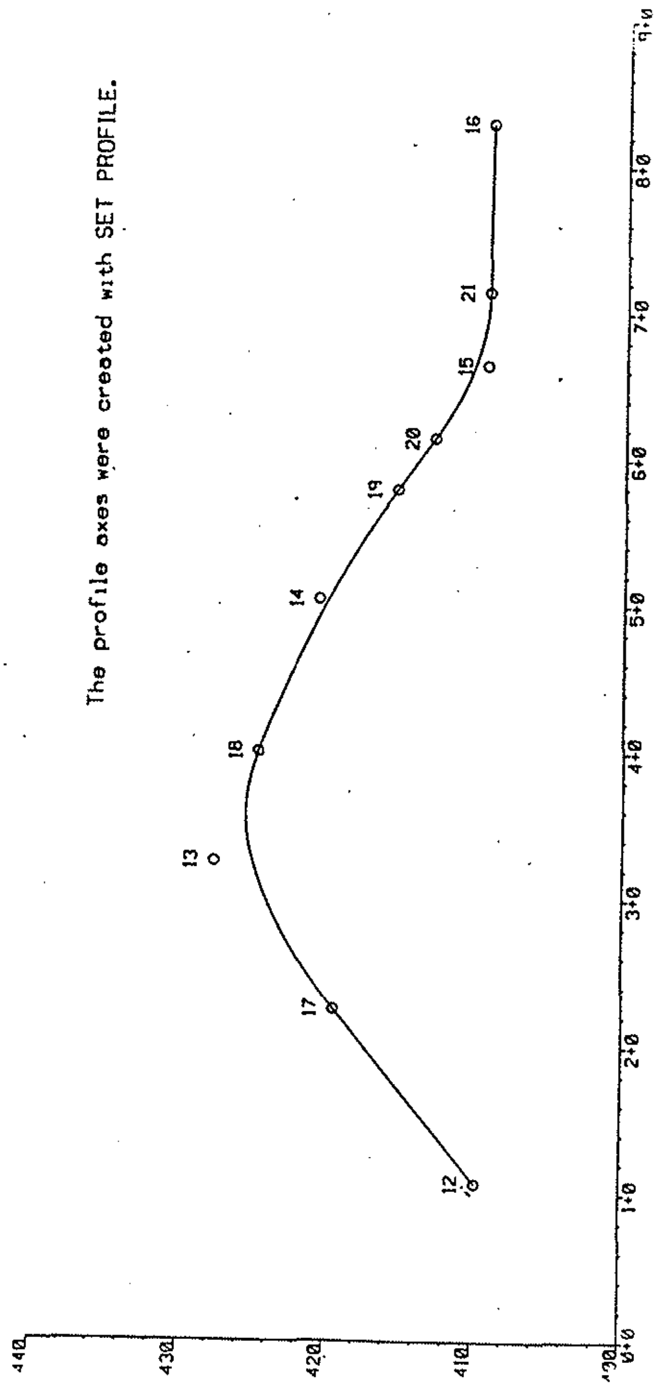



*Figure 9-19. VERTICAL START and VERTICAL END*

REPORT FILE LISTING:

```
SET ALIGNMENT  Hfg/pAL sAL /Vfg
   1 1 100
%SAL-I-HORALAS, horizontal alignment assigned successfully to figure          1
VERTICAL START  nPVI sAL elev /VCL1 /VCL2
   0 -7053.3199 1638.3429 0 0 1
        12      1177.8554    -7061.6280      409.7976 DEFAULT
VERTICAL START  nPVI sAL elev /VCL1 /VCL2
   0 -6834.5529 1818.1879 100 75 1
        13      1342.3088    -6922.5838      427.7821 DEFAULT
VERTICAL START  nPVI sAL elev /VCL1 /VCL2
   0 -6656.0499 1748.3969 0 75 1
        14      1365.8101    -6747.5263      420.8030 DEFAULT
VERTICAL START  nPVI sAL elev /VCL1 /VCL2
   0 160 -7 100 0  3
        15      1358.5150    -6587.6927      409.6030 DEFAULT
VERTICAL END  nPVI sAL elev Vfg /n
   0 -6331.2549 1634.3160 0 0 1
        16      1308.4407    -6433.0492      409.3949 DEFAULT
%VND-I-VERGEN, vertical alignment figure generated
         2 (12 17 V13 18 V14 19 20 V15 21 16)
```

*Figure 9-19. VERTICAL START and VERTICAL END (Continued)*

### 9.3.3 CREATE PROFILE (Graphics Only)

The CREATE PROFILE command plots a profile, given a description *des* of points with defined elevations. Only centerline points and distances are used in the creation of the profile, which is plotted as a complex string connecting consecutive centerline points in *des*.

Note: You must run the SET PROFILE command prior to this command in order to define the origin of the profile.

Additionally, you have the option to create a vertical figure *Vfg* using this command. You must run the SET ALIGNMENT and SET PROFILE commands before CREATE PROFILE if you want to generate a vertical figure.

Note: CREATE PROFILE uses the stationing on the active alignment that is in effect when the SET PROFILE command was executed.

Operator Sequence

1. Select CREATE PROFILE.

   *The system responds:*

   CREATE PROFILE des /Vfg /fnum or lfname

2. Identify the following:

   des — description of the centerline. Points with undefined elevations are ignored. Only points on the centerline are included in the profile: arc center points and spiral intersection points are ignored.

   Vfg — the ID of the figure to be defined.

   fnum — the feature number to be used when plotting.

   lfname — the feature name to be used when plotting.

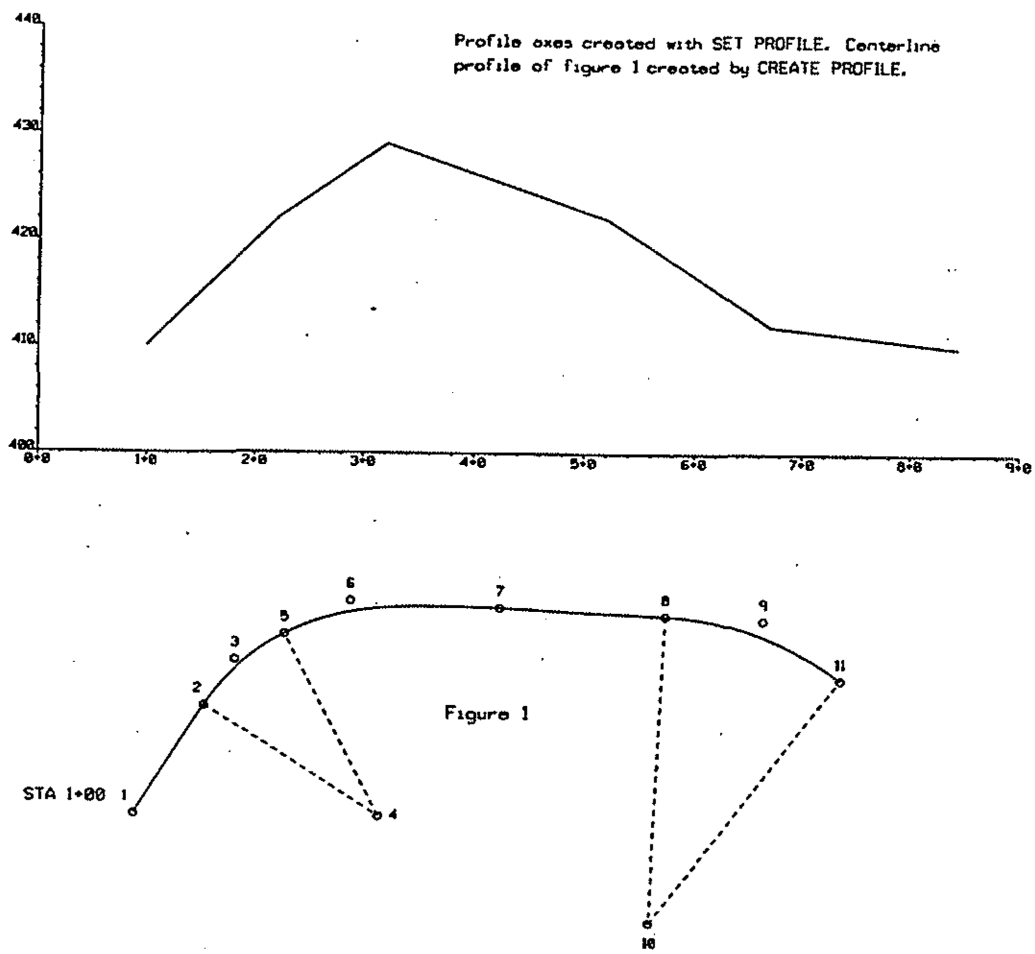



*Figure 9-20. CREATE PROFILE*

## 9.3.4 EVEN STATIONS

The EVEN STATIONS command computes and plots points along a vertical alignment at even intervals of *x*. The interval between points cannot be zero. These points are stored in the database and plotted along the vertical alignment. If you specify *n*, the first point generated is assigned the ID *n*, and the ID's of subsequent points generated are assigned the ID's *n*+1, *n*+2, etc. If you omit *n*, ICS uses the next available point ID's.

If you want to have points calculated on only a section of the alignment, you can use the optional parameters *sBEG* and *sEND*. ICS calculates points only on the portion of the alignment that falls between stations *sBEG* and *sEND* if you specify these parameters. You can assign zero (0) to *sBEG* and *sEND* as a wild card to indicate from the beginning, or to the end, respectively.

<u>Operator Sequence</u>

1. Select or key in EVEN STATIONS.

   *The system responds:*

   EVEN STATIONS x /sBEG sEND /n

2. Identify the following:

   x — the station interval between points.

   sBEG — the beginning station.

   sEND — the ending station.

   n — the ID for the first point to be defined.

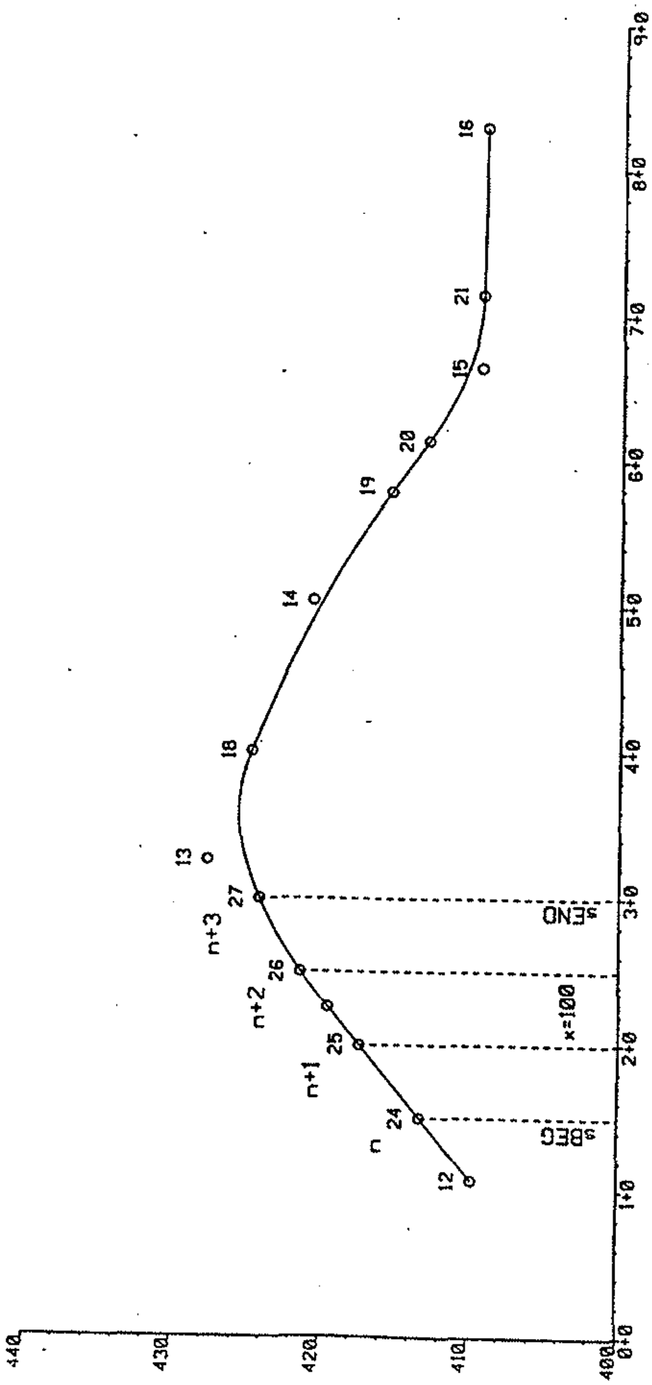



*Figure 9-21.* EVEN STATIONS

REPORT FILE LISTING:

```
EVEN STATIONS  x /sBEG /sEND /n
   50 150 350
     STATION            PT          COORDINATES
1+50                     24   1213.7613    -7038.6480      413.3021
2+00                     25   1255.8748    -7011.6951      417.4126
2+50                     26   1296.6663    -6982.8935      421.3750
3+00                     27   1329.4009    -6945.2877      424.2152
3+50                     28   1351.3914    -6900.5308      425.6407
```

*Figure 9-21. EVEN STATIONS (Continued)*

## 9.3.5 ODD STATIONS

The ODD STATIONS command stores and plots a point at station *sAL* on a vertical alignment. If you specify *n*, the generated point is assigned the ID *n*. If you omit *n*, ICS uses the next available point ID.

Operator Sequence

1. Select or key in ODD STATIONS.

   *The system responds:*

   ODD STATIONS sAL /n

2. Identify the following:

   sAL  the station of the new point.

   n  the ID of the first point to be defined.

REPORT FILE LISTING:

```
ODD STATIONS  sAL /n
   300 55 0
        55     1329.4009    -6945.2877       424.2152 DEFAULT
ODD STATIONS  sAL /n
   150 0 0
        23     1213.7613    -7038.6480       413.3021 DEFAULT
ODD STATIONS  sAL /n
   800 56 0
        56     1323.7951    -6458.1469       409.4321 DEFAULT
```

*Figure 9-22. ODD STATIONS*

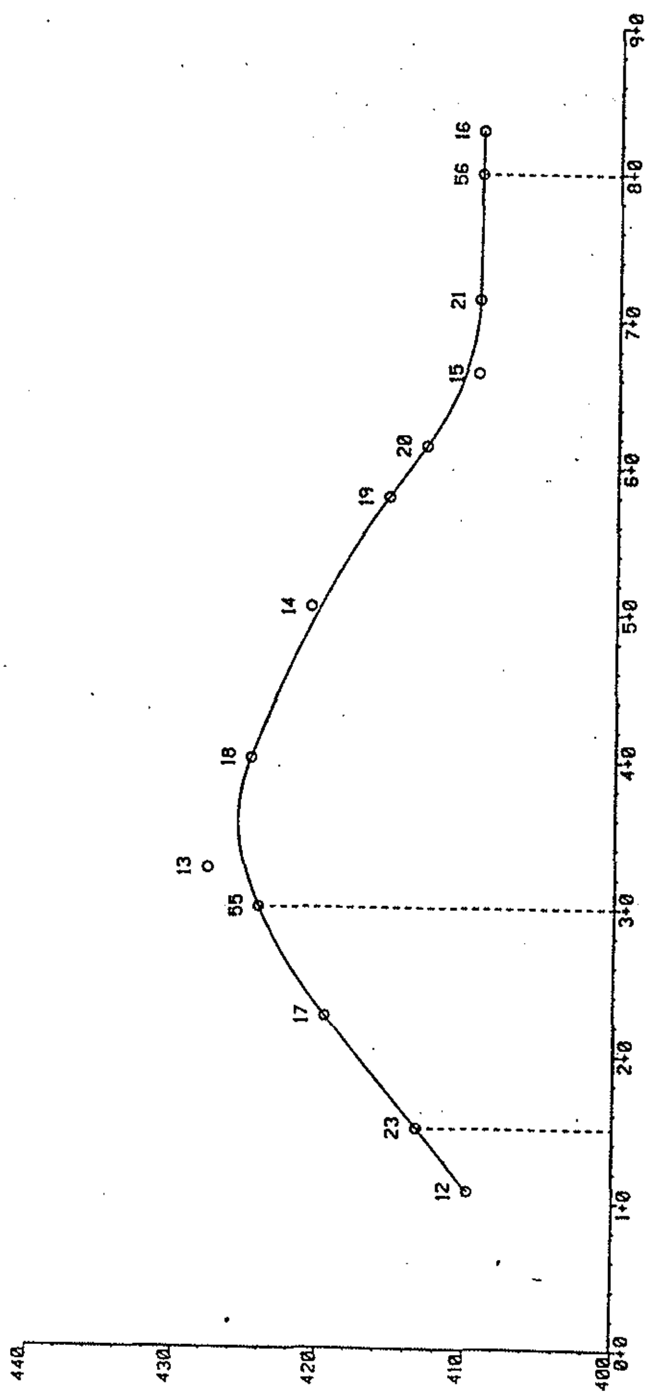



*Figure 9-22.* ***ODD STATIONS (Continued)***

## 9.3.6 CURVE DRAIN

The CURVE DRAIN command computes and labels all high and low elevations on the vertical alignment between *sBEG* and *sEND.* The high and low points are stored as a point ID beginning at *n.* If you omit *n,* ICS uses the next available point ID. The points are plotted and labeled as HI/LO with station and elevation. You can assign zero (0) to *sBEG* and *sEND* as a wild card to indicate from the beginning, or to the end, respectively.

Operator Sequence

1. Select or key in CURVE DRAIN.

   *The system responds:*

   CURVE DRAIN sBEG /sEND /n

2. Identify the following:

   sBEG the beginning station.

   sEND the ending station.

   n the ID of the first point to be defined.

REPORT FILE LISTING:

```
CURVE DRAIN  sBEG /sEND /n
  0 900 0
  HI/LOW is at station 3+58.8342       elevation        425.6767
        22      1354.1328    -6892.1334      425.6767 DEFAULT
     Plotting feature is       DEFAULT                          [    1]
```

*Figure 9-23. CURVE DRAIN*

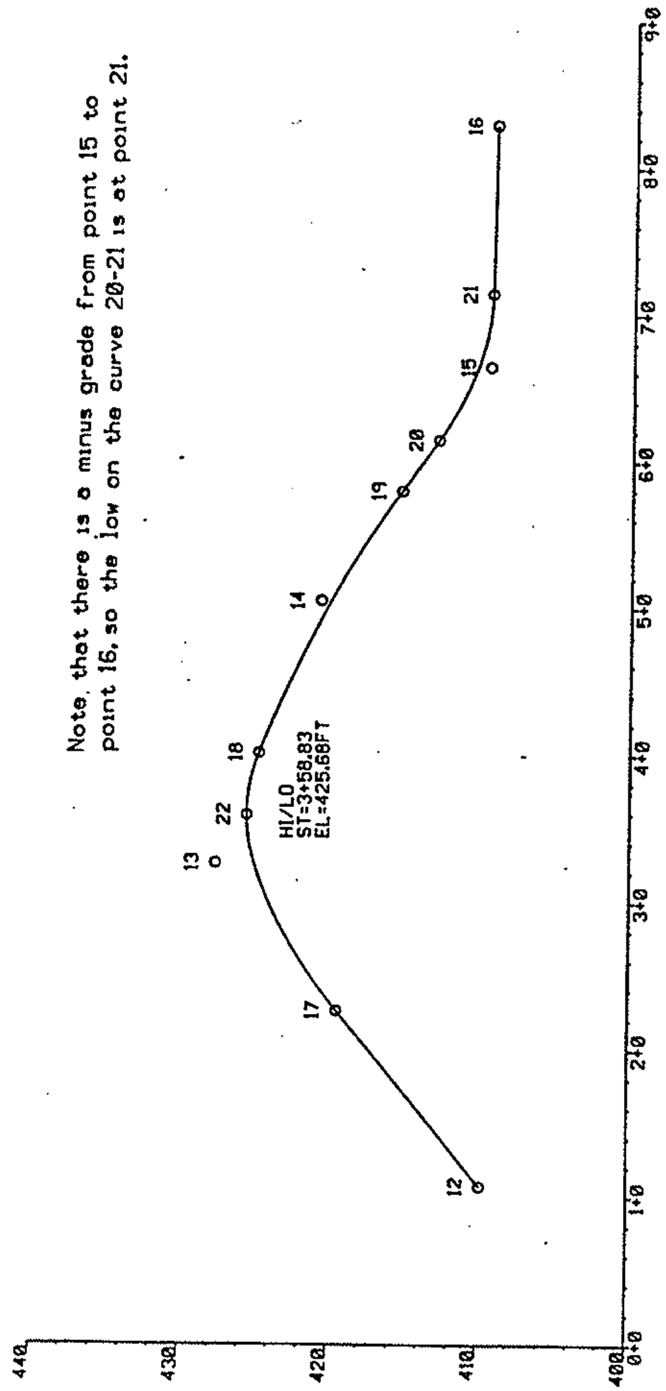



*Figure 9-23.* ***CURVE DRAIN*** *(Continued)*

# ADJUST/TRANSFORM Commands

10

# 10. ADJUST/TRANSFORM Commands

The ADJUST/TRANSFORM menu block includes commands that let you alter the values of point coordinates using several different methods. With these commands you can do traverse adjustments, vertical adjustments, area adjustments, and Helmert transformations. The commands in this menu block alter values in the database. If you want to copy the points that will be altered to other points, use the COPY POINTS command (see Section 8.1.3 for details).

## 10.1 Adjustment Commands

ICS has traverse adjustment commands, a vertical adjustment command, and area adjustment commands. Adjustment commands include the following:

| *Command* | *For* |
|---|---|
| TRAVERSE ADJUSTMENT: COMPASS, CRANDALL, and TRANSIT | adjusting or closing a traverse |
| VERTICAL ADJUST | adjusting level traverses |
| ADJUST AREA: ROTATE, PARALLEL, and NORMAL | adjusting the area of closed figures |

### 10.1.1 TRAVERSE ADJUSTMENT: COMPASS, CRANDALL, and TRANSIT

The COMPASS, CRANDALL, and TRANSIT commands adjust or close traverses. The commands calculate the positional error between two points known to occupy the same position. The first point is the endpoint in the traverse, and the second point *pFOC* is the corresponding field observation point.

The TRAVERSE ADJUSTMENT commands distribute the positional error among the points of the traverse described in *des*. The point numbers of the points occupied as observation stations should be included in the description *des* or in a figure you have previously stored. The last point in the description is the endpoint in the traverse, and *pFOC* will be adjusted to close to it.

ICS displays allowable and actual closures on the terminal screen. If the positional error tolerance test *Dtol* is met, ICS adjusts the positions contained in *des* using the selected traverse adjustment rule.

Note: If the positional error tolerance test is not met, the command does not execute unless the angular misclosure *Atol* is specified and met, in which case the new coordinates will reflect adjusted directions only. If you specify the optional *Atol* parameter, you must also specify the last sighted point *pLS* and the field position of the last sighted point *pFLS*. These two points provide the closing angle required to compute an angular misclosure.

Additionally, you have the option to set the scale reduction factor by using the parameters *Tdis* and *Mdis*. *Tdis* is the measured terrain distance between two points, and *Mdis* is the mapping plane distance between the same two known points.

After you have entered all the required data, ICS prints the adjusted data for angular adjustment (if applicable) and the selected adjustment. Then ICS prompts you to accept or reject the adjusted coordinates. If you accept the adjusted coordinates, ICS stores these adjusted coordinates for the points. If you do not want to be prompted, you can change a parameter in the ICS.PAR parameter file (see Section 3.6).

ICS protects against dividing by zero. This error occurs if the traverse adjustment commands encounter a perfect closure. These commands also protect against an integer flow error if a closure better than 1/2,000,000,000 occurs. The highest precision ICS will display is 1 in 2 billion (1/2,000,000,000).

Note: These commands use current values of the coordinates of points to compute new positions of the same points. Make sure that no current values are needed, since they are replaced in this process. Use the COPY POINTS command to save points that you do not want altered.

These commands delete the final occupied station point *pFOC* and the field location of the point last sighted *pFLS*. You should copy these points before executing this command if you want to use these points again.

Setup

Before executing any of the traverse adjustment commands, compute the approximate (field) positions by using a POINT LOCATION command.

Operator Sequence

1. Select or key in COMPASS, CRANDALL, or TRANSIT.

   *The system responds:*

   COMPASS CLOSURE des pFOC Dtol /pLS pFLS Atol /Tdis Mdis

   -OR-

   CRANDALL CLOSURE des pFOC Dtol /pLS pFLS Atol /Tdis Mdis

   -OR-

   TRANSIT CLOSURE des pFOC Dtol /pLS pFLS Atol /Tdis Mdis

2. Identify the following:

   des — the figure ID or list of points defining the traverse to be adjusted. If the traverse is a loop, the last point in *des* should match the first point. Also, make sure the last point corresponds to *pFOC*; the last point in *des* should be the actual location of the last occupied point.

   pFOC — the ID of the known point that is the field location of the final occupied station; this point is computed from field observational data. If the positional closure is successful, this point is deleted.

Dtol — the maximum allowable positional error between point number *pFOC* and the last point number in *des*. You can specify the tolerance as a distance in master units or as a ratio of the traverse length over that distance. For example, Dtol = 0.15 specifies that the traverse must close within 0.15 master units. If you specify *Dtol* to be 20000, the ratio between the total distance traveled by the traverse and the positional error must not fall below 20000. If this tolerance test is not met, the command does not execute unless *Atol* is specified and is met. If *Atol* is met, the new coordinates reflect adjusted directions only.

Note: If *Dtol* is less than 1000, ICS assumes it is a distance in master units; if *Dtol* is large, ICS assumes it is a ratio to 1.

pLS — the ID of the control or traverse point last sighted.

pFLS — the ID of the field location of *pLS* (the location of *pLS* computed using closing measurements). If the positional closure is successful this point is deleted.

Atol — the maximum allowable angular misclosure. If the angular misclosure exceeds this value, the command does not execute.

Tdis — the measured terrain distance between any two points. If you omit the fields *Tdis* and *Mdis*, the reduction factor is set equal to one.

Mdis — the mapping plane distance between the same two known points used for *Tdis*.

3. **Accept Adjusted Coordinates (y/n) ?**

Key in a *y* if you want the adjusted coordinates to be stored in the database.

-OR-

Key in an *n* if you want to reject the adjusted coordinates. *The unadjusted coordinates will be retained in the database.*

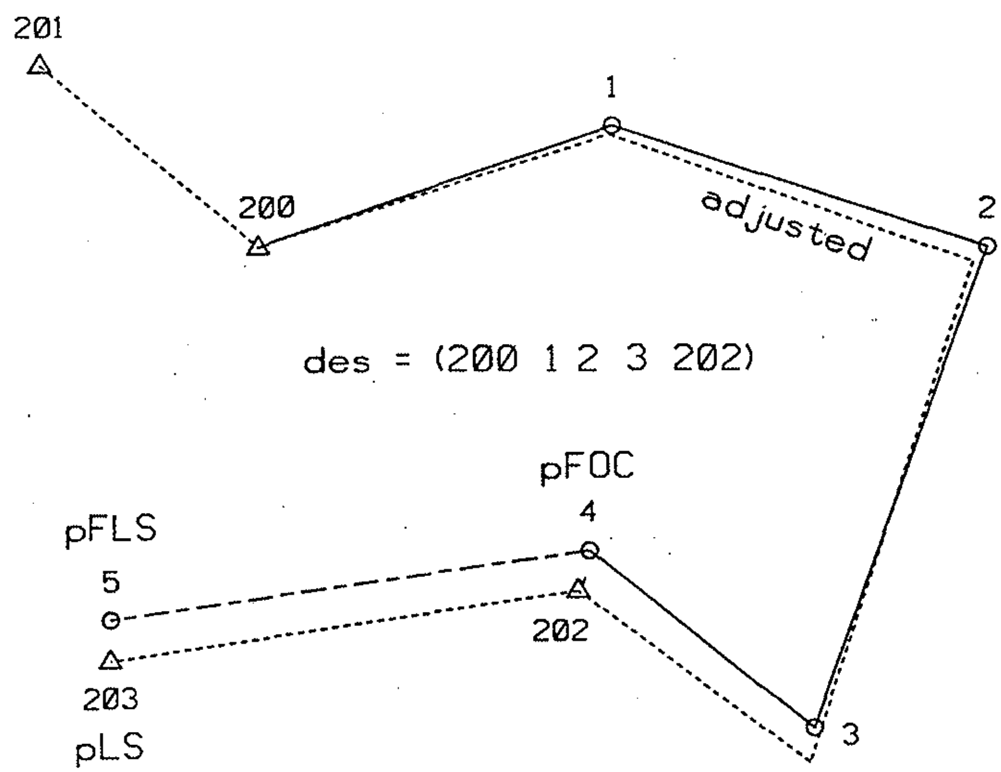



*Figure 10-1. COMPASS CLOSURE*

REPORT FILE LISTING:

```
COMPASS CLOSURE   des pFOC Dtol /pLS pFLS Atol /Tdis Mdis
   (200 1 2 3 202) 4 20 203 5 5.
  Angular Misclosure          000-21-43.1  Allowable              005-00- 0.0
  Correction                  000-04-20.6
  Traverse Length                637.1550
  Linear  Misclosure     -5.0383  -16.2731
  Total            17.0352 (1:         37) Allowable   20.0000 (1:        32)
  Length and Lateral Errors
  Total                          -14.1718            9.4529
-------Compass  Adjustment  Summary--------
     Point           Coordinates
---------------------------------------------
        200     4322.8399    6150.6119        0.0000
                                               72-11-40.8        148.2689FT
          1     4368.1782    6291.7789        0.0000
                                              109-49-14.4        159.9470FT
          2     4313.9438    6442.2504        0.0000
                                              199-44- 2.2        215.8703FT
          3     4110.7514    6369.3612        0.0000
                                              307-24-11.4        115.3644FT
        202     4180.8260    6277.7179        0.0000
%CLO-P-ACCEPT, Accept adjusted coordinates(Y,N) ?
Y
%CLO-I-UPDAT,    3 coordinate records updated
```

*Figure 10-1. COMPASS CLOSURE (Continued)*

REPORT FILE LISTING:

```
CRANDALL CLOSURE  des pFOC Dtol /pLS pFLS Atol /Tdis Mdis
  (200 1 2 3 202) 4 20 203 5 5.
 Angular Misclosure          000-21-43.1  Allowable          005-00- 0.0
 Correction                  000-04-20.6
 Traverse Length                637.1550
 Linear  Misclosure     -5.0383  -16.2731
 Total            17.0352 (1:         37) Allowable   20.0000 (1:        32)
 Length and Lateral Errors
 Total                       -14.1718             9.4529
-------C r a n d a l l  A d j u s t m e n t  S u m m a r y--------
   Point              Coordinates
-----------------------------------------------------------------
        200     4322.8399    6650.6119       0.0000
                                               70-56-23.5       145.8899FT
          1     4370.4818    6788.5035       0.0000
                                              108-17-24.7       161.0594FT
          2     4319.9365    6941.4261       0.0000
                                              199-48- 5.7       223.2353FT
          3     4109.9008    6865.8020       0.0000
                                              308-50-27.5       113.0893FT
        202     4180.8260    6777.7179       0.0000
%CLO-P-ACCEPT, Accept adjusted coordinates(Y,N) ?
                                                                  Y
%CLO-I-UPDAT,    3 coordinate records updated
```

*Figure 10-2. CRANDALL CLOSURE*

REPORT FILE LISTING:

```
TRANSIT CLOSURE   des pFOC Dtol /pLS pFLS Atol /Tdis Mdis
   (200 1 2 3 202) 4 20 203 5 5.
  Angular Misclosure          000-21-43.1  Allowable             005-00- 0.0
  Correction                  000-04-20.6
  Traverse Length                637.1550
  Linear  Misclosure     -5.0383  -16.2731
  Total             17.0352 (1:        37) Allowable  20.0000 (1:        32)
  Length and Lateral Errors
  Total                          -14.1718            9.4529
-------T r a n s i t   A d j u s t m e n t   S u m m a r y--------
     Point              Coordinates
---------------------------------------------------------------------
         200     4322.8399    7150.6119      0.0000
                                                71-31-51.5      148.4267FT
           1     4369.8603    7291.3939      0.0000
                                               109-14- 9.0      158.9283FT
           2     4317.5002    7441.4493      0.0000
                                               199-13-24.0      218.7241FT
           3     4110.9716    7369.4341      0.0000
                                               307-17-39.2      115.2887FT
         202     4180.8260    7277.7179      0.0000
%CLO-P-ACCEPT, Accept adjusted coordinates(Y,N) ?
                                                                    Y
%CLO-I-UPDAT,    3 coordinate records updated
```

*Figure 10-3. TRANSIT CLOSURE*

### 10.1.2 VERTICAL ADJUST

VERTICAL ADJUST adjusts the level traverse described by *des*. The last point in *des* is closed to the known point *pFOC* defining the final occupied station. You specify the maximum allowable misclosure between the elevations of these points in the parameter *tol*. If the misclosure is greater than *tol*, ICS will not perform the adjustment.

ICS can divide misclosures evenly among turns or proportionally to the distance between turns. By default ICS divides misclosures evenly among turns. If you want the misclosures divided proportionally, use the optional parameter *prop*.

Operator Sequence

1. Select VERTICAL ADJUST or key in VERTICAL CLOSURE.

   *The system responds:*

   VERTICAL CLOSURE des pFOC tol /prop

2. Identify the following:

   des — a description defining the traverse to be adjusted. If the traverse is a loop, the last point in *des* should match the first point. Also, make sure the last point corresponds to *pFOC*; the last point should be the actual location of the last occupied point.

   pFOC — the ID of the known point defining the field location of the final occupied station; this point is computed from field observational data. The *pFOC* x and y coordinates do *not* have to match those of the last point in *des*. If the closure is successful, this point is deleted.

   tol — the maximum allowable vertical misclosure between *pFOC* and the last point in *des* (database units). If this condition is not met, ICS does not perform the adjustment.

   prop — any numerical value in this field allows proportional division of the misclosure between each turn. You should use this command *only if* there are valid Northing and Easting positional values for the ID's contained in *des*.

REPORT FILE LISTING:

```
 VERTICAL CLOSURE (100-103 204) 104 0.55
  Vertical Misclosure:    -0.0800FT  in         0.0000FT
  Correction is          0.0200FT / Turn
--------V e r t i c a l  A d j u s t m e n t  S u m m a r y-------
     Point               Coordinates
--------------------------------------------------------------------
          100        0.0000        0.0000       500.0130
                                                   N 0- 0- 0.0E          0.0000FT
          101        0.0000        0.0000       501.8030
                                                   N 0- 0- 0.0E          0.0000FT
          102        0.0000        0.0000       504.9330
                                                   N 0- 0- 0.0E          0.0000FT
          103        0.0000        0.0000       512.1830
                                                   N 0- 0- 0.0E          0.0000FT
          204        0.0000        0.0000       507.3230
```

*Figure 10-4. VERTICAL ADJUST*

### 10.1.3 ADJUST AREA: ROTATE, PARALLEL, and NORMAL

The ADJUST AREA COMMANDS adjust the area described by *des1* to be a specified number (*area*) of square units. Three methods are available to adjust the area:

- ROTATE
  Rotate the sides in the figure description between *p1* and *p2* about *p3*. Point *p3* can be any defined point; it does not have to be in *des1*. Point *p4* must be assigned a value of zero (0).

- PARALLEL
  Moves the side between *p1* and *p2* in a direction parallel to the line defined by *p3* and *p4*. Points *p3* and *p4* can be any defined points; they do not have to be in *des1*.

- NORMAL
  Move the side in the figure description between points *p1* and *p2* in a direction perpendicular to the line defined by *p1* and *p2*. Points *p3* and *p4* must be assigned a value of zero (0).

For all three of these options, you can use the optional parameter *des2*. The points described in the optional *des2* can be moved in the same manner as the points between *p1* and *p2*. The first and the last point in *des2* cannot be adjusted. The points *p1* and *p2* must be specified in the same sequence used in *des1*.

Note: This command uses current values of the coordinates of points to compute new positions of the same points. Make sure that no current values are needed, since they are replaced in this process. Also, if you use a figure ID for a description, be sure you know the direction in which the figure was stored (use the LIST FIGURE command if you do not remember).

<u>Operator Sequence</u>

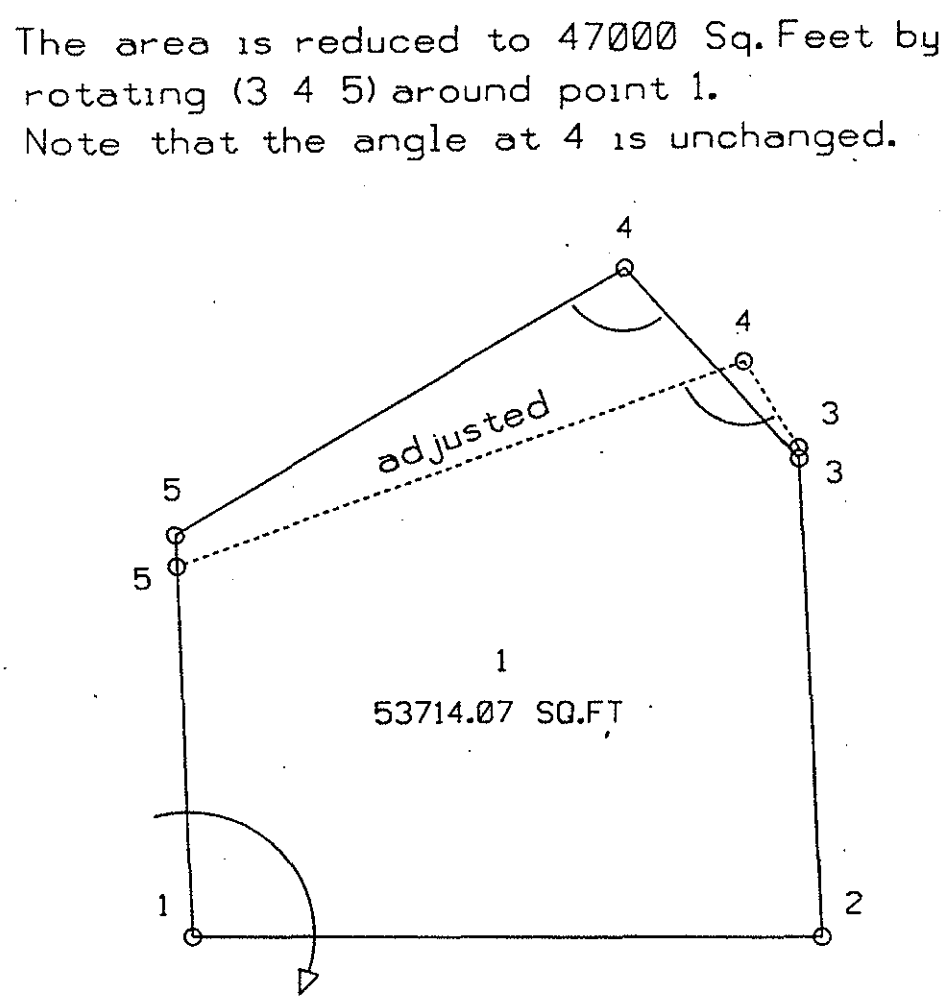
1. Select ROTATE, PARALLEL, or NORMAL; or key in ADJUST AREA.

   *The system responds:*

   ADJUST AREA des1 area p1 p2 p3 p4 /des2

2. Identify the following:

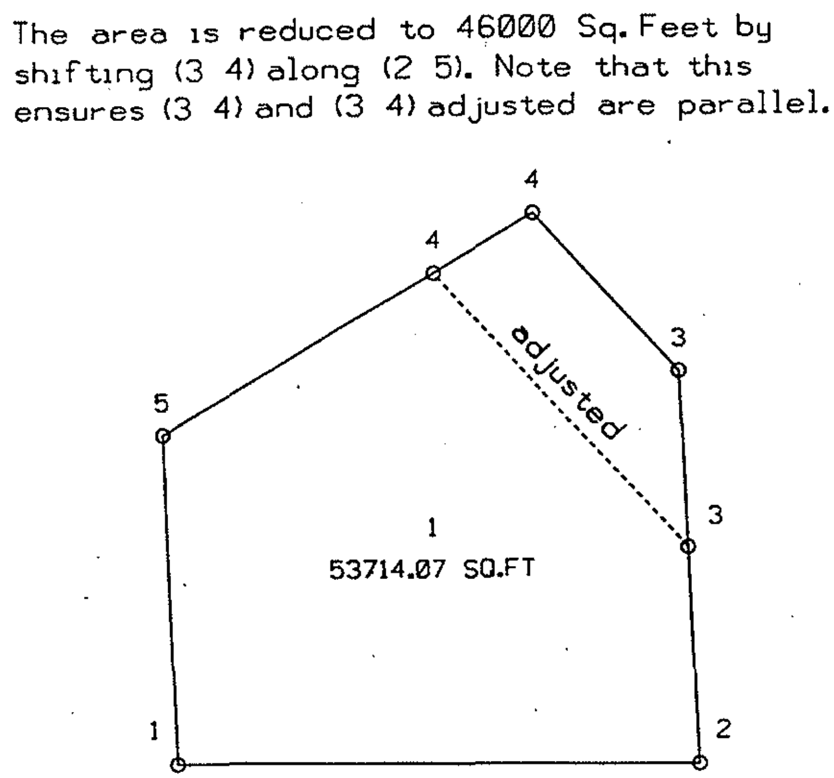
| | |
|---|---|
| des1 | the figure ID or list of points to be adjusted. |
| area | the desired area in square units. |
| p1 | the ID of the known point in the figure description that is the first point defining the sides to be moved. |
| p2 | the ID of the known point in the figure description that is the last point defining the sides to be moved. |
| p3 | the ID of the known point of rotation (rotate), 0 (normal), or the known point ID of the first point of a line (parallel). |
| p4 | 0 (rotate), 0 (normal), or the ID of the last point of a line (parallel). |
| des2 | a second figure ID or group of points to be adjusted. |




REPORT FILE LISTING:

```
ADJUST AREA  des1 ar p1 p2 p3 p4 /des2
   1 47000 3 5 1 0
     Original area =     53714.0686 square FT
        Final area = _   47000.0000 square FT
```

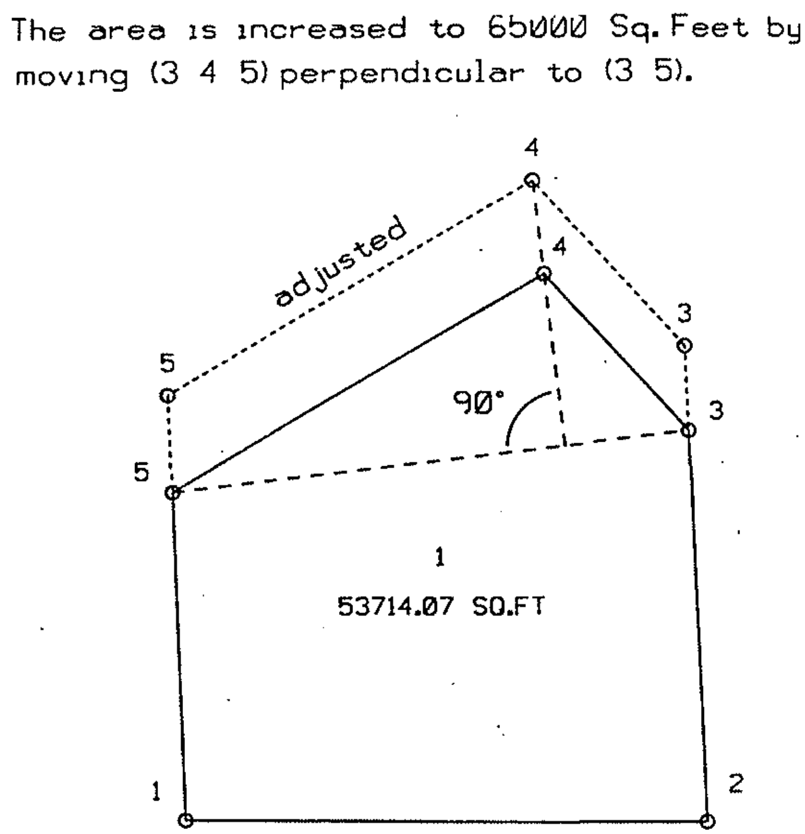
*Figure 10-5. ADJUST AREA ROTATE*




REPORT FILE LISTING:

```
ADJUST AREA  des1 ar p1 p2 p3 p4 /des2
   1 46000 3 4 2 5
     Original area =    53714.0686 square FT
        Final area =    46000.0000 square FT
```

*Figure 10-6. ADJUST AREA PARALLEL*




REPORT FILE LISTING:

```
ADJUST AREA  des1 ar p1 p2 p3 p4 /des2
    1 65000 3 5 0 0
     Original area =    53714.0686 square FT
        Final area =    65000.0000 square FT
```

*Figure 10-7. ADJUST AREA NORMAL*

## 10.2 Transformation Commands

ICS uses a Helmert transformation ICS that is a conformal transformation composed of uniform scale change in all axes, rotations about the axes, and an origin shift. In matrix rotation

$$x' = sMx + \Delta$$

where x' is the transformed coordinate vector, s is the scale factor, M is the orthogonal rotation matrix, and Δ is the origin shift vector. Thus, the Helmert transformation has four parameters in 2D (scale, rotation, x-y translations) and seven parameters in 3D (scale, three rotations, x-y-z translations). The number of parameters reduces to three and six, respectively, if the scale is known.

ICS allows you to compute and apply Helmert transformations in either two or three dimensions. In the graphics environment, you can use a tutorial to key in or graphically select transformation parameters. Once a transformation is computed, you can use the tutorial to apply the inverse transformation.

The transformation commands include the following:

| *Command* | *For* |
|---|---|
| COMPUTE TRANSFORM | establishing parameters for a coordinate transformation |
| TRANSFORM COORDINATES | transforming coordinate positions according to a set of parameters that have been either input using this command (tutorial review option in graphics), or calculated by the COMPUTE TRANSFORM command |

### 10.2.1 COMPUTE TRANSFORM

The COMPUTE TRANSFORM command determines the active ICS transformation parameters that are then used by the TRANSFORM COORDINATES command to transform the coordinate positions. In graphics, you can input known transformation parameters using the tutorial activated by the TRANSFORM COORDINATES command.

COMPUTE TRANSFORM computes the parameters for the transformation from the coordinate system defined by *desFROM* to the coordinate system defined by *desTO*. The data field arguments *desFROM* and *desTO* must each contain a minimum of two pass points (points common to the two coordinate systems). The pass points in each data field argument must appear in the same order, since they will be mapped one to one from *desFROM* to *desTO*. If more than two pass points are used (up to a maximum of 200), ICS computes a least squares solution of the transformation parameters.

If you want a fixed scale for the transformation, you can use the optional scale parameter *sca*. If you use this optional parameter, the transformation becomes a three-parameter or six-parameter transformation (for 3D) instead of a four-parameter or seven-parameter transformation, because the scale parameter is not calculated.

If you want to perform a 3D transformation (a seven-parameter transformation), you can key in a one (1) for the optional *3D* parameter. In order to use this optional parameter, all the input control points you specify *must* be stored as 3D points in the database. Also, if you use this parameter and you input only two pass points, ICS will calculate a third control point and use it automatically (this is valid only for transformations that do not involve rotations about the x or y axis).

Operator Sequence

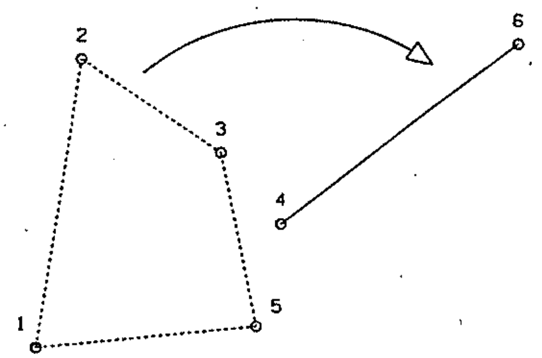
1. Select or key in COMPUTE TRANSFORM.

   *The system responds:*

   COMPUTE TRANSFORM desFROM desTO /sca /3D

2. Identify the following:

| | |
|---|---|
| desFROM | the figure ID or list of points containing the original pass points. |
| desTO | the figure ID or list of points containing the altered pass points. The points must correspond one-to-one with *desFROM*. |
| sca | the optional scale factor to be held fixed in the transformation process (0 is the default when using computed scale). |
| 3D | 1 when computing a seven-parameter transformation. |




REPORT FILE LISTING:

```
COMPUTE TRANSFORM  desFROM desTO /Sca /3D
   (1 2) (4 6)
                   Compute  Transformation
      Adjustment Results                       2  Control Points Used
    Point     C o o r d i n a t e s           R e s i d u a l s
      1       4740.9119       6201.6839        0.0000         0.0000
      2       4860.3609       6357.8869        0.0000         0.0000
   Computed scale factor is        1.0118767    ( 1 :          84 )
   Rotation Angle is .... -43 Degrees 48 Minutes 25.1 Seconds
   X Translation is  ....-1471.5476 FT
   Y Translation is  .... 5569.8632 FT
                   Minimum Solution.   No Statistics.
```

*Figure 10-8. COMPUTE TRANSFORM*

## 10.2.2 TRANSFORM COORDINATES

TRANSFORM COORDINATES transforms the positions defined by the description *des* according to a set of parameters you define. In alpha mode you must define these parameters using the COMPUTE TRANSFORM command, so you must execute COMPUTE TRANSFORM before initiating TRANSFORM COORDINATES. In graphics you have the option to define the parameters individually using the COMPUTE TRANSFORM command, or you can define the parameters through a tutorial that you can activate after selecting TRANSFORM COORDINATES.

Using the TRANSFORM COORDINATES Tutorial

The TRANSFORM COORDINATES tutorial, pictured in Figure 10-9, provides two ways for you to define the transformation parameters. The right side of the tutorial contains the column of DEFINE commands: SCALE, TRANSLATION, and ANGLE. These DEFINE commands allow you to define the parameters using data points; you do not have to know the values. The DEFINE commands work the same way as the IGDS SCALE and ANGLE commands; they allow you to define the transformation parameters by graphic selection *only*. You select these commands by placing the cursor on top of the command and pressing <D>. After you select a command and give the appropriate data points, ICS calculates the parameter and displays the result in the appropriate field in the lower portion of the tutorial.

The other way you can define parameters from this tutorial is by actually keying the values into the fields provided at the bottom of the tutorial. You press <D> in the field you want to enter data into, key in the data, and press <RETURN>.

In addition to the transformation parameter fields, there are four additional command boxes:

- Two/Three DIMENSION
  Allows toggling between 2D and 3D transformations.

- RESTORE
  Restores parameters to values present when tutorial was entered.

- APPLY DIRECT/INVERSE
  Allows toggling between direct and inverse type of transformation. Direct transformations can be performed repeatedly, but inverse transformations require that APPLY INVERSE be selected from the tutorial each time.

- QUIT
  Exits the tutorial and saves current parameter values.

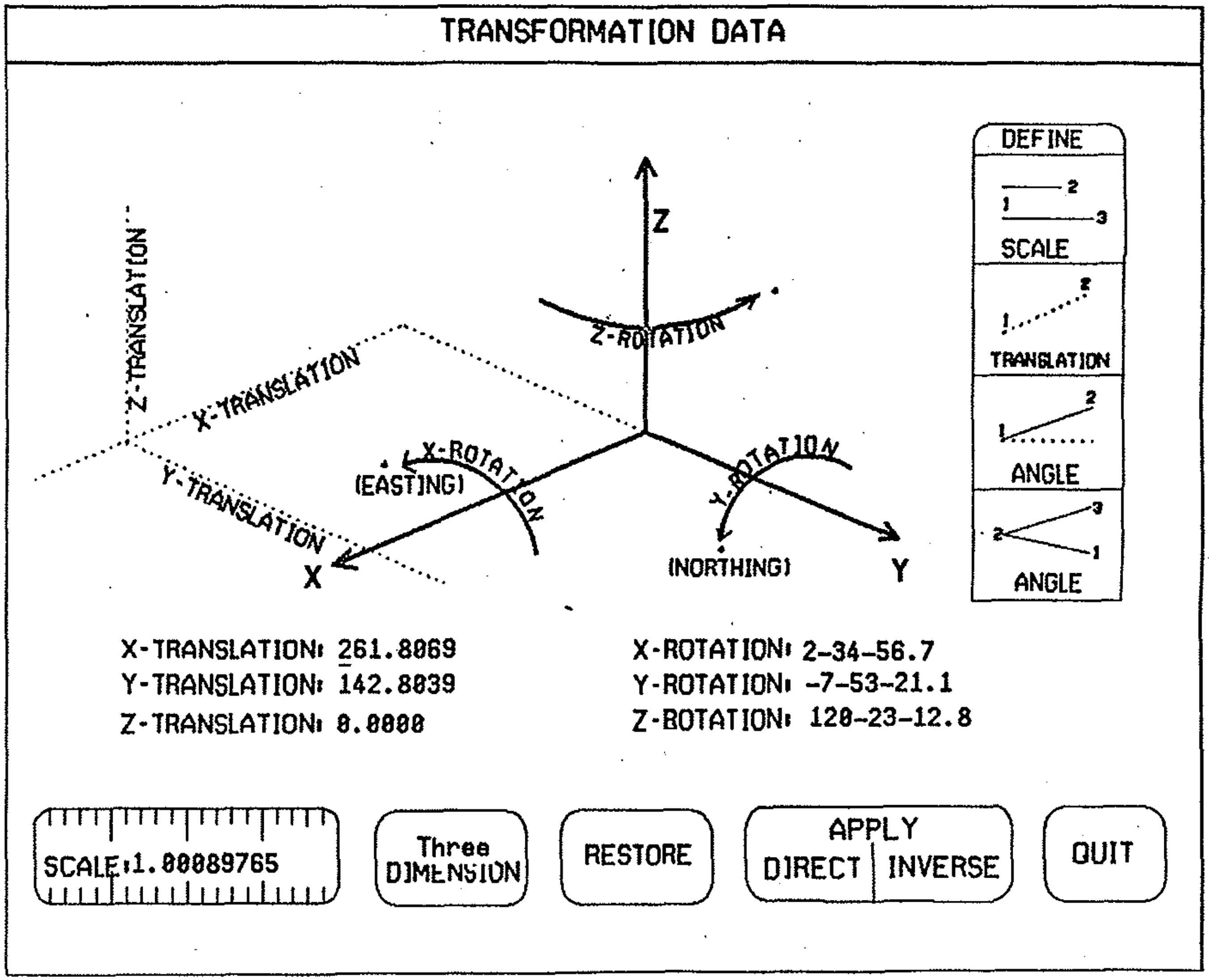



*Figure 10-9. TRANSFORM COORDINATES Tutorial*

Using the Distribution Option

ICS maintains a record of the pass points used to establish the transformation parameters and a record of the residuals associated with each pass point. If you want ICS to weight each of the pass point residuals and, according to these weights, distribute the contribution of each residual to the coordinates of the new points, you can set the optional distribute variable *distrib* to 1.

The weight of each residual is based on the inverse distance between it and the control points as computed in the original coordinate system. You must ensure that none of the original control points are transformed prior to applying the transform to all other desired points.

Residual distribution generally results in *forcing* control points to the positions defined in the new system, and this sacrifices the conformal nature of the transformation. Thus, the application of the active transformation to the original control is redundant.

Verifying the Transformed Coordinates

You can optionally view the transformed coordinates before they are accepted, if you use the optional *verify* parameter. This parameter allows you to view the transformed coordinates of each point and individually accept or reject them. If you set *verify* equal to 1, the following message appears:

Accept: (Y/N) <RETURN> = exit

Key in Y to accept the transformed positions, key in N to reject the transformed positions, or press <SPACE> followed by <RETURN> to exit the TRANSFORM COORDINATES command. In graphics, you can press <D> for yes, and <R> for no.

Note: On a graphics terminal, you must press <RETURN> to exit.

TRANSFORM COORDINATES alters the coordinates of points in the database. If you do not want the points in the database to be altered, you can use the optional parameter $n$. This parameter saves the transformed coordinates with new point ID's; the original point ID's are not altered. You specify $n$ to be the ID of the first transformed point, and subsequent new points are $n+1$, $n+2$, etc. If you give $n$ the value zero (0), ICS will assign the next available point ID's to the generated points.

Note: If you do not use the optional $n$ parameter, this command will alter the coordinates of points in the database. Be sure that none of the current values are needed, since they are replaced in this process. Use the COPY POINT command to save values which may be useful.

Setup

In alpha, execute the COMPUTE TRANSFORM command to calculate the transformation parameters.

Operator Sequence

1. Select or key in TRANSFORM COORDINATES. If you are working in alpha, go to Step 3. Otherwise, go to Step 2.

2. **<D> Apply current transformation <R> Review/Define transformation**

   Press <D> to accept the current parameters and apply the transformation. Go to Step 3.

   -OR-

   Press <R> to review or define the transformation parameters.

   *ICS will display the TRANSFORM COORDINATES tutorial.*

   You can alter the parameters and save the new ones as described in the previous section *Using the TRANSFORM COORDINATES Tutorial.* After exiting the tutorial, go to Step 3.

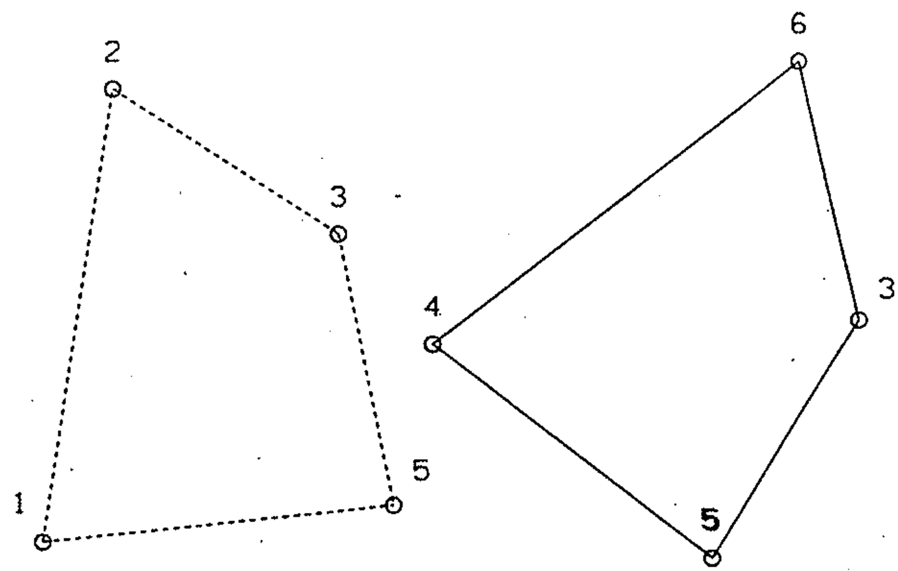
3. *The system responds:*

   TRANSFORM COORDINATES des /distrib /verify /n

   Identify the following:

   | | |
   |---|---|
   | des | the figure ID or list of points to be transformed. |
   | distrib | 0 (default) when not applying residual distribution (conformal transformation); 1 when applying residual distribution (nonconformal transformation). |
   | verify | 0 when adjusting all points; 1 when accepting/rejecting each point on an individual basis. |
   | n | the point ID for the first generated point; subsequent new points will be *n*+1, *n*+2, etc. 0 for ICS to assign the next available point ID numbers. |

Points 3 and 5 (des) are transformed to new positions based on the transformation which maps (1 2) onto (4 6).




REPORT FILE LISTING:

```
TRANSFORM COORDS  des /distrib /verify /n
   (3 5) 0 0
                          Transform Coordinates
   Point      C o o r d i n a t e s          Residual Contributions
         3     4749.9806      6382.0961         0.0000 DEFAULT
         5     4649.7231      6318.4501         0.0000 DEFAULT
```

*Figure 10-10. TRANSFORM COORDINATES*

REPORTS Commands

11

# 11. REPORTS COMMANDS

The REPORTS commands interpret and list information already stored in an ICS database. The information is displayed on the screen and optionally to an output file. The contents of an output file can then be listed at the terminal, printed on a line printer, and/or plotted in a design file using the PLOT FILE command. See Section 7.1 for details on the OUTPUT FILE and INPUT FILE commands. See Section 12.1.10 for details on the PLOT FILE command.

The REPORTS menu block is divided into four parts:

- LIST Commands
- ALIGNMENT Commands
- AREA Commands
- TRAVERSE Commands.

## 11.1 LIST Commands

Use the LIST commands to list coordinates, list figures, or display information that is in the feature table. The LIST commands include the following:

| *Command* | *For* |
|---|---|
| LIST COORDINATES | listing the coordinates of points |
| LIST FIGURES | listing the point ID's contained in a figure |
| SHOW FEATURE | displaying information about a feature in the feature table |

## 11.1.1 LIST COORDINATES

The LIST COORDINATES command lists the coordinates of the point ID's you specify in *des*; *des* may be a list of point ID's or figure ID's. If you want to get a listing of available point ID's, follow *des* with a -1.

Operator Sequence

1. Select or key in LIST COORDINATES.

   *The system responds:*

   LIST COORDINATES des

2. Identify the following:

   des description of the points whose coordinates are to be listed.

## 11.1.2 LIST FIGURES

The LIST FIGURES command lists the point ID's and geometry of each figure contained in *des*; *des* can be a figure ID or list of figure ID's.

Operator Sequence

1. Select or key in LIST FIGURES.

   *The system responds:*

   LIST FIGURES des

2. Identify the following:

   des description of the figure(s).

---

### 11.1.3 SHOW FEATURE

The SHOW FEATURE command displays information about features in the feature table. You can identify the feature you want information on by entering either the feature number or name.

Operator Sequence

1. Select or key in SHOW FEATURE.

   *The system responds:*

   SHOW FEATURE fnum !fname (alpha)
   fnum or !fname (graphics)

2. Identify the following:

   fnum the feature number.

   -OR-

   !fname the feature name. The name must be preceded by an exclamation mark (!).

REPORT FILE LISTING:

```
SHOW FEATURE 500
 F e a t u r e                          FIXED POINT                [ 500]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 Element line style               0  Pattern cell name
 Element level                   16  Pattern type                      0
 Element weight                   1  Pattern scale                  2.00
 Element color                   16  Pattern angle                  0.00
 Annotation font                 23  Pattern delta                  1.00
 Annotation justification        12  Cell/Point name              OCT140
 Annotation txt hght (MU's)       7  Cell library USR_DD_ICS:SURSYM.CEL
 Annotation txt wdth (MU's)       7  User Command
 Annotation level                17  Dmrs filename
 Annotation line weight           0  No annotation
 Annotation color                 2  Distance and direction annotation
 Annotation line spacing          3
 Symbol font                     90
 Symbol character                 '
 Symbol justification             8  Curve labels:  RLTD
 Symbol text height (MU's)        8
 Symbol text width  (MU's)        8
 Symbol level                    18
 Symbol weight                    0
 Symbol color                    18
```

*Figure 11-1. SHOW FEATURE*

## 11.2 ALIGNMENT Commands

Use the ALIGNMENT commands to describe horizontal and vertical alignments and to compute the stations and offsets of an alignment. ALIGNMENT commands include the following:

| *Command* | *For* |
|---|---|
| DESCRIBE ALIGNMENT | describing horizontal alignment features |
| DESCRIBE VERTICAL ALIGNMENT | describing vertical alignment features |
| STATIONS AND OFFSETS and RELOCATION OFFSET | computing the stations and offsets of an alignment |

### 11.2.1 DESCRIBE ALIGNMENT

The DESCRIBE ALIGNMENT command describes a horizontal alignment. If you specify *sBEG* and *sEND*, the command describes only the part of the alignment between stations *sBEG* and *sEND*. Stations are expressed in decimal units. For example, station 41+23.67 is entered as 4123.67. If you omit the interval *x* along the alignment, ICS only reports on the points in the description of the alignment.

<u>Operator Sequence</u>

1. Select or key in DESCRIBE ALIGNMENT.

   *The system responds:*

   DESCRIBE ALIGNMENT Hfg / x /sBEG /sEND

2. Identify the following:

   Hfg the horizontal figure.

   x the specified interval. It must be preceded by a space.

   sBEG the station which begins the description.

   sEND the station which ends the description.

Note: In graphics, even if *x* is not input, *Hfg* must be followed by a space for the command to execute.

REPORT FILE LISTING:

```
DESCRIBE ALIGNMENT  Hfg /x /sBEG /sEND
  1
 PI         1     1171.6479    -7065.6009          1+00
                                                N 32-37-10.4 E
 PC         2     1273.1639    -7000.6299     2+20.5269
                                                N 32-37-10.4 E
      Radial direction from PC to CC   S 57-22-49.6 E
 CC         4     1170.2114    -6839.7685
      Radius          190.9859FT  Degree      30- 0- 0.0
      Length          100.0000FT  Delta       30- 0- 0.0 RT
      Tangent          51.1745FT  Back      N 32-37-10.4 E
      External          6.7373FT  Ahead     N 62-37-10.4 E
      Long Chord       98.8616FT            N 47-37-10.4 E
      Mid. Ord.         6.5077FT
 PI               1316.2666    -6973.0439     2+71.7014
                                                N 62-37-10.4 E
      Radial direction from CC to PT   N 27-22-49.6 W
 CS         5     1339.8016    -6927.6023     3+20.5269
                                                N 62-37-10.4 E
 PI         6     1371.2852    -6866.8136
      Spiral Ahead                Degrees/station:     15.0000
      Length          200.0000FT  Throw       30- 0- 0.0
      L. Tangent      135.2997FT  Ahead     S 87-22-49.6 E
      S. Tangent       68.4579FT  Back      N 62-37-10.4 E
      Chord           197.5737FT            N 82-38-34.5 E
      Degree       30- 0- 0.0
 ST         7     1365.1015    -6731.6552     5+20.5269
                                                S 87-22-49.6 E
 PC         8     1358.2459    -6581.8119     6+70.5269
                                                S 87-22-49.6 E
      Radial direction from PC to CC   S  2-37-10.4 W
 CC        10     1072.0664    -6594.9051
      Radius          286.4789FT  Degree      20- 0- 0.0
      Length          175.0000FT  Delta       35- 0- 0.0 RT
      Tangent          90.3264FT  Back      S 87-22-49.6 E
      External         13.9026FT  Ahead     S 52-22-49.6 E
      Long Chord      172.2917FT            S 69-52-49.6 E
      Mid. Ord.        13.2591FT
 PI               1354.1176    -6491.5799     7+60.8533
                                                S 52-22-49.6 E
      Radial direction from CC to PT   N 37-37-10.4 E
 PT        11     1298.9810    -6420.0340     8+45.5269
```

*Figure 11-2. DESCRIBE ALIGNMENT*

### 11.2.2 DESCRIBE VERTICAL ALIGNMENT

The DESCRIBE VERTICAL ALIGNMENT command reports information about points of vertical intersection, curvature and tangency, grades, and topographic high and low points along the active vertical alignment. If tangent lengths are unequal, vertical curve lengths are computed. You can assign zero (0) to *sBEG* and *sEND* as a wild card to indicate from the beginning or to the end, respectively. The optional input of design speed and coefficient of friction is used to calculate the stopping sight distance for crest curves.

Operator Sequence

1. Select or key in DESCRIBE VERTICAL ALIGNMENT.

   *The system responds:*

   DESCRIBE VERTICAL ALIGNMENT sBEG /sEND /speed friction

2. Identify the following:

   sBEG the beginning station.

   sEND the ending station.

   Note: *sBEG* and *sEND* must both be defined on the same figure, either *Hfg* or *Vfg*.

   speed design speed.

   friction the coefficient of friction.

REPORT FILE LISTING:

```
DESCRIBE VERTICAL ALIGNMENT  sBEG /sEND /speed friction
   2
                Tangent Section
  Grade from PVI         12 to PVC         17 is          8.2208%
     Point                     Station                              Elevation
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   PVI        12               1+07.3700                          409.7976
   PVC        17               2+26.1370                          419.5613
         Crest Vertical Curve
  Grade from PVC         17 to PVI         13 is          8.2208%
```

*Figure 11-3. DESCRIBE VERTICAL ALIGNMENT*

```
  Point                        Station                          Elevation
-------------------------------------------------------------------------
 PVC        17                2+26.1370                         419.5613
 PVI        13                3+26.1370                         427.7821
 PVT        18                4+01.1370                         424.8498
 HI/LOW                       3+58.8342                         425.6767
Curve lengths        100.0000 and        75.0000
       Crest Vertical Curve
Grade from PVC        18 to PVI        14 is        -3.9098%
  Point                        Station                          Elevation
-------------------------------------------------------------------------
 PVC        18                4+01.1370                         424.8498
 PVI        14                5+04.6400                         420.8030
 PVT        19                5+79.6400                         415.5530
Curve lengths        103.5030 and        75.0000
             Tangent Section
Grade from PVT        19 to PVC        20 is        -7.0000%
  Point                        Station                          Elevation
-------------------------------------------------------------------------
 PVT        19                5+79.6400                         415.5530
 PVC        20                6+14.6400                         413.1030
       Sag Vertical Curve
Grade from PVC        20 to PVI        15 is        -7.0000%
  Point                        Station                          Elevation
-------------------------------------------------------------------------
 PVC        20                6+14.6400                         413.1030
 PVI        15                6+64.6400                         409.6030
 PVT        21                7+14.6400                         409.5399
Curve length        100.0000
             Tangent Section
Grade from PVT        21 to PVI        16 is        -0.1263%
  Point                        Station                          Elevation
-------------------------------------------------------------------------
 PVT        21                7+14.6400                         409.5399
 PVI        16                8+29.4350                         409.3949
```

*Figure 11-3. DESCRIBE VERTICAL ALIGNMENT (Continued)*

### 11.2.3 STATIONS AND OFFSETS

The STATIONS AND OFFSETS command computes and reports the perpendicular offsets from the active alignment (baseline) to a second alignment (centerline). The first point in the description *desAL2* of the centerline is defined as station 0+00. ICS computes the offsets to points on the centerline starting at *sBEG*, with interval *x*, until *sEND*, including all transition points along the centerline. Both the baseline and centerline stationing is reported for all computed points.

Note: STATIONS AND OFFSETS assumes that all straight line segments extend to infinity. If you are not careful, unexpected intersections may result.

<u>Setup</u>

Be sure to run SET ALIGNMENT before running STATIONS AND OFFSETS.

<u>Operator Sequence</u>

1. Select or key in STATIONS AND OFFSETS.

   *The system responds:*

   STATIONS AND OFFSETS desAL2 /x /sBEG /sEND

2. Identify the following:

   desAL2 description of the alignment (centerline) for which stations and offsets will be listed.

   x the distance between the stations on the centerline.

   sBEG the beginning station on the alignment *desAL2.*

   sEND the ending station on the alignment *desAL2.*

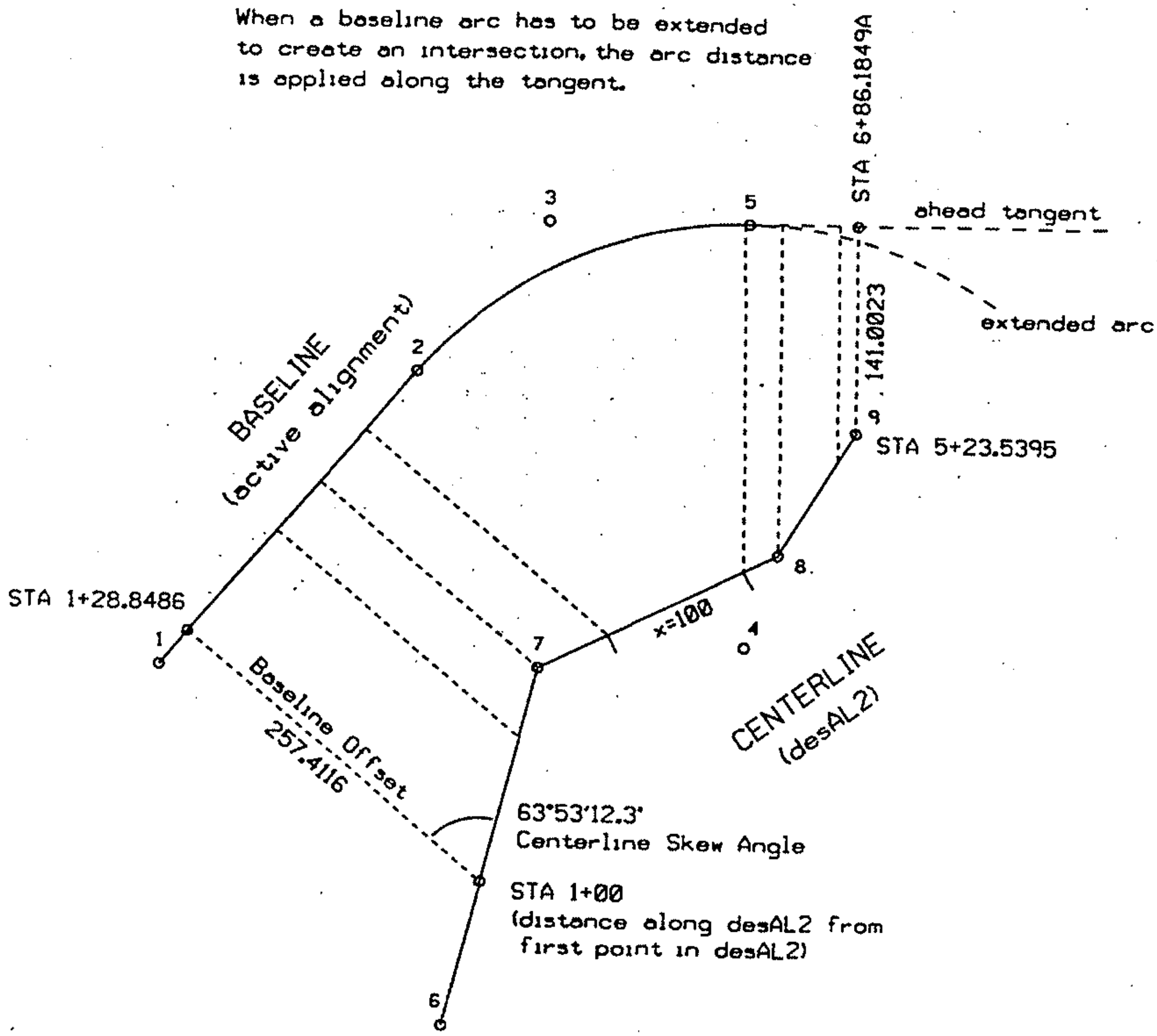



*Figure 11-4. STATIONS AND OFFSETS*

REPORT FILE LISTING:

```
STORE FIGURE  nfg des /pAL sAL or  fg des /pAL sAL
        1 (1 2 C4R 5)
STORE FIGURE  nfg des /pAL sAL or  fg des /pAL sAL
        2 (6-9)
STATIONS AND OFFSETS  desAL2 /x /sBEG /sEND
      Baseline figure is :          1
      Centerline figure is :          2
                    Baseline      Centerline    Baseline                 Centerline
      Point         Station       Station       Offset      Elevation  Skew Angle
-----------------------------------------------------------------------------------
  Points On Line           1          2
                 1+28.8486            1+00    257.4116                63-53-12.3
                 2+18.6412            2+00    213.3969                63-53-12.3
        7        2+62.3926       2+48.7250    191.9508       0.0000  113-54-40.6
                 3+09.2669            3+00    212.7336               113-54-40.6
  Points On Curve          2          5
                 6+09.0029            4+00    235.4228                64-29-32.2
        8        6+34.1440A      4+25.3351    224.2840       0.0000   32- 0- 1.4
                 6+73.7108A           5+00    160.9648                32- 0- 1.4
        9        6+86.1849A      5+23.5395    141.0023       0.0000   32- 0- 1.4
```

*Figure 11-4. STATIONS AND OFFSETS (Continued)*

### 11.2.4 RELOCATION OFFSET

The RELOCATION OFFSET command is the same as the STATIONS AND OFFSETS command except that the interval $x$ is defined along the baseline (active alignment), *not* the centerline (described by *desAL2*). See Section 11.2.3 for details.

Note: RELOCATION OFFSET assumes that all straight line segments extend to infinity. If you are not careful, unexpected intersections may result.

**In alpha, instead of keying in RELOCATION OFFSET, you must key in STATIONS AND OFFSETS, and precede the interval $x$ with a negative sign (-). The negative sign preceding the interval value indicates to ICS that you want the RELOCATION OFFSET command.**

<u>Setup</u>

Be sure to run SET ALIGNMENT before running RELOCATION OFFSET.

<u>Operator Sequence</u>

1. Select RELOCATION OFFSET or key in STATIONS AND OFFSETS.

   *The system responds:*

   RELOCATION OFFSET desAL2 /x /sBEG /sEND

2. Identify the following:

   desAL2 description of the alignment (centerline) to be assigned stations and offsets.

   x the distance between the stations on the baseline (active alignment).

   Note: In alpha, you must precede $x$ with a negative sign (-).

   sBEG the beginning station on the alignment *desAL2*.

   sEND the ending station on the alignment *desAL2*.

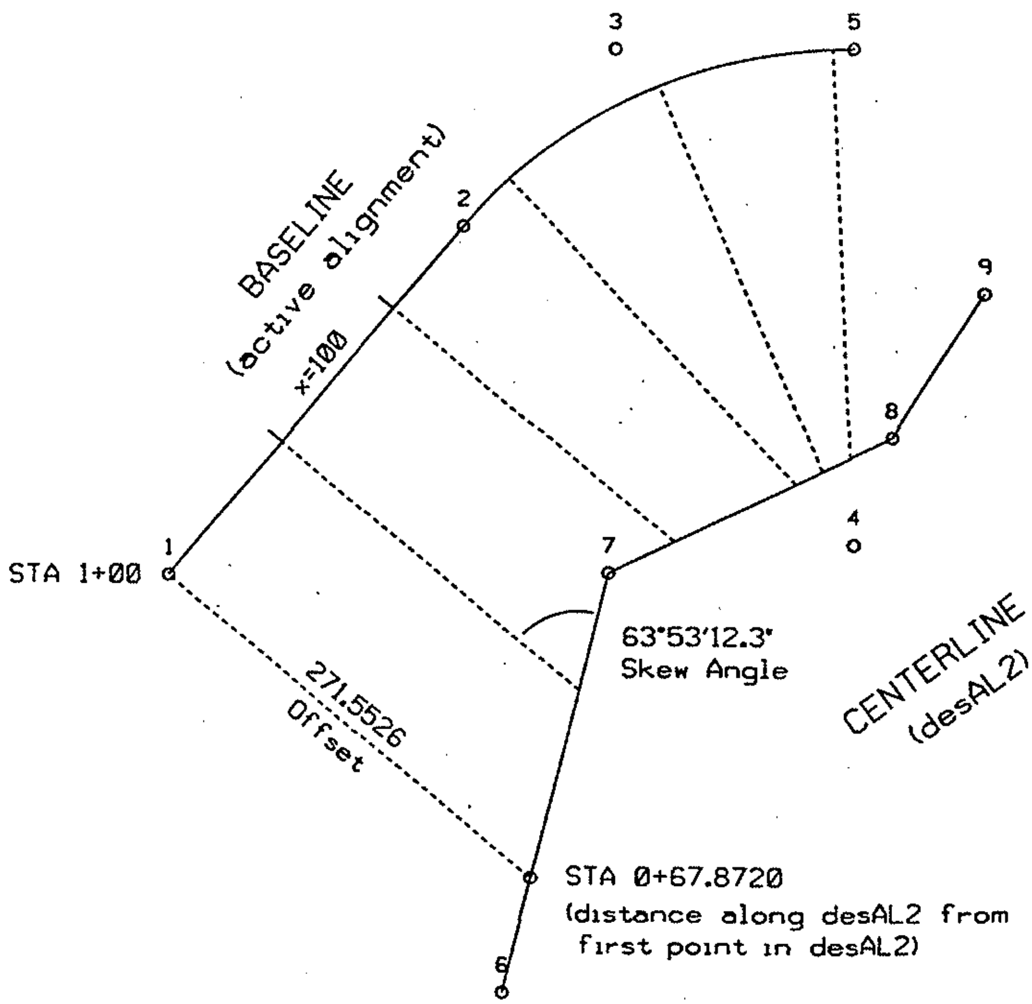



*Figure 11-5. RELOCATION OFFSET*

REPORT FILE LISTING:

```
STORE FIGURE  nfg des /pAL sAL or  fg des /pAL sAL
        1 (1 2 C4R 5)
STORE FIGURE  nfg des /pAL sAL or  fg des /pAL sAL
        2 (6-9)
STATIONS AND OFFSETS  desAL2 /x /sBEG /sEND
     Baseline figure is :         1
     Centerline figure is :         2
         Baseline     Centerline                 Skew Angle
         Station      Station       Offset       Deg Min Sec
--------------------------------------------------------------
             1+00     0+67.8720     271.5526      63  53 12.3
             2+00     1+79.2397     222.5345      63  53 12.3
             3+00     2+89.8631     208.6249     113  54 40.6
             4+00     3+64.5452     238.4714     106  17 34.3
             5+00     3+80.9992     240.3028      86  17 34.3
             6+00     3+98.2479     236.1525      66  17 34.3
```

*Figure 11-5. RELOCATION OFFSET (Continued)*

## 11.3 AREA Commands

Use the AREA commands for computing and reporting the area of closed polygons and circular segments. AREA commands include the following:

| *Command* | *For* |
|---|---|
| AREA DIRECTION | computing the area of a closed polygon and the directions and distances between the points |
| SEGMENT | computing the area of a circular segment with two specified arc and chord boundaries. |

**Note: ICS does not currently support the computation of areas bounded in part by parabolas (vertical curves) or spirals. ICS warns you if you attempt this operation.**

## 11.3.1 AREA DIRECTION

The AREA DIRECTION command computes and reports the area of a closed polygon described by *des*, as well as a brief direction and distance listing of the polygon boundary. The *des* must be a closed figure: the first and last points in *des* must be the same.

Operator Sequence

1. Select or key in AREA DIRECTION.

   *The system responds:*

   AREA DIRECTION des

2. Identify the following:

   des — description of the polygon. If you give a list of points, the last point you list must be the same as the first point.

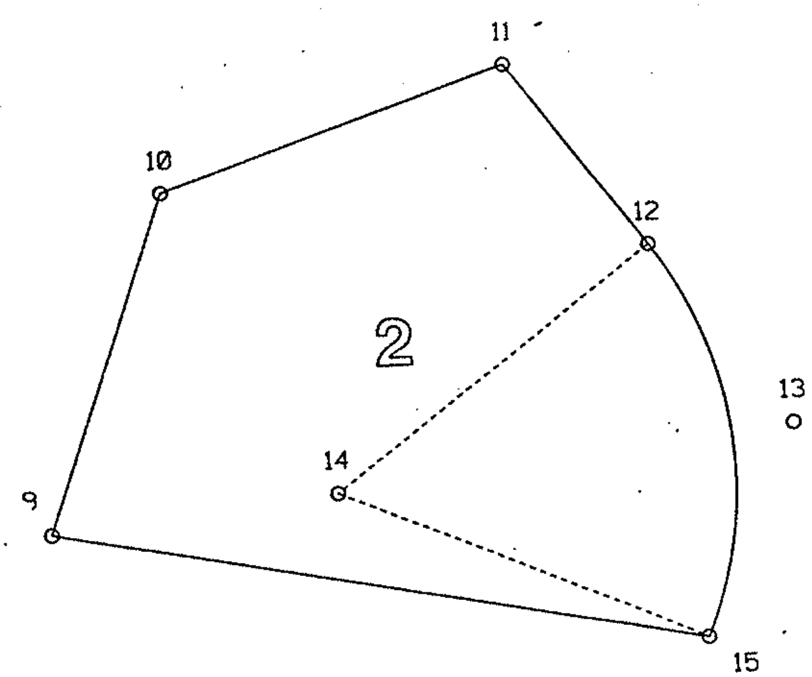



*Figure 11-6. AREA DIRECTION*

REPORT FILE LISTING:

```
STORE FIGURE  nfg des /pAL sAL or  fg des /pAL sAL
   0 (9 10 11 12 C14R 15 9)
         2 (9 10 11 12 C14R 15 9)
 AREA DIRECTION 2
     Point               Coordinates                 Direction       Distance
-----------------------------------------------------------------------------
              9     4686.7740     1559.8309       0.0000
                                                     N19-42-50.8E      172.5388FT
             10     4849.1999     1618.0329       0.0000
                                                     N71-33-53.8E      175.4928FT
             11     4904.6959     1784.5199       0.0000
                                                     S36-35-27.2E      111.2637FT
             12     4815.3609     1850.8439       0.0000
                                                     S53-24-32.8W      190.9859FT
 CC          14     4701.5147     1697.4990
                                                     Deflection        60- 0- 0.0R
                                                     Arc Length         200.0000FT
                                                     Long Chord         190.9859FT
                                                                      S 6-35-27.2E
                                                     S66-35-27.2E      190.9859FT
             15     4625.6372     1872.7651
                                                     N78-56-44.1W      318.8503FT
              9     4686.7740     1559.8309       0.0000
  Figure          2  AREA  =   61488.5412 SQ FT             1.4116 ACRE
```

*Figure 11-6. AREA DIRECTION (Continued)*

### 11.3.2 SEGMENT

The SEGMENT command computes and reports the area of a circular segment whose boundaries are the arc with radius *r* between *p1* and *p2* and the chord between *p1* and *p2*. (See also the SEGMENT PLUS and SEGMENT MINUS commands in Section 14.)

<u>Operator Sequence</u>

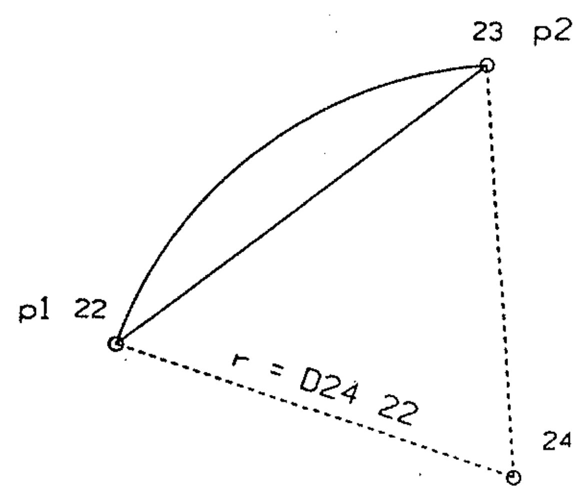
1. Select or key in SEGMENT.

   *The system responds:*

   SEGMENT p1 p2 r

2. Identify the following:

   | | |
   |---|---|
   | p1 | the point that is the starting point of the arc and of the segment. |
   | p2 | the point that is the ending point of the arc and segment. |
   | r | the radius of the arc. It can be expressed using the D delimiter. |




REPORT FILE LISTING:

```
SEGMENT 22 23 D24 22
     Segment AREA  =    5095.7999 SQ FT              0.1170 ACRE
       Length         235.8047
       Radius         200.0000FT  Delta        67-33-11.1
       Long Chord     222.3821FT              51-10-37.4
```

*Figure 11-7. SEGMENT*

## 11.4 TRAVERSE Commands

Use the TRAVERSE commands for computing and reporting distances and angles. The description *des* for the TRAVERSE commands can be a point listing or figure ID. If *des* is a figure ID, TRAVERSE ANGLES and TRAVERSE DEFLECTIONS report on circular and spiral arcs; DISTANCES and INVERSE DIRECTIONS do not.

Note: The T delimiter is useful for these commands; it allows you to get reports on traverses described in *des* in the reverse (transposed) direction.

TRAVERSE commands include the following:

| *Command* | *For* |
|---|---|
| DISTANCE | computing the distances between points |
| INVERSE DIRECTIONS | computing distances and directions between points |
| ANGLES | computing clockwise angles from a baseline to a group of points |
| TRAVERSE ANGLES | computing the angles and distances between points in a *des* |
| TRAVERSE DEFLECTIONS | computing the deflections and distances between points in a *des*. |

## 11.4.1 DISTANCE

The DISTANCE command computes and reports the distances between consecutive points in the description *des*.

Operator Sequence

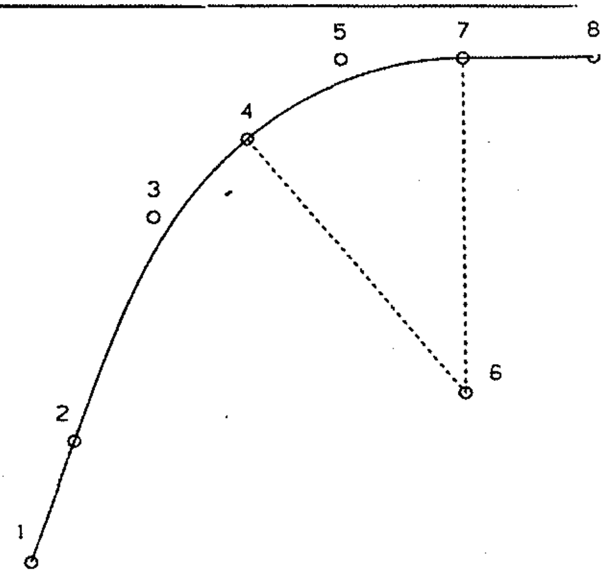
1. Select or key in DISTANCE.

   *The system responds:*

   DISTANCE des

2. Identify the following:

   des     description of the points.




REPORT FILE LISTING:

DISTANCE des
1

| From | To | Distance FT | Slope Distance FT | Elev Diff FT | Vert Angle |
|---|---|---|---|---|---|
| 1 | 2 | 73.2044 | 73.2044 | 0.0000 | 90- 0- 0.0 |
| 2 | 3 | 135.2997 | 135.2997 | 0.0000 | 90- 0- 0.0 |
| 3 | 4 | 68.4579 | 68.4579 | 0.0000 | 90- 0- 0.0 |
| 4 | 6 | 190.9859 | 190.9859 | 0.0000 | 90- 0- 0.0 |
| 6 | 7 | 190.9859 | 190.9859 | 0.0000 | 90- 0- 0.0 |
| 7 | 8 | 75.0000 | 75.0000 | 0.0000 | 90- 0- 0,0 |

*Figure 11-8. DISTANCE*

### 11.4.2 INVERSE DIRECTIONS

The INVERSE DIRECTIONS command computes and reports the distances and directions between consecutive points in the description *des.*

Operator Sequence

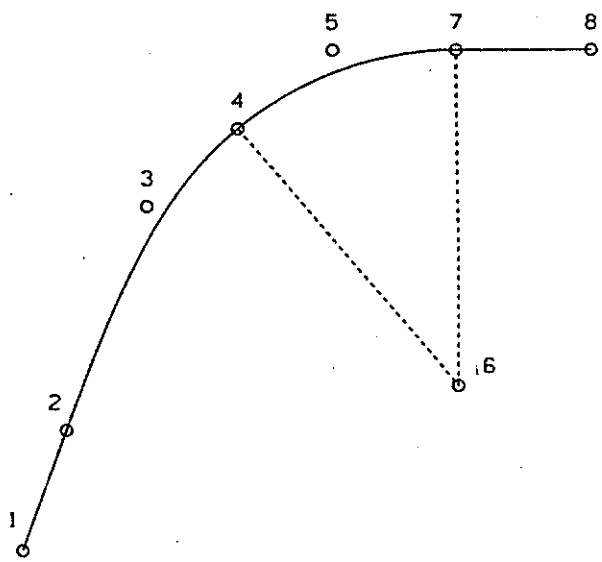
1. Select or key in INVERSE DIRECTIONS.

   *The system responds:*

   INVERSE DIRECTIONS des

2. Identify the following:

   des description of the points.




REPORT FILE LISTING:

INVERSE DIRECTIONS des
1

| From | To | Direction | Distance FT | Slope Distance FT | Elev Diff FT |
|---|---|---|---|---|---|
| 1 | 2 | N19-26-24.7E | 73.2044 | 73.2044 | 0.0000 |
| 2 | 3 | N19-26-24.7E | 135.2997 | 135.2997 | 0.0000 |
| 3 | 4 | N49-26-24.7E | 68.4579 | 68.4579 | 0.0000 |
| 4 | 6 | S40-33-35.3E | 190.9859 | 190.9859 | 0.0000 |
| 6 | 7 | N 0-33-35.3W | 190.9859 | 190.9859 | 0.0000 |
| 7 | 8 | N89-26-24.7E | 75.0000 | 75.0000 | 0.0000 |

*Figure 11-9. INVERSE DIRECTIONS*

### 11.4.3 ANGLES

The ANGLES command computes and reports the clockwise angles from the baseline (*pBS* to *pOC*) to the points in the description *desFS*. ANGLES also computes the length of each line segment *as well as* tangents and offsets.

Operator Sequence

1. Select or key in ANGLES.

   *The system responds:*

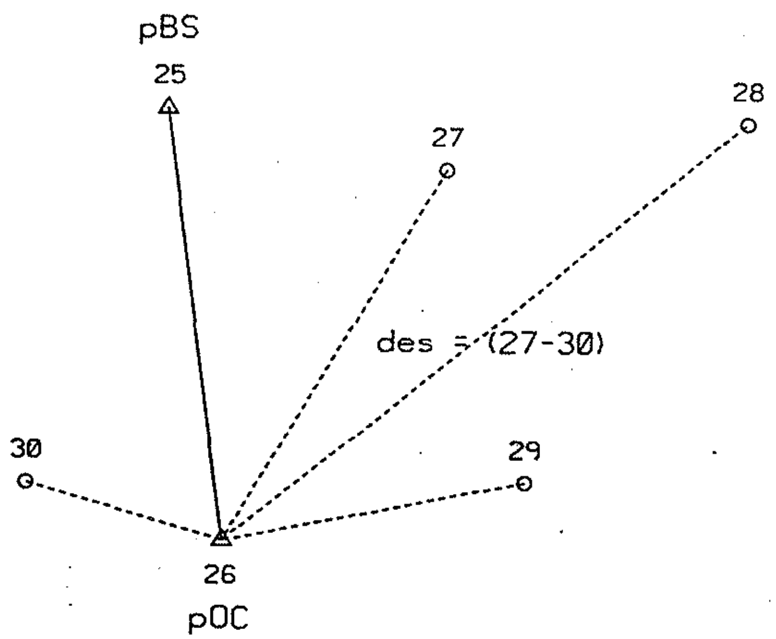
   ANGLES pBS pOC desFS

2. Identify the following:

   pBS the backsight point on baseline.

   pOC the occupied points from which angles will be turned.

   desFS description of the points to which angles will be turned.




REPORT FILE LISTING:

ANGLES 25 26 (27-30)

| | | | Layout From Point | | 26 | |
|---|---|---|---|---|---|---|
| At | From | To | Angle | Distance(FT) | Tangent(FT) | Offset(FT) |
| 26 | 25 | 25 | 0- 0- 0.0 | 210.3760 | 210.3760 | 0.0000 |
| 26 | 25 | 27 | 37-30-35.8 | 207.3407 | 164.4725 | 126.2496 |
| 26 | 25 | 28 | 58- 5- 0.7 | 320.0937 | 169.2280 | 271.7018 |
| 26 | 25 | 29 | 85-55-53.1 | 145.2091 | 10.3026 | 144.8431 |
| 26 | 25 | 30 | 294- 9-16.6 | 95.9395 | 39.2585 | -87.5395 |

*Figure 11-10. ANGLES*

### 11.4.4 TRAVERSE ANGLES

The TRAVERSE ANGLES command computes and reports the angles and distances between consecutive points in *des*.

Operator Sequence

1. Select or key in TRAVERSE ANGLES.

   *The system responds:*

   TRAVERSE ANGLES des

2. Identify the following:

   des description of the points.

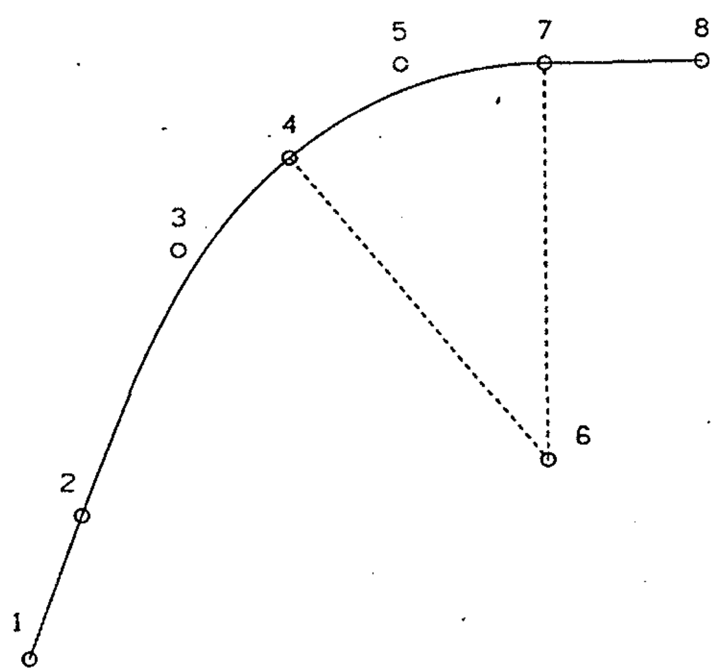



*Figure 11-11. TRAVERSE ANGLES*

REPORT FILE LISTING:

```
 TRAVERSE ANGLES  des
    1
     Point              Coordinates
 -----------------------------------------------------------
             1    4624.5109    1065.7849        0.0000
73.2044FT
             2    4693.5419    1090.1490        0.0000
                                                  Angle Right      180- 0- 0.0
135.2997FT
  PI          3    4821.1281    1135.1798
                                                  Spiral In
                                                  Deg of Curve     30- 0- 0.0
                                              a Degrees/station:    15.0000
                                                  Angle Right      210- 0- 0.0
68.4579FT
             4    4865.6423    1187.1892
                                                  Angle Right      270- 0- 0.0
190.9859FT
  CC          6    4720.5450    1311.3762
                                                  Deflection        40- 0- 0.0L
                                                  Arc Length         133.3333FT
                                                  Long Chord         130.6421FT
                                                                   N69-26-24.7E
                                                  Angle Right       40- 0- 0.0
190.9859FT
             7    4911.5218    1309.5102
                                                  Angle Right      270- 0- 0.0
75.0000FT
             8    4912.2546    1384.5066
```

*Figure 11-11. TRAVERSE ANGLES (Continued)*

## 11.4.5. TRAVERSE DEFLECTIONS

The TRAVERSE DEFLECTIONS command computes and reports the deflections and distances between consecutive points in *des*.

Operator Sequence

1. Select or key in TRAVERSE DEFLECTIONS.

   *The system responds:*

   TRAVERSE DEFLECTIONS des

2. Identify the following:

   des description of the points.

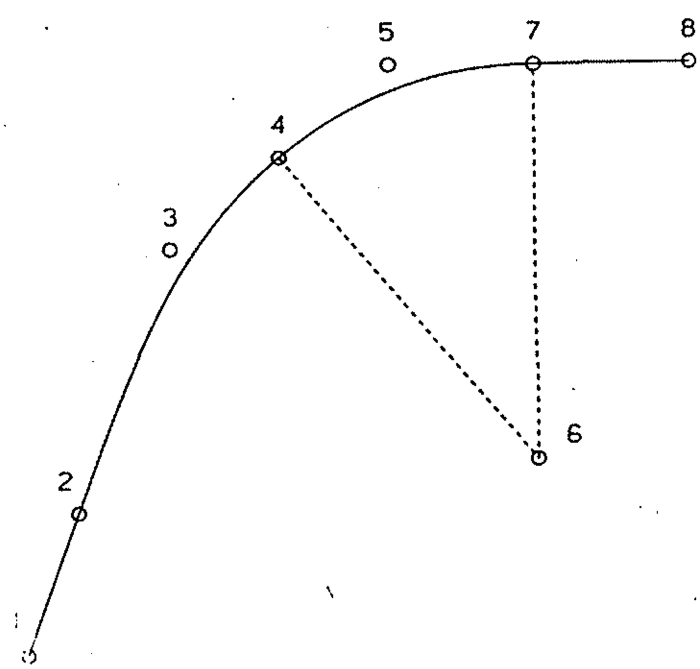



*Figure 11-12. TRAVERSE DEFLECTIONS*

REPORT FILE LISTING:

```
 TRAVERSE DEFLECTIONS  des
    1
      Point              Coordinates
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
             1      4624.5109    1065.7849       0.0000
73.2044FT
             2      4693.5419    1090.1490       0.0000
                                                   Deflection        0- 0- 0.0R
135.2997FT
  PI          3      4821.1281    1135.1798
                                                   Spiral In
                                                   Deg of Curve     30- 0- 0.0
                                              a Degrees/station:    15.0000
                                                   Deflection        30- 0- 0.0
68.4579FT
             4      4865.6423    1187.1892
                                                   Deflection        90- 0- 0.0R
190.9859FT
  CC          6      4720.5450    1311.3762
                                                   Delta             40- 0- 0.0
                                                   Arc Length         133.3333FT
                                                   Long Chord         130.6421FT
                                                                    N69-26-24.7E
                                                   Deflection       140- 0- 0.0L
190.9859FT
             7      4911.5218    1309.5102
                                                   Deflection        90- 0- 0.0R
75.0000FT
             8      4912.2546    1384.5066
```

*Figure 11-12. TRAVERSE DEFLECTIONS (Continued)*

PLOT Commands 12

# 12. PLOT COMMANDS

The PLOT commands menu block includes commands for placing graphic elements in IGDS design files and commands for creating coordinate, line, and curve tables. The PLOT menu block is divided into two parts:

- GENERAL Commands
- TABLE Commands.

## 12.1 GENERAL Commands

The GENERAL commands place graphic elements in IGDS design files at precise locations calculated by ICS. The characteristics of the graphic elements are determined by the parameters in the feature table. Design files do not have to be empty in order to place graphics in them; however, they must contain enough disk space to accept the graphic elements.

All GENERAL commands place annotation according to feature information stored in the database, unless a feature is specified in the command line. If a feature is specified in the command line, the PLOT commands use that feature. If you do not have a feature specified in the command line or the database, ICS uses the AUTO PLOT feature.

Note: Feature, justification, and elevation locks do not affect PLOT commands.

**Be careful when using 3D features in a 3D design file.** Figures plotted with 3D features in a 3D design file can be viewed three-dimensionally, but their plan (top) view might not be acceptable.

GENERAL commands include the following (see also the PLOT TEXT command in Section 14).

| *Command* | *For* |
|---|---|
| PLOT POINTS | placing point symbols with annotation |
| PLOT POV | placing points on a vertical alignment figure |
| PLOT CELL | placing cells (graphics only) |
| PLOT LINES | placing lines between points |
| PLOT SHAPE | placing complex shapes or connected line strings |
| PLOT CURVE | placing a curve string between points |
| PLOT ALIGNMENT | placing complex shapes or connected line strings when you are going to plot cross-section lines |
| PLOT VERTICAL ALIGNMENT | plotting vertical alignment figures |
| PLOT XSECTION | placing cross-section lines |
| PLOT FILE | placing ASCII files in a design file. |

## 12.1.1 PLOT POINTS

The PLOT POINTS command places the symbols and annotation for points described in *des*. The content of the information placed with PLOT POINTS depends on values in the feature table. If the ACTIVE FEATURE LOCK is ON, the content is determined by specifications in the active feature. If the ACTIVE FEATURE LOCK is OFF, the content is determined by specifications of the feature stored with the point. The justification of the annotation is also determined by that feature, unless the ACTIVE JUSTIFICATION LOCK is ON. If the ACTIVE JUSTIFICATION LOCK is ON, the justification from the IGDS menu overrides the justification in the feature table.

The feature table areas that affect the way points are placed include:

- Symbol area
  All the symbol characteristics apply.

- Annotation area
  All the annotation characteristics apply to the point ID except for the level and the justification. The justification of point ID's comes from the specifications in the Symbol Justification variable. The level comes from the 2nd Feature Number variable, or from ICS.PAR.

- Flag 2
  This field controls the content of the annotation placed with the point. The values are shown below.
  E (uppercase) plots elevation *only*
  X (uppercase) plots Northing and Easting
  Z (uppercase) plots Northing, Easting, and elevation
  N (uppercase) plots the name *only*
  e (lowercase) plots elevation *and* the name
  x (lowercase) plots Northing, Easting, *and* name
  z (lowercase) plots Northing, Easting, elevation, *and* name

- Misc 1 (character 4 only)
  Character 4 specifies if the point is 2D or 3D. The values are:
  2 for 2D
  3 for 3D

- 2nd feat num (2nd Feature Number)
  If the number in this field is in the range from 1 to 62, it is used at the active level for placement of the point ID. If the number in this field is 0 or greater than 62, the level for the point ID is taken from the ICS.PAR file.

Plotting the Feature Name with the Symbol

As shown above, you can plot the feature name with a symbol by placing an appropriate value in Flag 2 of the feature table. When plotted with coordinates, the feature name is included in the same text node as the coordinates. The name is separated from the coordinate(s) by a blank line.

The underbars in a feature name are considered as carriage returns. For example, the feature "centerline_road" would be plotted as:

centerline
road

whereas the feature "centerline road" would be plotted as

centerline road

Note: The *Annotation text width* field of the feature table controls the number of characters that will be placed in the design file. If the feature name has more characters than the *Annotation text width* field, the feature name will be truncated.

Operator Sequence

1. Select or key in PLOT POINTS.

   *The system responds:*

   PLOT POINTS des /fnum or !fname

2. Key in the following:

   des — description of the points to be plotted. This description *cannot* include more than 1000 points.

   fnum — the feature number.

   !fname — the feature name.

### 12.1.2 PLOT POV

The PLOT POV command places points on a vertical alignment figure. PLOT POV plots all points in the description *des* on the vertical alignment between *sBEG* and *sEND*. You can assign zero (0) to *sBEG* and *sEND* as a wild card to indicate from the beginning, or to the end, respectively.

Setup

Use SET ALIGNMENT and SET PROFILE before using this command.

Operator Sequence

1. Select or key in PLOT POV.

   *The system responds:*

   PLOT POV des /sBEG /sEND /fnum or lfname

2. Key in the following:

   des — the points to be plotted. The points must already have been located in the vertical alignment. The figure number of the vertical alignment can be used instead of *des*.

   sBEG — the beginning station.

   sEND — the ending station.

   fnum — the feature number.

   lfname — the feature name.

REPORT FILE LISTING:

```
PLOT POV  des /sBEG /sEND /fnum  or  lfname
   2 200 700
%VPL-I-BEGPLO, plot task in progress
     Plotting feature is        DEFAULT                        [    1]
%VPL-I-ENDPLO, plot task complete
```

*Figure 12-1. PLOT POV*

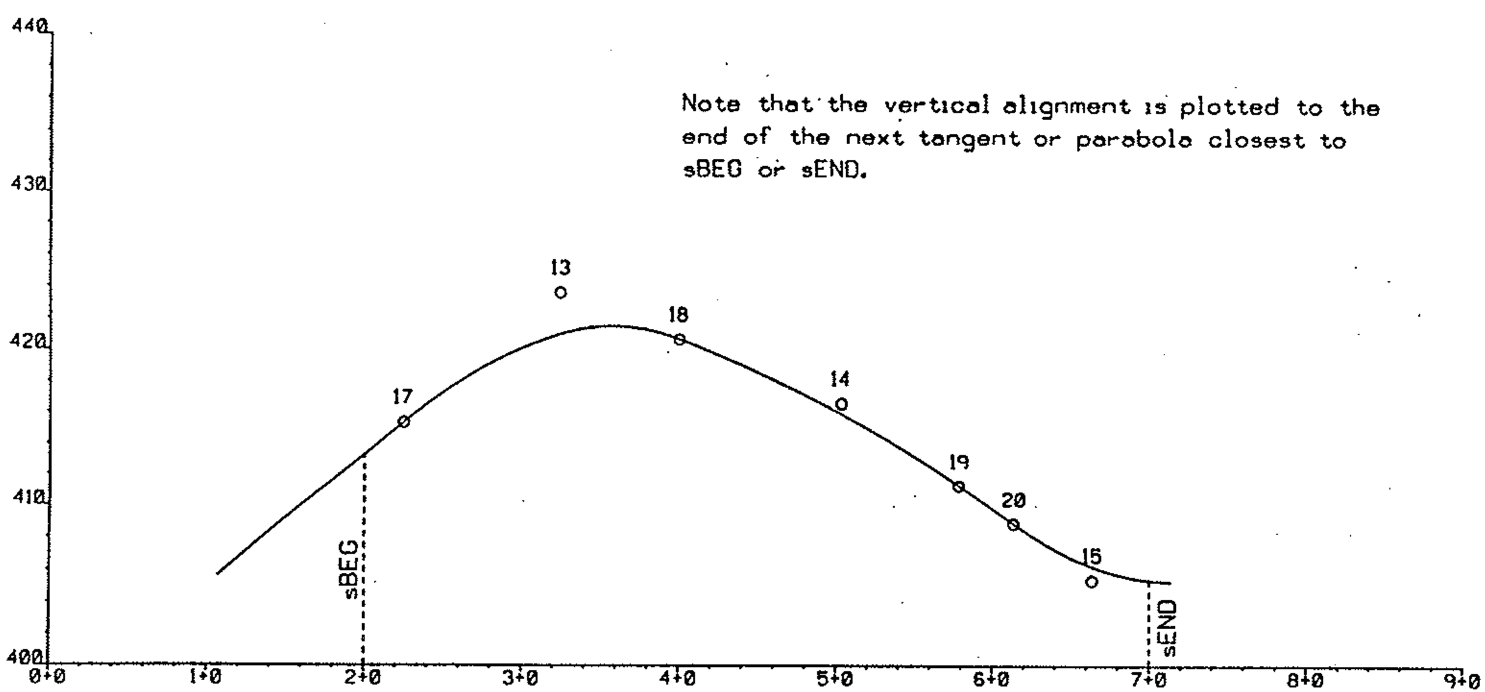



*Figure 12-1. PLOT POV (Continued)*

### 12.1.3 PLOT CELL (Graphics Only)

The PLOT CELL command places cells at an ICS point or range of points. You can specify the range by placing a fence around the desired points before you select the command, by keying in a range after you select the command, or by identifying each point with a data button.

The PLOT CELL command gives you the option to place the active cell or use information in the feature table to determine the cells to be placed. If you choose to use the active cell, you can review the active cell and either accept it or name a new active cell.

Note: Cell files that depict typical symbols and line patterns common to surveying are delivered with the product. Menus for activating these cell files are also delivered. For more information on how to install the point symbols, refer to the *Font Manager's User's Guide*. For more information on how to use the line pattern cells, refer to the FCPELT program documentation in Appendix C.

There are two ways to use PLOT CELL with the feature table. If the ACTIVE FEATURE LOCK is ON, the cell specified in the active feature is placed on all the points. If the lock is OFF, each point's feature is accessed to determine what cell (and from what library) is placed on that point. If the cell library or active cell in the feature table does not exist, you are prompted to change the feature table and to select PLOT CELL again.

<u>Operator Sequence</u>

1. Select PLOT CELL.

2. **<D> Active Cell / <R> Feature Table**

   Press <D> to place the active cell. *If there is an active cell, ICS prompts you to accept it by pressing <D> or reject it by pressing <R>. If you press <R> or if there is not active cell, ICS prompts you to key in the active cell name.*

   -OR-

   Press <R> to use the feature table to determine which cells are to be placed.

3. Identify the following:

   des      description of the points on which you want to place a cell.

### 12.1.4 PLOT LINES

The PLOT LINES command places lines between the points described in *des.*

Operator Sequence

1. Select or key in PLOT LINES.

   *The system responds:*

   PLOT LINES des /fnum or lfname

2. Key in the following:

   des the description of points between which the lines are to be plotted.

   fnum the feature number.

   lfname the feature name.

### 12.1.5 PLOT SHAPE

The PLOT SHAPE command places complex shapes (if the figure is closed) or connected line strings (if the figure is not closed) in the design file. The shape is closed if the first and last point numbers of the description are identical.

Operator Sequence

1. Select or key in PLOT SHAPE.

   *The system responds:*

   PLOT SHAPE des /fnum or Ifname

2. Key in the following:

   des the description of points that describe the shape to be plotted.

   fnum the feature number.

   Ifname the feature name.

### 12.1.6 PLOT CURVE

The PLOT CURVE command generates a curve string through the points listed in *des*. ICS complexes the curve string into a connected line string (open figure) or a complex shape (closed figure).

This command is useful for drawing lake or stream boundaries. You must turn the IGDS Fast Curve Display OFF to display this curve properly, and you should turn it ON to find the element for deletion.

Note: At least three points, none of which may be in a straight line, must exist for this command to work properly. Since this command is not intended for use with actual arcs, any figures which contain arcs (such as 2 C3R 4) will be plotted using the points only (2 3 4).

Operator Sequence

1. Select or key in PLOT CURVE.

   *The system responds:*

   PLOT CURVE des /fnum or lfname

2. Key in the following:

   des — the description of points that describe the curve.

   fnum — the feature number.

   lfname — the feature name.

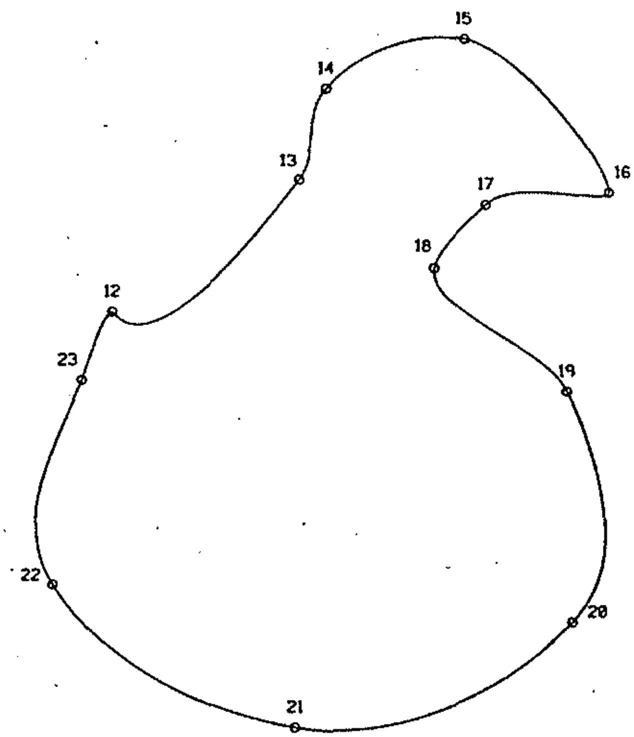



*Figure 12-2. PLOT CURVE*

## 12.1.7 PLOT ALIGNMENT

The PLOT ALIGNMENT command is very similar to the PLOT SHAPE command. The difference between the two commands is that if you are going to use the PLOT XSECTION command on the alignment, you use PLOT SHAPE, whereas if you are not going to use the PLOT XSECTION command on the alignment, you use PLOT ALIGNMENT.

PLOT ALIGNMENT forces a figure ID of less than 32768 for the alignment. This allows the figure ID to be the graphic group of the figure in subsequently placed SAC cross-sections—a SAC requirement.

Operator Sequence

1. Select or key in PLOT ALIGNMENT.

   *The system responds:*

   PLOT ALIGNMENT des /fnum or !fname

2. Key in the following:

   des description of the points that describe the alignment.

   fnum the feature number.

   !fname the feature name.

### 12.1.8 PLOT VERTICAL ALIGNMENT

The PLOT VERTICAL ALIGNMENT command plots the vertical alignment figure between *sBEG* and *sEND*. The vertical alignment figure is plotted to the end of the next tangent or parabola closest to *sBEG* or *sEND*, and the alignment is plotted as a profile in two dimensions in the x,y plane.

You can assign zero (0) to *sBEG* and *sEND* as a wild card to indicate from the beginning, or to the end, respectively. If you omit *sEND* from the data field, PLOT VERTICAL ALIGNMENT plots to the end of the alignment.

Setup

Use SET ALIGNMENT and SET PROFILE before using this command.

Operator Sequence

1. Select or key in PLOT VERTICAL ALIGNMENT.

   *The system responds:*

   PLOT VERTICAL ALIGNMENT sBEG /sEND /fnum or lfname

2. Identify the following:

   sBEG the beginning station.

   sEND the ending station.

   fnum the feature number.

   lfname the feature name.

REPORT FILE LISTING:

```
PLOT VERTICAL ALIGNMENT  sBEG /sEND /fnum or lfname
   200 700
%VPL-I-BEGPLO, plot task in progress
%VPL-I-ENDPLO, plot task complete
```

Note: See Figure 12-1 for a display of the vertical alignment.

*Figure 12-3. PLOT VERTICAL ALIGNMENT*

### 12.1.9 PLOT XSECTION

The PLOT XSECTION command places cross-section lines in the design file at all points of transition along an alignment and at user-specified intervals. It is also possible to plot single (plus) stations by station value. If you want to place only one cross-section line, set $x$ to zero (0) and *sBEG* to the station where you want the cross-section line (in this case, *sEND* must be the same as *sBEG*).

Note: In order to use PLOT XSECTION, you must have stored a figure with a station on a point or have performed SET ALIGNMENT on that figure.

Setup

Before selecting this command, you must store the figure (alignment) with a station on a point or perform SET ALIGNMENT on the figure (alignment).

Operator Sequence

1. Select or key in PLOT XSECTION.

   *The system responds:*

   PLOT XSECTION Hfg x offR /offL /sBEG /sEND

2. Identify the following:

   Note: In graphics, *offR* and *offL* are taken from active offset values, with *offR* equal to *off1* and *offL* equal to *off2*.

   Hfg — a figure or group of points defining an alignment.

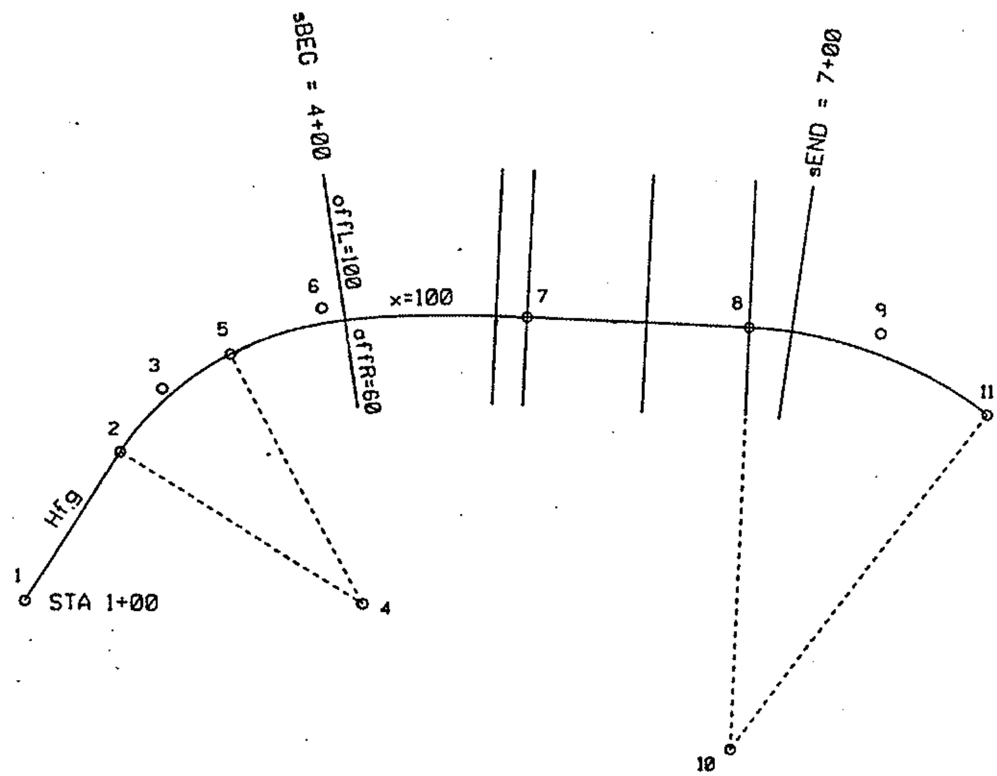
   x — the distance between section lines. If zero (0), a single (plus) station will be placed.

   offR — the distance to extend the section line to the right of the alignment.

   offL — the distance to extend the section line to the left of the alignment.

   sBEG — indicates (if $x$ is not zero) the station at which to begin placement of cross sections. If $x$ is zero, a single station is placed at *sBEG*.

   sEND — indicates (if $x$ is not zero) the station at which to terminate placement of cross sections.




REPORT FILE LISTING:

```
PLOT XSECTION  Hfg x offR/offL /sBEG /sEND
   1 100 60 100 350 750
     Plotting feature is       DEFAULT                        [    1]
%PXS-I-ENDLAB, label task complete
```

*Figure 12-4. PLOT XSECTION*

### 12.1.10 PLOT FILE (Graphics Only)

The PLOT FILE command plots ASCII files to a design file. This command is useful for plotting coordinate, line, and curve tables to your design file. These tables can be created using the TABLE commands described in Section 12.2.

**Note: The tables are all created using a monospace font. In order for these tables to display correctly, you must set the active font to a monospace font, such as font 50.**

Setup

Before selecting this command, set the active font equal to a monospace font.

Operator Sequence

1. Select PLOT FILE

2. **Keyin text filename Reset/exit dev:[UIC]filename.EXT**

   Key in the filename in the format shown.

   -OR-

   Press <R> to exit.

3. **Identify origin Reset/exit**

   Place the cursor where you want the text to be placed and press <D>.

   -OR-

   Press <R> to exit.

**Note: ICS places the text according to the IGDS active text node justification, line spacing, color, weight, text size, etc.**

## 12.2 TABLE Commands

The TABLE commands create or enable the creation of ASCII files that contain a table of information. These ASCII files can then be plotted to a design file using the PLOT FILE command (see Section 12.1.10). The TABLE commands include the following:

| *Command* | *For* |
|---|---|
| COORDINATE TABLE | creating a table of coordinate information |
| LINE TABLE | enable/disable the creation of a table of line information |
| CURVE TABLE | enable/disable the creation of a table of curve information |

### 12.2.1 COORDINATE TABLE (Graphics Only)

The COORDINATE TABLE command writes the coordinates of points you specify in *des* to an ASCII file containing the coordinate table. You can give a figure ID, point, or range of points in *des*. After creating the coordinate table, you can plot it to the design file using the graphics command PLOT FILE (see Section 12.1.10 for details).

You define the coordinate file format and content by editing the ICS_ANNOTATE.PAR file. This file contains a series of variables under the header $COORDINATE_TABLE. Refer to Section 3.7 for more details about ICS_ANNOTATE.PAR.

<u>Operator Sequence</u>

1. Select COORDINATE TABLE.

   *The system responds:*

   COORDINATE TABLE des

2. Identify the following:

   des description of the points to be listed in the coordinate table.

| POINT # | EASTING (FT) | NORTHING (FT) | ELEV. (FT) |
|---|---|---|---|
| 1 | -6065.6009 | 1171.6479 | 410.0000 |
| 2 | -6000.6299 | 1273.1639 | 422.0000 |
| 4 | -5839.7685 | 1170.2114 | |
| 5 | -5927.6023 | 1339.8016 | 429.0000 |
| 6 | -5866.8136 | 1371.2852 | |
| 7 | -5731.6552 | 1365.1015 | 422.0000 |
| 8 | -5581.8119 | 1358.2459 | 412.0000 |
| 10 | -5594.9051 | 1072.0664 | |
| 11 | -5420.0340 | 1298.9810 | 410.0000 |

*Figure 12-5. Plot of a Coordinate Table Generated with COORDINATE TABLE*

### 12.2.2 LINE TABLE and CURVE TABLE (Graphics Only)

The LINE TABLE and CURVE TABLE commands are toggle commands that allow you to enable the generation of line and curve tables. The line and curve data is written to ASCII files by selecting the LABEL LINE command. If the LINE TABLE option is active, *all* lines within the figure will be annotated with a line number. If the CURVE TABLE option is active, *all* curves within the figure will be annotated with a curve number.

The line data is written to an ASCII file containing the line table, and the curve data is written to an ASCII file containing the curve table. After creating one or both of these tables, you can plot them to the design file using the graphics command PLOT FILE (see Section 12.1.10 for details).

You define the format and content of the ASCII files by editing the ICS_ANNOTATE.PAR file. This file contains a series of variables under the headers $LINE_TABLE and $CURVE_TABLE. Refer to Section 3.7 for more details about ICS_ANNOTATE.PAR.

<u>Handling Sequential Line or Curve Data</u>

You have several options available to you for handling sequential line or curve data resulting from labeling more than one figure.

- Append new data to existing files
  ICS maintains an active curve or line table and automatically appends new line or curve data to existing files. You specify the filenames in the ICS_ANNOTATE.PAR file.

  Note: You should not modify the table formats and contents if you are appending data.

- Create new files for new figures
  By editing the filename in the ICS_ANNOTATE.PAR file, you can place data for a new figure in another file. And if desired, the new file can have a different format and content. In contrast to the option above, the line and curve data would not be appended because the filename would be different.

- Edit the curve and line labels
  You can edit the next_line and next_curve variables in ICS_ANNOTATE to control the line/curve labels. (See Section 3.7 for more details.)

Generating Curve and Line Tables by Default

You also have the option to have curve and line tables generated by default instead of explicitly requesting them. Like other annotation handling, having the tables created by default is controlled by the feature table variable Misc 1 (miscellaneous one). ICS version 8.8.3 allowed you to enable/disable scale and/or rotation of annotation text that did not fit the graphics element being annotated. This is now extended to allow you to default to a line and curve table. With this default set, if annotation text does not fit on the graphics element being annotated, ICS labels the lines and curves with the current line or curve label; ICS makes no attempt to scale or rotate the annotation text. ICS interprets field three of Misc 1 as:

0 disable rotation
1 enable rotation
2 disable rotation and scaling and default to a curve/line table.

The available curve parameters for a default curve table is limited to four items. It is recommended that these conform to the values specified in ICS.PAR and fields five through eight of Misc 1. Possible curve annotation parameters currently include radius, delta, arc length, and tangent. The default curve table is intended to hold these items or a subset that you have defined for curve annotation.

Note: ICS will maintain separate table and default table files. The next available curve or line number is common to the respective files. You can append default line/curve data to the line/curve table files by assigning the same name to the table and default table file. **It is your responsibility to ensure that curve and default curve format and content is consistent if you use a common file.**

ANNOTATE Commands 13

# 13. ANNOTATE COMMANDS

ANNOTATE commands include commands that place text in a design file to provide information on points, lines, areas, and alignments. Annotation does not follow the DATA READOUT FORMAT set in DESIGN OPTIONS--it is always plotted in terms of decimal master units. The ANNOTATE menu block is divided into four parts:

- LOCKS Commands
- LABEL Commands
- LEADERS Commands
- LAYOUT Commands.

## 13.1 LOCKS Commands

Use the LOCKS commands to specify the content of elevation and stations/offset annotation. LOCKS commands include the following:

| *Command* | *For* |
|---|---|
| ELEVATION ANNOTATION | specifying the content of elevation annotation that ICS will use when labeling points |
| STATIONS/ OFFSETS | specifying the content of stations/offsets annotation that ICS will use when labeling stations/offsets. |

### 13.1.1 ELEVATION ANNOTATION (Graphics Only)

The ELEVATION ANNOTATION command sets the content of elevation annotation that ICS will use when labeling points with the ANNOTATE COORDINATES command. ELEVATION ANNOTATION gives you three options for annotation:

- x, y, and z
- x and y only
- z only.

Note: ELEVATION ANNOTATION does not affect the PLOT POINTS command.

Operator Sequence

1. Select ELEVATION ANNOTATION.

   *The system displays the active annotation content.*

2. Press <D> to accept the current option.

   -OR-

   Press <R> the scroll to the next available option. Go back to Step 2.

### 13.1.2 STATIONS/OFFSETS (Graphics Only)

The STATIONS/OFFSETS command sets the content of point annotation that ICS will use when labeling stations/offsets. This command gives you three options for annotation:

- station and offset
- station only
- offset only.

Operator Sequence

1. Select STATIONS/OFFSETS.

   *The system displays active annotation content.*

2. Press <D> to accept the current option.

   -OR-

   Press <R> the scroll to the next available option. Go back to Step 2.

## 13.2 LABEL Commands

LABEL commands include the following:

| *Command* | *For* |
|---|---|
| COORDINATES | labeling coordinates of points |
| LABEL ALIGNMENT | labeling a horizontal alignment at all points of transition and at specified intervals |
| LABEL VERTICAL ALIGNMENT | labeling a vertical alignment at points of transition |
| AREA | computing and labeling the area of a closed polygon |
| LABEL LINES | labeling lines with their radius, length, tangent length, and/or central angle |
| LABEL VERTICAL STATIONS | labeling the station and elevation of points. |

### 13.2.1 COORDINATES (Graphics only)

The COORDINATES command annotates coordinates of points according to specifications in the feature table and active parameters. The content of the annotation is determined by the ELEVATION ANNOTATION LOCK (see Section 13.1.1). The justification of the annotation is determined by the ACTIVE JUSTIFICATION LOCK, or specifications in the feature table. If the ACTIVE JUSTIFICATION LOCK is ON, the justification from the IGDS menu overrides the justification of the feature table. If the ACTIVE JUSTIFICATION LOCK is OFF and the ACTIVE FEATURE LOCK is ON, the justification is determined by the annotation justification parameter of the active feature. If both these locks are OFF, the justification is determined by the annotation justification parameter of the feature stored with the point.

Operator Sequence

1. Select COORDINATES.

2. Identify the points to be annotated.

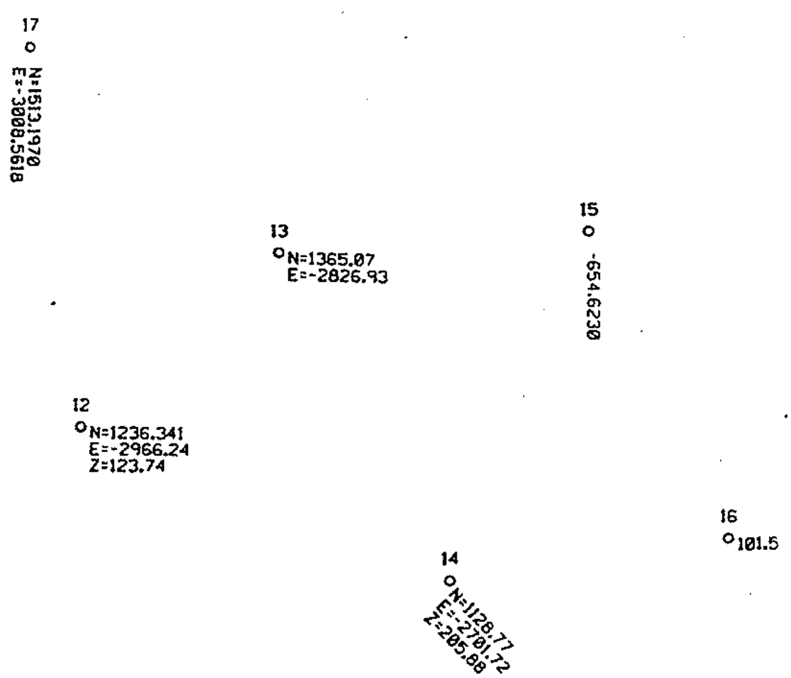



*Figure 13-1. COORDINATES (Showing Effect of ACTIVE ANGLE)*

### 13.2.2 LABEL ALIGNMENT

The LABEL ALIGNMENT command annotates an alignment at all points of transition along the alignment as well as at various user-specified intervals, starting at *sBEG* and ending at *sEND*. This command plots the beginning and ending stations only if those stations coincide with the increment you specify.

Note: Use the LABEL ALIGNMENT command in conjunction with the PLOT XSECTION command to label section lines.

Variables in the parameter file ICS_ANNOTATE.PAR affect how labels are placed by LABEL ALIGNMENT. A namelist called LABEL_ALIGN in ICS_ANNOTATE.PAR contains the parameters listed below (see Figure 13-2 for a diagram showing these parameters). You can edit ICS_ANNOTATE.PAR to set these parameters to the values you want (see Section 3.7 for instructions).

| | |
|---|---|
| label_reading | right or ahead; reading for PI, PC, PT, etc. station labeling |
| major_tic_x | the distance between major ticks on the alignment |
| ma_mi_scale | the ratio of major to minor tick marks (size only) |
| tic_side | right or left; the side where ticks are to be placed |
| label_side | for curves and spirals *only*: 1 to indicate that station labels are to be placed on the same side as tick marks; 2 to indicate that station labels are to be placed at the opposite side |
| station_orient | 'PA' for labels to be placed parallel to the alignment; 'PE' (default) for labels to be placed perpendicular to the alignment |
| full_sta_format | 1 for full station format (with +00); 0 for not the full station format (without +00) |
| with_leader | Y for labels to be placed with leaders; N for labels not to be placed with leaders. |

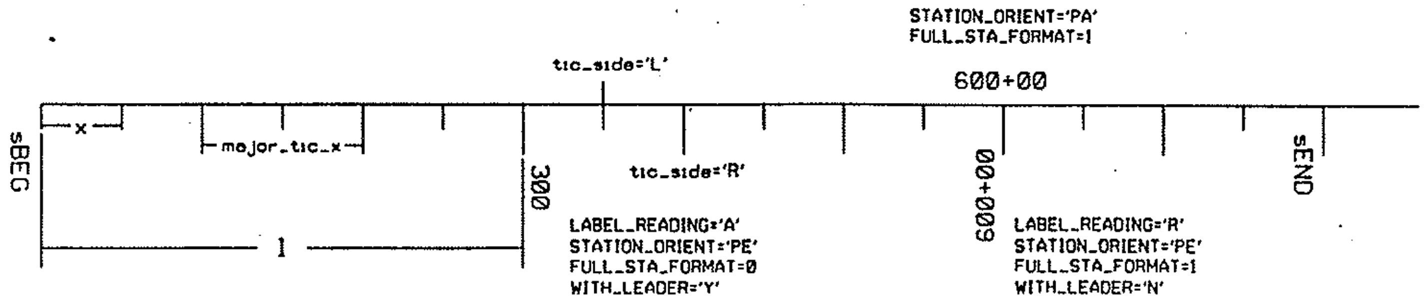



*Figure 13-2. Diagram of ICS_ANNOTATE.PAR Parameters*

Operator Sequence

1. Select or key in LABEL ALIGNMENT.

   *The system responds:*

   LABEL ALIGNMENT Hfg /x /l /fnum /sBEG /sEND /lfname

2. Identify the following:

   Hfg the horizontal figure or group of points defining the alignment.

   x the distance between ticks on the alignment.

   l the distance between labels along the alignment.

   fnum a feature number, ignored if zero (0).

   sBEG the beginning station.

   sEND the ending station. If *sBEG* and *sEND* are omitted, all stations are labeled.

   lfname the feature name.

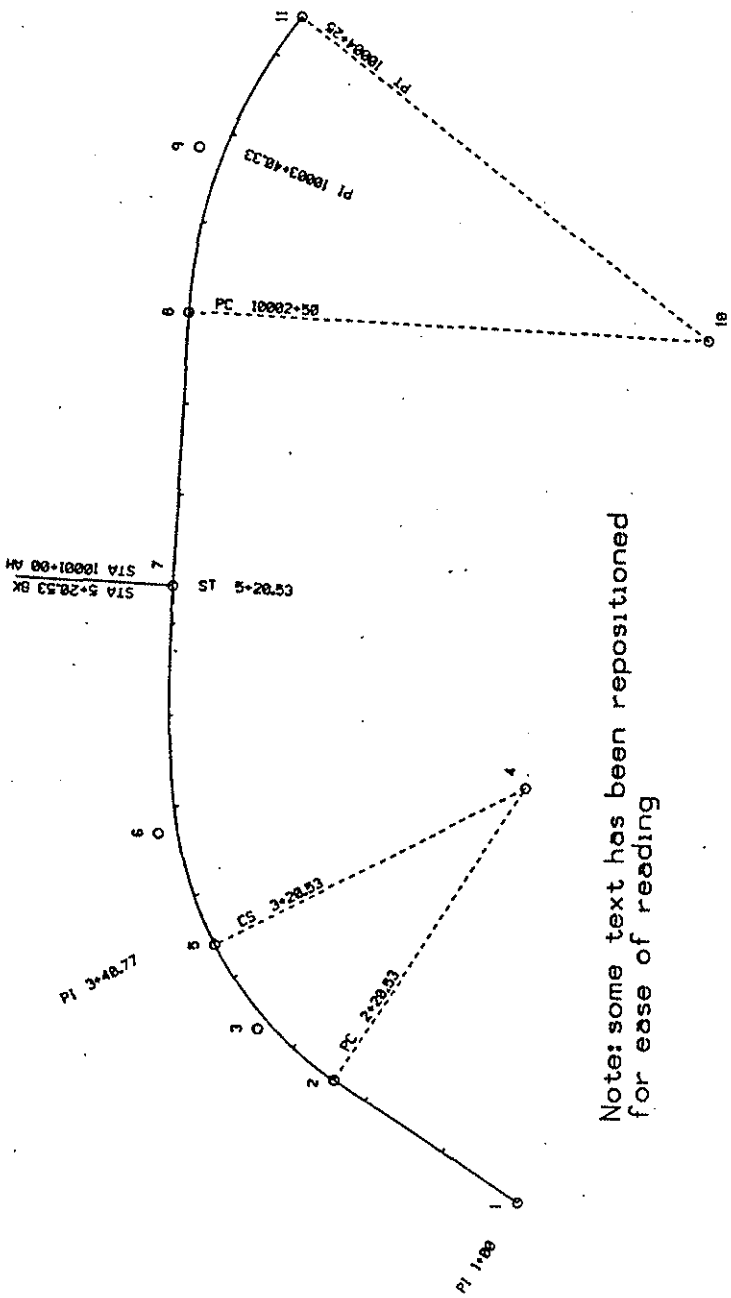
---

REPORT FILE LISTING:

```
LABEL ALIGNMENT  Hfg /x /l /fnum /sBEG /sEND /lfname
   1 50 200
%LAL-I-BEGLAB, label task in progress
     Plotting feature is       DEFAULT                        [ 628]
%LAL-I-ENDLAB, label task complete
LABEL LINES  des /fnume  or  lfname
   1
     Plotting feature is       DEFAULT                        [ 628]
%PLS-I-BEGLAB, label task in progress
%PLS-I-ENDLAB, label task complete
```

*Figure 13-3. LABEL ALIGNMENT*

---




Note: some text has been repositioned
for ease of reading

*Figure 13-3.* *LABEL ALIGNMENT (Continued)*

### 13.2.3 LABEL VERTICAL ALIGNMENT

The LABEL VERTICAL ALIGNMENT command labels the *PVI*, *PVC*, *PVT*, and the grade of the tangents between *sBEG* and *sEND*. If you omit *sBEG* and *sEND* or assign zero (0) to each, this command labels the whole vertical alignment. You have the option to specify the number of decimal places *dec* used for the grade annotation, which is placed on vertical alignment segments.

Note: All labels placed by this command are text nodes, and they can be modified in ICS.PAR (see Section 3.6).

Setup

Run SET ALIGNMENT and SET PROFILE before selecting this command.

Operator Sequence

1. Select or key in LABEL VERTICAL ALIGNMENT.

   *The system responds:*

   LABEL VERTICAL ALIGNMENT sBEG /sEND /dec /fnum or lfname

2. Identify the following:

   sBEG the beginning station.

   sEND the ending station.

   dec the number of decimal places for the grade annotation.

   fnum the feature number.

   lfname the feature name.

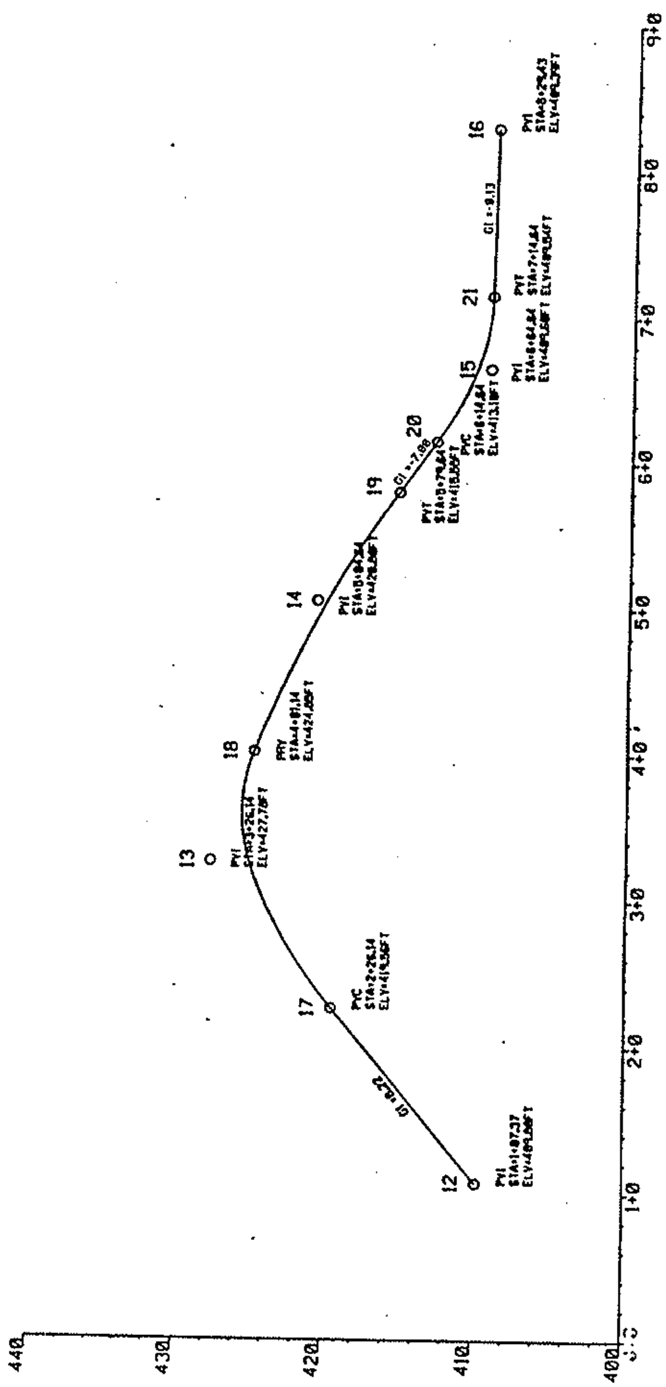



*Figure 13-4. LABEL VERTICAL ALIGNMENT*

### 13.2.4 AREA

The AREA command computes the area of a closed polygon defined in *des*. The last point number in the list of points must be the same as the first point to properly define a closed polygon. The units of measure used for the label are defined in ICS.PAR. You can plot master units, subunits, or both. In graphics, the area labels are placed at the center of the polygon. In alpha, the area is displayed on the screen.

Operator Sequence

1. Select AREA.

   *The system responds:*

   AREA des

2. Identify the following:

   des    the description of the polygon.

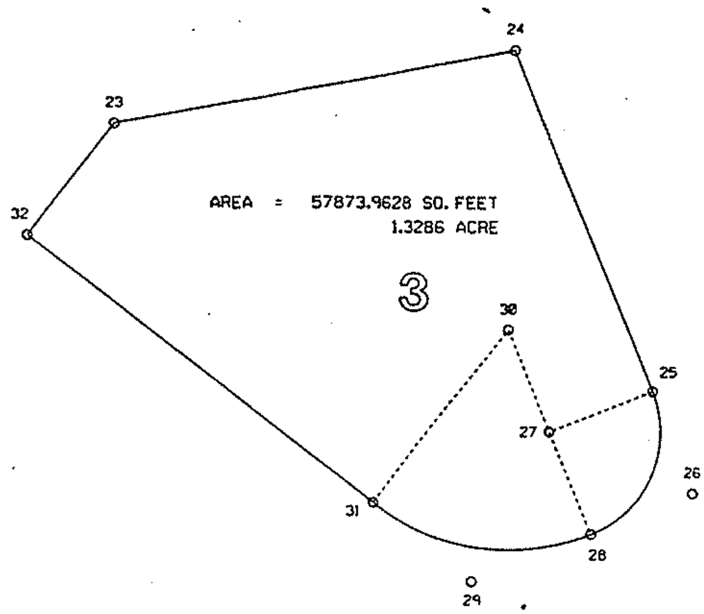



*Figure 13-5. AREA*

### 13.2.5 LABEL LINES

The LABEL LINES command annotates lines and curves defined in *des* according to specifications in ICS.PAR and the feature table variable Misc 1 (miscellaneous one). The units of measurement are defined in ICS.PAR (see Section 3.6). The content of line annotation is defined in the feature table variable Flag 3 (see Appendix A). The contents of curve annotation and how the annotation is placed are defined in the feature table variable Misc 1 (see Appendix A). The text justification is affected by the ACTIVE JUSTIFICATION command (see Section 7.3.6).

For line annotation, LABEL LINES places text groups (as IGDS text nodes) that form descriptive labels directly on the line. When you move a line and its label in a design file, you can move the label as a text group without breaking it up. To break up text, simply perform the IGDS DROP STATUS command on the text node. LABEL LINES places cells when annotation is placed *along* a circular or spiral arc.

Text placed along a line or curve is placed from the first point defining the line or curve to the second point. For example, for a description (1 2) the text is read from left to right from point 1 to point 2. For a description (2 1) the text is read from left to right from point 2 to point 1.

The dimensional accuracy of annotation depends on the decimal data readout that is defined in the design file. Directional annotation depends on the angle format of the design file (bearings or azimuths).

<u>The Feature Table Variable Flag 3</u>

Flag 3 controls whether the line annotation includes distance, direction, or distance and direction. By default, ICS labels both the distance and direction.

<u>The Feature Table Variable Misc 1 (Miscellaneous One)</u>

The content of the curve annotation is controlled by fields 5 through 8 of Misc 1. You can annotate the lines with any or all of the following in any order:

- Radius of Curve (R)
- Length of Curve (L)
- Tangent Length of Curve (T)
- Central Angle of Curve (D).

By default, LABEL LINES labels the lines with all the items listed above in the order listed above. Also, by default the labels used for each is the letter shown to the right such as D= for the central angle. You can change these letters in ICS.PAR.

Characters 1 through 3 of Misc 1 control how the annotation is placed if it will not fit on the line. If the annotation does not fit, you can:

- Scale only
- Rotate only
- Scale and rotate
- Label the line/curve and generate tables (see Section 12.1.2).

If text will not fit along a line or curve, ICS first attempts to scale the text (if scaling is enabled). If the text still will not fit, ICS rotates the text without scaling. As a last resort, ICS will rotate and scale the text. For lines, ICS places the text perpendicular to and centered about, the line. For curves, ICS places the text perpendicular to and centered about an imaginary line running from the center point of the curve to the PI. If the text will not fit even with rotating and scaling, ICS places the text upright in the design file somewhere close to the line or curve.

Note: If the LINE TABLE command is active, *all* lines are annotated with a line number, and if the CURVE TABLE command is active, *all* curves are annotated with a curve number. The data is placed in a line or curve table. See Section 12.2.2 for more details.

Operator Sequence

1. Select or key in LABEL LINES.

   *The system responds:*

   LABEL LINES des /fnum or lfname

2. Identify the following:

   des description of the line or figure to be labeled.

   fnum the feature number.

   lfname the feature name.

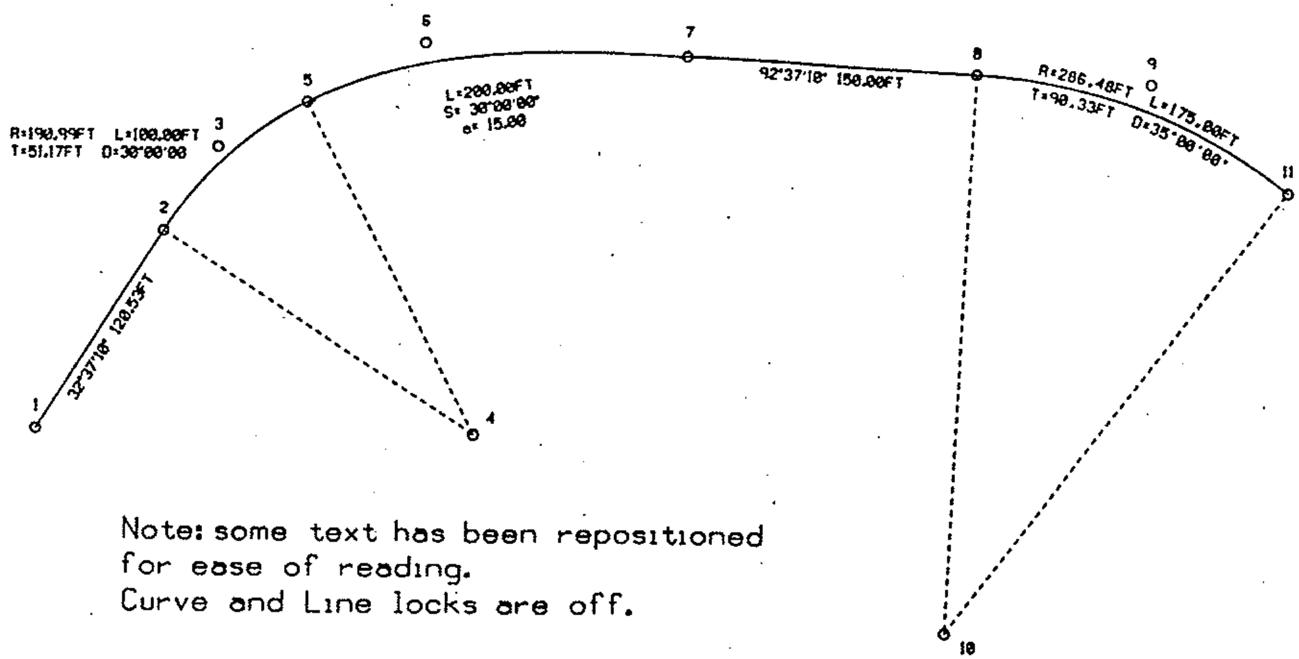
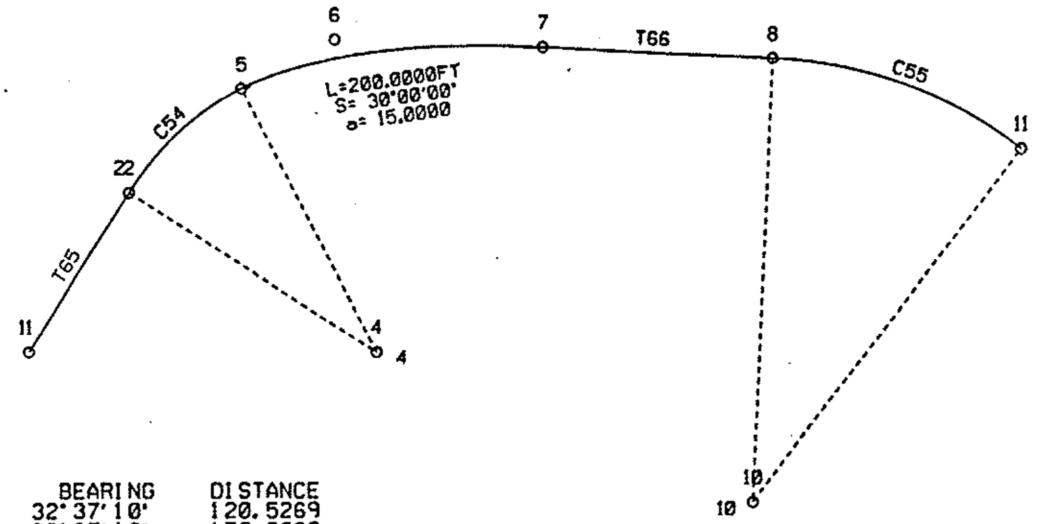



*Figure 13-6. LABEL LINES with LINE and CURVE LOCKS OFF*




| NO. | BEARING | DISTANCE |
|---|---|---|
| T65 | 32°37'10" | 120.5269 |
| T66 | 92°37'10" | 150.0000 |

| NO. | RADIUS | DELTA | LENGTH | TANGENT | TANGENT BRNG | CHORD | CHORD BRNG |
|---|---|---|---|---|---|---|---|
| C54 | 30°00'00" | 30°00'00" | 100.0000 | 51.1745 | 32°37'10" | 98.8616 | 47°37'10" |
| C55 | 20°00'00" | 35°00'00" | 175.0000 | 90.3264 | 92°37'10" | 172.2917 | 110°07'10" |

*Figure 13-7. LABEL LINES with LINE and CURVE LOCKS ON*

### 13.2.6 LABEL VERTICAL STATIONS

The LABEL VERTICAL STATIONS command labels points in the description *des* with the station and elevation.

Operator Sequence

1. Select or key in LABEL VERTICAL STATIONS.

   *The system responds:*

   LABEL VERTICAL STATIONS des /fnum or lfname

2. Identify the following:

   des — description of the points to be labeled.

   fnum — the feature number.

   lfname — the feature name.

REPORT FILE LISTING:

```
LABEL VERTICAL STATIONS  des /fnum or lfname
   (1 2 3 5 7 8 11)
%VPL-I-BEGLAB, label task in progress
%VPL-I-ENDLAB, label task complete
```

*Figure 13-8. LABEL VERTICAL STATIONS*

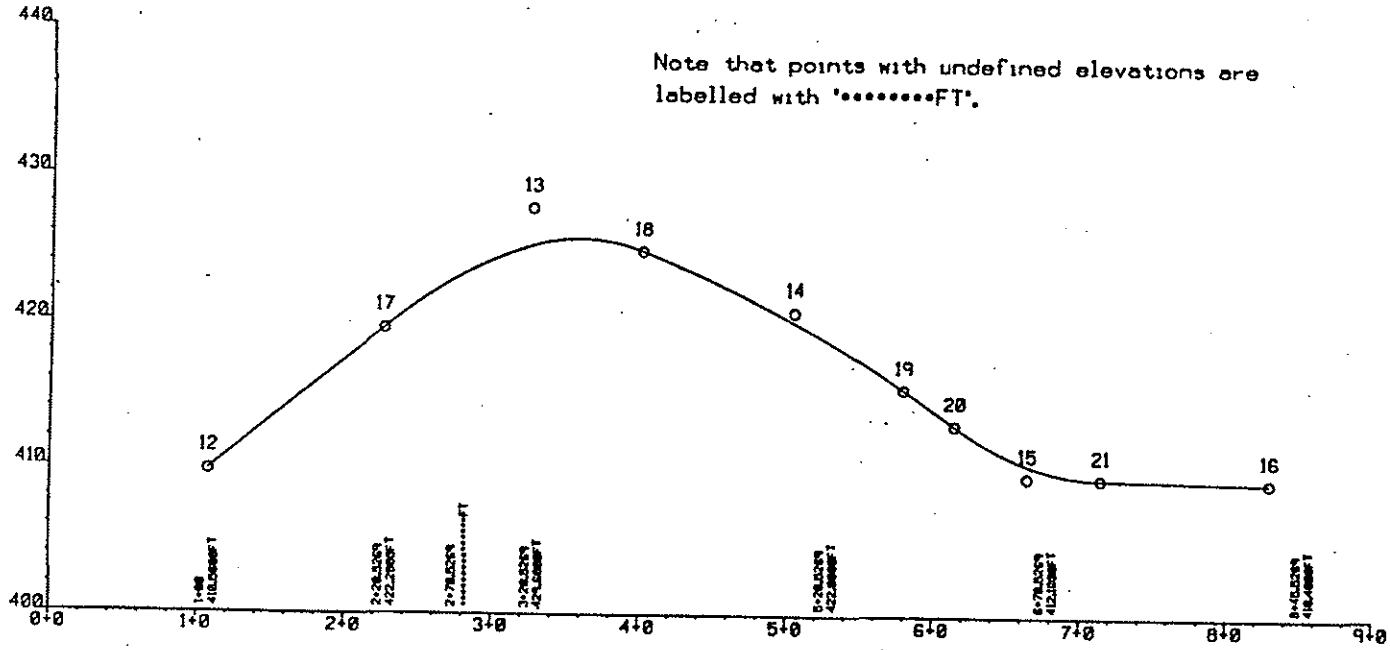



*Figure 13-8.* ***LABEL VERTICAL STATIONS*** *(Continued)*

## 13.3 LEADERS Commands

All the LEADERS commands are affected by the ELEVATION ANNOTATION LOCK. If this lock is set to *coordinates with elevation*, both the coordinates and height annotation is placed. LEADERS commands include the following:

| *Command* | *For* |
|---|---|
| COORDINATES ANGLED | placing coordinates of points along angled leaders |
| COORDINATES STRAIGHT | placing coordinates of points along straight leaders |
| STATIONS ANGLED | placing stations of points along angled leaders |
| STATIONS STRAIGHT | placing stations of points along straight leaders. |

### 13.3.1 COORDINATES ANGLED LEADER (Graphics Only)

The COORDINATES ANGLED LEADER command places the coordinates of specified points along angled leaders. If you want to place identical leaders on a set of points, use a fence. Then you only have to indicate the position of a single leader, and the system attaches identical leaders running from each point in the fence.

Operator Sequence

1. Select COORDINATES ANGLED LEADER.

2. **Identify point to annotate.**

   Position the cursor on the point you want annotated (or any point in the fence) and press <D>. (Then accept by pressing <D> again.)

3. **Identify vertex point of leader**

   Position the cursor on the vertex of the leader for the point identified and press <D>.

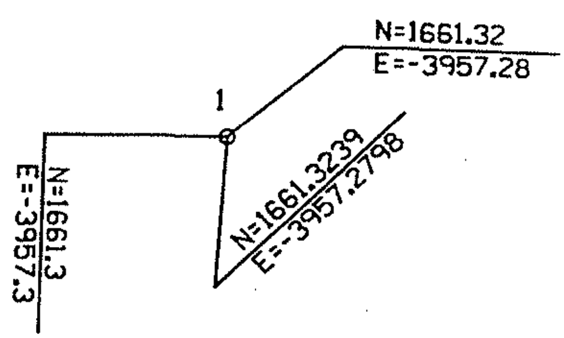



*Figure 13-9. COORDINATES ANGLED LEADER*

## 13.3.2 COORDINATES STRAIGHT LEADER (Graphics Only)

The COORDINATES STRAIGHT LEADER command places the coordinates of specified points along straight leaders. If you want to place identical leaders on a set of points, use a fence. Then you only have to indicate the position of a single leader, and the system attaches identical leaders running from each point in the fence.

Operator Sequence

1. Select COORDINATES STRAIGHT LEADER.

2. **Identify point to annotate.**

   Position the cursor on the point you want annotated (or any point in the fence) and press <D>. (Then accept by pressing <D> again.)

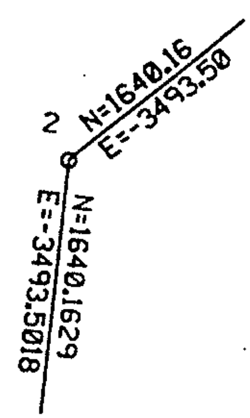



*Figure 13-10. COORDINATES STRAIGHT LEADER*

### 13.3.3 STATIONS ANGLED LEADER (Graphics Only)

The STATIONS ANGLED LEADER command places the stations and/or offsets of specified points along angled leaders. If you want to place identical leaders on a set of points, use a fence. Then you only have to indicate the position of a single leader, and the system attaches identical leaders running from each point in the fence.

<u>Operator Sequence</u>

1. Select STATIONS ANGLED LEADER.
2. **Identify point to annotate.**

   Position the cursor on the point you want annotated (or any point in the fence) and press <D>. (Then accept by pressing <D> again.)
3. **Identify vertex point of leader**

   Position the cursor on the vertex of the leader for the point identified and press <D>.

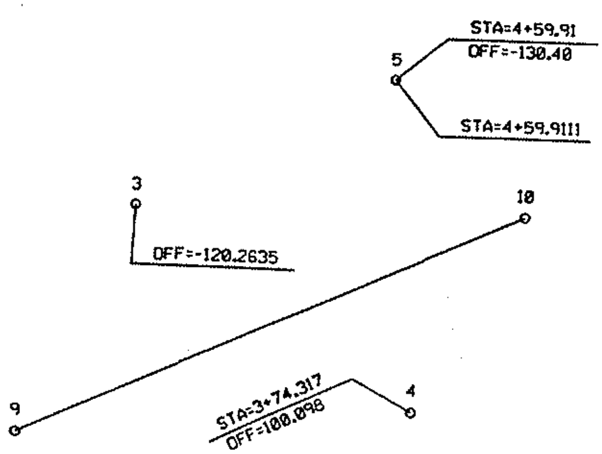



*Figure 13-11. STATIONS ANGLED LEADER.*

## 13.3.4 STATIONS STRAIGHT LEADER (Graphics Only)

The STATIONS STRAIGHT LEADER command places stations and/or offsets of specified points along a straight leaders. If you want to place identical leaders on a set of points, use a fence. Then you only have to indicate the position of a single leader, and the system attaches identical leaders running from each point in the fence.

Operator Sequence

1. Select STATIONS STRAIGHT LEADER.

2. **Identify point to annotate.**

   Position the cursor on the point you want annotated (or any point in the fence) and press <D>. (Then accept by pressing <D> again.)

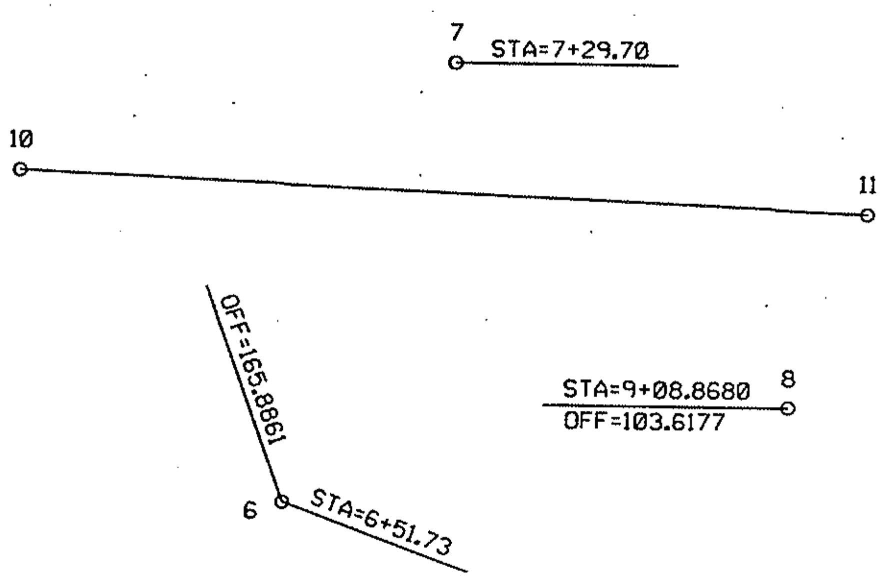



*Figure 13-12. STATIONS STRAIGHT LEADER*

## 13.4 LAYOUT Commands

LAYOUT commands include the following:

| *Command* | *For* |
|---|---|
| ANGLES AND DISTANCES | annotation lines with angles and distances |
| TANGENTS AND OFFSETS | annotation lines with tangents and offsets. |

### 13.4.1 ANGLES AND DISTANCES (Graphics Only)

The LAYOUT ANGLES AND DISTANCES command annotates the line segments between the occupied point *pOC* and the foresight points in *des*. The angle right (from *pBS*) and distance annotation are placed on the lines between the *pOC* and the points in *des*.

LAYOUT ANGLES AND DISTANCES uses the element and annotation weight and color values set for the active feature to display the linework and annotation. For the line between the backsight and the occupied point, it uses the element weight plus one for emphasis. The line style is always medium dash.

Operator Sequence

1. Select ANGLES AND DISTANCES.

2. Identify the following points:

| | |
|---|---|
| pBS | the known backsight point. |
| pOC | the known occupied point. |
| desFS | description of the foresight points. |

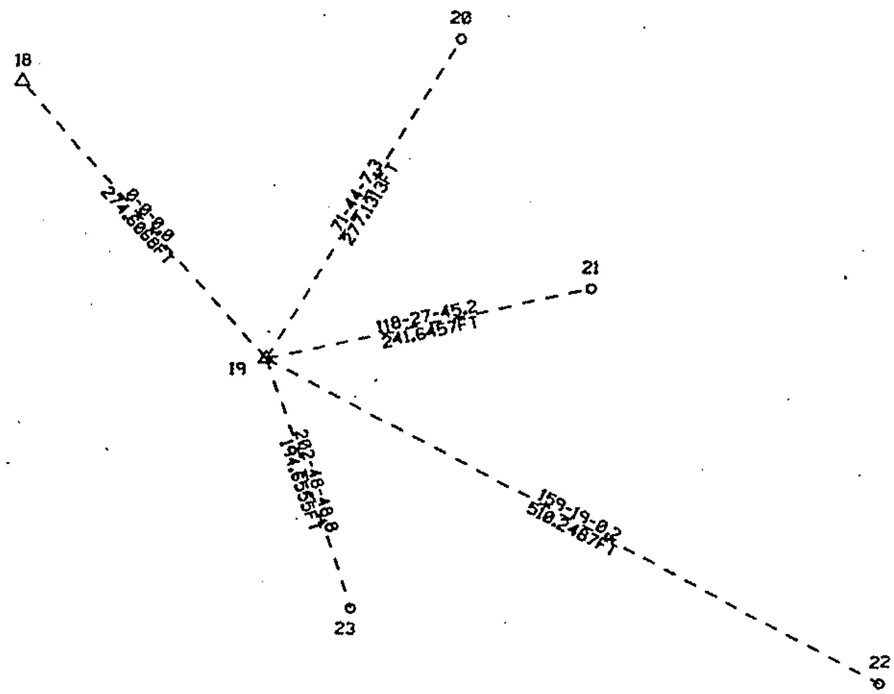



*Figure 13-13. ANGLES AND DISTANCES*

### 13.4.2 TANGENTS AND OFFSETS (Graphics Only)

The TANGENTS AND OFFSETS command annotates the baseline between the backsight point *pBS* and the occupied point *pOC* with the distances offsets. The command places offset annotation along perpendicular dashed lines and corresponding tangent annotation on the opposite side of the baseline.

TANGENTS AND OFFSETS uses the element and annotation weight and color values set for the active feature to display the linework and annotation--except line style, which is always medium dash. The baseline is placed with element weight plus one for emphasis.

Offsets are positive if they are to the right of the line extending from the occupied point toward the backsight point, and offsets are negative if to the left. Tangent distances are positive as you move along the line from the occupied point toward the backsight point. Tangent distances are negative if you are moving away from the occupied point, looking at the backsight point.

Operator Sequence

1. Select TANGENTS AND OFFSETS.

2. Identify the following points:

   pBS the known backsight point.

   pOC the known occupied point.

   desFS description of the foresight points.

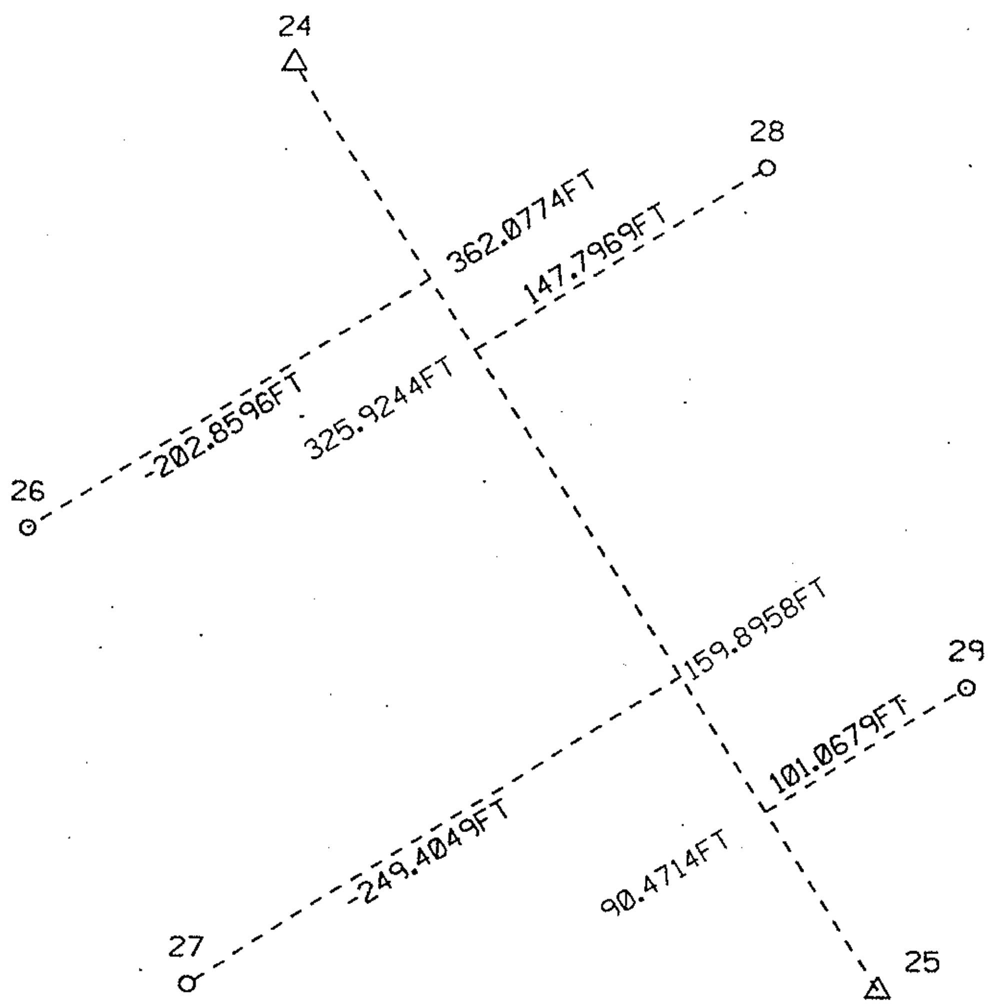



*Figure 13-14. TANGENTS AND OFFSETS*

## Additional Alpha Commands

14

## 14. ADDITIONAL ALPHA COMMANDS

In addition to the commands shown on the ICS menu, there are several other commands available to you in alpha mode. These commands can be executed in graphics if you key the *entire* command line in prefixed with a backslash (\). An alphabetical listing of these commands follows.

| *Command* | *For* |
|---|---|
| ALIGNMENT | creating a circular curve |
| CHANGE FEATURE | changing point features in the database |
| END OF JOB | updating the ICS database and completing plotting at the end of a job |
| PLOT TEXT | placing text in a design file according to feature table specifications |
| RENUMBER POINTS | renumbering specified points in the database |
| SEGMENT MINUS | calculating the area of the segment and subtracting the area from a cumulative polygon area |
| SEGMENT PLUS | calculating the area of the segment and adding the area to a cumulative polygon area |
| SET DISPLAY | turning reporting on and off (in graphics, it enables/disables reporting to the ALPHA SCROLL tutorial) |
| SPIRAL LENGTH | computing a spiral between two points |
| SPIRAL OFFSET | locating a point on a spiral given a point offset from the spiral |
| START OF JOB | setting up ICS for a new job |
| TRAVERSE DIRECTIONS | computing the directions and the lengths of each line segment defined by consecutive points in a description |

## 14.1 ALIGNMENT

The ALIGNMENT command stores parameters that define a circular curve. If some of the parameters must be calculated, use this command. If all of the curve parameters are known, use the DEFINE CURVE command instead.

The parameters required by the ALIGNMENT command include the curve identification number *cId*, a known point *pBT* on the back tangent, the known point *pPI* of intersection between two tangents to the curve, a known point *pAT* on the ahead tangent, and one of four additional values: *r*, *tan*, *sBT*, or *x*, where:

r    is the radius of the curve.

tan    is the tangent length of the curve.

sBT    is the station at point *pBT*.

x    is the fixed distance between *pBT* and *nPC*.

Note: *cId* has no significance for this command. You can enter any number from 0 to 999 to simply fill the field.

ALIGNMENT calculates coordinates of the point *nPC* where the tangent meets the curve, the center *nCC* of the curve, and the point *nPT* where the curve meets the ahead tangent. The stored curve remains in memory until replaced by a new curve defined by another ALIGNMENT command.

All curves computed by these commands are circular, and stations are expressed in decimal units. For example station 41+23.67 is entered as 4123.67.

Note: Any one of the three values *r*, *tan*, or *x* defines the curve completely. ICS uses the first of these values that is not equal to zero (0) and disregards the other values. If *r*, *tan*, and *x* are zero, the curve is compounded or reversed with the previous curve (so that *pBT*=*pPT* of previous curve).

<u>Setup</u>

Locate all tangent intersections (*pPI*), using LOCATE or another command, before using ALIGNMENT.

Operator Sequence

1. Key in ALIGNMENT.

   *The system responds:*

   ALIGNMENT cid pBT pPI pAT nPC nCC nPT r tan sBT x

2. Identify the following:

| | |
|---|---|
| cid | the curve identification number; a value between 0 and 999. |
| pBT | the ID of a known point on the back tangent. |
| pPI | the ID of the known point of intersection of the tangents. |
| pAT | the ID of a known point on the ahead tangent. |
| nPC | the ID of the point to be defined as the transition from tangent to curve. |
| nCC | the ID of the point to be defined as the center of the curve. |
| nPT | the ID of the point to be defined as the transition from curve to tangent. |
| r | the radius of curve; enter 0 if unknown. |
| tan | the tangent length of curve; enter 0 if unknown. |
| sBT | the station number at *pBT*; enter 0 if unknown. To carry stationing forward automatically, define *sBT* as -1 and set *pBT* equal to the *pPT* (point of transition for curve to tangent) of the previous curve. *sBT* is defined as the station of the previous PT; that is, x = 0. |
| x | the fixed distance from *pBT* to *nPC*; enter 0 if unknown. If *r* and *tan* are 0 and *x* is 150, the curve is computed such that *nPC* is 150 feet from *pBT*; this is usually the *pBT* of the previous curve. |

## 14.2 CHANGE FEATURE

The CHANGE FEATURE command changes a feature associated with points in the database.

Operator Sequence

1. Key in CHANGE FEATURE.

   *The system responds:*

   CHANGE FEATURE des /fnum or !fname

2. Identify the following:

   des — description of points whose feature is to be changed.

   fnum — the feature number.

   !fname — the feature name.

## 14.3 END OF JOB

The END OF JOB command must be the last command in an ICS job with one exception: if the job is the last job in a run, use the END OF RUN command. The END OF JOB command updates the ICS database, completes any plotting, sends the message ICS PROCESSING COMPLETE, and pauses. If you resume ICS after an END OF JOB command, ICS expects a control command (such as START OF JOB or OUTPUT FILE) to follow the END OF JOB command.

Operator Sequence

1. Key in END OF JOB.

   *The system responds:*

   END OF JOB

## 14.4 PLOT TEXT

The PLOT TEXT command places text at point *p*, and in direction *dir*, in a design file with the justification, color, font, size, etc. specified in the feature table.

Operator Sequence

1. Key in PLOT TEXT.

   *The system responds:*

   PLOT TEXT p dir /vdep /fnum or !fname *text

2. Identify the following:

   p — the ID of the point that indicates where to place the text.

   dir — the direction for the text. It can be expressed as an azimuth or a bearing or can be calculated using the A delimiter.

   vdep — a flag to indicate view-dependent or view-independent. If omitted or assigned 0, view-independent. 1 for view-dependent.

   fnum — the feature number.

   !fname — the feature name.

   *text — the text to be placed preceded by an asterisk. This is a required data field.

## 14.5 RENUMBER POINTS

The RENUMBER POINTS command renumbers the points in the database described by *des*. If *n* is specified, ICS will start numbering those points with *n*; otherwise ICS will start numbering those points with the first available point ID. For example

RENUMBER (1-10) 50

renumbers points 1 through 10 to 50 through 60, whereas

RENUMBER (1-10)

renumbers points 1 through 10 to next available point ID's.

Note: RENUMBER POINTS does *not* update the graphics point symbol ID's. If AUTO PLOT is ON, this command plots points with new ID's, but it does not delete existing graphics.

Operator Sequence

1. Key in RENUMBER POINTS.

   *The system responds:*

   RENUMBER POINTS des /n

2. Identify the following:

   des — description of the points to be renumbered.

   n — the ID of the first renumbered point.

## 14.6 SEGMENT MINUS

The SEGMENT MINUS command calculates the area of the segment as in the SEGMENT command. However, this command enables you to subtract segment area from a cumulative polygon area.

Remember that the SEGMENT command computes the area of a circular segment, whose boundaries are an arc (with radius *r* between *p1* and *p2*), and a chord (between *p1* and *p2*). SEGMENT MINUS subtracts the area of the segment from the previous cumulative area. The cumulative area is a running total resulting from an AREA type command and all the SEGMENT PLUS or SEGMENT MINUS commands following the AREA type command.

Operator Sequence

1. Key in SEGMENT MINUS.

   *The system responds:*

   SEGMENT MINUS p1 p2 r

2. Identify the following:

   p1 the ID of the known point that is the starting point of the arc and segment.

   p2 the ID of the known point that is the ending point of the arc and segment.

   r the radius of the arc.

   *ICS displays the chord length, arc length, segment area, and cumulative area on the screen.*

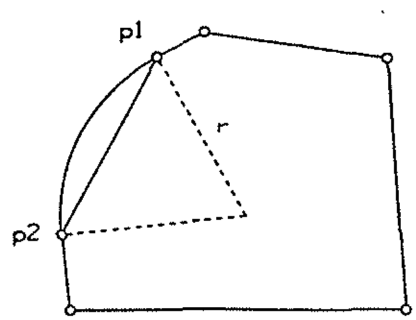



*Figure 14-1. SEGMENT MINUS*

## 14.7 SEGMENT PLUS

The SEGMENT PLUS command calculates the area of the segment as in the SEGMENT command. However, this command enables you to add segment area to a cumulative polygon area.

Remember that the SEGMENT command computes the area of a circular segment, whose boundaries are an arc (with radius *r* between *p1* and *p2*), and a chord (between *p1* and *p2*). SEGMENT PLUS adds the area of the segment to the previous cumulative area. The cumulative area is a running total resulting from an AREA type command and all the SEGMENT PLUS or SEGMENT MINUS commands following the AREA type command.

Operator Sequence

1. Key in SEGMENT PLUS.

   *The system responds:*

   SEGMENT PLUS p1 p2 r

2. Identify the following:

   p1 the ID of the known point that is the starting point of the arc and segment.

   p2 the ID of the known point that is the ending point of the arc and segment.

   r the radius of the arc.

   *ICS displays the chord length, arc length, segment area, and cumulative area on the screen.*

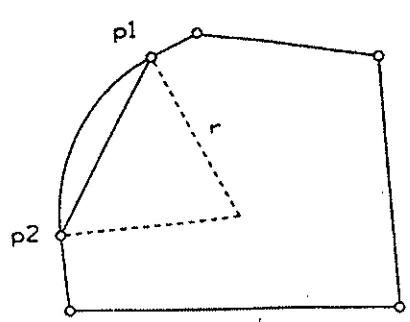



*Figure 14-2. SEGMENT PLUS*

## 14.8 SET DISPLAY

The SET DISPLAY command controls output to the ICS MEASURE/REPORT tutorial. By default, the display mode is ON when you enter the ICS graphics environment. When the display mode is ON, if you have the MEASURE/REPORT tutorial activated, ICS displays output to this tutorial. When the display is mode is OFF, ICS does not write output to the tutorial; however, it does update the current tutorial contents in memory. Turning the display mode ON after it has been OFF results in the tutorial being refreshed with the current tutorial page contents from memory. By default, display mode is ON.

Note: This command has no effect in the ICS alpha environment.

This command does not affect system output to either a report/output file or the audit trail file.

<u>Operator Sequence</u>

1. Key in SET DISPLAY.

   *The system responds:*

   SET DISPLAY mode

2. Identify the following:

   mode 0 turns display OFF; 1 turns display ON

## 14.9 SPIRAL LENGTH

The SPIRAL LENGTH command computes the spiral between point *pTS* and point *pSC* (for spiral in). For spiral out, SPIRAL LENGTH computes the spiral between point *pST* and point *pCS*. The direction of the long tangent is described by *dir*, and *sign* determines clockwise or counterclockwise curvature. SPIRAL LENGTH also calculates the coordinates of point *nSIT*, where the two tangents intersect.

Note: *cid* has no significance to this command. You can enter any number from 0 to 999 to simply fill the field.

Operator Sequence

1. Key in SPIRAL LENGTH.

   *The system responds:*

   SPIRAL LENGTH (in) cid pTS pSC nSIT dir sign

   -OR-

   SPIRAL LENGTH (out) cid pST pCS nSIT dir sign

2. Identify the following:

   cid — the curve identification number; a value between 0 and 999.

   pTS — the known point of transition from tangent to spiral (for spiral in).

   pST — the known point of transition from spiral to tangent (for spiral out).

   pSC — the known point of transition from spiral to curve (for spiral in).

   pCS — the known point of transition from curve to spiral (for spiral out).

   nSIT — the ID of the point of intersection of spiral tangents.

   dir — the direction of the long tangent at pTS or pST. It may be expressed as an azimuth or a bearing or may be calculated using the A delimiter.

   sign — 1 for a clockwise spiral; -1 for a counterclockwise spiral.

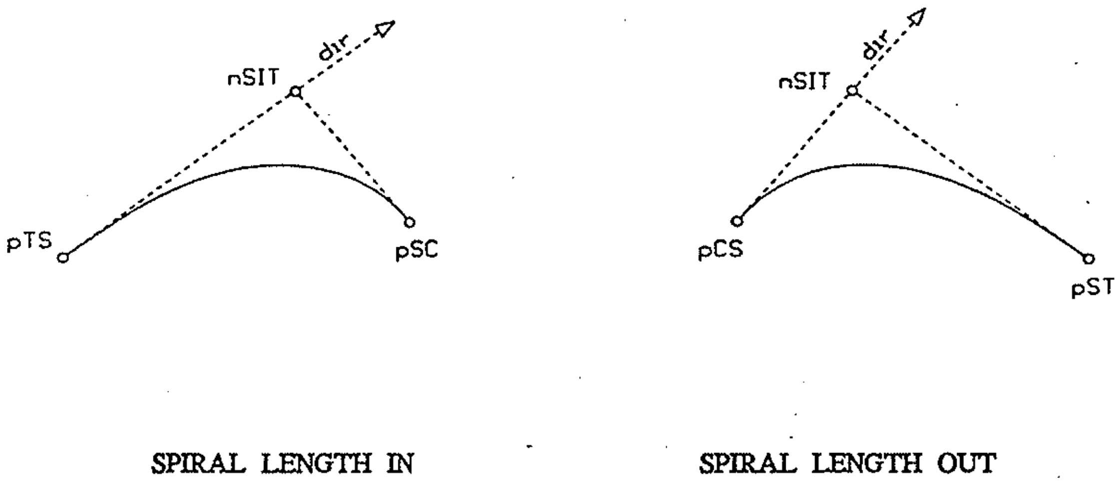



*Figure 14-3. SPIRAL LENGTH*

## 14.10 SPIRAL OFFSET

The SPIRAL OFFSET command locates point *nSP* on a spiral given point *pOF* offset from the spiral. Point *nSP* is defined by the projection of point *pOF* onto the spiral. The spiral itself must be previously defined by a SIMPLE SPIRAL or SPIRAL LENGTH command.

<u>Operator Sequence</u>

1. Key in SPIRAL OFFSET.

   *The system responds:*

   SPIRAL OFFSET nSP pOF

2. Identify the following:

   nSP    the ID of the point whose coordinates are to be located.

   pOF    the ID of the point offset from the spiral.

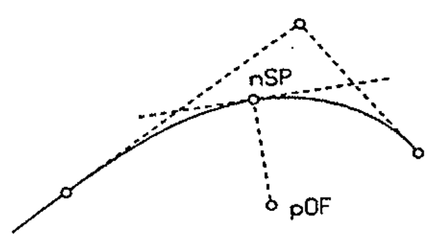



*Figure 14-4. SPIRAL OFFSET*

## 14.11 START OF JOB

The START OF JOB command must be the first command in any ICS job (with the exception of the optional OUTPUT FILE and INPUT FILE commands). The START OF JOB command instructs ICS to reset all indicators and to prepare for a new job. The command also requires you to specify a job number *j1 j2* and an ICS database *fn.dbs,* and to place headings at the beginning of each page. One useful technique is to use a job number equivalent to your UIC; that is, use [group,member] to assign *j1 j2.*

When running ICS in graphics, NEW JOB is performed automatically during initialization. Upon receiving the ACTIVE message, you can select any ICS command. Select the NEW JOB option from the matrix menu if you want to change jobs after initialization.

Note: The largest job number you can specify for a START OF JOB command is 32767.

Operator Sequence

1. Select NEW JOB or key in START OF JOB.

   *The system responds:*

   START OF JOB j1 j2 dev:[UIC]fn.dbs *heading

2. Identify the following:

| | |
|---|---|
| j1 j2 | the job number (two integers). |
| dev:[UIC] | the device and user identification code where the database is located. |
| fn.dbs | the filename of the database. |
| *heading | the comments to be printed at the top of each page of output. In the ICS environment, ICS uses the text between the asterisk and column 67 as the comments. In the graphics environment, only comments through column 40 are used as the heading. |

## 14.12 TRAVERSE DIRECTIONS

The TRAVERSE DIRECTIONS command computes the directions and distances between consecutive points in *des*. Unlike the AREA DIRECTIONS command, this command does not compute an area and does not require the first and last points in the description to be the same.

Operator Sequence

1. Key in TRAVERSE DIRECTIONS.

   *The system responds:*

   TRAVERSE DIRECTIONS des

2. Identify the following:

   des description of points.

A: Feature Table and EFEED

# Appendix A
# Feature Table and EFEED

## A. FEATURE TABLE AND EFEED

A feature table (FEATURES.TBL) and the feature table editor (EFEED) are delivered with the following packages:

- Electronic Theodolite Interface (ETI)
- Feature Coding
- Geodetic Network Analysis (GNA)
- Interactive Coordinate Geometry Subsystem (ICS)
- Map Feature Coding (MFC)
- Stereoplotter Alignment Cross-Sectioning (SAC)

The feature table stores an ordered set of map features and the graphic characteristics that define them. Using EFEED, you can edit the existing feature table or create your own.

A screen editor displays all information about a feature on the screen, one feature at a time. You can then edit the information to customize the feature characteristics.

The main components of the feature table are the files described below.

| | |
|---|---|
| Feature table file | contains the map feature characteristics. |
| Feature table editor | is an executable that performs the editing options. |
| ASCII report/list files | contain formatted ASCII output of the information in the feature table. You can create these files using a l=filename or b=filename qualifier (see Section A.2). They can be used as a reference to the contents of the feature table. |

## A.1 Feature Table

The feature table is a direct access binary file where information is stored record by record, with one feature for each record. All of the information for one feature is stored back to back in the record (that is, with no spaces between characteristics). Figure A-1 shows a delivered sample feature. The default feature table (FEATURES.TBL) is located in the ICS product directory. It is recommended that you copy this file into your default directory and make changes to the copy. Also, the file SURFEA.DGN, an IGDS design file containing a feature matrix menu cell that has been installed in ICS.CEL, is located in the product directory (see Figure A-2). The menu selections represent the standard features in FEATURES.TBL.

Feature number 0001

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Type | M | Symbol font | 090 | Cell/pnt name | | DMRS entity # | 0000 |
| Ele type | 03 | Symbol char | Z | Cell/pnt scale | 0.00000000 | Anno Line Spac | 3.5000000 |
| Ele line style | 0 | Symbol just | 08 | Digitizing num | 000 | Stream rate | 0 |
| Ele level | 04 | Symbol height | 6.00000000 | Stream delta | 1.00000000 | Stream Delta x | 000000000 |
| Ele weight | 00 | Symbol width | 6.00000000 | Stream tol | 1.00000000 | Stream Delta y | 000000000 |
| Ele color | 000 | Symbol level | 02 | Snap tolerance | 000 | Stream Delta z | 000000000 |
| Anno text font | 000 | Symbol weight | 00 | Annot flag | | IMA Stream tol | 000000000 |
| Anno text just | 12 | Symbol color | 003 | Flag 1 | | IMA Max Dist | 000000000 |
| Anno text ht | 7.00000000 | Pattern cell | | Flag 2 | | IMA Display # | 00 |
| Anno text wid | 7.00000000 | Pattern type | 0 | Flag 3 | | | |
| Anno level | 01 | Pattern scale | 0.00000000 | Misc 1 | | | |
| Anno weight | 00 | Pattern angle | 0.00000000 | Misc 2 | | | |
| Anno color | 000 | Pattern delta | 0.00000000 | 2nd feat num | 0012 | Fname # chars | 07 |

| | |
|---|---|
| Cell library filename | |
| User command filename | |
| DMRS command filename | |
| Feature name | DEFAULT |
| Secondary feature name | |

*Figure A-1. Sample Feature Table*

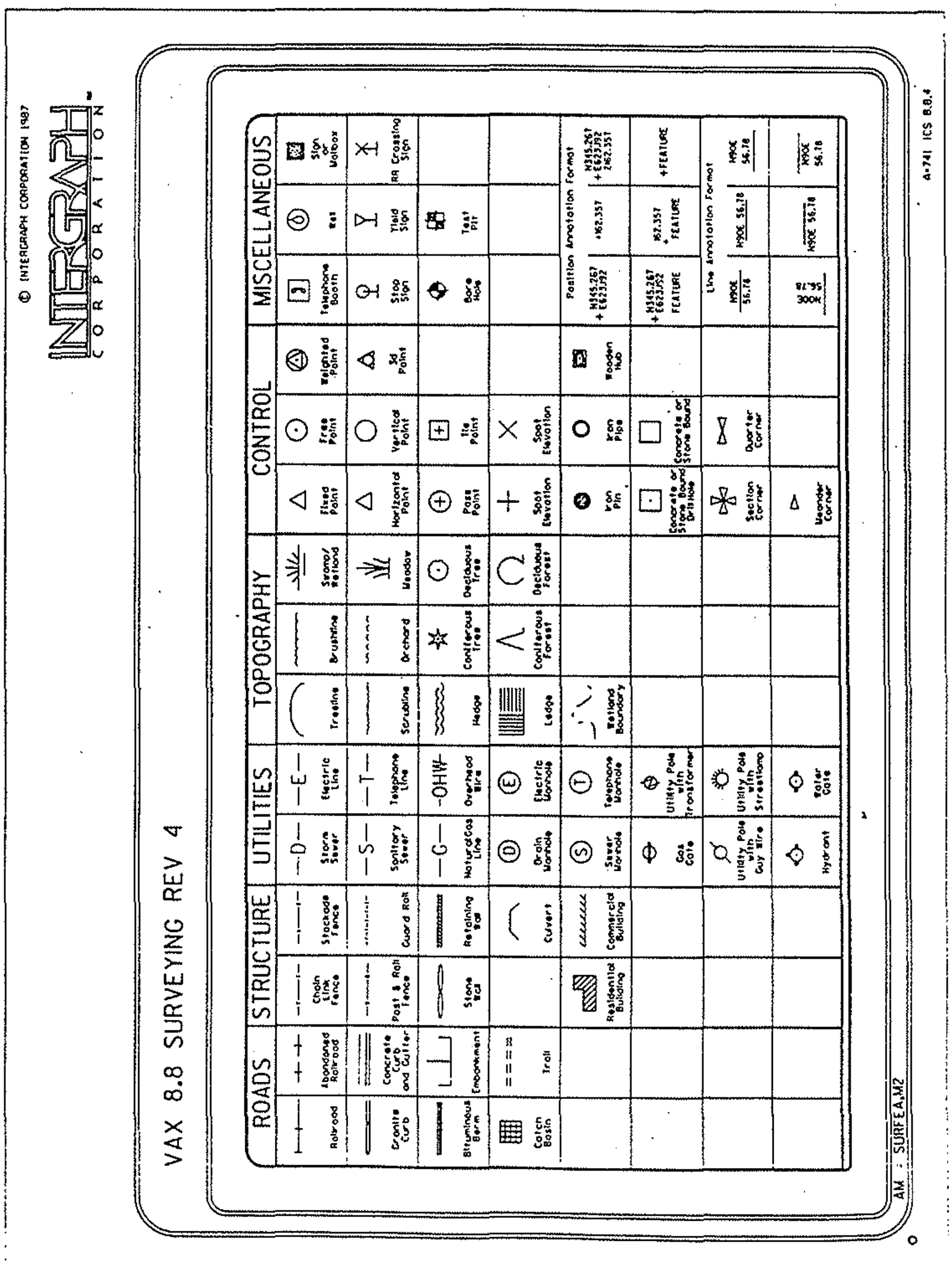



**Figure A-2. The Feature Table Menu**

## Feature Table Parameters

The parameters that are defined in the feature table to specify the characteristics of each feature are described below. Additional information on IGDS terms can be found in the *Interactive Graphics Design Software (IGDS) User's Guide.*

Table A-1 lists the range of acceptable values for feature parameters. The pages that follow this table describe each parameter in greater detail.

### General Parameters

Feature Number:

Assign a number for each feature. Valid numbers are 1 through 9999. If you key in a number that is already in use, a message alerts you to that fact.

### Element Parameters

Element Type:

This is an IGDS term. It refers to the internal format of information represented by an element. Your choices include:

| Element | Type |
|---|---|
| Cell | 2 |
| Line | 3 |
| Line String | 4 |
| Group Data | 5 |
| Simple Shape | 6 |
| Text Node | 7 |
| Curve | 11 |
| Complex String | 12 |
| Cones | 13 |
| Complex Shape | 14 |
| Ellipse | 15 |
| Arc | 16 |
| Text | 17 |
| Surface (3D) | 18 |
| Solid (3D) | 19 |

*Table A-1. Range of Acceptable Values for Feature Parameters*

| Parameter | Acceptable Range |
|---|---|
| Feature Number | 1 - 9999 |
| Type | any single alphabetic character |
| | |
| Element Type | 2 - 19 |
| Element Line Style | 0 - 7 |
| Element Level | 1 - 62 |
| Element Weight | 0 - 31 |
| Element Color | 0 - 255 |
| | |
| Annotation Text Font | 0 - 83 |
| Annotation Justification | 0 - 14 |
| Annotation Text Height | 10-character real number$^{+}$ |
| Annotation Text Width | 10-character real number$^{+}$ |
| Annotation Level | 1 - 62 |
| Annotation Weight | 0 - 31 |
| Annotation Color | 0 - 255 |
| | |
| Symbol Font | 84 - 126 |
| Symbol Character | any single character, octal value |
| Symbol Justification | 0 - 14 |
| Symbol Text Height | 10-character real number$^{+}$ |
| Symbol Text Width | 10-character real number$^{+}$ |
| Symbol Level | 1 - 62 |
| Symbol Weight | 0 - 31 |
| Symbol Color | 0 - 256 |

*expressed in working units
$^{+}$expressed in master units

*Table A-1. Range of Acceptable Values for Feature Parameters* (*Continued*)

| Parameter | Acceptable Range |
|---|---|
| Pattern Cell Name | any 6-character cell name |
| Pattern Type | 0 - 9 |
| Pattern Scale | 10-character real number* |
| Pattern Angle | 10-character real number* |
| Pattern Delta | 10-character real number* |
| Cell/Point Name | any 6-character cell name |
| Cell/Point Scale | 10-character real number* |
| Digitizing Number | 0 - 126 |
| Stream Delta | 10-character real number* ( $\geq$ .0001) |
| Stream Tolerance | 10-character real number* ( $\geq$ .0001) |
| Snap Tolerance | any single alphabetic character |
| Annotation Flag | any single alphabetic character |
| Flag 1,2,3 | any single alphabetic character |
| Misc 1,2 | 8-character string, maximum |
| Cell Library Filename | 30-character string, maximum |
| User Command Filename | 30-character string, maximum |
| DMRS Command Filename | 30-character string, maximum |
| Feature Name | 30-character string, maximum |
| No. of Annotation Characters | 0 - 30 |
| Secondary Feature Name | 30-character string, maximum |
| No. of Secondary Feature Characters | 0 - 30 |

*expressed in working units
+expressed in master units

Element Line Style:
This is an IGDS term. It refers to a style of line. Valid line style values are 0 through 7.

0=solid
1=dotted
2=medium dashed
3=long dashed
4=dash-dotted
5=short dashed
6=dash-dotted-dotted
7=long dashed-short dashed

Element Level:
This is an IGDS term. It refers to the design file level on which the feature will be placed. Valid level values are 1 through 63.

Element Weight:
This is an IGDS term. It refers to the line thickness for the feature. Valid weights are 0 to 31, with 31 being the thickest. Any value over 5 is thick.

Element Color:
This is an IGDS term. It refers to the color of the feature. Valid element color values are 0 to 255 (0 to 31 for 32's and 32C's). A color table must be set up or the IGDS default table will remain active.

Annotation Parameters

Annotation Text Font:
This refers to a set of type of a particular size and face. Fonts 0 to 83 represent potential fonts in IGDS.

Annotation Text Justification:
This refers to the relationship between the position of text and its placement point. Generally, values 0 through 14 are valid.

| | |
|---|---|
| 0 | upper left justification |
| 1 | center left justification |
| 2 | bottom left justification |
| 3 | upper left margin justification |
| 4 | center left margin justification |
| 5 | bottom left margin justification |
| 6 | upper center justification |
| 7 | center center justification |
| 8 | bottom center justification |
| 9 | upper right margin justification |
| 10 | center right margin justification |
| 11 | lower right margin justification |
| 12 | upper right justification |
| 13 | center right justification |
| 14 | lower right justification |

Note: In ICS point ID's are placed according to symbol justification. Also, see the ACTIVE JUSTIFICATION command for details on justification for ANNOTATE commands (see Section 7.3.6).

Annotation Text Height:
This refers to the height of characters or numerals in a text string that is used to label a feature. Enter the value of text height in decimal master units format.

Annotation Text Width:
This refers to the width of characters or numerals in a text string that is used to label a feature. Enter the value of text width in decimal master units format.

Annotation Level:
This refers to the design file level on which to place your text label. Valid level values are 1 to 63.

Annotation Weight:
This refers to the thickness of the lines that compose text. Valid annotation weight values are the same as line weight values. They are 0 to 31, with 31 the thickest.

Annotation Color:
This refers to the color for text. Valid annotation color values are 0 to 255 (0 to 31 for 32's and 32C's).

*The following fields are related to annotation, but they appear in the lower portion of the third column in the feature table.*

Annotation Flag:
This field is not used in ICS.

Flag 1:
This field is not used in ICS.

Flag 2:
This field indicates the content of annotation to be placed with a point when using the PLOT POINTS or LABEL COORDINATES commands. Your choices include:

| Value | For Plotting |
|---|---|
| E | elevation *only* |
| X | Northing and Easting |
| Z | Northing, Easting, *and* elevation |
| N | the feature name *only* |
| e | elevation *and* the name |
| x | Northing, Easting, *and* name |
| z | Northing, Easting, elevation, *and* name |

Flag 3:
This field indicates the content of line annotation. Your choices include:

| Value | For Plotting |
|---|---|
| S | distance *only* |
| D | direction *only* |
| blank | distance *and* direction |

Miscellaneous 1:

This field indicates how annotation is placed when it will not fit on a line/curve and what type of annotation is placed for curves.

Characters 1 and 2—

Indicate the degree of scaling allowed. Set both to zero if you want no scaling.

Character 3—

Indicates whether rotation is allowed or not. Set this character to 1 to allow rotation and 0 to disallow rotation. Additionally, if you set this character to 2, ICS disallows rotation and scaling and defaults to curve/line tables. In other words, if this character is set to 2, when annotation will not fit along a line, ICS will place line and curve data in default tables (see Section 12.2.2 for more details).

Character 4—

Indicates whether the point is 2D or 3D. Set this character to 2 for 2D and to 3 for 3D. If you leave this field blank, ICS reads Miscellaneous 1 from features created from previous versions of ICS to determine if the feature is 2D or 3D.

Characters 5 through 8—

Indicate what type of curve annotation you want. Key in the parameters in the order you want them placed. For example, if you key in 1432, you are specifying labels 1, 4, 3, 2 in that order. Labels 1 through 4 are defined in the parameter file ICS.PAR.

Miscellaneous 2:

Fields reserved for future use. Leave these fields blank.

Symbology Parameters

Symbol Font:

This is an IGDS term for a font whose characters display as user-defined symbols. For example, you may place the letter P from a specially defined symbol font to display a pond. Symbols are defined using the procedures you would use to create a text font. Valid symbol font values are 84 to 126. See Appendix D for additional information on symbol fonts and their creation.

Symbol Character:

This is an IGDS term. It refers to an alphabetic or numeric character that represents a symbol, such as P for pond, T for tree, and so forth. Valid symbol characters are any single character, represented as an octal value.

Symbol Justification:

This is an IGDS term. It refers to the relationship between the position of a symbol and its placement point. In general, valid symbol justification values are 0 to 14.

| | |
|---|---|
| 0 | upper left justification |
| 1 | center left justification |
| 2 | bottom left justification |
| 3 | upper left margin justification |
| 4 | center left margin justification |
| 5 | bottom left margin justification |
| 6 | upper center justification |
| 7 | center center justification |
| 8 | bottom center justification |
| 9 | upper right margin justification |
| 10 | center right margin justification |
| 11 | lower right margin justification |
| 12 | upper right justification |
| 13 | center right justification |
| 14 | lower right justification |

Symbol Height:
This is an IGDS term. It refers to the height of text used in a symbol. A valid symbol text height value is any 10-character real number, expressed in master units.

Symbol Width:
This is an IGDS term. It refers to the width of text used in a symbol. A valid symbol text width value is any 10-character real number, expressed in master units.

Symbol Level:
This is an IGDS term. It refers to the design file level on which a symbol is located. Valid symbol level values are 1 to 63.

Symbol Weight:
This is an IGDS term. It refers to the line thickness of a symbol. Valid symbol weight values are 0 to 31.

Symbol Color:
This is an IGDS term. It refers to the color of a symbol. Valid symbol colors are 0 to 255 (0 to 31 for 32's and 32C's).

Pattern/Cell Parameters

The pattern parameters are used when patterning with FCPELT (see Appendix C for details). The cell parameters are used with the PLOT CELL command (see Section 12.1.3).

Pattern Cell Name:
This is an IGDS term. It refers to the name given to a group of complex design features used as a pattern for shading an area in a design. Use any six-character cell name.

Pattern Type:
This is an IGDS term. It refers to any graphically distinctive and regular arrangement of markings (that is, cross-hatching, shading, and so forth). Valid pattern type values are 0 to 9.

Pattern Scale:
This is an IGDS term. It refers to the relative size of a pattern, expressed as a scale factor applied to the pattern cell during patterning. A valid pattern scale value is any 10-character real number, expressed in working units.

Pattern Angle:
This is an IGDS term. It refers to the angular orientation of a pattern. A valid pattern angle value is any 10-character real number, expressed in working units. Pattern angle can be expressed as degrees (0 to 360), decimal degrees, or degrees-minutes-seconds; the angular measurement determines whether the readout is measured clockwise from the X-axis (conventional), clockwise from the Y-axis (azimuth), or as a bearing.

Pattern Delta:
This is an IGDS term. It refers to the regular distance between columns and rows of markings that compose a pattern (that is, cross-hatching, shading, and so forth). A valid pattern delta value is any 10-character real number, expressed in working units.

Cell/Point Name:
This is an IGDS term. It refers to the name given to a cell or point. A valid cell/point name consists of up to six characters.

Cell/Point Scale:
This is an IGDS term. It refers to the scale given to a cell or point. A valid cell/point scale value is any 10-character real number, expressed in working units.

Feature Coding/Digitizing Parameters

Digitizing Number
Stream Delta
Stream Tolerance
Snap Tolerance:

None of these parameters are used in ICS.

File Name Parameters

Cell Library File Name:
This is the filename of the cell library file that contains the cell used in the feature. If it is left blank, the currently attached cell library is used.

User Command File Name:
This field is not used by ICS.

DMRS Command File Name:
Field reserved for future use. Leave this field blank.

Feature Name/Description Parameters

Feature Name:
The name of the feature. A valid feature name consists of a character string up to 30 characters long. This is the feature name that can be placed in the design file when placing points with PLOT POINTS or LABEL COORDINATES.

Secondary Feature Number:
This field indicates the design file level on which point ID's will be placed. Valid levels are 1 to 63.

Fname # chars:
The number of alphanumeric characters in the feature name that will be used as annotation. A valid response is any number up to 30. For example, if the feature name is STREAM - RAINBOW TROUT, and the number of characters is 6, the annotation text is placed as STREAM.

Secondary Feature Name:
Field reserved for future use. Leave this field blank.

## A.2 Running EFEED

EFEED must be run on a VT100 compatible terminal. You can set your graphics workstation to emulate a VT100. To learn more about the VT100 emulation capabilities of graphics workstations, from a VAX window, key in

TYPE PRO_DD_MIC:VT100.DOC

Refer to this file (on the screen of your workstation or as a hardcopy) to help you understand the alphanumeric processing capabilities of the 68000-based workstations.

Note: It is recommended that you copy the feature table, features.tbl, to another. filename before making changes. This allows you to retain the original feature table delivered with the product for reference.

The EFEED command line consists of the command and any number of valid qualifiers. The following list describes the qualifiers that are used in the EFEED command line.

<u>Command Line Qualifiers</u>

| <u>Qualifier</u> | <u>Explanation</u> |
|---|---|
| /c<br>(optional) | specifies that the feature names are to be written in the existing case. If this qualifier is not used, all feature names are converted to uppercase. All feature name searches require exact matches (including case). |
| /b or<br>/b=[filename]<br>(optional) | specifies that a brief list of the feature table including only feature names and numbers should be directed to the file specified. You can then print the file in order to use it as a reference. If the filename does not contain the .tbl extension, the software automatically appends it to the filename. If no filename is specified, the information will be printed to the screen. The listing will be in numerical order starting with the lowest feature number in the feature table. |

| | |
|---|---|
| /l or /l=[filename] (optional) | specifies that a complete list of the feature table is directed to the file specified or, if no filename is specified, to the screen. If the filename does not contain the .tbl extension, the software automatically appends it to the filename. This listing will be in numerical order starting with the lowest feature number in the feature table. |
| /m | specifies that a new features menu file will be generated. |
| /out=[filename] | copies the feature table to the file specified by the filename and makes all changes to the copy. Specifies the output feature table dataset as [filename]. |

| Parameter | Explanation |
|---|---|
| [input Feature Table dataset name] | specifies the input feature table dataset name. If you do not key in the input feature table dataset name, you will be prompted for it. |

Running EFEED

To run EFEED, set your working directory to the directory of the feature table and key in the command line using the qualifiers described above. Be sure to leave a space between EFEED, the qualifier(s), and the filename. Key in:

EFEED /[qualifier/qualifier=] [feature_table_name]

If you do not key in the extension to the feature table name, the software will automatically append .tbl.

If no filename is given in the qualifier statements /b or /l, formatted output is displayed on the screen. If a filename is given, an ASCII file is generated that can be printed and used as a reference.

When you key in the command line, the lowest number feature description from the feature table appears in formatted display.

Figure A-3 illustrates the function keypad. Table A-2 explains the function keys that allow you to move about in the feature table. You can also press [PF2] [2] twice for information on keypad function keys.

Use arrow keys to position the cursor over data to be edited; press the spacebar to clear the field and then key in changes and press <RETURN> or any arrow key. The cursor is re-positioned on the next field. Information is automatically written to the feature table when you select any keypad function, including exit.

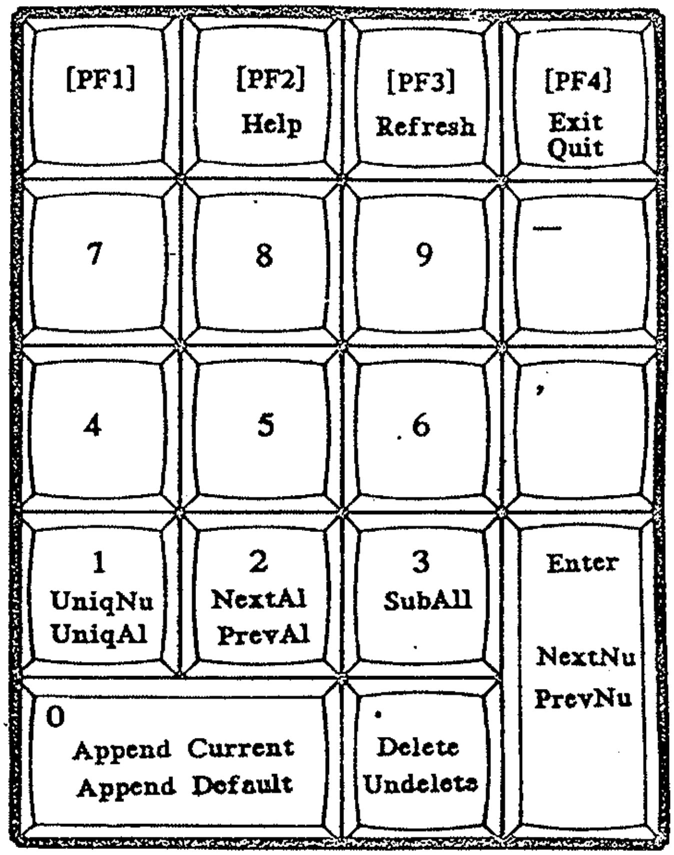
If you do not include a decimal point when entering numerical parameter values that are real numbers, and press any arrow key or <RETURN> to enter the value, the software will place the decimal for you. When you return to the feature later, the software will have automatically padded the field with zeros.

To display maximum and minimum parameter values, press [PF2]. If any value that you key in is invalid, you will be informed by a message and the default value will be returned to the field.

If at any time you try to exceed a boundary (for example, press the left arrow when you are at the left margin or move forward beyond the end of the last record in the feature table), you are alerted that you have reached a boundary.

To add a new feature to the feature table use [PF1] [0]. This appends a feature to the end of the feature table with a feature number one digit larger than the current highest feature number. This feature will appear on the screen with default values for all parameters. You can edit any of these fields including the feature number.

For example, if the last feature number is 3011, when you add a new feature its default value will be 3012. If you know that feature 3003 was deleted and want to use that number for this feature, you can change the feature number field to 3003.




Append Current copies current feature to end of file
Append Default copies default feature to end of file

Delete deletes current feature, advance to next feature by number
Undelete restores the last deleted feature

NextNu finds the next feature by feature number
PrevNu finds previous feature by feature number

UniqNu finds a unique feature when you key in a feature number
UniqAl finds a unique feature when you key in a feature name

NextAl finds the next feature by feature name
PrevAl finds the previous feature by name

SubAll replaces all, or a range of occurrences of the current field

*Figure A-3. Function Keypad*

EFEED Keypad Programmed Function Keys

When running EFEED, specific keypad functions are active. Table A-2 lists these keys and their functions.

Note: When you key in a change in a feature table parameter, pressing <RETURN> or any arrow key enters your change. If you do not press <RETURN> or an arrow key, your change will not be saved.

*Table A-2. Keypad Key Functions*

| Key | Function |
|---|---|
| [PF2] | displays a help line to indicate the range of acceptable values for the current field. |
| [PF2] [PF2] | displays a picture of the keypad. Press the key on your actual keyboard to display the help information that corresponds to that key. |
| [PF4] | exits EFEED and saves changes. |
| [PF1] [PF4] | quits EFEED without saving changes. |
| [PF3] | refreshes screen. |
| [1] | prompts for a feature number, then displays the data for the specified feature. |
| [PF1] [1] | prompts for a feature name, then displays the data for the specified feature. |
| [2] | finds the next feature (in alphabetical order) by feature name. |

*Table A-2. Keypad Key Functions (Continued)*

| Key | Function |
|---|---|
| [PF1] [2] | finds the previous feature (in alphabetical order) by feature name. |
| [3] | replaces the current field in every feature or a range of features with the data in that field. First clear the field you want to change, key in the change and press <RETURN> to enter. You must then reposition the cursor over that field. Press [3] (SUBALL) and key in a range of feature numbers at the prompt to update values in this field. This field for all features in the specified range will reflect this change and hold the same value. Instead of entering a range, you can press <RETURN> to change the value for the entire table. |
| [ENTER] | finds the next feature by feature number. |
| [PF1] [ENTER] | finds the previous feature by feature number. |
| [0] | copies the current feature to the end of the feature file. |
| [PF1] [0] | copies the default feature to the end of the feature file. You can then edit it and change any parameter(s). |
| [.] | deletes the current feature and resets screen to next feature by number. |
| [PF1] [.] | restores the last deleted feature and resets screen to that feature. |

B: Modifying ICS Menus

# Appendix B
# Modifying ICS Menus

# B. MODIFYING ICS MENUS

A feature matrix menu and screen menus are delivered in the ICS product delivery directory. This appendix lists those menus and describes how you can modify them.

## B.1 The ICS Feature Matrix Menu

The IGDS design file SURFEA.DGN contains the feature matrix menu graphics. The menu selections represent the standard features in FEATURES.TBL. They provide a quick means to set an active feature without having to remember specific feature numbers or names. When you select from the menu, an AUTO PLOT command with the appropriate feature is passed to ICS. To attach the menu from within ICS, key in:

SURFEA,Mn

-OR-

SURFEA,Sn

where *n* is 1 or 2 or 3. At a workstation, the menu is available directly from the ICS control strip.

<u>Modifying the Feature Menu</u>

To modify the feature menu, first you must enter the design file SURFEA.DGN. Each menu box on SURFEA contains, on level 63, a text string that contains the menu box location and the command issued to ICS when that box is selected. Typically the ICS command is of the form:

\ AU PL *###*

where *###* is the feature number in FEATURES.TBL for that particular feature.

To modify the feature menu, you can:

- Change a feature number:
  The feature number can be changed with the IGDS EDIT TEXT command if you want a different numbering convention in your feature table.

- Add a feature:
  To add to the menu you must run the user command MATMEN.UCM, in PRO_DD_IGDS. Refer to the *IGDS Application Software Interface User's Guide* for instructions on using MATMEN.UCM. For each new menu box you must identify the box location and add terminated text in the form:

  \ AU PL ###

  where ### is the new feature number in FEATURES.TBL.

Note: **It is imperative that the menu servicing software name (.IRB) not be modified.**

After changing or adding to the menu you must re-insert the menu cell into ICS.CEL. Make sure that the cell name remains SURFEA.

## B.2 ICS Screen Menus

The 8.8.4 release of ICS has swappable ICS and IGDS control strips, and a pop-up IGDS command menu. The delivered ICS screen menu files are in the workstation directory /usr/ip32/igds/ICS. The syntax and nomenclature of these files is straightforward:

Prefixes:
- ics__ implies an ICS pulldown menu
- igds__ implies an IGDS pulldown menu
- feat__ implies a Feature Table menu.

Suffixes:
- __CS implies a control strip
- __HMENU or __hmenu implies a hierarchical menu
- __syms implies a symbol file

For example, ics__feat__hmenu is a hierarchical ICS Feature Table pulldown menu.

The delivered ICS screen menu files are:

1. ICS__CS, the primary ICS control strip containing eight pulldown menus and a toggle for the IGDS control strip. The ICS pulldowns correspond to the ICS matrix menu command blocks, with the exception that the CONTROL block and the ENVIRONMENT block have been combined. The pulldown menu files are:

   1. ics__feat__hmenu (SURFEA: FEATURE TABLE)
   2. ics__points (POINTS/COORDINATES)
   3. ics__figures (FIGURES/ALIGNMENTS)
   4. ics__environmen (ENVIRONMENT + CONTROL)
   5. ics__annotate (ANNOTATION)
   6. ics__reports (REPORTS)
   7. ics__plot (PLOT)
   8. ics__adjust (ADJUST/TRANSFORM)

   With the exception of ics__feat__hmenu, all of the above pulldown menus use symbols from ics__syms.

2. IGDS_CS, the IGDS control strip containing all of the IGDS pulldowns found on the /CG control strip CS2, and a toggle for the ICS control strip. The pulldown menu files are:

   1. igds_create (CREATE)
   2. igds_measure (MEAS/MANIP)
   3. igds_dimension (DIMENSIONING)
   4. igds_text (TEXT)
   5. igds_dmrs (DMRS)
   6. igds_tds (TDS)
   7. igds_view_on2 (VIEW SELECTIONS)

   The hierarchical pulldown ics_feat_hmenu currently has six feature categories, requiring the following menu files:

   1. feat_roads (ROADS)
   2. feat_structure (STRUCTURE)
   3. feat_utilities (UTILITIES)
   4. feat_topograph (TOPOGRAPHY)
   5. feat_control (CONTROL)
   6. feat_miscellan (MISCELLANEOUS)

   All of these feature table files use symbols from feat_syms.

3. ICS_HMENU, the ICS hierarchical menu which is the same as the IGDS pop-up menu CS2, except that its servicing task is JRB.

Modifying ICS Screen Menus

The size, shape, color, location, hierarchy, and function of each menu as well as its components can be modified to meet your needs. Of particular interest will be modification of the Feature Table pulldown menu, the introduction of different colors, elimination of rarely used commands, and the incorporation of specialized menus.

Workstation menu modification requires the use of Menubuilder, and possibly Procasso and Snapshot. The construction of menus involves some or all of the following phases, and some may have to be performed more than once:

1. Planning
   At this stage, the structure, layout, fonts, vlts, symbol usage, button shapes and sizes, etc., are decided.

2. Symbol Creation
   Creation of symbols via Snapshot and Procasso is usually preferable to utilizing the limited graphics capabilities within Menubuilder. In addition, a symbol file provides a portable and editable approach to menu graphics. Refer to *Procasso User's Guide* and *Menubuilder User's Guide* for information on Procasso and Menubuilder.

3. **Menu File Creation**
   Each menu file is created with a specific vlt, servicing software, background color, and menu type.

4. **Testing and Modification**
   This is probably the first stage you will use because it requires only the use of Menubuilder.

Each of these phases is described in detail below.

Phase 1: Planning

Available fonts are in /usr/ip32/resrc/rfont, vlts can be found in /usr/ip32/resrc/vlt, and a blank symbol file (called menusyms) can be found in /usr/ip32/resrc/rfont. Recommendations include the use of fix7.vlt or fix14.vlt, and drake.10 and nonie.r.12 fonts. Examination of ICS and IGDS menus using the EDIT SELECT AREA command in Menubuilder will allow you to see the syntax required for select areas, and the various menu types. The ICS menus are in /usr/ip32/igds/ICS and the IGDS menus are in /usr/ip32/igds/CG.

Phase 2: Symbol Creation

If the menu buttons require icons, or if you wish to modify an existing menu, the symbols can be created using Procasso, usually in conjunction with Snapshot. If Snapshot is to be used, create the graphics for the symbols using IGDS, then take a *snapshot* of the view containing the symbol *graphics—at the desired scale.* This raster image can then be displayed in Procasso and its contents fenced and saved as individual symbols in the symbol file. Alternately, the symbols can be drawn entirely within Procasso. Vlt's can also be created and edited within Procasso.

Phase 3: Menu File Creation

The servicing sofware for each ICS menu is JRB, not FB. In strict terms (by *menu type*), the menus in /usr/ip32/igds/ICS are either hierarchical (hi) or pop-up (pu), but they will appear as one of the following four categories:

| *Menu Appearance* | *Type* | *Example* |
|---|---|---|
| pulldown | hi | ics control |
| pop-up | hi | ICS_HMENU |
| control strip | pu | ICS CS |
| hierarchical pulldown | hi | ics feat hmenu |

The contents of the select areas for these four menu categories are:

| *Menu Appearance* | *Command String* | *Type* | *Next Menu Name* |
|---|---|---|---|
| pulldown & pop-up | A,application__num<br>-OR-<br>P,primitive__num | 1<br><br>1 | |
| control strip | TM,/ICS/pd__fname,PD<br>-OR-<br>T,AM=/ICS/cs__fname,CS | 1 | |
| hierarchical pd | | 1 | ./ICS/hi__fname |

Thus, application commands or primitive commands can be selected from pulldown and pop-up menus; control strips or pulldown menus can be selected from control strips; and hierarchical menus can be selected from pulldowns. Other menu structures not found in the ICS menus can be created by combining the types of Command Strings and Next Menu Names listed above.

Phase 4: Testing and Modification

Menus can be tested by attaching them with an AM=menuname,menutype key-in during a graphics session. The menus can be modified in numerous ways, employing the methods found in Phases 1 throu 3 above. For example, a *streamlined* IGDS pop-up menu that contained the commands used most frequently during an ICS run could be made. Another example is moving commands in the ENVIRONMENT block to the block(s) they affect, such as moving TRAVERSING MODE to the LOCATE menu block.

C: ICS Utilities

# Appendix C
# ICS Utilities

# C. ICS UTILITIES

The following sections describe the utilities: I2SDB, ICS Customer Support Library, ICS Database to DTM Points File, Database Upgrade Utility, the Linkage User Command, and Feature Code Patterning by Element List.

## C.1 I2SDB Utility

The utility program I2SDB facilitates the population of a survey database from an IGDS design file. It converts node information contained in IGDS graphic elements to point coordinate information. Snapped nodes are purged; as a result, there is a one-to-one correspondence to each unique coordinate value in the design file and database. The utility is useful for loading approximate coordinates for GNA (A*797) or design coordinates for ICS (A*741). The graphic element types and nodal information extracted are:

| *Element Type* | *Extracted Information* |
|---|---|
| cell | cell origin |
| line | endpoints |
| line string | vertices |
| shape | vertices |
| curve | vertices |
| conic | vertices |
| ellipse | origin |
| arc | endpoints and origin |
| text | origin |
| point string | point coordinates |

* Only 1-character text elements (symbols) are considered eligible.

<u>Operation</u>

To run I2SDB, key in RUN PRO_DD_ICS:I2SDB. You will be prompted for a design filename. All other inputs are derived from the header of this file, including destination database, working units, and level mask. Elements that are displayed in IGDS view 1 are considered eligible. Point ID's in the survey database are assigned on a minimum-next-available basis.

## C.2 ICS Customer Support Library

### C.2.1 ICSCSL

The ICSCSL is an object library much like the CSL; it provides you with tools for customizing alternate user interfaces and applications based on ICS. Routines to initialize, communicate with, and terminate the ICS image from within a user program are included in ICSCSL (see Section C.2.2). Use of this CSL requires a license for the C Runtime Library.

Synopsis The ICSCSL consists of the following files:

o ICSCSL.OLB, which contains an number of interfering routines to the ICS.EXE task;

o ICSRDEF.INC, an include file suitable for C programs which contains return code definitions from the ICS task;

o ICSCSL.OPT, a VAX/VMS linker options file required by all programs linking with the ICSCSL;

o ICSCOM.UCM, ICSCOM.EXE, and ICSCOM.C, which serve as an example of the use of the ICSCSL in a traditional user command/task interface scenario;

o ICSCSL.DOC, this library's documentation.

Features ICSCSL allows the full set of ICS commands as documented in the this user's guide to be accessed by the user. The implementation of this interface is to allow programs to send commands to the ICS task much as an alphanumeric user would, and to return to the user the records normally sent to the terminal, with an appropriate status code. Return codes allow a moderate amount of control over the behavior of the ICS task by the user. Care should be taken when reformating return records for further processing and numerical calculations, as the format and order of output records is not guaranteed from release to release. Alternatively, users may access ICS data by using various levels of DMRS interfaces available in that product.

Bugs It is the responsibility of the user to insure that ICS executes properly. This requires that the user check for and remedy such things as errors encountered during processing, and insuring that valid databases and design files exist, as well as valid feature codes, prompts, and error messages.

Comments ICSCSL is an enhancement to, and a replacement for, the 8.8.2 custom ICS user commands (formerly in Appendix G of the 8.8.2 document).

### C.2.2 ICSCSL Routines

| | | | |
|---|---|---|---|
| Name | long ICS_start(graphics, ICS_chan) | | |
| Purpose | Initiate the ICS task, map communication buffers, and associate event flag clusters. | | |
| Synopsis | o graphics | 1 byte | 0 parent isn't an FB subprocess; 1 parent is an FB subprocess; passed by value. |
| | o ICS_chan | 2 bytes | VMS channel number of ICS termination mailbox; passed by reference. |
| On Exit | 1: | success | |
| | 2: | error associating event flag cluster | |
| | 4: | error creating termination mailbox | |
| | 6: | error creating ICS subprocess | |
| | 8: | error mapping global section | |
| Bugs | Graphics == 0 does not work | | |

| | | | |
|---|---|---|---|
| Name | long ICS_comm(RW, ICS_msg) | | |
| Purpose | Send or receive a buffer from ICS. The RW variable defines the operation. | | |
| Synopsis | o RW | 1 byte | 0 write;<br>1 read;<br>passed by value. |
| | o ICS_msg | n bytes<br>n >= 82 | ICS command (write);<br>or ICS return record (read)<br>passed by reference. |
| On Exit | defined in ICSRDEF.INC | | |

| | |
|---|---|
| Name | void ICS_setef() |
| Purpose | Set the appropriate event flag within the communication routines to synchronize reads from the ICS task. |
| Synopsis | It is the user's responsibility to call this routine between successive reads from the ICS task. |

## C.3 ICS Database to DTM Points File

There are two methods available for loading Intergraph DTM point files from an ICS project. One method involves running ICSTODTM on an alphanumeric terminal, and it is typically used to transfer a whole block of points defined by a range. The other method is more versatile, and it involves bringing up the design file in graphics and running the DTM DGNXYZ processor (this method is recommended in most cases).

### C.3.1 Running ICSTODTM

To create a DTM point file (.XYZ) from an ICS survey database on an alphanumeric terminal, key in the following, then press <RETURN>:

RUN PRO_DD_ICS: ICSTODTM

ICS responds with a series of screen messages in the following operator sequence:

Operator Sequence for ICSTODTM

1. **Enter database name (dev:[uic]name.DBS)**

   Key in the database filename that stores ICS coordinates, then press <RETURN>. Go to Step 2.

   For example, key in QS3:[227,12]TALKEETNA.DBS.

2. **Enter beginning point id (<RETURN> for all)**

   Key in the first point ID number, then press <RETURN>. Go to Step 4.

   -OR-

   Press <RETURN> to select all point ID's. Go to Step 4.

3. **Enter ending point id**

   Key in the last point ID number, then press <RETURN>. Go to Step 4.

4. **Enter DTM point (.XYZ) filename**

   Key in the DTM filename, then press <RETURN>. Go to Step 5.

   For example, key in QS3:[227,12]TALKEETNA.XYZ.

5. **Enter elevation multiplier (<RETURN> for 100)**

   Key in a value that represents an elevation multiplier, or vertical exaggeration factor, then press <RETURN>. Go to Step 6.

   -OR-

   Press <RETURN> for the default vertical exaggeration of 100. Go to Step 6.

6. **Enter design filename for working unit definition >**

   Key in the name of your working design file, then press <RETURN>.

   *ICSTODTM creates a DTM point file.*

ICSTODTM Error Messages

The screen can display several error messages if an error occurs during ICSTODTM processing. Possible error messages are:

DTM FILE ERROR; STATUS = ...
DATABASE NOT FOUND
ERROR OPENING ICS DATABASE
ERROR ACCESSING ICS HEADER
DMRS ERROR NUMBER ...
EXITING ICSTODTM
NO POINTS FOUND IN ICS DATABASE
ERROR OPENING DESIGN FILE
ERROR OPENING DESIGN FILE -- WRITE LOCKED
ERROR ACCESSING POSITIONAL UNITS IN DESIGN FILE
ERROR CLOSING DESIGN FILE

---

## C.3.2 Running DGNXYZ

To run DGNXYZ, on a graphics terminal, key in the command:

UC=PRO_DD_DTM:DGNXYZ

Refer to the *Digital Terrain Modeling (DTM) User's Guide* for more information.

## C.4 8.8.3 Database Upgrade Utility

The Database Upgrade Utility is SDB_82TO83.COM. ICS 8.8.3 uses an upgraded database schema for compatibility with the Stereo Alignment Cross-Sectioning (SAC) software. 8.8.3 and newer databases have additional entities for SAC cross-sections, SAC shots, SAC skew shots, and ICS station equations.

In order to upgrade 8.8.2 ICS databases to the 8.8.3 format, you must invoke the command procedure PRO_DD_ICS:SDB_82TO83.COM as follows:

Operator Sequence

1. Key in: @PRO_DD_ICS:SDB_82TO83.COM

2. **Database filename:**

   Key in the name of the database to be upgraded, and press <RETURN>. If not specified, the command procedure defaults the database device and UIC to the current directory and the database schema file extension to .dbs.

3. **Number of x-sections or parameter filename:**

   Key in the approximate number of SAC cross-sections you initially intend to create. It is best to estimate relatively closely the number of cross-sections. However, underestimation is not critical because of the the "auto extend" file space reallocation feature in ICS.

   Alternatively, a parameter file, PRO_DD_ICS:SURVEYDB.PAR, may be edited and submitted to SDB_82TO83.COM at this prompt. SURVEYDB.PAR is essentially a list of occurrence numbers for each entity to be inserted in the ICS database. SURVEYDB.PAR is edited rather straight-forwardly through the EDT editor.

4. **Number of shots:**

   Key in the approximate total number of ordinary SAC cross-section shots. Auto-extend will compensate if the number of shots is underestimated.

5. **Number of skew shots:**

   Key in the approximate total number of SAC skew shots. Again, auto-extend will compensate upon underestimation of the total number of skew shots.

6. Number of station equations:

   Key an estimate of the total number of station equations that will be utilized for the job. A new occurrence is created for each individual alignment where a station equation is defined. Auto-extend will compensate for underestimation.

   It should be noted that SURVEYDB.COM, which creates 8.8.3 databases from scratch, defaults the number of stations equations. Thus, the station equation prompt is never seen when running SURVEYDB.COM.

Notes:

User input is complete after Step 5. Numerous informational messages will be spooled to the screen as the old database files are upgraded.

Entity files (.ent extension) that were present in the 8.8.2 database are not recreated, merely modified. Entity files that are unique to 8.8.3 are created according to the size specified in steps 2 through 5.

All other database files, with the exception of the compilation listing, are recreated. These files are not purged; therefore, you will see duplicate <dbname>.att, <dbname>.dbs, <dbname>.ddl, <dbname>.par, <dbname>.sym, and <dbname>.win files in the database directory. A new compilation listing with the extension .lis is created for 8.8.3 upgrades.

Following is an example 8.8.3 ICS database upgrade using DB_82TO83.COM.

```
$ dir texas*

Directory ZG0:[110020]

TEXAS.ATT;1           TEXAS.DBS;1           TEXAS.DDL;1
TEXAS.LST;1           TEXAS.PAR;1           TEXAS.SYM;1
TEXAS.WIN;1           TEXASCIRC__CURV.ENT;  TEXASCOORD__DIFF.ENT;1
TEXASFIG__HEAD.ENT;1  TEXASFIG__POINT.ENT;  TEXASHEADER.ENT;1
TEXASLINE.ENT;1       TEXASOBS.ENT;1        TEXASPNT.ENT;1
TEXASREMAP.ENT;1      TEXASSPIRAL.ENT;1     TEXASVERTICAL.ENT;1

Total of 18 files.
$ @pro__dd__ics:sdb__82to83

                    ETI / GNA / ICS
                      8.2 -> 8.3
                Database Upgrade Utility

$__Database filename                       : texas
$__Number of X-sections or Parameter filename : 1
$__Number of shots                           : 1
$__Number of skew-shots                      : 1
$__Number of station equations               : 1

%SUR-I-CIDDL, creating ddl image file for  ZG0:[110020]TEXAS.dbs
TYPE "/HELP" FOR ASSISTANCE

%DELETE-I-FILDEL, ZG0:[110020]TEXAS.COM;1 deleted (1 block)
%SUR-I-UPDDL, updating ddl image file for  ZG0:[110020]TEXAS.dbs
%SUR-I-CUDDL, compiling updated ddl image file for
ZG0:[110020]TEXAS.dbs
    DDL STOP
%SUR-I-WUPG, writing database upgrade files for
ZG0:[110020]TEXAS.dbs %SUR-I-EUPG, executing database upgrade
file ZG0:[110020]TEXAS.UPG
TYPE "/HELP" FOR ASSISTANCE

%DELETE-I-FILDEL, ZG0:[110020]TEXAS.UPG;1 deleted (1 block)
%DELETE-I-FILDEL, ZG0:[110020]TEXAS60.COM;1 deleted (1 block)
%DELETE-I-FILDEL, ZG0:[110020]TEXAS62.COM;1 deleted (1 block)
%DELETE-I-FILDEL, ZG0:[110020]TEXAS63.COM;1 deleted (1 block)
%DELETE-I-FILDEL, ZG0:[110020]TEXAS64.COM;1 deleted (1 block)
%DELETE-I-FILDEL, ZG0:[110020]TEXAS65.COM;1 deleted (1 block)
%DELETE-I-TOTAL, 5 files deleted (5 blocks)
%SUR-I-NEWDDL, use dmrs SCHDDL to generate new DDL file
%SUR-S-SUCC, successful completion
```

```
$ dir texas*

Directory ZG0:[110020]

TEXAS.ATT;2              TEXAS.ATT;1              TEXAS.DBS;2
TEXAS.DBS;1              TEXAS.DDL;3              TEXAS.DDL;2
TEXAS.DDL;1              TEXAS.LIS;1              TEXAS.LST;1
TEXAS.PAR;2              TEXAS.PAR;1              TEXAS.SYM;2
TEXAS.SYM;1              TEXAS.WIN;2              TEXAS.WIN;1
TEXASCIRC__CURV.ENT;1                             TEXASCOORD__DIFF.ENT;1
TEXASFIG__HEAD.ENT;1  TEXASFIG__POINT.ENT;1
TEXASHEADER.ENT;1        TEXASLINE.ENT;1          TEXASOBS.ENT;1
TEXASPNT.ENT;1           TEXASREMAP.ENT;1         TEXASSHT.ENT;1
TEXASSKW.ENT;1           TEXASSPIRAL.ENT;1        TEXASSTA__EQUA.ENT;1
TEXASSTA__HEAD.ENT;1  TEXASVERTICAL.ENT;1  TEXASXSE.ENT;1

Total of 31 files.
```

## C.5 The Linkage User Command: LINKAGE.UCM

The ICS 8.8.4 version provides a user command, LINKAGE.UCM, that allows you to review and/or set the design file TCB variable, LKGNMD. This variable controls how IGDS operations work on graphics elements that possess DMRS database linkages. The most significant effect of this variable for ICS users occurs when you use IGDS to COPY or to FENCE COPY graphics elements produced by ICS.

The LKGNMD variable has the following possible linkage generation modes.

- NEW LINKAGE MODE LKGNMD=0
  In this case a *new* entity occurrence is created in the ICS database for the copied element. Attribute values are copied from the existing element. See warning below.

- INFORMATION LINKAGE MODE LKGNMD=1
  In this case no new entity occurrence is created in the ICS database but the copied element will have a DMRS informational linkage that is identical to the existing element linkage.

- DUPLICATE LINKAGE MODE LKGNMD=2
  In this case a *duplicate* entity occurrence is created in the ICS database for the copied element. Attribute elements are copied from the existing element. See warning below.

- NULL LINKAGE MODE LKGNMD=-1
  In this case the copied graphics element contains no database linkage so there is no effect on the ICS database.

WARNING: **DUPLICATE linkage mode can result in ICS database corruption. The preferred mode is NULL (LKGNMD=-1) for ICS users. This permits you to copy ICS graphics elements for display purposes without affecting the ICS database.**

Operator Sequence

1. To invoke the user command key in:

   UC=PRO_DD_ICS:LINKAGE.UCM

2. At the prompt key in N, I, D, or X to set LKGNMD to either NEW, INFORMATIONAL, DUPLICATE, or NULL mode, respectively

   -OR-

   Press <R> to review the current linkage mode setting.

## C.6 Feature Code Patterning by Element List

Feature Code Patterning by Element List (FCPELT) is a batch program that performs essentially the same operation as Delayed Patterning at the workstation. FCPELT sets up the type 5 patterning elements so that the Pattern Driver (PDV) task can then be run to pattern the elements in the design file. In combination with a feature table, FCPELT can be used to pattern both linear and area figures created using ICS.

Unlike patterning at the workstation, FCPELT does not require you to identify with a data point each element to be patterned. Instead, you indicate the elements by assigning feature codes and creating a list file. FCPELT takes as input this universe list file (created at the workstation with the "EL=" command), a feature table describing the elements involved, and the design file itself.

The list file contains pointers to each element to be patterned. These pointers are used to access the element in the design file and retrieve its feature code. The feature code is then used as an index into an entry in the feature table, which contains patterning information for all elements *with the same feature code.* One type 5 patterning element is written to the design file for every unique feature code encountered. Therefore, FCPELT is very useful and efficient if a large number of elements are to be patterned the same way.

### C.6.1 Setup Notes: What You Need To Know Before Running FCPELT

<u>The Design File</u>

FCPELT will pattern elements in either a 2D or a 3D file, with either a 2D or 3D pattern cell. However, patterning is strictly 2D in nature, so the z coordinates of 3D cells are ignored. Patterning takes place in the plane of a 3D element.

Note: Currently, the underbar character is *not* allowed in the design filename. Using an underbar in the file name will cause a fatal error in FCPELT.

<u>The Feature Table</u>

There should be one feature for each distinct feature code among the elements in the list file. Feature table items pertinent to FCPELT are: Feature Code, Pattern Cell, Pattern Type, Pattern Scale, Pattern Angle, Pattern Delta, and Cell Library. Each of these is described below.

FEATURE CODE

Make sure that the feature code for each feature IS THE SAME AS the feature code of the element(s) to be patterned. ALL elements with this feature code will be patterned according to the specifications in this feature.

PATTERN CELL

The name of the pattern to be used, up to 6 characters long.

PATTERN TYPE

The pattern type is extremely important! There is only one way to specify area patterning, but there are four ways to specify linear patterning:

| *Pattern Type* | *Value* |
|---|---|
| area | 1 |
| unscaled linear | 0 |
| scaled linear | 2 |
| multiple segment (linear) | 4 |
| single segment (linear) | 5 |

PATTERN SCALE

Pattern scale describes how much the pattern is to be scaled before placement. If scale=2, the pattern will be twice its normal size.

PATTERN ANGLE

This number is only used when doing area patterning, and specifies the angle at which the pattern will be placed inside the area. Enter the number in degrees.

PATTERN DELTA

Pattern delta, also used only for area patterning, describes the row and column spacing between the pattern cells placed. Enter the number in master units.

CELL LIBRARY

The cell library name. Specify DEV:[UIC]FILE.EXT (unless the library is in the default directory, in which case you may leave off the device and UIC). All the pattern cells used for one element list need not be in a single cell library. No library need be attached to the design file, and you may specify as many libraries as needed in the feature table.

Note: If NO cell library is specified in the feature table, the library attached to the design file will be used.

Eligible Elements

The following elements are the only ones that FCPELT (as well as PDV) considers patternable:

| *Type* | *Element* |
|---|---|
| 3 | line |
| 4 | line string |
| 6 | shape |
| *11 | curve |
| 12 | complex string |
| 14 | complex shape |
| 15 | ellipse/circular ellipse |
| 16 | arc |

(*Curves may only be patterned using scaled or unscaled linear pattern types.)

All elements must be assigned a feature code and a corresponding entry in the feature table.

Note: For error-free processing, do not give linear elements the same feature code as areas (such as shapes or ellipses). Since only one patterning element is generated for each feature code, some elements would contradict the pattern type specification in the feature table. Linear elements cannot be patterned as areas and would not be changed during PDV processing, although an error message would be produced.

An area can be patterned as a linear element, and would be in this case unless some action were taken before running PDV.

To determine if any type contradictions exist, first run FCPELT in CHECK-ONLY mode.

Running FCPELT in CHECK-ONLY Mode

A very useful feature of FCPELT is the CHECK-ONLY mode. If this mode is specified in the command line, FCPELT will run normally except that it will *not* write any type 5 elements to the design file. In this way, the setup of the feature codes can be checked for errors without actually tampering with the design file.

It is recommended that FCPELT always be run in CHECK-ONLY mode first, to save time and trouble if there are errors. Whatever the mode, after processing, FCPELT displays the number of elements in the list file, and the number of type 5 elements actually written.

## C.6.2 Patterning ICS Figures with FCPELT

After setting up a feature table and running ICS on the design file (see Section C.6.1), three major operations follow:

1. In the graphics environment, you specify the elements to be patterned by creating a universe list file with the "EL=" command.

2. In the alpha environment, you run the program FCPELT.

3. In the alpha environment, you run the IGDS pattern task program PDV.EXE.

<u>Creating the List File</u>

When all the elements to be patterned have been added to the design file, create a universe list file of those elements. To do this, make sure that *only* the desired levels are turned on. (Turn off any levels containing elements that are not to be patterned.) Set the appropriate locks, and place a fence around (or on, depending on the locks) the desired elements. Next, key in

EL=filename

where *filename* is the name of the list file to be created.

Note: *Not all* elements to be patterned must be in the list file. It is enough to have one element from each distinct feature code in the list. Then, elements with the same feature code that were not actually included in the list file will still be patterned.

If some elements have associated text on the same level, the text will not be affected if it is included in the element list file. Text and some other types of elements cannot be patterned (see Section C.6.1), and will simply be ignored by the patterning process.

In addition, elements with no feature code are not patterned. Elements that are not to be patterned can be assigned a zero (0) feature code and included in the list file without error.

Be careful not to compress the file before you run FCPELT, as this could easily upset the block and byte pointers just established in the list file. This would cause FCPELT to access the design file incorrectly, and would result in a cascade of errors.

Running FCPELT

1. Key in:

   SET COMMAND PRO_DD_ICS:FCPELT.CLD

2. Key in:

   FCPELT

   for interactive prompting. You are prompted for the design filename (.DGN), universe list filename (.LIS), feature table file (.TBL), and check/no check option.

   -OR-

   You can key in the full command line:

   FCPELT design_file list_file feature_table [check/nocheck]

Note: Intergraph recommends running FCPELT first in CHECK-ONLY mode (see Section C.6.1) to make sure you've made no setup errors.

Remember to specify device and UIC for the file and feature table names, if they are not in the current device and directory. The extension on the feature table name always defaults to .TBL, that of the design file defaults to .DGN, and that of the list file to .LST.

If you want to run FCPELT in CHECK-ONLY mode, enter *Y*, *YES*, or *CHECK* in the last field. If you DO NOT want CHECK-ONLY mode, enter *N*, *NO*, or *NOCHECK*. The key-ins can be upper or lowercase. The default is NOCHECK.

Running the Pattern Driver Task

1. Key in:

   RUN PRO_DD_IGDS:PDV

2. At the PDV prompt key in the design file name.

### C.6.3 FCPELT Error Messages

As FCPELT runs, the block, byte, and feature code of each element are displayed on the screen. (Block and byte numbers start with 1, not 0.) Messages may also be displayed. Some of the possible messages are explained below.

"%FCPELT_DR-I-NOPTY, Element type ?? cannot be patterned"
Some element types, such as text, are not eligible for patterning. If one is encountered, this statement is printed, no type 5 is written, and processing continues.

"%FCPELT_DR-I-ZOGRG, Zero feature code - element will not be patterned"
Since there is no zero feature code in a feature table, any elements with no feature code assigned will not be patterned. Processing is not interrupted.

"%FCPELT_READFEA-W-RDFETB, ERROR ?? READING FEATURE TABLE"
This error typically indicates that the feature code requested could not be found. Make sure that there is one feature for every feature code. The current element is skipped, and processing continues.

"%FCPELT_DR-W-LNASAR - LINEAR ELEMENT CANNOT BE PATTERNED AS AREA"
This error results from mixing different element types under the same feature code. If the first element found with a particular feature code is an area, the type 5 element will be written for area patterning. If a linear element with the same code is then found, the type 5 also applies to it, but a linear element cannot be patterned as an area. This will cause an error when running PDV, and the linear element will not be patterned or changed at all. This will not interfere with the processing of any other elements.

"%FCPELT_DR-W-ARASLN - AREA AT THIS BLOCK/BYTE WILL BE PATTERNED AS LINEAR UNLESS FEATURE CODE IS CHANGED"
This is the same sort of error as above, except in reverse. However, an area CAN be patterned as linear (see examples), so this condition will NOT cause an error in PDV. If you want the area to be patterned along its border instead of inside it, nothing needs to be done. But if not, your *must* change the element's feature code or take some other action before running PDV and ruining the element. As before, this problem will not interfere with the processing of any other elements.

Each time FCPELT is run, a file "FCPELT ERR.MSG" is created. Each time one of the last two errors above is encountered, the error message and the element's feature code, type, and block and byte pointers are written to the file. If there are errors, you should consult the file and correct the problems (if necessary). FCPELT displays the number of errors after processing all the elements.

D: How to Create Symbols

# Appendix D
# How to Create Symbols

# D. HOW TO CREATE SYMBOLS

Symbols are specially created cells in a cell library which has been inserted into a font library. The following steps can be used to create a font library with symbol characters inserted.

1. Create a 2D cell file from the UTILITIES environment.

    UTILITIES: CRE File.CEL 2C 50

2. Create a 2D design file for the symbols.

    UTILITIES: CRE File.DGN 2D 50

3. Set up design file.

    a. Datapoint DESIGN OPTIONS to set working units in the design file.

    | | |
    |---|---|
    | - | MU |
    | 1 | SU |
    | 254 | PU |

    Other working units can be used. See the *Font Utilities User's Guide.*

    b. Key in LV=63.

    c. Datapoint the Grid display on. (You may have to zoom in or out to see the grid display.)

    d. Datapoint Delays off.

    e. Datapoint the Grid lock on.

    f. Key in NC=File.CEL to initialize a new cell file or RC=File.CEL if the cell file has been used previously.

    g. Use PLACE BLOCK to place square shapes (WT=0,LC=0) on level 63 by giving a datapoint at two diagonally opposite grid nodes of the grid. The squares have to be 254 PU by 254 PU. Copy a square for each symbol to be created. A symbol font cell file can have 256 symbols. Up to 127 fonts (0 through 126) can be inserted into the font library.

    h. Change active level to any number from 1 to 62.

    i. Datapoint the Grid lock off.

5. Create a cell within a square.

   a. Use lines, line strings, shapes, arcs or circles to create the desired point feature. The area within a shap will be displayed as filled. The graphics should not extend beyond the square and should be on any level from 1 to 62.

   b. Place a fence around the square.

   c. Symbol cell names are a single alphanumeric character or an octal or decimal number as shown below. Create the cell by using an acceptable cell name. Key in CC="cell name". An acceptable name is one of the following:

   *Cell Name*

   | | |
   |---|---|
   | 0-9 | Keyboard number 0 through 9 |
   | A-Z | Keyboard uppercase A through Z |
   | .A-.Z | Keyboard lowercase a through z |
   | OCTnnn | 'nnn' can be from 0 to 377 with no 8's or 9's in the number |
   | DECnnn | 'nnn' can be from 0 to 255 |
   | % | PRCENT |

   d. Select the DEFINE ORIGIN command from the IGDS menu and give a datapoint *near* one of the nine acceptable cell origin locations: one of the four corners of the square shape, one of the four midpoints between the four corners, or the center of the square.

6. The cell file can then be inserted into a font library. The cell file can be inserted into an existing font library such as PRO_DD_IGDS:FONTLIB.; or a font library can be created. To create a font library key in the following.

   FLIB/CREATE File.FLB

   To insert the cell file into the font library, key in:

   FLIB/INSERT/NUMBER=n/CLASS=SYMBOL File.FLB File.CEL

   where *n* is a number from 0 to 126. (The fast symbol font 127 will be inserted automatically when the font library is created.)

   If the cell file is to be inserted into PRO_DD_IGDS:FONTLIB.;, no one can be in graphics.

To list fonts in a font library, key in:

FLIB/LIST File.FLB

To replace existing fonts, key in:

FLIB/REPLACE/FONT=n File.FLB File.CEL

To delete fonts from the font library, key in:

FLIB/DELETE/FONT=n File.FLB

The cell file can be saved for future symbol additions or for insertion into font libraries on other systems. If a design file containing the symbols is to be displayed on another system, it is necessary to have the correct symbol font in the font library.

7. Place the symbols in the feature table using the ICS feature table editor EFEED. See Appendix A.

8. To attach the font library to a design file, from the GRAPHICS prompt key in:

GRAPHICS: File.DGN/FONT=File.FLB

It is not necessary to specify the FONT qualifier, if the symbol font was inserted into the default font library PRO_DD_IGDS:FONTLIB.;.

To list the fonts which have been downloaded in graphics, key in:

DF=$

Datapoint SLOW FONT DISPLAY on to display the symbols correctly.

Note: If you want to run the ICS input file from thealpha environment, it is necessary to attach the font library first as described above.

E: Sample Project

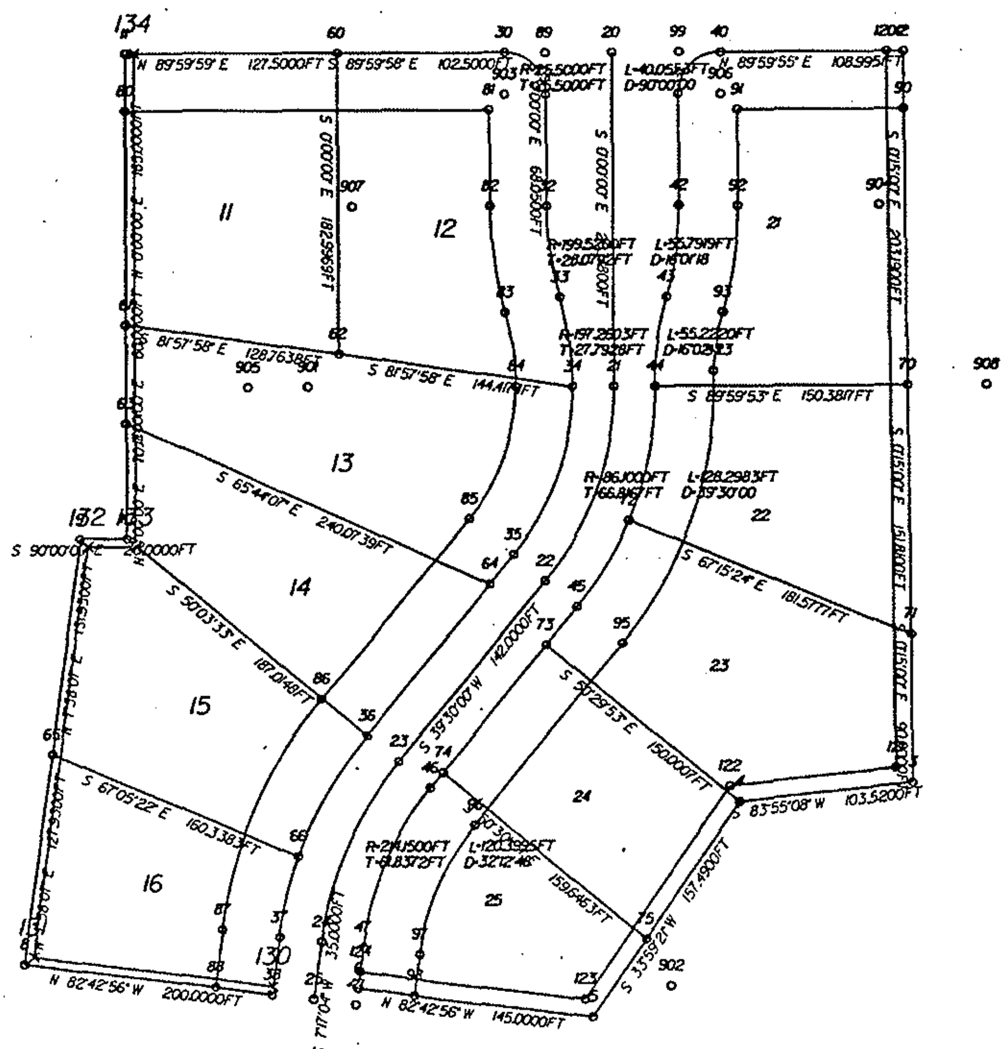
# Appendix E
# Sample Projects

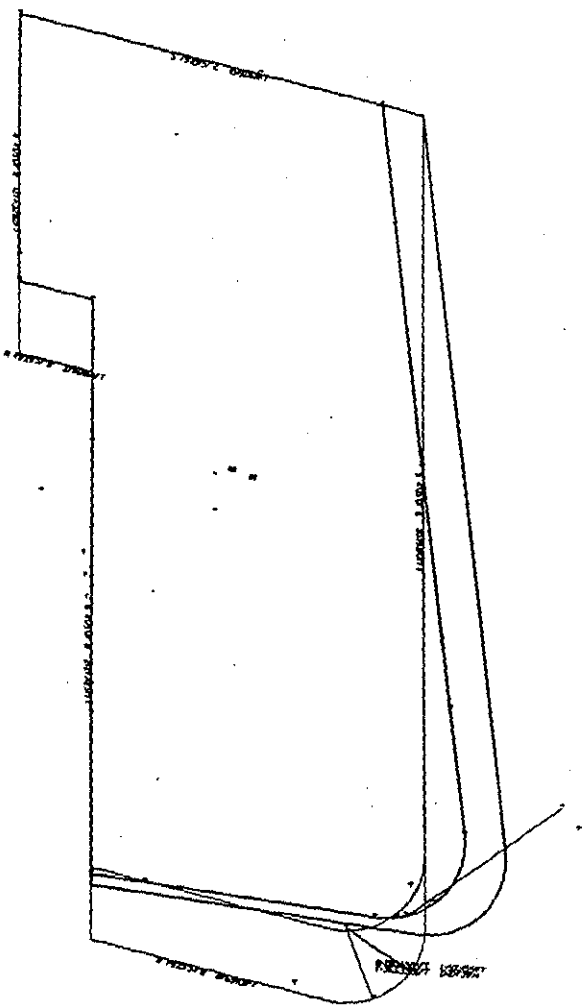
# E. SAMPLE PROJECTS

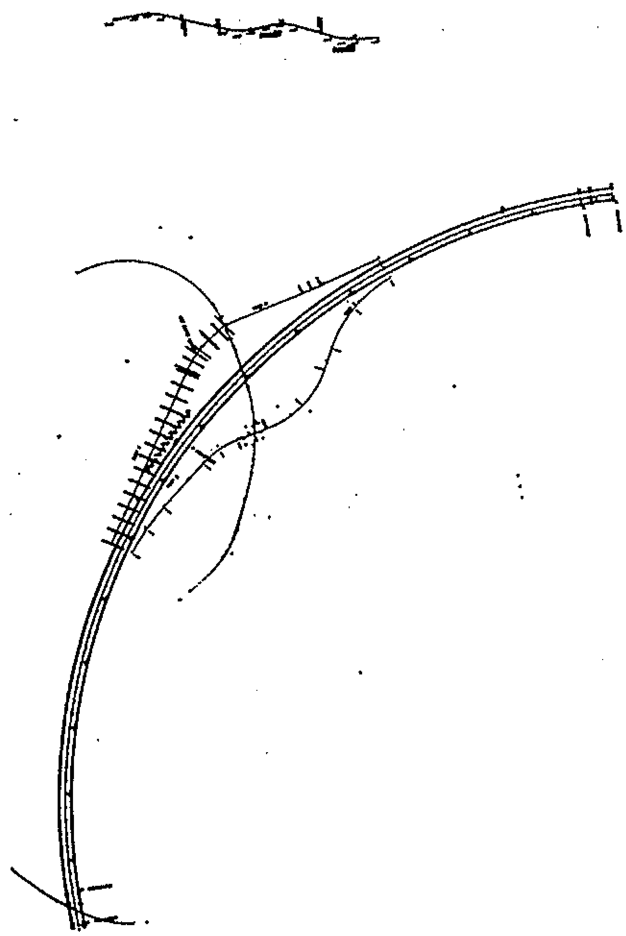
The file PRO_DD_ICS:EXAMPLE.CMD contains the input commands for three separate projects:

- a property survey
- a highway interchange
- a subdivision layout.

The input file contains numerous comments and includes usage of most ICS commands within a project workflow. Documentation for the use of this input file can be found in PRO_DD_ICS:EXAMPLE.DOC. The graphics output for each of the three sample projects follow.







F: ICS Error Messages

# Appendix F
# ICS Error Messages

# F. ICS ERROR MESSAGES

$$$$ ERRORS, JOB TERMINATED

Reason: The maximum number of errors allowed per job as specified in ICS.PAR has been exceeded. The ICS image will terminate itself.

Recovery: Correct the reported errors and rerun the input or increase the number of errors permitted by modifying ICS.PAR.

$$$$$$$$ IS INVALID

Reason: Some item of input data or value retrieved from the database is invalid.

Recovery: This error is command specific. Insure that the input data given makes sense for that command.

X IS INVALID

Reason: An argument in the data entry field may be wrong, or the data input format may be incorrect.

Recovery: Specify the correct arguments or consult the user's guide for the correct format.

ACCURACY IS POOR FOR THIS SOLUTION

Reason: A command has detected that the computed results are of poor quality.

Recovery: Inspect the results of the command and evaluate the impact of the poor results.

ALIGNMENT ID MUST BE LESS THAN 32768

Reason: The ID assigned to the alignment figure exceeds allowable Stereoplotter Alignment Cross-Sectioning (SAC) range.

Recovery: Renumber the figure.

ARC GREATER THAN SPIRAL LENGTH

Reason: The distance along the spiral as specified in the input is greater than the length of the spiral that is presently stored.

Recovery: Specify a distance that is less than the length of the spiral.

ARCHAIC COMMAND

Reason: The command selected has been replaced by a more robust command.

Recovery: Consult this User's Guide for an appropriate replacement.

ARCHAIC COMMAND, USE ALIGNMENT OFFSET

Reason: The command selected has been replaced by a more robust command.

Recovery: Use the command indicated.

ARCHAIC COMMAND, USE LOCATE FROM ALIGNMENT

Reason: Invalid command.

Recovery: Use current command.

ARCHAIC COMMAND, USE STATIONS AND OFFSETS

Reason: The command selected has been replaced by a more robust command.

Recovery: Use the command indicated.

AREA NOT FOUND AFTER NN ATTEMPTS

Reason: ICS has not been able to find a position for the side being adjusted that gives the desired area.

Recovery: Re-examine the area and desired area of the polygon, the method chosen to adjust the area, and the constraints given on the side of the polygon.

CLOSING DIRECTION NOT SPECIFIED, ANGLES ARE UNADJUSTED

Reason: A horizontal closure command has been run without specifying angular closure information or criteria. The angles are therefore assumed errorless and will not influence the calculation of the traverse closure.

COMMAND BETWEEN JOBS IS NOT CONTROL COMMAND

Reason: An attempt was made to perform an ICS command without having started a job and assigned an ICS table.

Recovery: Issue a START OF JOB command.

COMMAND DISABLED WITH DATABASE UPGRADE

Reason: An archaic command has been selected which is no longer valid for this version of ICS. It has been kept for transitional purposes.

Recovery: Do not use this command.

COORDINATE AREA FULL

Reason: The point entity in the database is full and cannot be extended.

Recovery: Insure that autoextend is enabled and that there is ample disk space for the extension. Contact your system manager if this problem persists.

COORDINATE RECORDS PARTIALLY UPDATED

Reason: A command has encountered a nonfatal error or stopping criteria but has already modified coordinates in the database due to intermediate calculations. This typically happens with adjustment or transformation commands whereby some tolerance has not been met.

Recovery: Copy the coordinates to a temporary point range prior to running the command if you suspect this may occur, or rerun the audit trail as input to restore the original coordinates if it already has happened.

DATABASE NOT FREE

Reason: The database is currently locked for write access by another user. This may be a temporary condition due to another ICS user, or a more permanent condition caused by a DMRS user issuing a lockout.

Recovery: Request that your system manager inspect those processes accessing the database and their status.

DEFAULT FEATURE DEFAULT NOT FOUND

Reason: No feature called "default" is found in the feature table.

Recovery: Key in the proper feature name.

DESIGN FILE IS 2-D

Reason: A 3D feature is specified for a 2D table.

Recovery: None

DESIGN FILE LIMITS EXCEEDED

Reason: The design file is not large enough to accommodate the requested elements to be placed.

Recovery: Exit ICS and increase the design file size using ICE utilities.

DESIRED OFFSET CREATES INVALID RADIUS FOR CURVE OR SPIRAL

Reason: An offset has been specified which collapses a curve or spiral.

Recovery: When specifying parallel offsets, specify an offset that does not exceed the radius of the largest curve or spiral.

DISTANCE IS ZERO

Reason: The distance specified as input to this command is zero.

Recovery: None

ELEVATION OF POINT X UNDEFINED

Reason: There is no elevation of the instrument station to transfer.

Recovery: Use the STORE command to specify the elevation of the instrument. For the VERTICAL CLOSURE command, inspect the list of points for the missing elevation.

END OF FILE NOT REACHED AFTER READING $$$$$ FEATURES

Reason: The feature table being used has lost its end-of-file sentinel, or the maximum number of features allowed per table (9999) has been exceeded.

Recovery: The file may have been corrupted, in which case it should be restored from an archive tape. Otherwise, consult EFEED documentation on possible remedies for this situation.

ENDING STATION GREATER THAN SPIRAL PT

Reason: You requested that points be computed on a spiral at stationing which extends beyond the station at which the spiral ends.

Recovery: Specify an ending station that is less than the spiral PT.

ERROR CALCULATING SPIRAL LENGTH, UNEXPECTED RESULTS MAY RESULT

Reason: The length of a spiral could not be determined. The effect on the results is unpredictable.

Recovery: Check for the validity of the given description.

ERROR CREATING NEW POINT

Reason: An error occurred while attempting to enter a new point in the database.

Recovery: Insure that autoextend is enabled and that there is ample disk space for the extension. Contact your system manager if this problem persists.

ERROR IN CGPAR

Reason: An error has occurred during initialization due to an improper parameter value.

Recovery: Inspect the contents of the file PRO_DD_ICS:ICS.PAR for the validity of the parameters therein.

ERROR IN PLOT, REQUEST ABORTED

Reason: An error occurred while placing graphics.

Recovery: Insure that enough free space exists in the design file. Also validate the feature table entry for the feature specified. If the problem persists, contact your system manager.

ERROR INITIALIZING GRAPHICS PLACEMENT

Reason: An error occurred initializing a design file for graphics placement.

Recovery: Try the command again. Insure that there is no conflict accessing the design file for write. Also, have your system manager check the integrity of the design file (contiguous, type 9 header).

ERROR OPENING $

Reason: An error occurred while opening the database indicated.

Recovery: Insure that the database in question is valid for the version of software you are running, and that you have the required permission to access it.

ERROR OPENING $$$$$$ AS OUTPUT

Reason: An error occurred while attempting to open the specified file for writing.

Recovery: Insure that directory/file protections do not prohibit this, and that there is ample disk space for the file.

ERROR OPENING FEATURE TABLE

Reason: The feature table specified could not be opened for read access.

Recovery: Insure that the given feature table exists and that you possess the required privileges to access it.

ERROR PLACING GRAPHICS

Reason: An error occurred during the placement of graphic elements.

Recovery: Try the command again. Inspect the input data for validity. If the problem persists, contact your system manager.

ERROR UPDATING VERTICAL CURVE LENGTH IN DATABASE

Reason: An internal error occurred while updating the database.

Recovery: Try the command again. If the problem persists, contact your system manager.

EVEN STATIONS INTERVAL MUST BE GREATER THAN ZERO

Reason: A value less than or equal to zero was entered for the interval in the EVEN STATIONS command.

Recovery: Use the proper interval, greater than zero.

EXAGGERATION MUST BE GREATER THAN 0 AND LESS THAN 30,000

Reason: Vertical exaggeration out of range (0 to 30,000).

Recovery: Change the value for vertical exaggeration (vEX) to a valid value (0 to 30,000).

EXECUTIVE REJECTED SCAN REQUEST

Reason: A scan request was rejected by the system.

Recovery: Try the command again. If the problem persists, contact your system manager.

FEATURE NOT FOUND

Reason: The feature name or number is not in a table.

Recovery: Consult the feature listing and plot with the correct name.

FEATURE TABLE INFORMATION IS UNRELIABLE

Reason: A list features command has been issued during an alphanumeric run with no design file associated to it. Some information displayed may depend on design file parameters and as such may be unreliable.

Recovery: If the questionable values are important, insure that a design file is attached to avoid this situation.

FIGURE $$$$$ IS ALREADY DEFINED

Reason: The user has requested that a figure be stored in the database under an id currently in use. If quiet mode has been enabled, ICS destroys the previous description.

Recovery: Disable quiet mode if you wish to indicate whether or not you want the previous description overwritten.

FIGURE $$$$$ IS UNDEFINED

Reason: The user requested that an operation be performed on a figure which could not be found in the database.

Recovery: Define the figure and select the command again.

FIGURE AREA FULL

Reason: More room is needed in the figure area to execute the specified command.

Recovery: Delete some figures from the table or make the table larger. In extreme cases you must make the table larger because there is not enough room to execute a delete command.

FIGURE LENGTH IS ZERO

Reason: An attempt has been made to store or process a figure without any elements.

Recovery: Specify a valid figure (one that contains elements).

FIGURES HAVE DIFFERENT LENGTHS

Reason: This command expects two descriptions with the same number of coordinates in each. This was found not to be the case.

Recovery: Insure that the descriptions define the same number of coordinates. An easy way to determine this is to run the LIST COORDINATES command on each.

FILE NAME SPECIFICATION ERROR

Reason: A key-in error has been made in entering a file specifier.

Recovery: Key in the file name correctly.

FILE NOT FOUND

Reason: The file specified could not be found on disk.

Recovery: Insure that the specified file exists and that its name is properly spelled.

FILE OPEN FAILURE

Reason: The file specified could not be opened for read, write, or both.

Recovery: Insure that the specified file exists and that directory/file protections do not prohibit this operation.

FILE READ/WRITE FAILURE

Reason: A fatal I/O error occurred.

Recovery: Contact your system manager to verify the integrity of the system and disk drive on which the file resides.

FINAL FIELD FORESIGHT STATION UNDEFINED
FINAL FIELD INSTRUMENT STATION UNDEFINED
FINAL RECORD FORESIGHT STATION UNDEFINED

Reason: One or more coordinates were not present for point(s) specified in the data list.

Recovery: Store or compute these points, or key in the correct point numbers.

FIRST POINT DIFFERENT FROM LAST

Reason: You requested that ICS compute the area on an open traverse.

Recovery: Specify computation of areas on closed elements only, where the first and last point in the description are the same.

FONT $$$$ IS UNDEFINED, DEFAULTING TO FONT O

Reason: The font specified in the feature table for the feature used could not be found in the font library associated with the design session.

Recovery: Exit ICS and attach the a font library containing the font(s) needed, or change feature tables with the new feature table command.

HORIZONTAL ALIGNMENT FIGURE IS UNDEFINED

Reason: The user requested that an operation be performed on a figure that could not be found in the database.

Recovery: Verify that you have given the correct figure ID.

HORIZONTAL ALIGNMENT UNDEFINED - SET ALIGNMENT NOT CALLED

Reason: The SET ALIGNMENT command was not completed correctly.

Recovery: Issue the SET ALIGNMENT command.

HORIZONTAL AND VERTICAL ALIGNMENT MUST BE A PERMANENT FIGURE

Reason: The figure used in Hfg of Vfg was not stored in the figure table.

Recovery: Store the figure in the figure table, and key in the figure ID instead of the actual figure.

I/O ERROR IS XXXX

Reason: FORTRAN I/O runtime error

Recovery: Use the VAX/VMS error messages to decode the error number.

ICS INTERNAL ERROR - TOO MANY TABLES OPEN

Reason: ICS allocates six units for table I/O. This amount has been exceeded.

Recovery: Issue an END OF JOB command and continue.

ILLEGAL FIGURE - CURVE PRESENT IN FIGURE

Reason: An invalid vertical alignment figure was used. The figure is contained in a curve.

Recovery: Use the correct vertical alignment figure, one generated using VERTICAL START and VERTICAL END.

IMPROPER USE OF BRACKETS IN COMMAND

Reason: The specified command can only process one figure at a time: multiple figures were specified within brackets.

Recovery: Re-enter command without brackets.

INITIALIZATION ERROR

Reason: Error in ICS.PAR file parameters.

Recovery: Check the ICS.PAR file for improper parameter values.

INPUT DATA ERROR

Reason: The data supplied for this command is not in accordance with the data required for this particular command.

Recovery: The point in the input where the error was detected is underlined. If no underline appears, the error was caused by not enough data. If the cause of the error is not apparent, compare the input against the data field arguments. Then re-enter the command with the correct data.

INSUFFICIENT NUMBER OF POINTS FOR A VERTICAL CURVE

Reason: Fewer than two points are in the vertical alignment.

Recovery: Use three or more PVIs in the vertical alignment.

INSUFFICIENT NUMBER OF POINTS FOR TRANSFORMATION

Reason: Fewer than two points in each coordinate system were specified.

Recovery: Specify more than two points in each coordinate system.

INTERNAL ERROR PROCESSING LINKED LIST $

Reason: This is an internal problem resulting from the corruption of a data structure.

Recovery: Try the command again. If the problem persists, contact your system manager.

INTERSECTION ANGLE IS $

Reason: A steep intersection has been computed for which the numerical results may be questionable.

Recovery: Inspect the geometry of the intersection, perhaps selecting an alternate way of computing it if required.

INVALID COMMAND

Reason: The first non-blank column of this record contains an alphabetic character indicating that it is the first letter of a command, but the command as entered cannot be recognized. If the job was started with a numeric start of job command, column one contains a character other than an asterisk or blank. However, the command cannot be recognized.

Recovery: Refer to Section 2.3 for valid abbreviations.

INVALID ELEMENT IN FIGURE; CURVE OR SPIRAL PRESENT

Reason: A figure containing a curve or spiral was specified for a command expecting vertical data.

Recovery: Re-enter the command with a different figure.

INVALID ELEMENT IN FIGURE; VERTICAL CURVE PRESENT

Reason: A figure containing a vertical curve was specified for a command expecting horizontal data.

Recovery: Re-enter the command with a different figure.

INVALID ENTRY - ALL 4 POINTS ON CIRCLE COTANGENT

Reason: The resection cannot be performed if all 4 points lie on a circle.

Recovery: Re-enter the command using different points.

INVALID ENTRY, NO DATABASE SPECIFIED

Reason: A command requesting that a valid database be specified has not been fulfilled.

Recovery: Indicate a valid database filename.

INVALID FIGURE X

Reason: The number specified as a figure number is not in the range from 1 to 9999.

Recovery: The range must be 1 to 9999.

INVALID FIGURE LIST, NO CURVES, SPIRALS, OR VERTICAL CURVES

Reason: A description contains invalid geometric elements for this command. This can occur with vertical alignment commands encountering circular curves or transition spirals, for example.

Recovery: Insure that the geometric elements comprising the description make sense for the command.

INVALID NUMBER OF PARTS X

Reason: The number of parts specified is not positive.

Recovery: Make the number of parts positive.

INVALID NUMBER OF SECTION CORNERS

Reason: An invalid number of section corners has been specified.

Recovery: The section subdivision command requires that all four section corners are given. Up to a maximum of four quarter corners may also be input. Inspect and correct any input data errors.

INVALID POINT X

Reason: An attempt has been made to store a point with a point number that is zero (0), negative, or greater than 2,147,483,647.

Recovery: Specify points that are in the range from 1 to 2,147,483,647.

INVALID POINT RANGE FROM X TO Y

Reason: The points to be defined, as specified by the number of parts and the starting point number, include at least one invalid point.

Recovery: Re-execute with a valid range.

INVALID QUARTER SECTION POINT

Reason: An invalid quarter section point has been input.

Recovery: Check the validity of the quarter section points given. They should be roughly at the midpoint of any two section corners given. Re-execute with valid points.

INVALID RADII IN REVERSE CURVES

Reason: One (or both) of the radii specified for the reverse curve command is invalid.

Recovery: Inspect the radii given for the geometric problem entered, insuring that they are reasonable.

INVALID RANGE FROM sBEG to sEND, NO POINTS COMPUTED

Reason: The user has specified invalid station limits for the operation requested on the alignment specified.

Recovery: Insure that the station limits specified fall within the station limits of the alignment specified.

INVALID SECTION NUMBER

Reason: An invalid section number was specified.

Recovery: Section numbers range from 1 to 36. Inspect and correct any input data error.

INVALID SPIRAL IN FIGURE, UNABLE TO COMPUTE RADIUS

Reason: An invalid spiral within a figure has been encountered.

Recovery: Insure that either a tangent and/or circular curve(s) bound each spiral within a figure.

INVALID STATION EQUATION IN DATABASE

Reason: A station equation which causes a fatal stationing conflict has been detected in the database.

Recovery: Overlap station equations cannot exceed the overlap constant defined in ICS.PAR. Adjust this value or redefine stationing along the alignment as required.

INVALID STATION INTERVAL X

Reason: The station interval specified is not positive.

Recovery: Make the station interval positive.

INVALID STATION NUMBER

Reason: An invalid station has been specified for the alignment given.

Recovery: Check the validity of the input data for the alignment in question and its stationing.

INVALID VERTICAL CURVE LENGTHS BETWEEN POINTS N1 AND N2

Reason: Vertical curve length is less than zero or is overlapping other vertical curves.

Recovery: Key in a smaller vertical curve length if Vcl is too large; the Vcl must be greater than zero.

ISSUE VERTICAL START FIRST

Reason: The user has attempted to append to or construct a vertical alignment without first having issued a VERTICAL START command.

Recovery: Issue the VERTICAL START command as required.

JOB X TERMINATED

Reason: A START OF JOB was encountered without an END OF JOB command having been executed. The current job is being terminated so that the new one can be started. The table in use by the terminated job is updated as if an END OF JOB had been read.

Recovery: Issue an END OF JOB command before issuing a START OF JOB.

JOB NUMBERS FOR THIS DATABASE ARE $

Reason: The user entered incorrect job numbers for this database via the start of job command. The correct numbers are reported.

Recovery: Insure that this is in fact the database you wish to use and if so, re-enter the START OF JOB command with the correct job numbers.

LONGER RADIUS MUST BE FIRST

Reason: The first radius entered is smaller than the second, and sign and cross have been specified.

Recovery: Refer to Section 6.12 for valid input.

LINE AT NNNNN HAS CHANGED DIRECTION

Reason: The adjustment of a parcel for area has caused the specified line to change direction.

Recovery: None.

MAXIMUM ANGULAR MISCLOSURE EXCEEDED

Reason: The user-supplied angular misclosure tolerance has not been satisfied.

Recovery: Inspect the input data for possible errors. Survey observations may be poor.

MAXIMUM ITERATIONS EXCEEDED FOR ROOT

Reason: An internal numerical routine did not converge.

Recovery: Inspect the input data for validity. If it is valid, contact your system manager.

MAXIMUM LINEAR MISCLOSURE EXCEEDED

Reason: The linear misclosure tolerance that you specified has not been met.

Recovery: Inspect the input data for possible error. Survey observations may be poor.

MAXIMUM NUMBER OF STATIONS ALLOWED IS $

Reason: An invalid number of stations has been given.

Recovery: Insure that the number of stations does not exceed the value indicated by the error message.

MINIMUM RELATIVE PRECISION EXCEEDED

Reason: The minimum relative precision that you specified has not been met.

Recovery: Inspect input data for possible errors. Survey observations may be poor.

MODE MUST BE ZERO OR 1

Reason: Invalid key-in.

Recovery: Key in 0 or 1.

NO XXXX FILE NAMED

Reason: An input source file or output destination file was not specified.

Recovery: Include a file name with the command.

NO ACTIVE TRANSFORMATION

Reason: At least one HELMERT TRANSFORM must precede the TRANSFORM COORDINATES command.

Recovery: Do the HELMERT TRANSFORM before executing TRANSFORM COORDINATES.

NO ACTIVE TRANSFORMATION

Reason: The user has requested that a set of coordinates be transformed without first having run the COMPUTE TRANSFORM command.

Recovery: You must establish an active transformation with the COMPUTE TRANSFORM command prior to using the transform coordinates command. A graphics tutorial allows you to review any active transformation.

NO ANGLE

Reason: One of the sides of the angle to be divided has zero length, and therefore has no direction.

Recovery: Both sides must have a defined length.

NO COORDINATES FOUND FOR DELETION

Reason: The user requested deletion of a series of coordinates for which no records were found in the database.

Recovery: Check to be sure you are entering the correct points.

NO CURVE DEFINED

Reason: These commands require that a curve be previously stored by an alignment or define curve command.

Recovery: Store a curve.

NO DESIGN FILE SPECIFIED

Reason: The user has initiated a plot or annotate command from an alphanumeric terminal without having specified a design file during initialization.

Recovery: Exit ICS and re-enter the program, specifying a destination design file for plot and annotate commands.

NO ELEVATION EXISTS FOR POINT: $$$$$$$$

Reason: A command which may need an elevation encountered a point for which no elevation was defined.

Recovery: Inspect the results and insure that the results are reasonable.

NO FEATURE SPECIFIED

Reason: No arguments were specified, or the incorrect arguments were specified.

Recovery: Specify the correct arguments in the command's data field.

NO HORIZONTAL ALIGNMENT DEFINED

Reason: A command has been entered which requires that an active horizontal alignment first be defined with the STORE FIGURE or SET ALIGNMENT command.

Recovery: Issue the appropriate command first to activate a horizontal alignment.

NO INTERSECTION

Reason: The specified elements do not intersect, or the intersection cannot be computed because of some condition noted in an earlier error message.

Recovery: Refer to the previous error messages.

NO LINE

Reason: The points specified to define the line have the same coordinates, and therefore do not define a line.

Recovery: Specify two points having different positions.

NO OFFSET IS POSSIBLE

Reason: No offset can be computed because of a previously noted error; in the case of tangent offset, the two points defining the line are actually the same point; in the case of spiral offset, the offset does not fall on the spiral; or the offset has not been found after 1000 tries.

Recovery: Refer to previous error messages. Note the positions of the points if in graphics.

NO PREVIOUS CURVE TO DEFINE STATIONING

Reason: The station has been specified as -1, meaning that stationing is to be carried forward from the previous curve, but no previous curve has been stored by an ALIGNMENT or DEFINE CURVE command.

Recovery: Store a curve.

NO SOLUTION

Reason: There is no mathematical solution possible for the data given, or the solution is beyond the capabilities of the algorithm used.

Recovery: Inspect the geometry to be solved for and the possibility of a solution. Consult your system manager if you feel that a solution is possible.

NO SPIRAL IS DEFINED

Reason: These commands require a spiral to be stored by a SIMPLE SPIRAL, SPIRAL LENGTH, or COMPOUND SPIRAL command. This has not been done.

Recovery: Store a spiral using one of the above commands.

NO TANGENT POSSIBLE

Reason: No tangent is being computed because of the condition noted in a previous message, because one circle lies entirely within the other, or because a cross tangent between intersecting circles has been requested.

Recovery: Refer to previous messages. Note the positions of the points in graphics.

NO TRANSFORMATION ACTIVE

Reason: The user has requested that a set of coordinates be transformed without first having run the COMPUTE TRANSFORM command.

Recovery: Recall that there may be one and only one transformation active at a time, and that transformations are not preserved from run to run. You must establish an active transformation with the COMPUTE TRANSFORM command prior to using the TRANSFORM COORDINATES command. A graphics tutorial allows you to review any active transformation.

NUMBER OF POINTS IN EACH SYSTEM IS UNEQUAL

Reason: The number of points in each coordinate system does not match the others.

Recovery: Insure that there are equal numbers of points in each coordinate system.

OVER XXXX ERRORS. JOB TERMINATED

Reason: The user-defined maximum number of errors per job has been reached. The job is terminated.

Recovery: Correct the errors.

PERPENDICULAR TO ALIGNMENT DOES NOT EXIST

Reason: The user requested that a perpendicular be computed which does not exist.

Recovery: Inspect the geometry of the solution desired and the input data given.

PLOT CURVE ERROR; LESS THAN 3 POINTS OR 3 POINTS COLINEAR POINTS

Reason: Either fewer than three points are in des, or three points in the description are in a straight line

Recovery: Either increase the length of des, or use PLOT SHAPE to plot points in a straight line.

POINT $$$$$ HAS UNDEFINED Z-COORDINATE

Reason: A command requiring that elevations exist for each point to be processed has determined that a point or points did not have an elevation assigned.

Recovery: Insure that the points to be processed have elevations.

POINT $$$$$ IS ALREADY DEFINED

Reason: The user has requested that a point be stored in the database under an id currently in use. If quiet mode has been enabled, ICS destroys the previous description.

Recovery: Disable quiet mode if you wish to indicate whether or not you want this overwrite to occur.

POINT $$$$$$ IS NOT ON HORIZONTAL ALIGNMENT

Reason: The point given in SET ALIGNMENT is not on the horizontal alignment figure, or the point on the vertical alignment is not on the horizontal alignment.

Recovery: Check the points to make sure they are on the horizontal or vertical alignment figure.

POINT $$$$$ IS UNDEFINED

Reason: The user has requested that a command operate on a coordinate which does not exist.

Recovery: Insure that the command and the point numbers specified were entered correctly, or define the points in question.

POINT TS NOT BETWEEN PI AND PBT

Reason: The tangent to spiral specified does not lie on the tangent between the point on the back tangent and the spiral intersection of tangents given.

Recovery: Inspect the input data for validity.

POINT X OR POINT Y IS OUT OF SEQUENCE

Reason: Points X and Y were specified as the starting and ending points of the sides to be adjusted. However, either one of these points is not in the description, one of them is the first or last point in the description, or Y precedes X in the description.

Recovery: Note the description via a LIST FIGURES command.

PREVIOUS VALUE OF FIGURE $$$$$$$$ DESTROYED

Reason: This warns the operator that a previously defined figure has been destroyed and replaced with a new figure.

Recovery: Insure quiet mode is disabled if you are concerned about destroying previously defined points.

PREVIOUS VALUE OF POINT $$$$$$$$ DESTROYED

Reason: This warns the operator that a previously defined position has been destroyed and replaced with a new position.

Recovery: Insure quiet mode is disabled if you are concerned about destroying previously defined points.

RADIUS IS UNDEFINED

Reason: An error occurred while ICS was processing a curve for which the radius is needed but could not be determined.

Recovery: Insure that the PC, CC, and PT are valid and that the distance between each is >0.

RADIUS LESS THAN HALF THE CHORD LENGTH

Reason: The distance between the two points specified is greater than twice the radius specified. Therefore, the points cannot be on the curve.

Recovery: Note the positions of the points if in graphics.

SECTION DIMENSION LARGER THAN 110% THEORETICAL

Reason: The subdivision of a section has resulted in questionable dimensions.

Recovery: Check to make sure the input data is correct. Be wary of sections that do not follow reasonable dimensions.

# Glossary

# ICS GLOSSARY

The following terms and definitions, in some instances, are specific to Intergraph Corporation or have different meanings as they apply to other fields of interest.

These definitions are not meant to be universal, but are expressed in the context of their usage within Interactive Coordinate Geometry Subsystem software.

| | |
|---|---|
| ahead tangent | the tangent exiting from a curve |
| alignment | a series of tangents and circular curves that describe the centerline of a highway or easement |
| angle | circular measurement taken from the intersection of two lines, given in degrees, or gons |
| audit trail file | a file optionally created every time ICS is used. The file records all inputs to ICS. |
| back tangent | the tangent entering a curve |
| batch mode | entry of input using an input file |
| closure line | the line between the actual endpoint of a traverse and the desired endpoint |
| command | instructions from the user to ICS to perform a function on the specified data |
| compound curve | a curve with two or more curves of differing degrees of curvature, both in the same direction |
| coordinate | the location of a point along a reference axis |
| data field | the second component of a command; this component specifies the data which the command is to use |
| data types | the different categories of information that can be entered in the data field of an ICS command |
| DCL | Digital Control Language; the terminal is in this mode when the $ prompt appears |

| | |
|---|---|
| default | a preassigned value that is assigned if a user-specified value is not entered |
| deflection angle | angular measurement between the extension of a line and another line |
| degree of curvature | central angle of a circle which subtends a 100-foot arc and is equal to $18000/(\pi^*r)$ degrees where $r$ is the radius of the circle |
| figure | list of point numbers and geometrical delimiters. The figure number may be used repeatedly instead of listing the point numbers each time they are needed. |
| forward tangent | the tangent exiting a curve |
| graphics environment | execution of ICS while in a design file with the tutorial displayed |
| host | the device and UIC a user is logged in to |
| input file | a file that contains the ICS commands necessary to execute a job |
| integer | a whole number, with no fractional part |
| interactive mode | input and output performed at the keyboard of a terminal; the process of sending one command to the computer at a time |
| output file | a file which may be created by the user. It records all output resulting from ICS commands and (optionally) records the input. |
| reverse curve | a curve which changes direction from clockwise to counterclockwise or vice versa |
| segment | figure made by the intersection of a circular arc and a chord |
| spiral in | a transition spiral with decreasing radius; that is, going from a line (tangent) to a circular arc |
| spiral out | a transition spiral with increasing radius; that is, going from a circular arc to a line (tangent) |

| | |
|---|---|
| station | a relative location on an alignment, measured in terms of path distance along the alignment |
| transposed figure | a figure in reverse order |