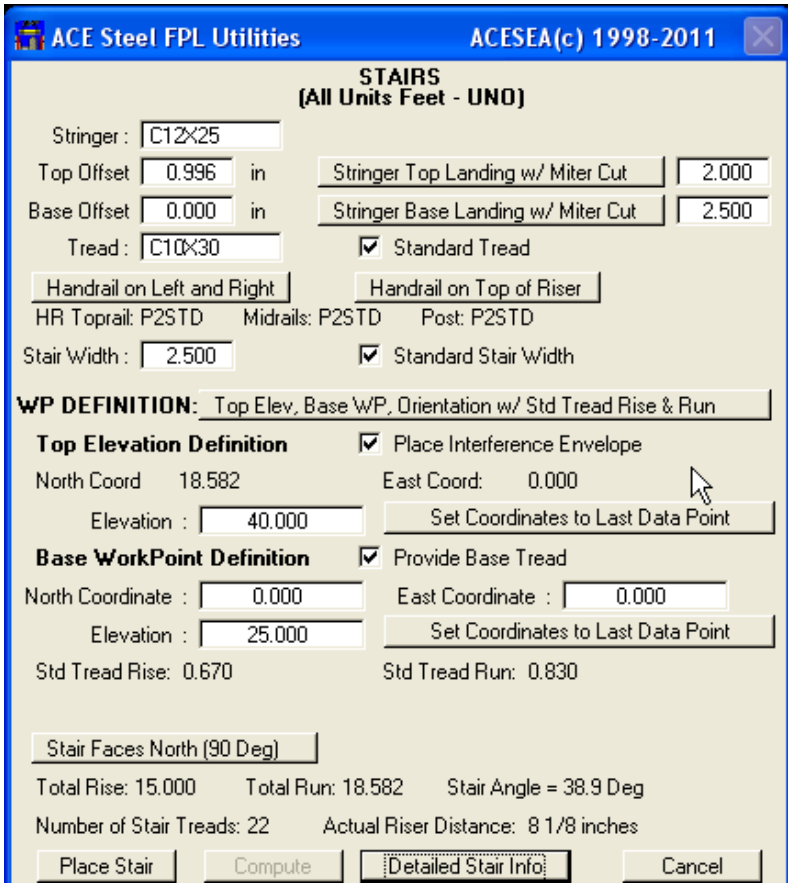
ACE Structural Engineering Applications LLC
ACE FrameWorks Utilities
Stairs Documentation

Mar 15, 2013

Stairs (ACE_ST.MA)

(Current Versions: FWP 3.1.x.x/3.2.x.x rel 2.1.5 & FWP 7.0.x.x rel 7.1.5 & FWP 7.1/7.2/7.3 rel 7.1.5 & FWP 8.0.x.x rel 8.1.5 & FWP 9.0.x.x rel 9.1.5 & FWP 10.0.x.x rel 10.1.5 & FWP 11.0.x.x rel 11.1.5 & FWP 12.0.x.x rel 12.1.5)

The stairs program facilitates the placement of stairs. The stairs consists of two stringers with treads and may optionally be placed with or without handrails and/or landings. The handrail (pipe or angle sections) can be supported on the top or side of the stringer. In addition, handrail may be placed on the left side only, right side only, left & right side or neither side. The handrail may consist of a single rail, two rails or three rails. An additional "Handhold" rail, which is inset from the handrail, may be specified (see page 23). The top of the stringer may be vertical (flush) cut or mitered with a landing. The base of the stringer may be horizontal cut, vertical (flush) cut or mitered with a landing. There are two top treatment options and three base treatment options allowing six different stair configurations (see figures 1 - 6). The stairs rise from the XY plane at any desired angle. The treads may be inset from the edge of the stringer (six methods provided). Treads may be inset by moving the treads (see figures 8 - 10) or by moving the stringer ends (see figures 11 - 15). Both a top and/or bottom surface offset may be specified (see figure 7). An interference envelope (FWP solids) may be placed. The stairs application also supports an immediate undo option after placement.



Numerous techniques (thirteen total) are provided to define the stair coordinates & orientation. The stairs are placed by center top and bottom workpoints, which is illustrated in the figures in this document. The method of defining the top and bottom WP's is dependent upon the stair definition technique chosen. The various placement techniques are listed in the ensuing sections. This utility also features the "Last Data Point" technique, which can be utilized to place the last data point coordinates into the dialog keyin boxes. After stair coordinate data is provided, pressing the Compute button will display stair data, which includes: number treads, stair angle, total rise, total run and actual riser distance. If the computed values are satisfactory, the stairs can be placed with the Place Stair button.

The utility has internal defaults for stringer, tread and handrail section sizes, types, classes and grades. Stair & handrail dimensions also have internal default values. In addition all input parameters shown in the dialog box shown above have default values. The defaults may be overridden with user defined defaults by using a definition file which is discussed in detail later in this document. This utility also features limit values and standard value definitions for stairs.

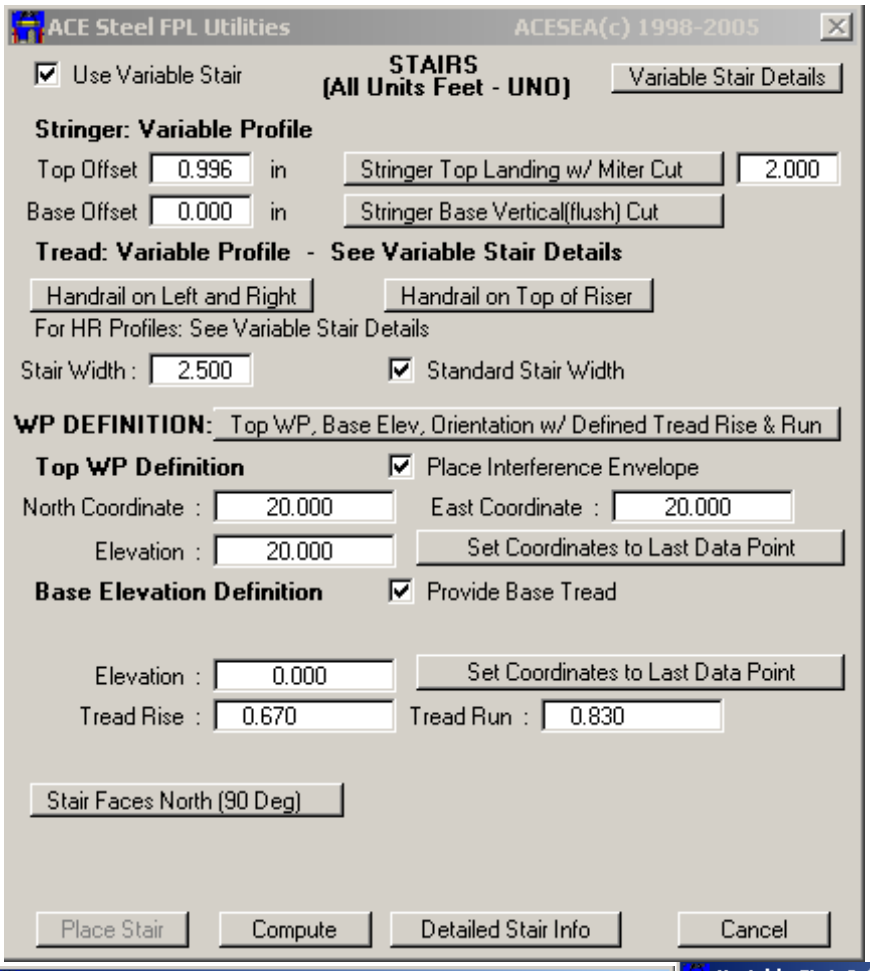
Stair component profiles (stringer, tread, hrp, hrr & hrm) can be based on stair length if the Variable Stair Capability is activated (see next page).

ACE FWP Stairs Documentation

Variable Stair Capability (Stair Profiles Based on Property Specification Case - PSC)

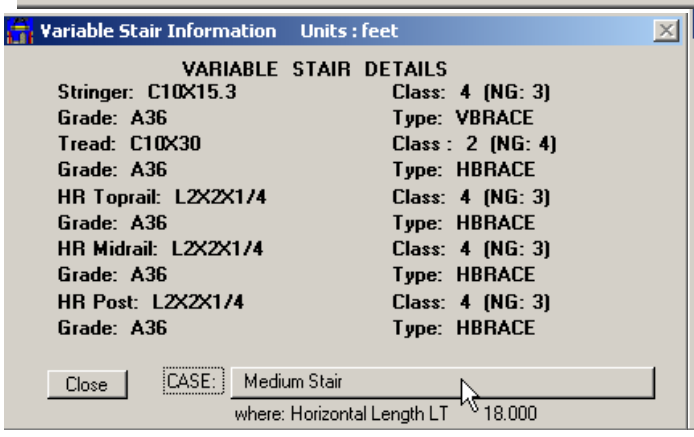
(Optional Capability)

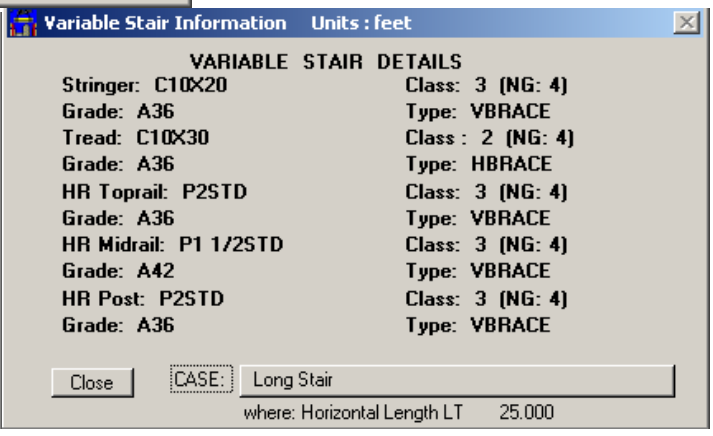
The Variable Stair Capability can be activated by specifying a valid PSC command in the definition file. A PSC command points to a PSC file, which contains cases defining a set of profiles for a specific length condition. The details of the PSC file are included at the end of the document following section on definition files. If a valid PSC file with at least one case (maximum 10) is defined in the definition file, the start-up dialog box will look as follows:



The dialog box includes the "Use Variable Stair" toggle and the "Variable Stair Details" command. The variable stair capability can be toggled off at any time with the "Use Variable Stair" toggle. The following dialog box appears when the "Variable Stair Details" button is pressed.

The cases contained in the PSC file can be viewed through this dialog box using the CASE option button at the bottom of the dialog box. The default case (profiles & properties specified in the definition file) can also be viewed. Cases are examined from the 1st to the last. The first case, which satisfies the length condition, determines the stair component properties.

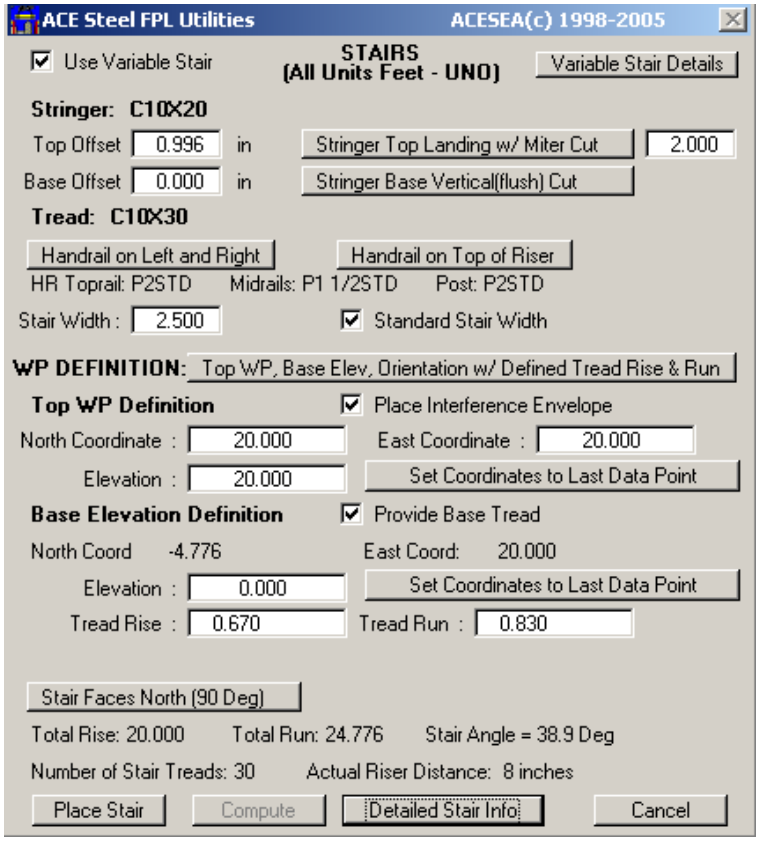




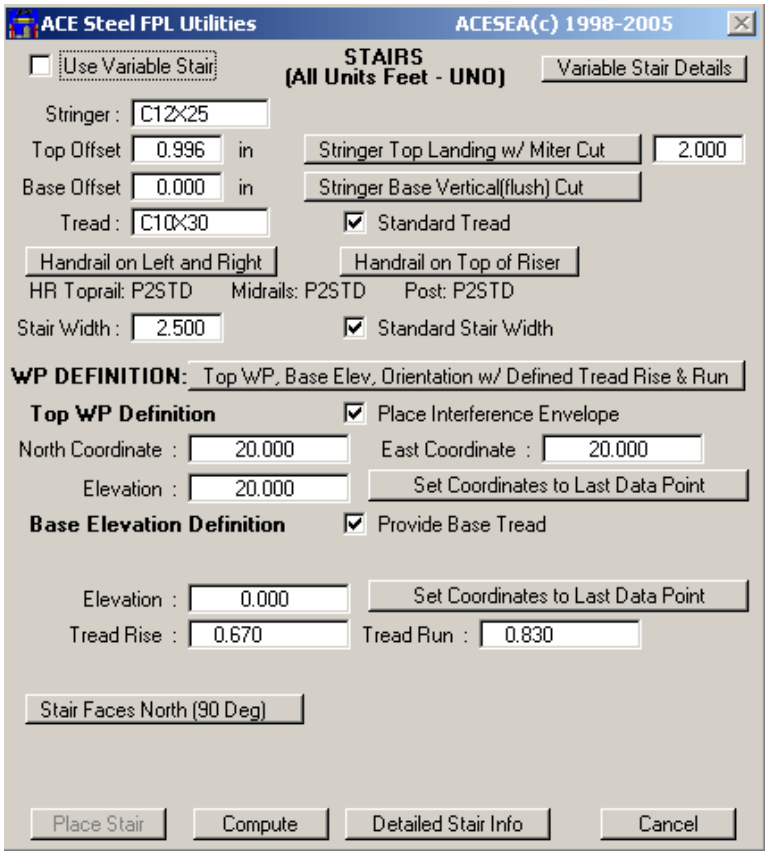
ACE FWP Stairs Documentation

Variable Stair Capability (con'd) – OPTIONAL CAPABILITY

When the variable stair capability is toggled on, the selected profiles will be displayed on the dialog box as shown left. If the "Place Stair" button is pressed the displayed profiles are those that will be utilized for stair placement.



If the "Use Variable Stair" is toggled off, the handrail profiles & properties are displayed and the Stringer & Tread profile keyin-dialog boxes appear. The Variable Stair Capability can be toggled on or off as desired at any time. When toggled off, the dialog box will appear as shown below.



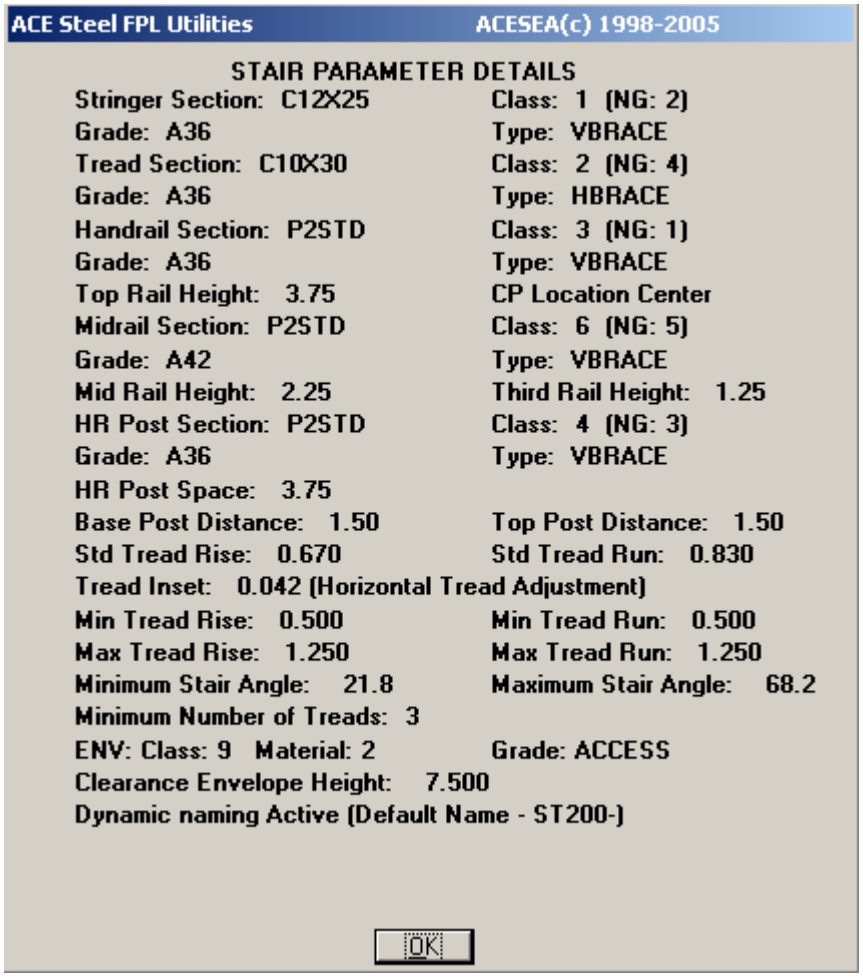
ACE FWP Stairs Documentation

Naming Options

For a given stair, all components are given the same name by concatenating the FWP ID to a prefix. The default prefix is STAIR, however a different prefix may be specified in the definition file. Other naming options include 1) a constant specified name for all stairs, 2) dynamic naming where the stair name with or without FWP_ID appended is defined at placement time or 3) FrameWorks normal naming for individual components (autoname).

Information Dialog Box

The Detailed Stair Info button yields the following information:



The data shown on this dialog box is the profiles, properties & stair parameters specified in the definition file and any interactive changes to the riser & tread profiles. If the Variable Stair Capability is active, the stair component profiles & properties will come from the PSC file providing a case length condition is met. If not met, the stair component profiles & properties will be those specified in the definition file (riser & tread interactive profile changes will be ignored).

The variables on both the primary dialog box (definable in definitions file & changeable interactively) and the information dialog box (definable in definitions file) are discussed in the next section.

ACE FWP Stairs Documentation

Stairs Variables and Options

The Stair application, ACE_ST.MA, has been designed to allow great flexibility (i.e. many options) in the placement of stair stringers. This FrameWorks Plus FPL application is limited to a single stringer run but allows the following interactive changes:

- Stringer Section Size (stringers are expected to be channels)
- Tread Section Size
  (Note: stringer & tread may be keyed-in with Variable Stair Capability (PSC file) active)
- Handrail Options - none, left only, right only, or both left & right.
- Handrail position (top or side)
- Stair Width
- Stair Location & orientation (thirteen methods of definition).
- Interference Envelope
- Option for Stringer Top Treatment (vertical (flush) cut or base landing w/ miter)
- Option for Stringer Base Treatment (horizontal cut, vertical (flush) cut or base landing w/ miter)
- Top Offset (vertical distance from top WP to actual stringer TOS) - normally floor thickness
- Base Offset (vertical distance from base WP to actual stringer TOS) - normally floor thickness
- Base Tread Option (for base vertical cut or base w/ landing)
- Top & Base String Shift Option for Inset w/ Stringer Shift (inset handling options 3 through 5)

The following parameters may be defined in the definitions file (note that all interactive definition variables may also be defined in the definitions file and are not listed below):

- Stringer grade, class, type & namegroup
- Tread grade, class, type & namegroup
- HR post section, grade, class, type & namegroup
- HR rail section, grade, class, type & namegroup
- HR midrail section, grade, class, type & namegroup (optional specification)
  (Note: above profiles & properties may be overridden with Variable Stair Capability (PSC file))
- HR handhold rail section, grade, class, & type (optional specification)
- Standard tread rise & run
- Tread inset distance
- Tread inset handling option (six methods with & without stringer shifting)
- Maximum tread rise & run
- Maximum & Minimum stair angles
- Minimum number of treads
- Naming options for stair components
- HR Post maximum spacing
- HR rail top, midrail, & third rail heights
- HR handhold rail height, offset, CP & support option (optional specification)
- Base HR post offset
- Top HR post offset
- Default HR option & Handrail position at startup
- Default stringer top and base treatment at startup
- Method of Stair location at startup
- Envelope Class, Grade & Material
- Height of access envelope

Many of the above listed options are shown graphically in the following illustrations. Subsequently, a sample definitions file is shown with expanded explanation.

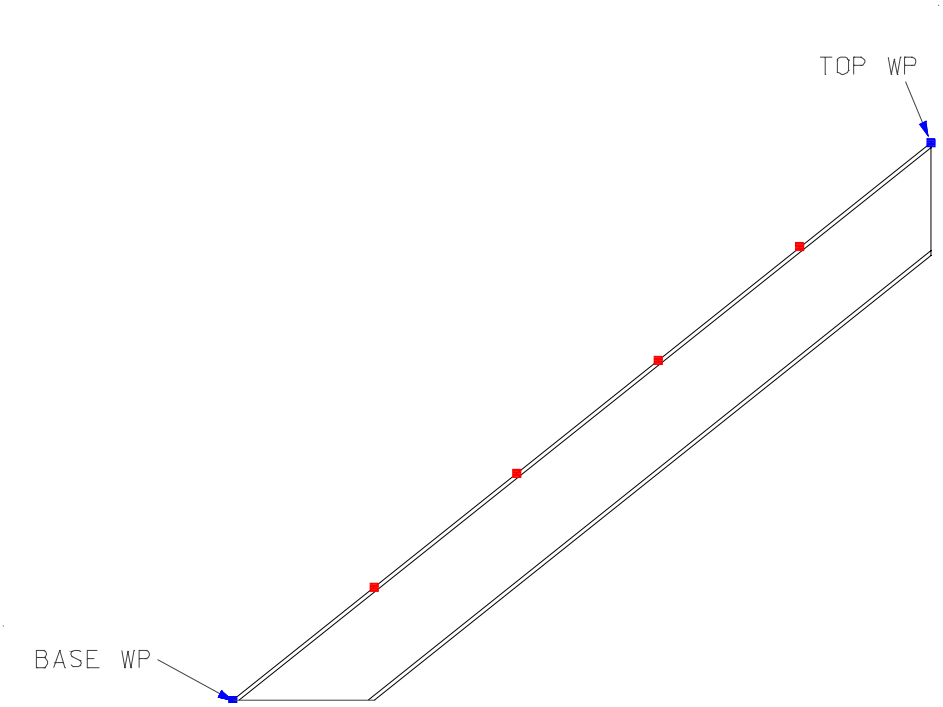


Figure 1: Horizontal Cut Base with Vertical Cut Top

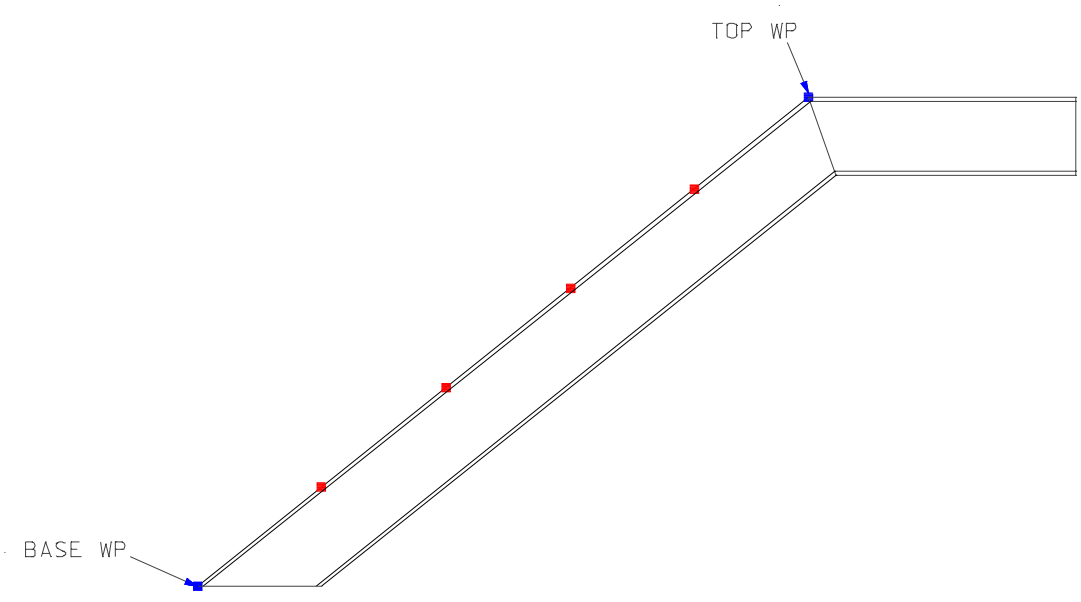


Figure 2: Horizontal Cut Base with Top Landing

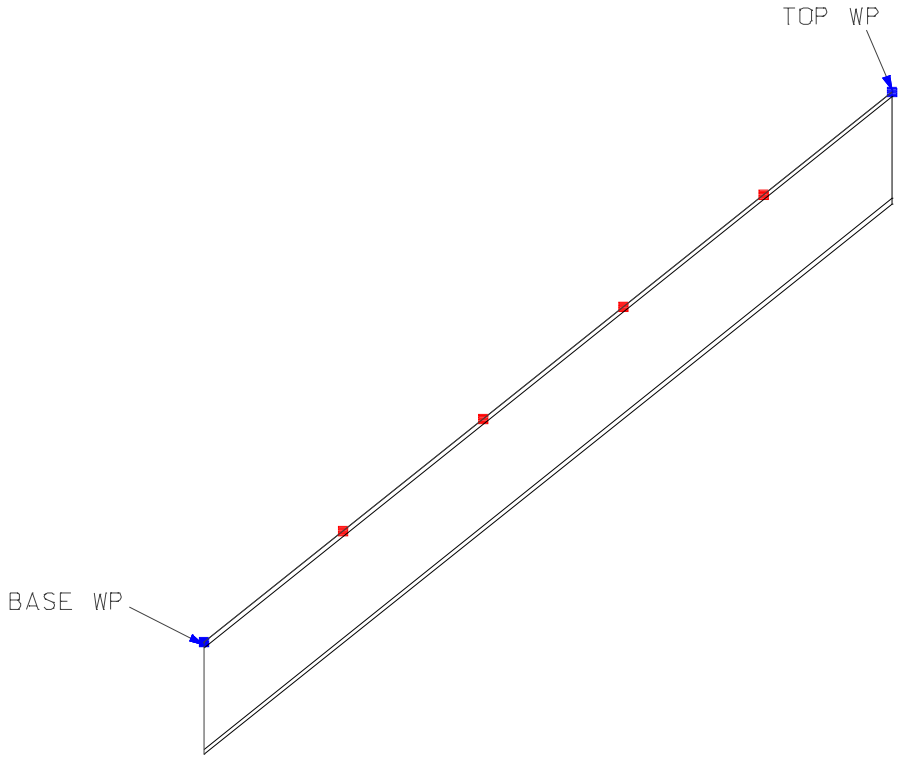


Figure 3: Vertical Cut Base with Vertical Cut Top

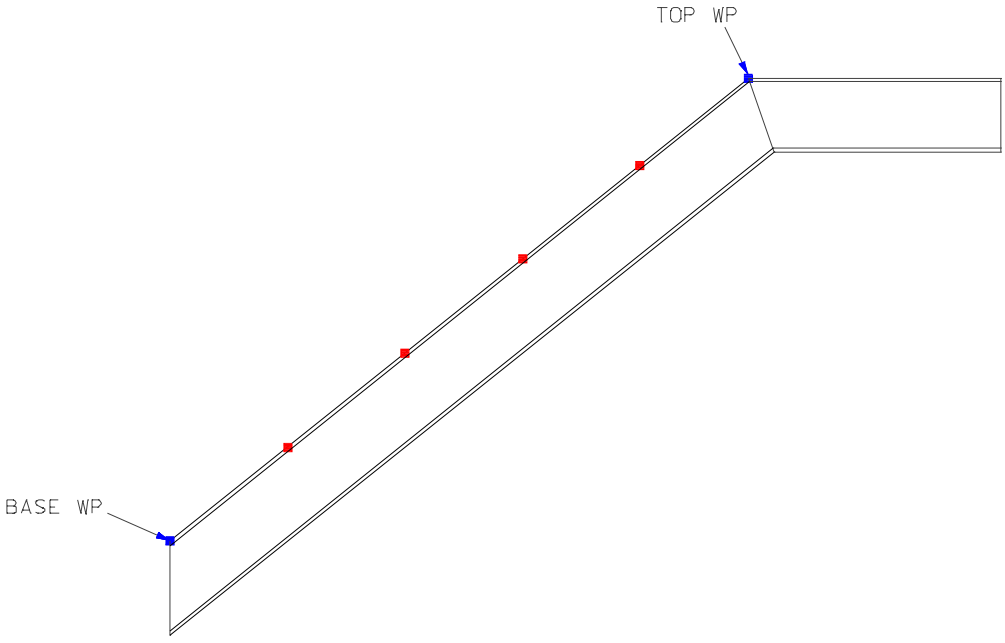


Figure 4: Vertical Cut Base with Top Landing

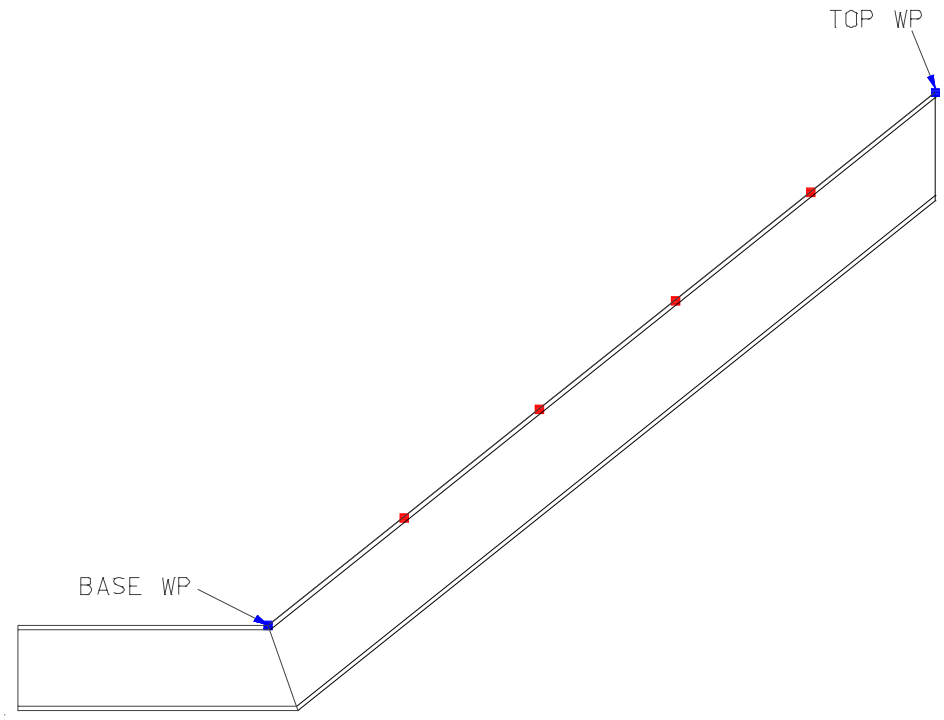


Figure 5: Base Landing with Vertical Cut Top

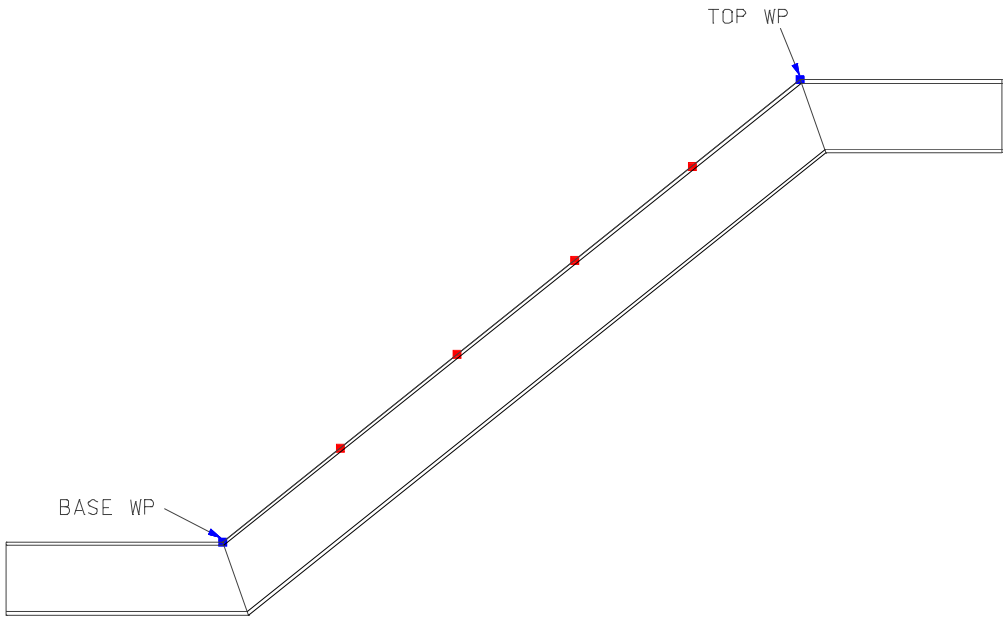


Figure 6: Base Landing with Top Landing

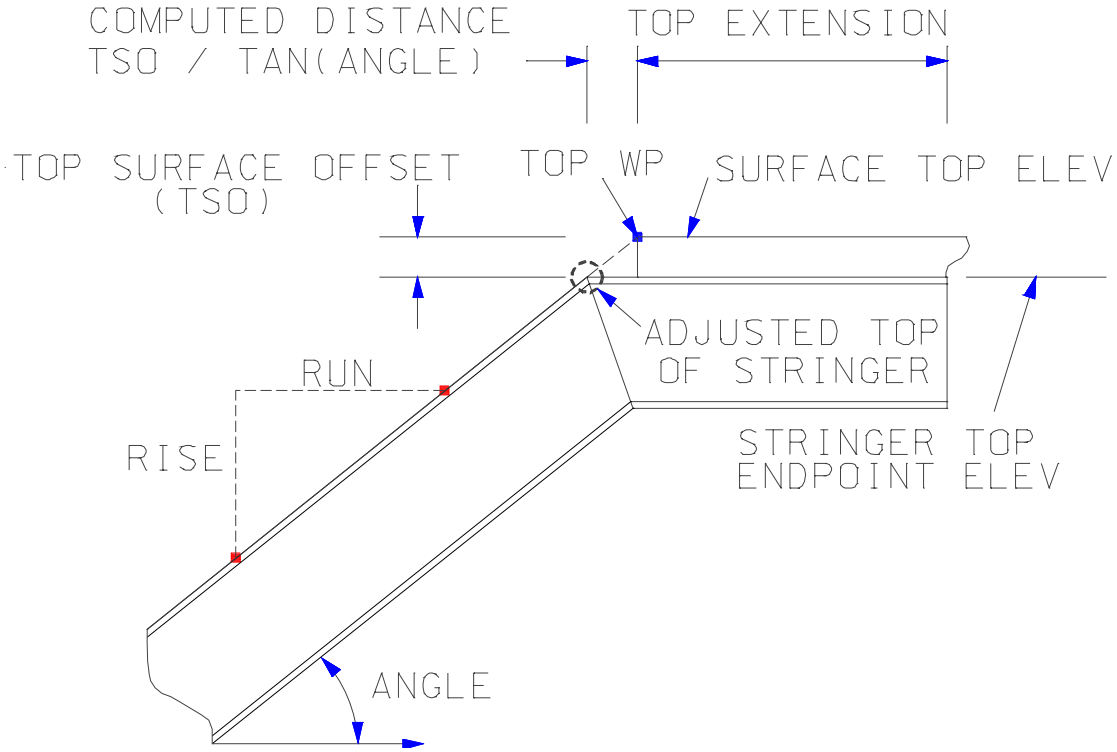


Figure 7: Top Surface Offset

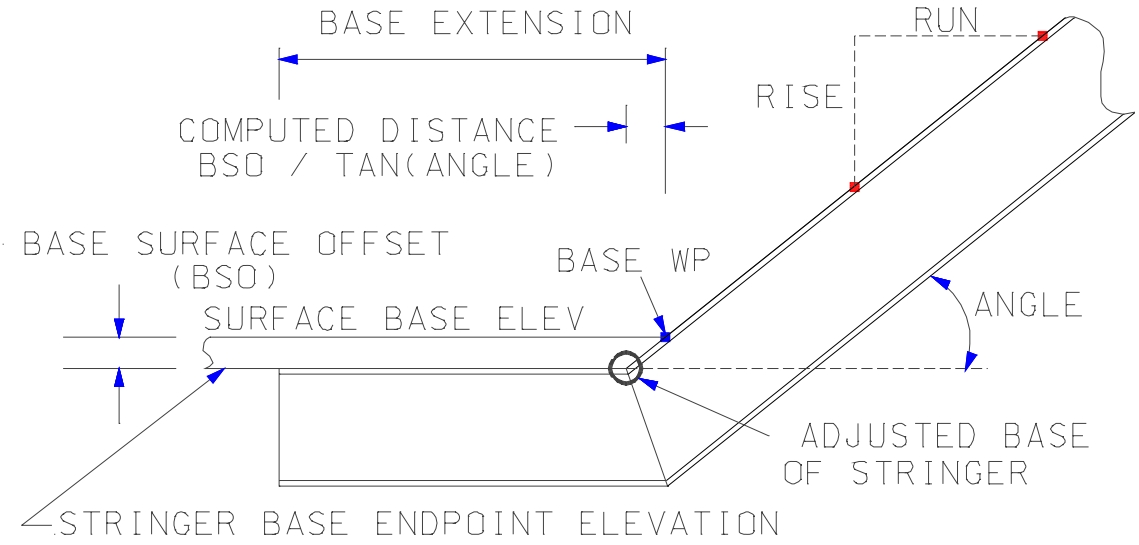


Figure 8: Base (Bottom) Surface Offset

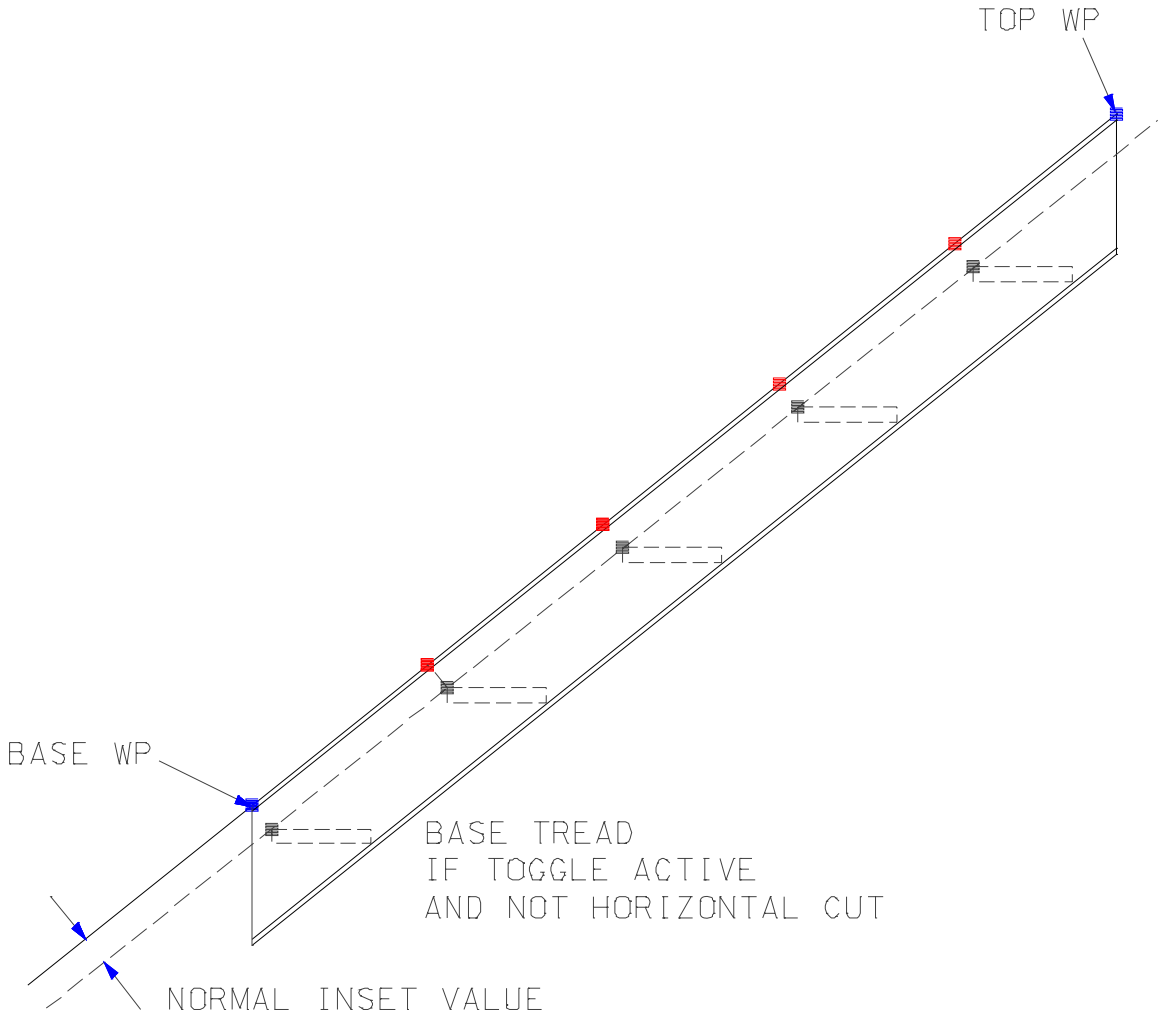


Figure 9: Normal Inset (Inset Handling Option 0)
Treads are Moved - Stringer Stationary

This is the original inset capability provided with the STAIRS application. There are some potentially undesirable features with this technique. The rise and run from the last tread to the top workpoint does not match the other treads. The configuration show has a vertical cut base where a base tread can be optionally provided (this is also true with miter cut and base w/ landing). When a base tread is provided, all treads except the top tread to top WP have equal rise & run. However the base must be handled carefully as the base tread location is shifted from the base WP. Generally speaking this is probably not a good inset technique. It works fine for visual purposes with a small inset value (DesignReview), but is not too realistic.

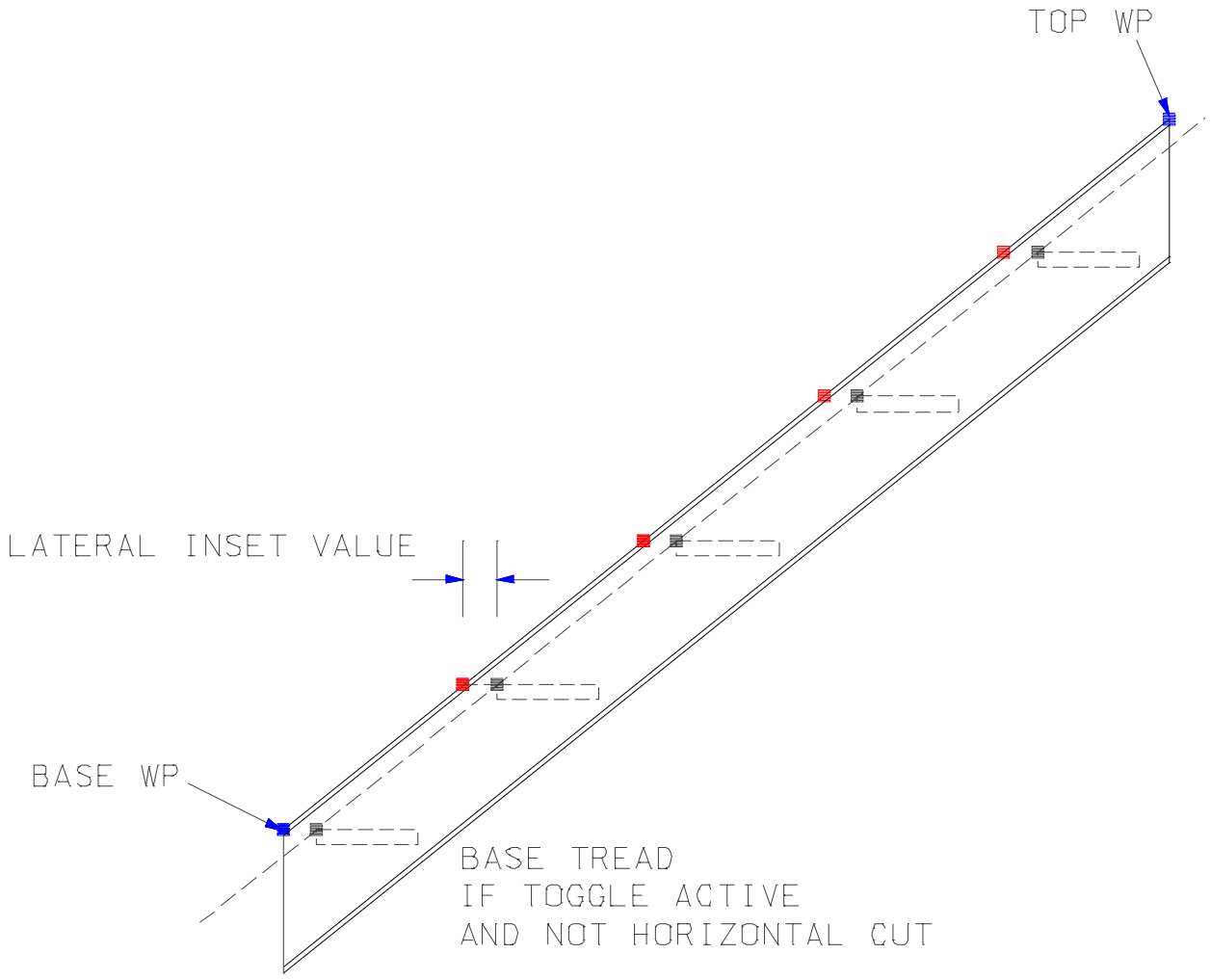


Figure 10: Horizontal Inset (Inset Handling Option 1)
Treads are Moved - Stringer Stationary

The horizontal inset option was added to version 2.0.7 of the STAIRS application. This generally produces a much more acceptable inset. The rise for all treads is maintained. Both the base and the top horizontal runs are different from the interior. At the base the run is lengthened, however if the base is a surface this is not significant. The run at the top is shortened and this can only be rectified by shifting the actual top landing if equal run distances are to be maintained. This is probably the best inset option assuming that the stringer CP is to lie on a line from the top WP to the bottom WP (i.e. stringer is not shifted – for shifted stringer insets see options 3, 4 & 5 in figures 12, 13 , 14 & 15).

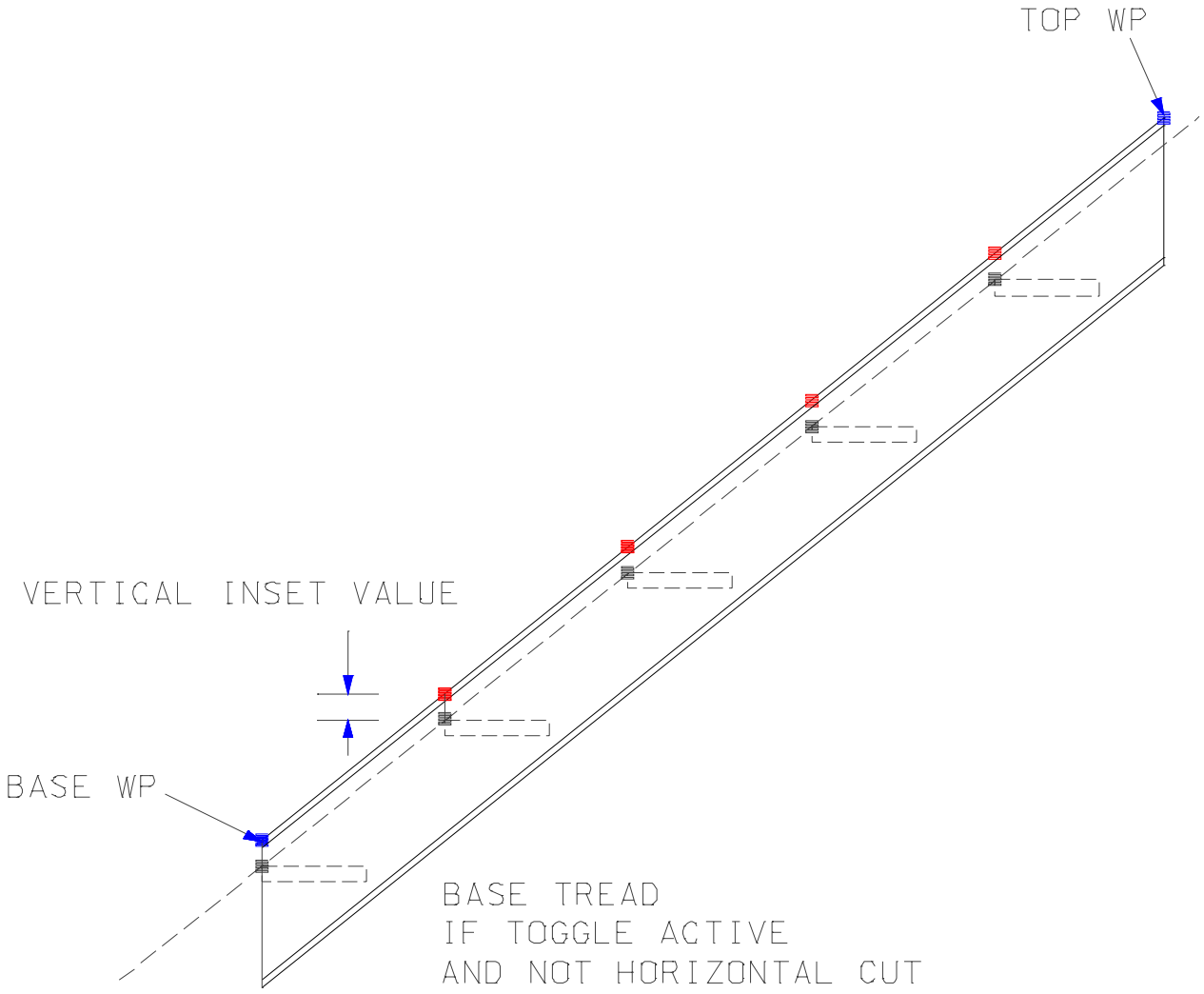


Figure 11: Vertical Inset (Inset Handling Option 2)
Treads are Moved - Stringer Stationary

The vertical inset option was added to version 2.0.7 of the STAIRS application. The run for all treads is maintained. Both the base and the top rise are different from the interior. At the base the rise is shortened. The rise at the top is lengthened. This is probably not a good inset option and is provided only for completeness.

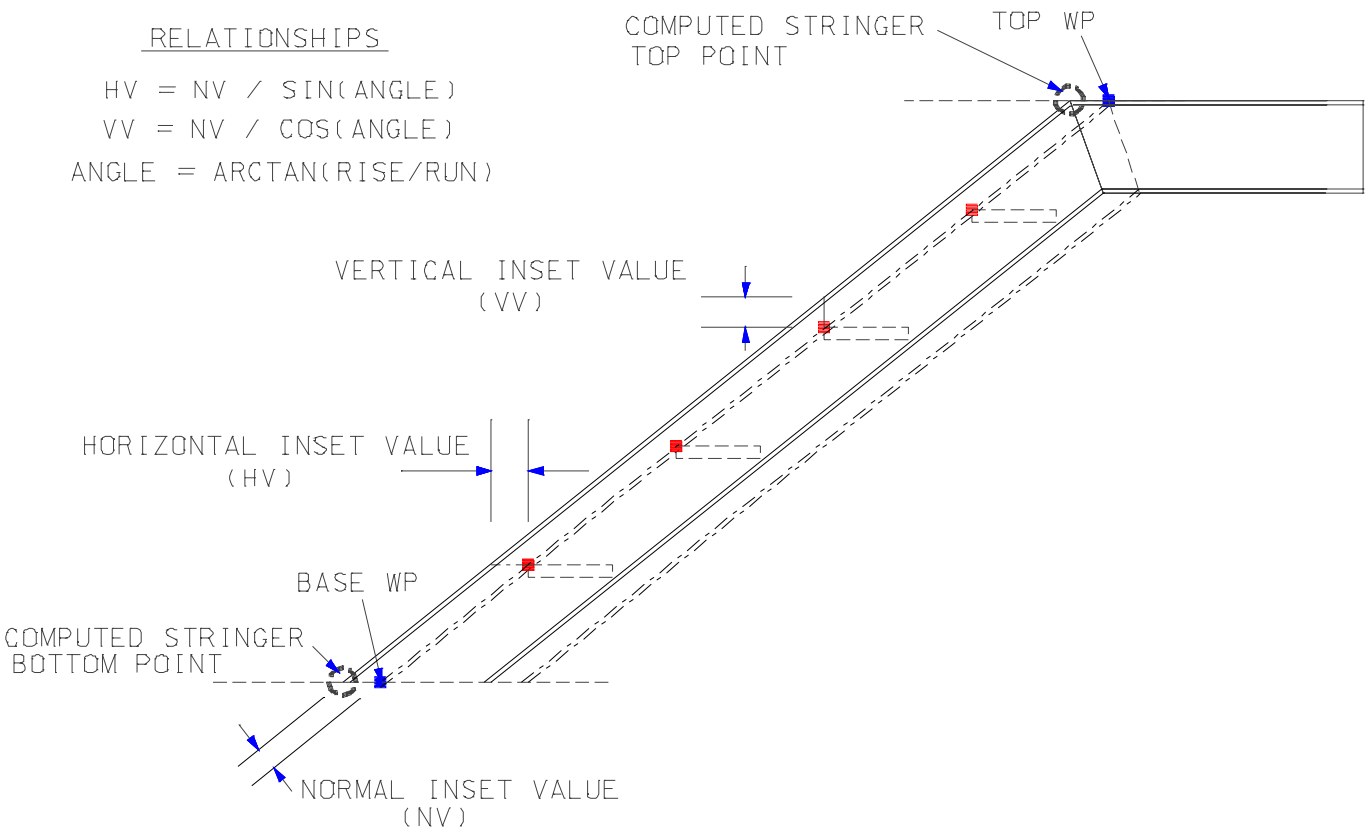


Figure 12: Inset w/ Stringer Shift - Horizontal Shift @ Both Ends
(Inset Option: Normal-3, Horizontal-4 or Vertical-5)
Treads are Stationary - Stringer Ends Adjusted

The above illustrates the method for inset handling when horizontal shifting at both ends is requested. The inset may be specified as normal, vertical or horizontal. No matter which method is used for specification, the basic movement is very similar. The equations relating horizontal inset value (HV), vertical inset (VV) and normal inset value (NV) are show on the figure above. For each method of specification, the offset stringer CP line is parallel to a line from the top WP to the bottom WP. This figure illustrates the case where a horizontal shift is requested for both the top and the bottom of the stringer. The amount of the horizontal shift can be computed using the equations show above (note for a horizontal inset, the shift is the value of the horizontal inset). The stringer CP line remains parallel to the line from the top WP to the bottom WP, it is simply shifted to accommodate the inset. The tip (CP lines) for all treads lie on a line from the top WP to the base WP. The rise and run from the top to the bottom remains consistent. Horizontal top shift is a good option for top landing situations. A base horizontal shift is good base option when base has a landing or a horizontal cut. This figure is typical of the case where there is no top or bottom surface offsets (note that in the case above WP's are at original stringer ends – see figures 7 & 8). If top and or bottom surface offsets are specified, neither the original stringer endpoint nor the shifted stringer endpoint is on a horizontal line through the top or bottom WP. If a surface offset is involved, the new stringer endpoint location is located by shifting the end in question laterally along the stringer top and/or base endpoint elevation line an amount equal to the lateral offset value (specified or computed).

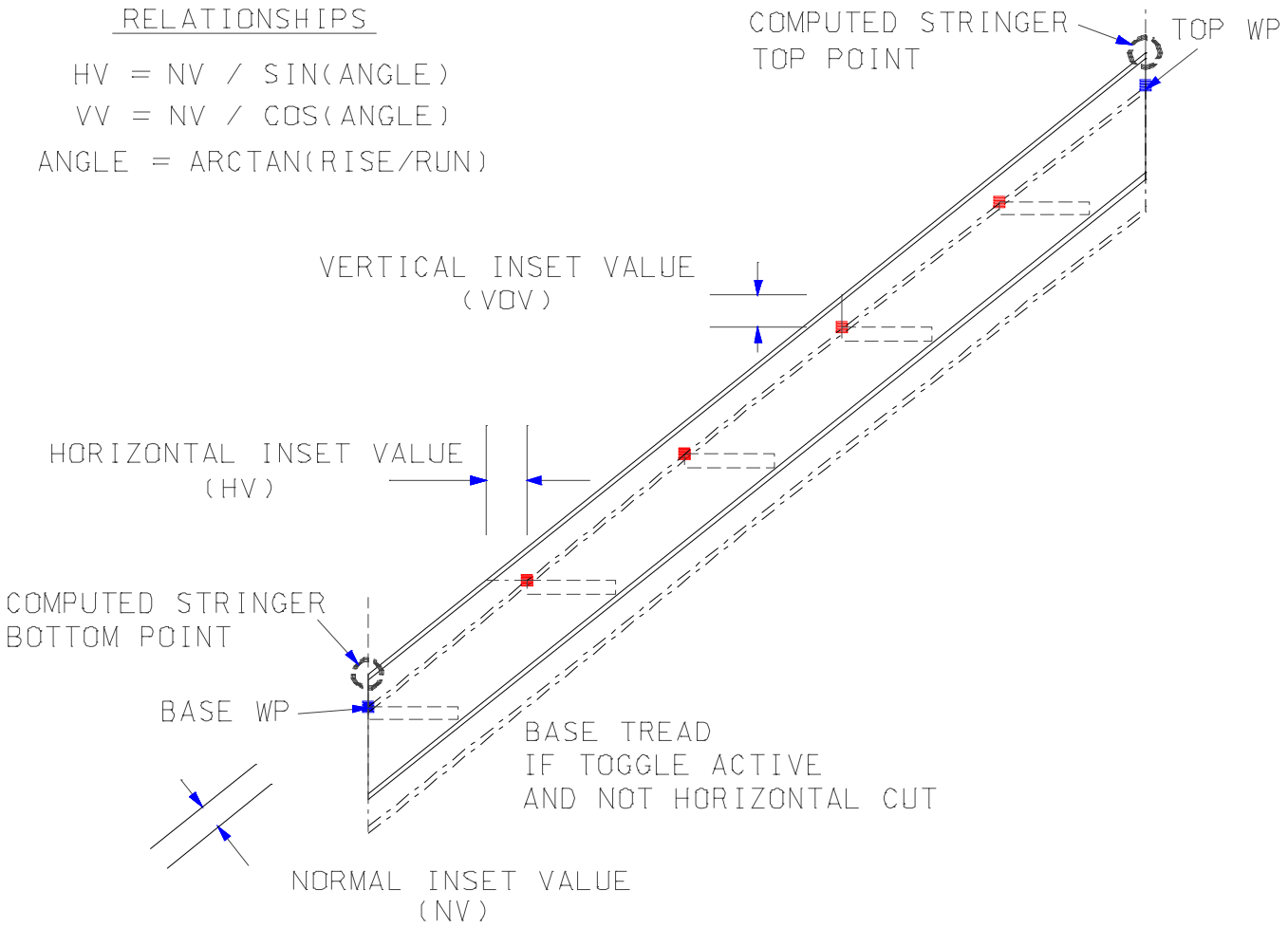


Figure 13: Inset w/ Stringer Shift - Vertical Shift @ Both Ends
(Inset Option: Normal-3, Horizontal-4 or Vertical-5)
Treads are Stationary - Stringer Ends Adjusted

The above illustrates the method for inset handling when vertical shifting at both ends is requested. The inset may be specified as normal, vertical or horizontal. No matter which method is used for specification, the basic movement is very similar. The equations relating horizontal inset value (HV), vertical inset (VV) and normal inset value (NV) are show on the figure above. For each method of specification, the offset stringer CP line is parallel to a line from the top WP to the bottom WP. This figure illustrates the case where a vertical shift is requested for both the top and the bottom of the stringer. The amount of the shift can be computed using the equations show above (note for a vertical inset, the shift is the value of the vertical inset). The stringer CP line remains parallel to the line from the top WP to the bottom WP, it is simply shifted to accommodate the inset. The tip (CP lines) for all treads lie on a line from the top WP to the base WP. The rise and run from the top to the bottom remains consistent. Vertical top and/or bottom shift is a good option when there is a vertical (flush) cut end present. This figure is typical of the case where there is no top or bottom surface offsets (note that in the case above WP's are at original stringer ends – see figures 7 & 8). If top and or bottom surface offsets are specified, neither the original stringer endpoint nor the shifted stringer endpoint is on a vertical line through the top or bottom WP. If a surface offset is involved, the new stringer end location is located by shifting the end in question vertically along a line through the original stringer endpoint and shifting an amount equal to the vertical offset value (specified or computed).

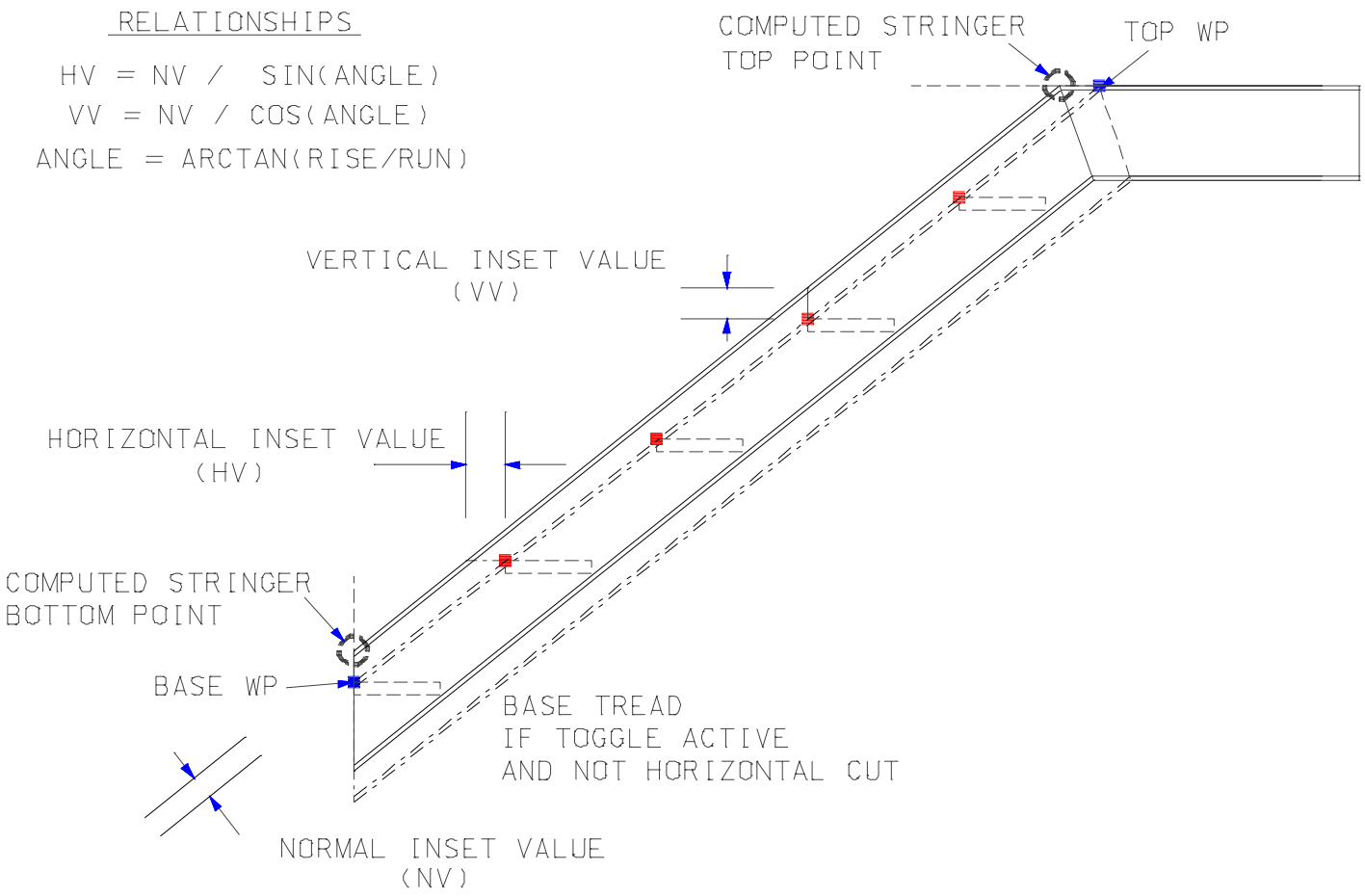


Figure 14: Inset w/ Stringer Shift - Horizontal @ Top & Vertical @ Base
(Inset Option: Normal-3, Horizontal-4 or Vertical-5)
Treads are Stationary - Stringer Ends Adjusted

The above illustrates the method for inset handling when horizontal shift is requested at the top and vertical shift is requested at the base. The inset may be specified as normal, vertical or horizontal. No matter which method is used for specification, the basic movement is very similar. The equations relating horizontal inset value (HV), vertical inset (VV) and normal inset value (NV) are show on the figure above. For each method of specification, the offset stringer CP line is parallel to a line from the top WP to the bottom WP. This figure illustrates the case where a horizontal shift is requested at the top and a vertical shift is requested at the bottom of the stringer. The amount of the shift can be computed using the equations show above. The stringer CP line remains parallel to the line from the top WP to the bottom WP, it is simply shifted to accommodate the inset. The tip (CP lines) for all treads lie on a line from the top WP to the base WP. The rise and run from the top to the bottom remains consistent. . Horizontal top shift is a good option for top landing situations. Vertical base shift is a good option when there is a vertical (flush) cut end at the base. This figure is typical of the case where there is no top or bottom surface offsets (note that in the case above WP's are at original stringer ends – see figures 7 & 8).

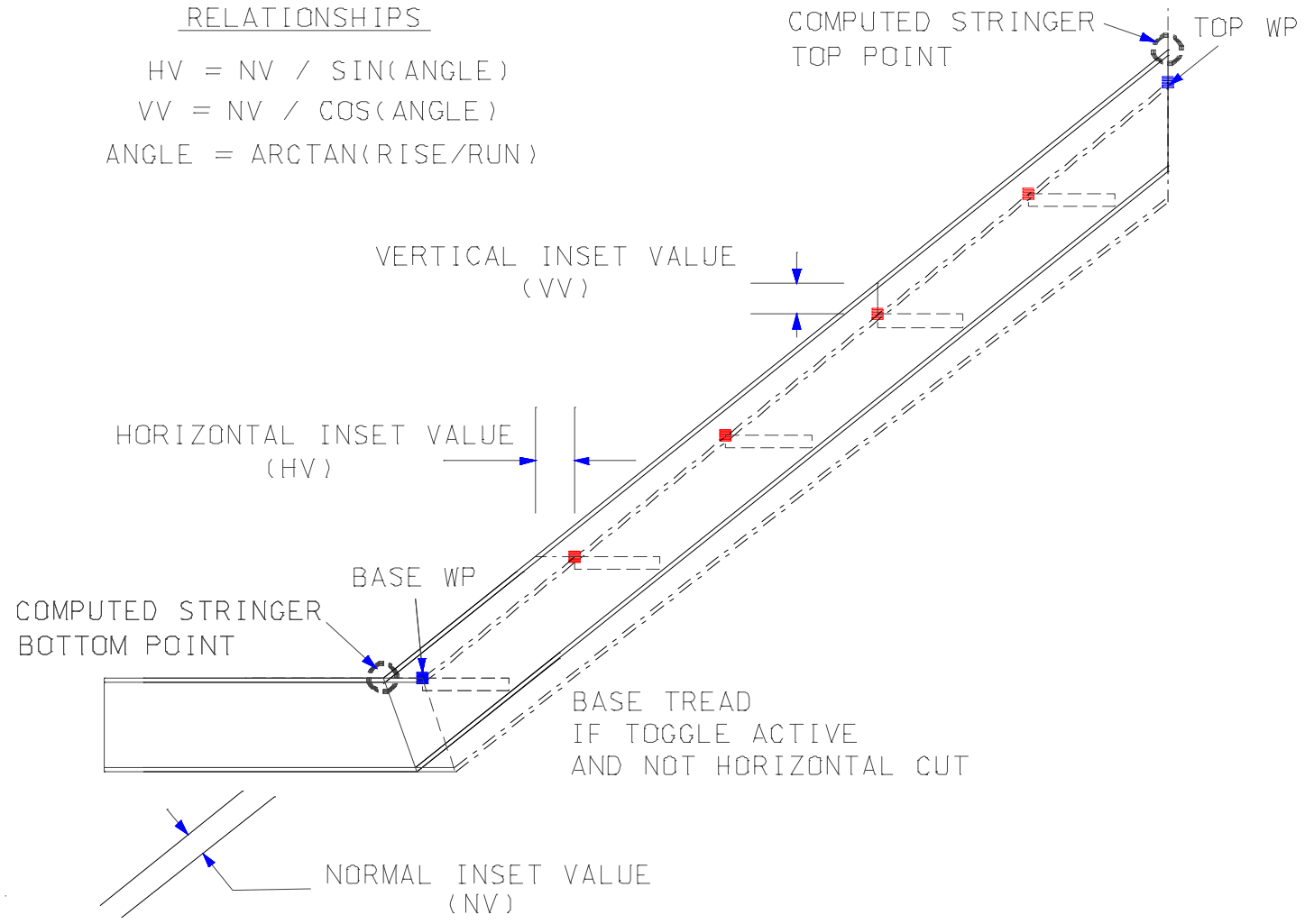


Figure 15: Inset w/ Stringer Shift - Vertical @ Top & Horizontal @ Base
(Inset Option: Normal-3, Horizontal-4 or Vertical-5)
Treads are Stationary - Stringer Ends Adjusted

The above illustrates the method for inset handling when a vertical shift is requested at the top and a horizontal shift is requested at the base. The inset may be specified as normal, vertical or horizontal. No matter which method is used for specification, the basic movement is very similar. The equations relating horizontal inset value (HV), vertical inset (VV) and normal inset value (NV) are show on the figure above. For each method of specification, the offset stringer CP line is parallel to a line from the top WP to the bottom WP. This figure illustrates the case where a vertical shift is requested at the top and a horizontal shift is requested at the base of the stringer. The amount of the shift can be computed using the equations show above. The stringer CP line remains parallel to the line from the top WP to the bottom WP, it is simply shifted to accommodate the inset. The tip (CP lines) for all treads lie on a line from the top WP to the base WP. The rise and run from the top to the bottom remains consistent. Vertical shift is a good option when there is a vertical (flush) cut end at the top. A base lateral shift is good base option when base has a landing or a horizontal cut. This figure is typical of the case where there is no top or bottom surface offsets (note that in the case above WP's are at original stringer ends – see figures 7 & 8).

ACE FWP Stairs Documentation

Handrail Notes

There are numerous configurable handrail options. Some of the options are:

Handrail Location on Stairs

- Handrail may be on both right and left
- Handrail may be on right side only
- Handrail may be on left side only
- Handrail may be omitted entirely

Handrail Mounting

- Handrail may be mounted on top of stringer (figure 16)
- Handrail may be mounted on side of stringer (figure 17)

Rail CP location

- Rail CP location may be top
- Rail CP location may be center
- Rail CP location may be bottom for an "L" profile

Handrail Profile

- Handrail may be Pipe, Bar or Tube profiles
- Handrail may be "L" (angle) profiles

(Note if handrail is "L", all handrail components must be "L" profile or a warning message is displayed)

Rail Options

- There may be single a single top rail
- There may be a top rail and a midrail
- There may be a top rail, mid rail and a third rail.
- Midrail (2^{nd} and 3^{rd}) may have different profile, class, grade or namedgroup from top rail
- A Handhold rail may be specified in the definition file (see page 23)

Post Location and Spacing Options

The following variables control post spacing:

base_post_dis	distance from bottom stringer tip to first post
top_post_dis	distance from top stringer tip to last post
max_hr_space	maximum distance for post spacing

1. base_post_dis & top_post_dis both greater than or equal to zero

base_post_dis & top_post_dis will locate the first and last posts respectively (Normally both of the variables will be greater than 0, however 0 may be specified). The interior posts will be equally spaced between the first and last. The post spacing will not exceed max_hr_space. A small post connecting top rail & midrails is provided at extreme ends if a full post is not provided. A full post at the extreme ends is generally undesirable but attainable.

2. base_post_dis less than zero

For this case the top_post_dis >= 0. The top post is located and posts are then spaced at max_hr_space toward the bottom. A small post connecting top rail & midrails is provided at extreme ends if a full post is not provided.

3. top_post_dis less than zero

For this case the base_post_dis >= 0. The base post is located and posts are then spaced at max_hr_space toward the top. A small post connecting top rail & midrails is provided at extreme ends if a full post is not provided.

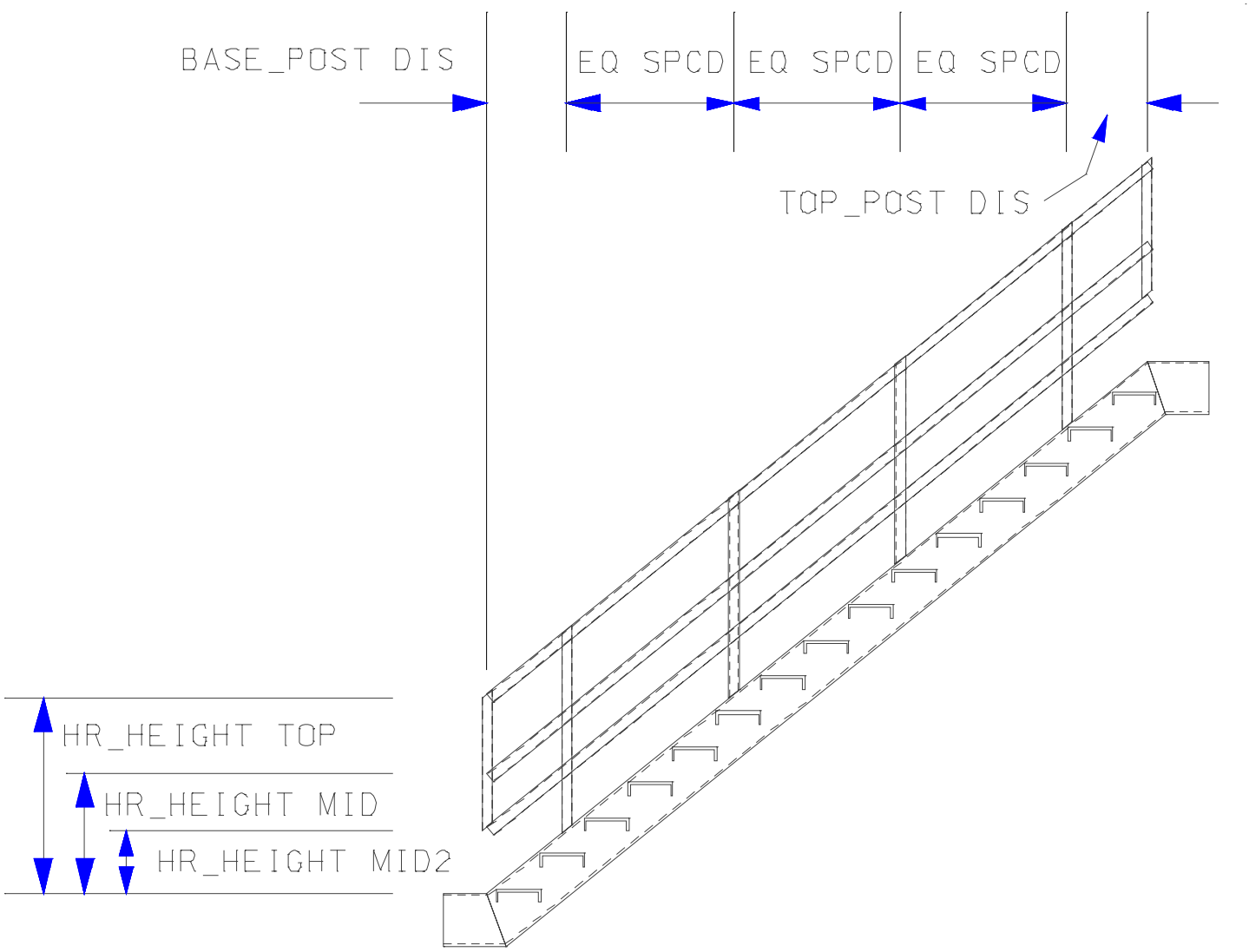


Figure 16 Handrail Mounted on Top of Stringer (Pipe Handrail w/ Top CP)

The above figure shows the situation where the handrail is mounted on top of the stringer. In the figure above, both a start base post position (BASE_POST_DIS) and a start top post position (TOP_POST_DIS) has been specified. This is the most normal case. The interior HR posts are spaced evenly between the base and top post at a distance not to exceed MAX_HR_SPACE. If the BASE_POST_DIS or the TOP_POST_DIS is specified as less than 0.0, the following special handling occurs. A post is located at the base or top (whichever has a positive value – both cannot be negative) and the remainder of the posts are spaced at MAX_HR_SPACE.

Note that the handrail is tied to the stringer endpoint and not necessarily the top and base WP's. The stringer endpoints will match the WP's if 1) the top and base surface offsets are 0.0 and 2) the inset is zero or uses methods 0, 1 or 2.

From 1 to 3 rails may be specified. Set HR_HEIGHT_MID2=0.0 for two rails. Set HR_HEIGHT_MID=0.0for 1 rail. The rail CP may be specified as top(8) or center(5). In this case, top was specified.

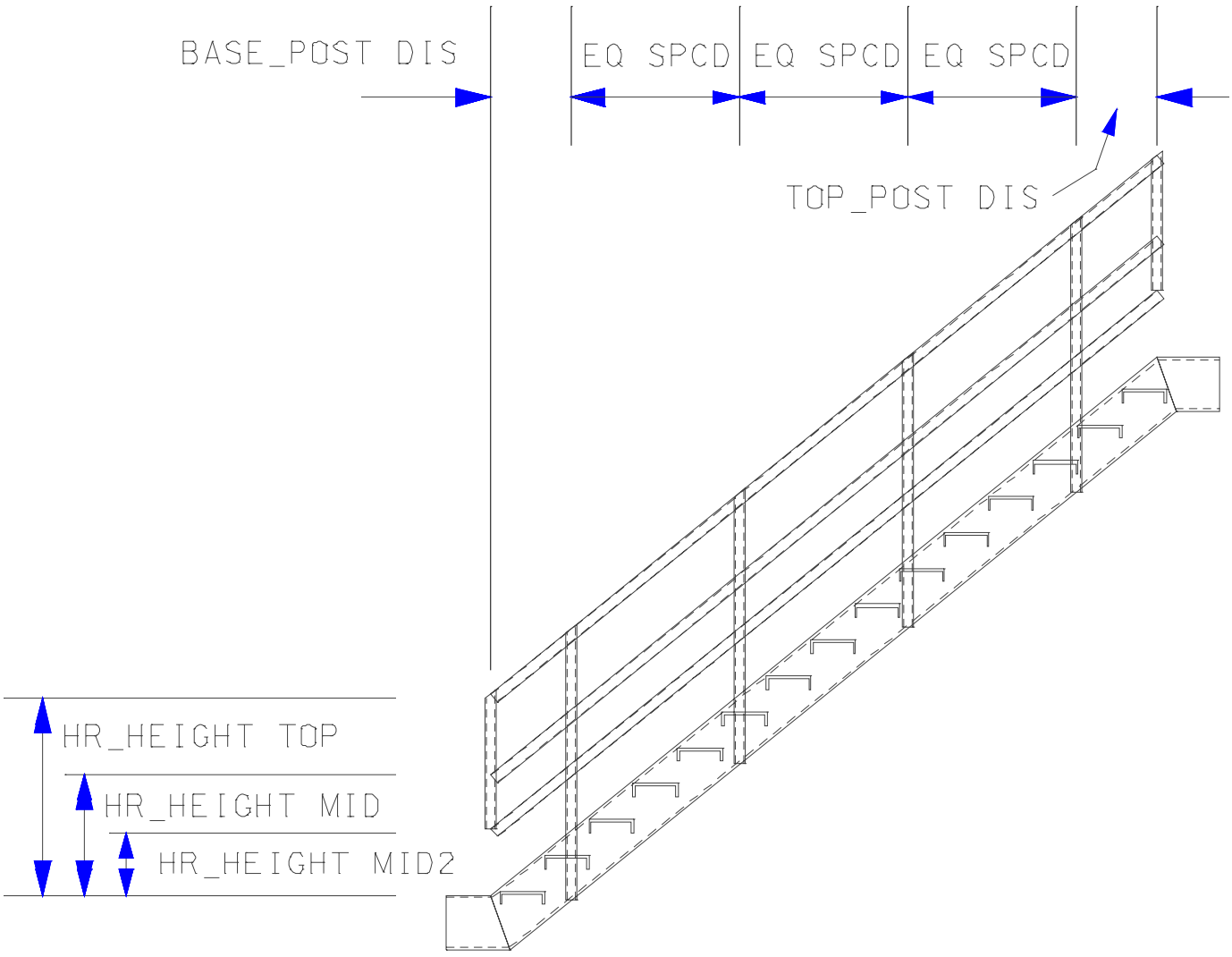


Figure 17 Handrail Mounted on Side of Stringer (Pipe Handrail w/ Top CP)

The above figure shows the situation where the handrail is mounted on side of the stringer. In the figure above, both a start base post position (BASE_POST_DIS) and a start top post position (TOP_POST_DIS) has been specified. This is the most normal case. The interior HR posts are spaced evenly between the base and top post at a distance not to exceed MAX_HR_SPACE. If the BASE_POST_DIS or the TOP_POST_DIS is specified as less than 0.0, the following special handling occurs. A post is located at the base or top (whichever has a positive value – both cannot be negative) and the remainder of the posts are spaced at MAX_HR_SPACE. Posts should not be placed at the base endpoint for the side mounted condition.

Note that the handrail is tied to the stringer endpoint and not necessarily the top and base WP's. The stringer endpoints will match the WP's if 1) the top and base surface offsets are 0.0 and 2) the inset is zero or uses methods 0, 1 or 2.

From 1 to 3 rails may be specified. Set HR_HEIGHT_MID2=0.0 for two rails. Set HR_HEIGHT_MID=0.0 for 1 rail. The rail CP may be specified as top(8) or center(5). In this case, top was specified.

ACE FWP Stairs Documentation

Definitions File

Due to the dissimilar nature of the variables in the steel utilities, each steel utility has a separate definition (DEF) file. While each file is distinctly different, each file is similar in the basic method of definition. Each definition file may optionally be controlled with either of two environment variables. Thus a project specific definition file for each project may be easily specified. The environment variables may be specified in numerous ways (similar to any MicroStation variable), however the utilization of a project.pcf is highly recommended. Environment variable definition is discussed in detail in the installation notes provided with the ACE FrameWorks utilities. The default name and location for the definition file for this utility are: C:\ACE_ST.DEF. A directory for the definition file may be specified with the environment variable ACE_DEF_PATH (will look in specified path for file ACE_ST.DEF). A complete name and location of a definition file may be also specified with the environment variable ACE_ST_DEF. The first valid definition file found is utilized. The search for a definition file happens in the following order or priority:

1. If the variable ACE_ST_DEF is specified, the named file at this location will be used if found.
2. If the variable ACE_DEF_PATH is specified and ACE_ST.DEF is found in this directory, it is used.
3. If there is a c:\ace_st.def file it is utilized.
4. If none of the above, internal program defaults are utilized – a warning message will be displayed.

(if environment variables in 1 and/or 2 above are specified and corresponding DEF file is not found, a warning is displayed)

A sample default file is provided in later sections of this document. Toward the end of this document, the commands for the definition file are outlined in detail.

The steel definition files allow the specification of units (Metric (meters or mm) or English (ft or in)). Thus a given default file may be utilized in either a Metric or English project. The units may be changed throughout the definition file. If units are not specified, it is assumed that the definition file units match the units of the model (feet/in-English & meters/mm-Metric). If units are defined and they do not match the model, the variables after the units command are converted to match the model units.

Sample Definitions File

Typical Definition File :

```
UNIts  ENGlish FEEt
STD   2.5   .67   .83   .0415  .0833  2.0   2   2    1  75  7.5   1   1   .0833 2.5
LIM   .5          1.25    .5        1.25      20      70      3
STR   C12X25            1       A36      VB
TRE   "C10X30"          2       A36      VB
HRR   "P1 1/2STD"       3       A36      VB
HRM   "P1 1/2STD"       6       A36      VB
HRP   "P1 1/2STD"       4       A36      VB
HRS   2.75        1.25    0.0      3.75      2.0      1.5      1      1
ENV   8  ACCESS  2
NGP   3     5     2     1     6     7
NAME  DYNamic     S200_
```

Note that section profiles for the STR, TRE, HRR, HRM & HRP commands may be surrounded in quote marks (i.e. "C12X25"). If the profile name has blank spaces, the section profile must be surrounded by quote marks (i.e. "P1 1/2STD").

ACE FWP Stairs Documentation

Stairs Definition File

Command Definition

- **Valid Primary Keyword Commands : (UNI, STD, LIM, STR, TRE, HRR, HRM, HRP, HRS, ENV, NGP, NAME, PSC)**
- **Each record must begin with a valid primary keyword or it is ignored**
- **All records that start with a blank are considered comments**
- **The commands/keywords (records) may be placed in any order however the order is significant**
- **All values for a given command must be defined in order shown above. If default values are acceptable, only the changed values must be given. However all values up to that point must be defined whether changed or not.**
- **The components of a given command (record) must all be present and in the order shown**
- **The units command is special and may be repeated and located as required. While commands may be in any order, it should be obvious that the location of the units command is extremely important.**
- **By default application looks for C:\ACE_ST.DEF definition file**
- **Definition file path may be defined with environment variable ACE__DEF_PATH**
- **ACE_DEF_PATH=d:\mydir**
  (the DEF file ACE_ST.DEF will be looked for in the directory d:\mydir)
- **Definition file may be defined with environment variable ACE_ST_DEF**
- **ACE_ST_DEF = d:\mydir\mydef_file *(highest priority definition)***
  (the DEF file mydef_file will be looked for in the directory d:\mydir)
- **NOTE : Components shown in bold may only be specified in the definitions file**

UNIT Command - Units Command (optional command)

UNIT {UNITTYPE} {UNIT}

where :

{UNITTYPE} May be ENGlish (feet) or METric (meters).
If units is not specified it is assumed that the units match the current model units.

{UNIT} Must be FEEt or INCh for ENGlish (default feet) or must be METer or MM for METric (default meters).
If unit is not specified, it is assumed that the units are feet for English & meters for metric.

Units may be changed at any time but be aware that the properties (in attached library) for the member specified must match the current model units.

STR Command - Stringer Command defines stringer parameters

STR str_section str_class str_grade str_type

where :

str_section **:** Stringer section (default value - C10X30 or "C10X30")
Section profile name may be enclosed in quotes
If name includes spaces, quotes must be utilized (i.e. "P1 1/2STD")

str_class **:** Stringer class valid options 0 - 9 (default value - 6)

str_grade **:** Stringer grade any valid FrameWorks grade (default value - A36)

str_type **:** FrameWorks member type - valid options : VB, HB, BE, CO (default VB)

TRE Command - Tread Command defines tread parameters

TRE tre_section tre_class tre_grade tre_type

where :

tre_section **:** Tread section (default value - Std_Tread)
Section profile name may be enclosed in quotes
If name includes spaces, quotes must be utilized (i.e. "P1 1/2STD")

tre_class **:** Tread class valid options 0 - 9 (default value - 6)

tre_grade **:** Tread grade any valid FrameWorks grade (default value - A36)

tre_type **:** FrameWorks member type - valid options : VB, HB, BE,CO (default VB)

ACE FWP Stairs Documentation

Stairs Definition File (continued)

HRR Command - Handrail Rail Command defines rail

HRR hrr_section hrr_class hrr_grade hrr_type

where

hrr_section : Section size for handrail rail (Pipe or Angle) (default value - P2STD)
Section profile name may be enclosed in quotes
If name includes spaces, quotes must be utilized (i.e. “P1 1/2STD”)

hrr_class : The class for the handrail rail (0 =< class < 10) (default value - 6)

hrr_grade : The grade for the handrail rail (i.e. A36 etc) (default value - A36)

hrr_type : The type for the handrail rail (BE,CO,VB,HB) (default VB)

HRP Command - Handrail Post Command defines post

HRP hrp_section hrp_class hrp_grade hrp_type

where

hrp_section : Section size for handrail post (Pipe or Angle) (default value - P2STD)
Section profile name may be enclosed in quotes
If name includes spaces, quotes must be utilized (i.e. “P1 1/2STD”)

hrp_class : The class for the handrail post (0 =< class < 10) (default value - 6)

hrp_grade : The grade for the handrail post (i.e. A36 etc) (default value - A36)

hrp_type : The type for the handrail post (BE,CO,VB,HB) (default VB)

HRM Command - Handrail Midrail Command defines midrail(s)
(OPTIONAL COMMAND IF NOT PRESENT - HRR USED FOR ALL RAILS)

HRM hrm_section hrm_class hrm_grade hrm_type

where

hrm_section : Section size for handrail midrail (Pipe or Angle) (default P2STD)
Section profile name may be enclosed in quotes
If name includes spaces, quotes must be utilized (i.e. “P1 1/2STD”)

hrm_class : The class for the handrail midrail (0 =< class < 10) (default 7)

hrm_grade : The grade for the handrail midrail (i.e. A36 etc) (default A36)

hrm_type : The type for the handrail midrail (BE,CO,VB,HB) (default VB)

LIM Command - Limits Command defines stair limits

LIM min_trd_rise max_trd_rise min_trd_run max_trd_run min_angle max_angle min_num_treads

where

min_trd_rise : Minimum tread rise value (default .5 ft)

max_trd_rise : Maximum tread rise (default 1.5 ft)

min_trd_run : Minimum tread run value (default .5 ft)

max_trd_run : Maximum tread run (default 1.5 ft)

min_angle : Minimum stair angle in degrees (default 20 deg)

max_angle : Maximum stair angle in degrees (default 70 deg)

min_num_treads : Minimum number of treads (default - 3)

ACE FWP Stairs Documentation

Stairs Definition File (continued)

HRH Command - HandHold Rail Command defines optional handhold rail

(OPTIONAL COMMAND IF EITHER HRH or HHD NOT PRESENT - HRH NOT PROVIDED)

HRH hrh_section hrh_class hrh_grade hrh_type

where

hrh_section : Section size for handhold rail (Pipe or Angle) (default value - P2STD)
Section profile name may be enclosed in quotes
If name includes spaces, quotes must be utilized (i.e. “P1 1/2STD”)

hrh_class : The class for the handhold rail (0 =< class < 10) (default value - 6)

hrh_grade : The grade for the handhold rail (i.e. A36 etc) (default value - A36)

hrh_type : The type for the handhold rail (BE,CO,VB,HB) (default VB)

Note: A Pipe section is recommended for the Handhold rail profile, however any profile may be utilized.

HHD Command - Handhold Data Command defines handhold data

(OPTIONAL COMMAND IF EITHER HRH or HHD NOT PRESENT - HRH NOT PROVIDED)

HHD hrh_height hrh_offset hrr_cp hrh_sup

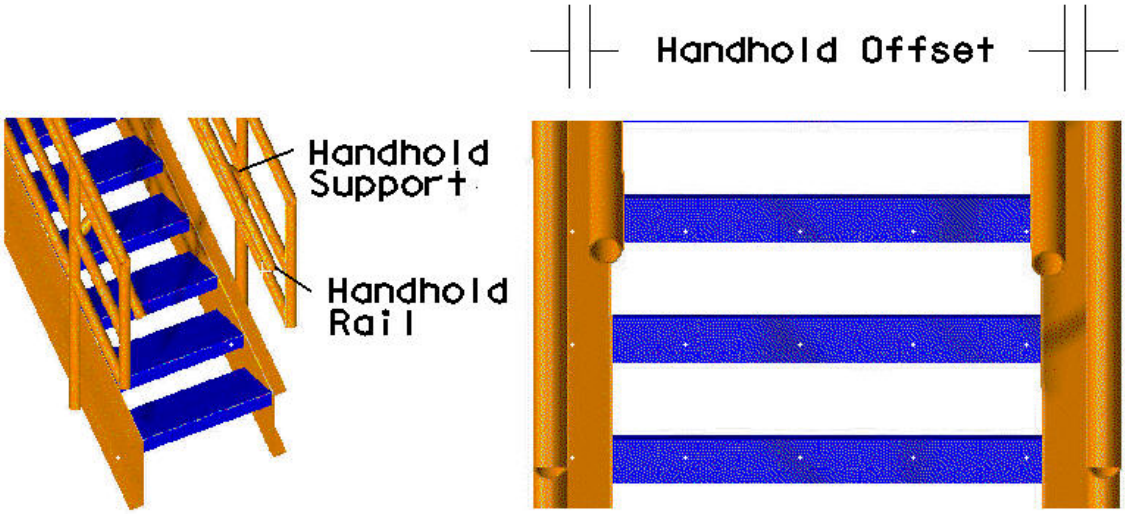
where

hrh_height : Distance from top of stringer to CP of handhold rail (default 0.0 ft)

hrh_offset : Handhold Offset **(**clearance between post edge and handhold rail edge) (default 0.0 ft)

hrh_cp : Cardinal Point of handhold rail (default 0-center)
0 - For non L profile CP = 4 left side & 6 right side
0 - For L profile CP = 4 both sides (center placement)
1 - For non L profile CP = 7 left side & 9 right side (top placement)
1 - For L profile CP = 7 both sides (bottom placement)
2 - For non L profile CP = 7 left side & 9 right side (top placement)
2 - For L profile CP = 1 both sides (top placement)
NOTE: a hrh_cp=2 is the recommended setting as this value provides accurate TOS for all profiles at any stair slope.

hrh_sup : Support from hrh to post (1 - place (default) 0 - do not place)
The handhold support profile is the same as the handhold rail.
The support rail is place at a 45 degree angle from handhold rail CL to post CL using CP 5
Both ends have a vertical cut. (If post is an “L” profile, support placed to the face of “L”)
(A Support is NOT placed with “L” handhold rail)



Handhold Rail Images

ACE FWP Stairs Documentation

Stairs Definition File (continued)

STD Command - Standards Command defines stair standards

STD std_stair_width std_tread_rise std_tread_run inset top_offset top_extension stair_def extension_opt interference_opt iwarn envelope_height inset_opt base_opt base_offset base_extension

where

std_stair_width : Inside to inside width of stair stringers (default 2.5 ft)
std_tread_rise : Riser dimension for tread (default .67 ft)
std_tread_run : Tread dimension for treads (default .83 ft)
inset : Tread inset dimension (default 0 inches)
top_offset : Vertical distance from WP to top of stringer - typically the floor thickness (default .083 ft)
top_extension : Top Landing width from WP to end of stringer extension (default 1.0 ft)
stair_def : Default option for stair definition (default 0)

0 - Top WP, Base Elev, Orientation w/ Std Tread Rise & Run
1 - Top WP, Base Elev, Orientation w/ Defined Tread Run
2 - Top WP, Base Elev, Orientation w/ Defined Tread Rise & Run
3 - Top Elev, Base WP, Orientation w/ Std Tread Rise & Run
4 - Top Elev, Base WP, Orientation w/ Defined Tread Run
5 - Top Elev, Base WP, Orientation w/ Defined Tread Rise & Run
6 - Define top & bottom WP's with Std Tread Rise & Run
7 - Define top & bottom WP's with Defined Tread Run
8 - Define top & bottom WP's with Defined Tread Rise
9 - Top WP, Slope, Base Elev, Orientation w/ Defined Tread Rise
10 - Top WP, Slope, Base Elev, Orientation w/ Defined Tread Run
11 - Top Elev, Slope, Base WP, Orientation w/ Defined Tread Rise
12 - Top Elev, Slope, Base WP, Orientation w/ Defined Tread Run

extension_opt : Default option for top extension (default 0)

0 - No extension – vertical (flush) cut stringer (i.e. vertical plane cut)
1 – Provide stringer landing with a miter cut at top

interference_opt : Default option for interference envelope (default 0)

0 - Interference Envelope
1 - No interference envelope

iwarn : Member placement limit without onscreen warning (default 50) (if number of members exceeds or equals iwarn - a warning is displayed before placement)

envelope_height : Access envelope height (default 0)
Vertical distance from TOS riser to top of access envelope

inset_opt : Method for inset handling (default 0)

0 – inset is normal from CP line (stringer NOT moved)
1 – inset is horizontal from CP line (stringer NOT moved)
2 – inset is vertical from CP line (stringer NOT moved)
3 – inset by moving stringer normal from CP line
4 – inset by moving stringer horizontal from CP line
5 – inset by moving stringer vertical from CP line
NOTE: see inset diagrams

base_opt : Default option for stringer base bevel treatment (default 0)

0 – base of stringer is horizontal cut (flush w/ floor)
1 – base of stringer is vertical (flush) cut
2 – Provide stringer landing with a miter cut at base

base_offset : Vertical distance from WP to base of stringer - typically the floor thickness (default .083 ft)
base_extension : Base Landing width from WP to end of stringer extension (default 1.0 ft)

ACE FWP Stairs Documentation

Stairs Definition File (continued)

HRS Command - Handrail standards Command defines handrail standards

HRS hr_height_top hr_height_mid hr_height_third max_hr_space base_post_dis top_post_dis hr_pos hrr_cp iWarn hr_sides

where

hr_height_top : Vertical distance from Wp line to top rail CP (default 3.0 ft)
hr_height_mid : Vertical distance from Wp line to mid rail CP (default 1.5 ft)
hr_height_third : Vertical distance from Wp line to third rail CP (default 0.0 ft)
max_hr_space : Maximum horizontal spacing for HR posts (default 4.0 ft)
base_post_dis : Horizontal distance from base WP to first post (default 2.0 ft)
top_post_dis **:** Horizontal distance from top WP to last post (default 2.0 ft)
hr_pos : HR on Top (0) or Handrail on Side (1) (default - 0)
hrr_cp : Cardinal Point of HRR & HRM handrail rail (default 0-center)
0 - For non L profile CP = 4 left side & 6 right side
0 - For L profile CP = 4 both sides (center placement)
1 - For non L profile CP = 7 left side & 9 right side (top placement)
1 - For L profile CP = 7 both sides (bottom placement)
2 - For non L profile CP = 7 left side & 9 right side (top placement)
2 - For L profile CP = 1 both sides (top placement)
NOTE: a hrh_cp=2 is the recommended setting as this value provides accurate TOS for all profiles at any stair slope.
iWarn : Warning message for mismatched rail & post types (0-ON 1-OFF)
hr_sides : Handrail Side of Stairs Option (default 0-both sides)
0 - Place both sides (default)
1 - Place left side only
2 - Place right side only
3 - Do Not Place Handrail

ENV Command - Envelope Command defines interference envelope parameters

ENV envclass envgrade envmaterial

where :

envclass : The class for the interference envelope (Default = 9)
(0 =< class < 10)
envgrade : The grade for the interference envelope (Default: Access)
envmaterial : The material type for the interference envelope (Default 2 (usually Aluminum)

NGP Command - Named Group Command defines namedgroups

NGP iNGP_str iNGP_tre iNGP_hrp iNGP_hrr iNGP_env iNGP_hrm

where

iNGP_str **:** Named group for stair stringer (default -1 which is none)
iNGP_tre **:** Named group for stair tread (default -1 which is none)
iNGP_hrp **:** Named group for stair handrail post (default -1 which is none)
iNGP_hrr **:** Named group for stair handrail rail (default -1 which is none)
iNGP_env **:** Named group for stiar solid envelopes (default -1 which is none)
iNGP_hrm **:** Named group for platform handrail midrail(s) (default -1 which is none)

NOTE: Namedgroups are defined globally for a project. The iNGP_xxx value is an integer value that corresponds to the index of the global namedgroups. The first namedgroup is 0, the next is 1 and so on up to a maximum integer value of the number of namedgroups minus one. If a name group does not exist for the integer value specified, the member type in question will simply not be placed in a named group. A value of –1 specifies that the member type in question is not to be put in a namedgroup. In FWP namedgroups are specified by an alpha name so be careful when selecting integers. **SOLID NAMEDGROUPS ARE FUNCTIONAL with FWP version 7.0.0.17 and later**

ACE FWP Stairs Documentation

Stairs Definition File (continued)

NAME Command - Name Command defines method of naming components

NAME {NAME_OPTION} name_prefix

where

{NAME_OPTION} : Keyword - must be AUT or DYN or SPE or CON

SPEcified : Use the supplied name and append the member ID for first stringer placed
Thus each stair will have a different name
However all components of a given platform will have same name
(This is the default option with the name "STAIR")

DYNamic : At placement time will display the last name(prefix) used with following options
1) option to supply new name/prefix
2) option to append the member ID for first stringer placed
Thus each stair can have a unique name
However all components of a given stair will have same name
3) option to abort placement of stair

AUTo : FrameWorks assigns names by type and sequence number
(name_prefix not required or utilized)

CONstant : Use this name for all stair placed for all components

PSC Command - Property Specification Case Command defines PSC file

(optional command to define variable stair profile specification file)

PSC sPSCfile

where

sPSCfile : Name of PSC (Property Specification Case) file
Name includes file path, name and extension
Optional command:

Note 1: If command not present, definition file is used for component properties (riser & tread profiles may be interactively defined)

Note 2: If command is present, PSC file is studied for valid case. If valid case is found, the component properties for the valid case is utilized. (the PSC file may be "turned off" interactively)

ACE FWP Stairs Documentation

PSC Files

A file termed a "Property Specification Case file" can be utilized to specify properties for Stairs application. A Property Specification Case file has a "PSC" file extension. This file can be utilized to specify stringer, tread, handrail rail, mid-rail, & post properties for Stairs application (including profile, class, grade, type & named group) as a function of length. For the Stairs application, length is the distance from the top WP to the base WP. The length may be total length or horizontal projected length.

This file conditionally defines a set of cases, which are essentially conditions with a corresponding set of member properties for Stairs. When a Stair is being processed, the cases are studied starting with the first case and proceeding downward through the file. The first case that applies will dictate the properties used for the corresponding application. If none of the cases are valid, the properties specified in the definition file, ACE_ST.DE, are utilized.

The case structure is initiated with a CASE statement that specifies a length condition. The CASE statement is followed by property definitions (valid property keywords). The CASE is terminated with the ENDCase command. A typical CASE statement would look as follows:

```
CASE "Sample Case 1"      LEN LT   10.0
HRR  "P1 1/2STD"    3     A36 VB    3
HRM  "P1 1/2STD"    6     A42 VB    3
HRP  "P1 1/2STD"    4     A36 VB    3
ENDC
```

Note that if a property is not specified, the value specified in the definition file is utilized. The PSC feature is be toggled off interactively. If toggled off, the properties specified in the definition file will be utilized. In this case (PSC feature is be toggled off), the stringer and/or tread may be optionally keyed-in.

CASE statements are studied from the first to the last. The first case to be TRUE is utilized for FWP member placement. Thus the order of case statements is very important. There can be from 1 to 10 case statements in a PSC file

A sample PSC files for the Stairs application is shown at the end of this document.

ACE FWP Stairs Documentation

Property Specification Case File Format

PSC File - Command Definition

- **Has Only Three Valid Primary Keyword Commands: (CASE, UNIT & ENDC)**
- **Valid Property Keyword Commands: (HRR, HRP, HRM, STR & TRE)**
- **NOTE: HRR, HRP, HRM, STR & TRE commands all specify named group**
- **A minimum of 1 CAS command is required, a maximum of 10 CAS commands are permitted**
- **All records that start with a blank (or incomplete) are considered comments**
- **The components of a given command (record) must all be present and in the order shown**
- **The full path & name location of the PSC file is specified in the ACE_ST definition file**
- **The Property Specification Case file must have a .PSC extension**
- **The commands/keywords (records) may be placed in any order however the order is significant**
- **All values for a given command must be defined in order shown above. If default values are acceptable, only the changed values must be given. However all values up to that point must be defined whether changed or not.**
- **The units command is special and may be repeated and located as required. While commands may be in any order, it should be obvious that the location of the units command is extremely important.**

CASE Command - Case Command Structure

CASE "sCaseName" {Condition Statement}

where :

sCaseName: Reference name for case up to 32 characters in quotes (spaces/ blanks are allowed)

{Condition Statement} The condition may only be LEN for length

{Condition Statement} explanation

where:

LENGTH Conditions (total length-LEN or horizontal length-HLE)

LEN	EQ	fValue	
LEN	LE	fValue	
LEN	LT	fValue	
LEN	GT	fValue	
LEN	GE	fValue	
LEN	BE	fValueLow	fValueHigh
HLE	EQ	fValue	
HLE	LE	fValue	
HLE	LT	fValue	
HLE	GT	fValue	
HLE	GE	fValue	
HLE	BE	fValueLow	fValueHigh

Where:

fValue	Floating point value
fValueLow	Starting (lower) floating point value
fValueHigh	Ending (higher) floating point value
LEN	Length is total length between work points
HLE	Length is projected horizontal length between work points

When a CASE command is processed, the applicable property specifications that follow (HRR, HRP, HRM, STR & TRE) will be attributed to that case until a ENDCase command is encountered.

ACE FWP Stairs Documentation

PRIMARY KEYWORD COMMANDS

UNIT Command - Units Command (optional command)

UNIT {UNITTYPE} {UNIT}

where :

{UNITTYPE} May be ENGlish or METric
If UNIT is not specified, the units are assumed to match the current model units.

{UNIT} Must be FEEt or INCh for ENGlish (default feet) or
must be METer or MM for METric (default meters).
If {UNIT} is not specified, it is assumed that the units are feet for English & meters for metric models.

Units may be changed at any time but be aware that the properties (in attached library) for the member specified must match the current model units.

ENDC Command - End Case Command Structure

ENDC

The property keyword commands must be sandwiched between the CASE and the ENDC primary commands. Property keyword statements which are not between CASE and ENDC commands are ignored. The UNIT command may be given at any time.

ACE FWP Stairs Documentation

PROPERTY KEYWORD COMMANDS (must be between CASE and ENDC primary commands)

HRR Command - Handrail Rail Command defines rail

HRR hrr_section hrr_class hrr_grade hrr_type hrr_ngp

where

hrr_section : Section size for handrail rail (Pipe or Angle) (default value - P2STD)
Section profile name may be enclosed in quotes
If name includes spaces, quotes must be utilized (i.e. “P1 1/2STD”)

hrr_class : The class for the handrail rail (0 =< class < 10) (default value - 6)

hrr_grade : The grade for the handrail rail (i.e. A36 etc) (default value - A36)

hrr_type : The type for the handrail rail (BE,CO,VB,HB) (default VB)

hrr_ngp : Named group for stair handrail rail (default -1 which is none)

HRP Command - Handrail Post Command defines post

HRP hrp_section hrp_class hrp_grade hrp_type **hrp_ngp**

where

hrp_section : Section size for handrail post (Pipe or Angle) (default value - P2STD)
Section profile name may be enclosed in quotes
If name includes spaces, quotes must be utilized (i.e. “P1 1/2STD”)

hrp_class : The class for the handrail post (0 =< class < 10) (default value - 6)

hrp_grade : The grade for the handrail post (i.e. A36 etc) (default value - A36)

hrp_type : The type for the handrail post (BE,CO,VB,HB) (default VB)

hrp_ngp : Named group for stair handrail post (default -1 which is none)

HRM Command - Handrail Midrail Command defines midrail(s)

(OPTIONAL COMMAND IF NOT PRESENT - HRR USED FOR ALL RAILS)

HRM hrm_section hrm_class hrm_grade hrm_type hrm_ngp

where

hrm_section : Section size for handrail midrail (Pipe or Angle) (default P2STD)
Section profile name may be enclosed in quotes
If name includes spaces, quotes must be utilized (i.e. “P1 1/2STD”)

hrm_class : The class for the handrail midrail (0 =< class < 10) (default 7)

hrm_grade : The grade for the handrail midrail (i.e. A36 etc) (default A36)

hrm_type : The type for the handrail midrail (BE,CO,VB,HB) (default VB)

hrm_ngp : Named group for stair handrail mid-rail (default -1 which is none)

ACE FWP Stairs Documentation

PROPERTY KEYWORD COMMANDS (con'd)

STR Command - Stringer Command defines stringer parameters

STR str_section str_class str_grade str_type str_ngp

where :

str_section : Stringer section (default value - C10X30 or "C10X30")
Section profile name may be enclosed in quotes
If name includes spaces, quotes must be utilized (i.e. "P1 1/2STD")

str_class : Stringer class valid options 0 - 9 (default value - 6)

str_grade : Stringer grade any valid FrameWorks grade (default value - A36)

str_type : FrameWorks member type - valid options : VB, HB, BE, CO (default VB)

str_ngp : Named group for stair stringer (default -1 which is none)

TRE Command - Tread Command defines tread parameters

TRE tre_section tre_class tre_grade tre_type tre_ngp

where :

tre_section : Tread section (default value - Std_Tread)
Section profile name may be enclosed in quotes
If name includes spaces, quotes must be utilized (i.e. "P1 1/2STD")

tre_class : Tread class valid options 0 - 9 (default value - 6)

tre_grade : Tread grade any valid FrameWorks grade (default value - A36)

tre_type : FrameWorks member type - valid options : VB, HB, BE,CO (default VB)

tre_ngp : Named group for stair tread (default -1 which is none)

NOTE: Namedgroups are defined globally for a project. The xxx_ngp value is an integer value that corresponds to the index of the global namedgroups. The first namedgroup is 0, the next is 1 and so on up to a maximum integer value of the number of namedgroups minus one. If a name group does not exist for the integer value specified, the member type in question will simply not be placed in a named group. A value of –1 specifies that the member type in question is not to be put in a namedgroup. In FWP namedgroups are specified by an alpha name so be careful when selecting integers. **SOLID NAMEDGROUPS ARE FUNCTIONAL with FWP version 7.0.0.17 and later**

ACE FWP Stairs Documentation

Sample PSC File for Stairs Application

Typical PSC File w/ name *anything*.PSC (no spaces/blanks):

```
UNIT   ENG  FEEt
CASE   “Small Stair ”              LEN     LE       10
HRR    “P1 1/2STD”     3     A36   VB     2
HRM    “P1 1/2STD”     6     A42   VB     2
HRP    “P1 1/2STD”     4     A36   VB     2
STR    C9X13.4            2     A36   VB    2
ENDC
CASE   “Medium Stair”              LEN     LT       18.0
HRR    L2X2X1/4        3     A36   HB   3
HRM    L2X2X1/4        1     A36   HB   3
HRP    L2X2X1/4        4     A36   HB   3
STR    C10X15.3        2     A36   VB   3
ENDC
CASE   “Long Stair”     LEN     LT       25.0
HRR    P2STD           3     A36   VB     2
HRM    “P1 1/2STD”     6     A42   VB     2
HRP    P2STD           4     A36   VB     2
STR    C10X20     2     A36   VB
ENDC
CASE   “Very Long Stair”           LEN     GE       25.0
HRR    P2STD     3     A36   VB     2
HRM    P2STD     6     A42   VB     2
HRP    P2STD     4     A36   VB     2
STR    C10X30    2     A36   VB     2
ENDC
```

The Stairs application would for each stair set placed, start with the first case looking for a fulfilled condition. The four cases above are carefully ordered and cover the entire range on possibilities. If a case condition matches, the specified component properties are utilized. If the case condition fails, the next case condition is investigated. If all case conditions fail (which can not happen for the cases shown above), the default Stairs properties specified in the Stairs definition file (ACE_ST.DEF) would be utilized.

ACE FWP Stairs Documentation

LOG FILES

All applications can write log files if the environment variable ACE_DUMP is set to 1. There have been reports that some sites lock the C root drive and under certain conditions a locked C drive can cause a system fault 5.

All applications have been modified to warn of a locked drive/file and then gracefully exit. All applications now look for the environment variable ACE_LOG_PATH. If it is found, that is the directory where the log files will be placed. If the directory is locked or non-existent or if file is locked a warning will be given and the C drive will be tried. If it is locked or the file is locked a warning will be given and application will gracefully exit.

Usage of the variable ACE_LOG_PATH to control log file locations is similar to ACE_DEF_PATH to control DEF files. However there is one very important difference: ACE_LOG_PATH should NEVER point to a network drive (this is highly recommended for ACE_DEF_PATH). Everyone writes to the same named log file and if they are on a network drive there will be bad consequences. ALWAYS point ACE_LOG_PATH to a local drive (perhaps a temp off C root).