

```
/* Author - Chuck Lawson, PE */  
/* Company - Bentley Systems, Inc. */  
/* Creation Date - 9/19/2006 */
```

```
/*  
The objectives of this criteria is to handle various options when drawing a  
line with criteria where the end point's offset and elevation may be  
established by several possible means. The first sub routine entitled  
"Draw_Any_Line", is used to set the variables required for the primary  
subroutine that establishes the end point.
```

```
The subroutine "ppl_chain_profile_length_qualifier" uses the variables from  
the sub routine "Draw_Any_Line". These variables can be set in your criteria  
by any means. The best way to do this that grants the user maximum  
flexibility is to read the variables as adhoc attributes from a plan graphic  
line. If a particular variable is not needed, just hard code the variable to  
the default null value shown in the comments above the variable.
```

```
The purpose or intent of this criteria is not to replace the normal basic draw  
line commands we are all use to. This criteria should only be considered,  
when the end point needs to be controlled by a profile and or a chain. Where  
the chain may be used as a station reference and also possibly the control for  
the horizontal location.
```

```
This criteria supports the following possible combinations listed in the order  
of priority.
```

1. End point chain, profile and stationing are all coincident and set. Draw to the endpoint's chain and profile.
2. End point chain and profile are set, but the user chose to NOT use the end point chain stationing. The profile stationing is the evaluated based on the shape cluster baseline stationing. If the station of the profile matches up with the shape cluster station, then this profile will be used. This will lock the elevation only. Then if the user also sets the option to use the chain as an offset, this will then lock the offset.

3. If options one and two above fail because of the stations not matching up, it is still possible to specify an ending chain and profile, but only use the chain as an offset. This will lock the offset but the elevation will still be undefined at this point. Of course, you must specify an ending chain name and the ending chain name must be within the limits of the pattern line station.
4. If no chain name is specified but a profile name is specified, then the criteria will see if the profile stationing matches up with the shape cluster baseline stationing. If it does, this will set the elevation only but not the offset.
5. The next test is to see if the offset has been computed in steps 1 - 4. If the offset has not been computed, then the following test are conducted.
 - Is there a slope to use given?
 - Is there a width qualifier to use given?
 - Is there a marked point to use given?

These three items are tested to set the delta offset. If they all fail then the delta Offset will default to zero distance from the first point.

6. The next test is to see if the elevation has been computed in steps 1 - 4. If the elevation has not been computed, then the following test are conducted.
 - Is there a slope to use given?
 - Is there a height qualifier to use given?
 - Is there a marked point to use given?

These three items are tested to set the elevation. If they all fail then the delta height will default to zero distance from the first point.

This concludes all of the possible test. The ending point will then be marked based on the computed delta offset and delta elevation.

*/

```
sub draw_any_line
{
  /* Optional PGL Chain. Set to ^none^ if not used. */
  _s_end_point_chain_name      = _s_top_berm_pgl_chain

  /* Optional PGL Profile. Set to ^none^ if not used. */
  _s_end_point_profile_name    = _s_top_berm_profile

  /* ^Yes^ or ^No^ to allow the PGL chain to set the end point offset */
  _s_use_chain_offset          = ^no^

  /* ^Yes^ or ^No^ to use the PGL chain stationing for profile station reference */
  _s_use_chain_stationing      = ^yes^

  /* Slope in percent for the line to be drawn. Set to 999 if not used. */
  _d_normal_slope_to_end_point = _d_ditch_back_slope

  /* Delta x WIDTH qualifier set to 999 if not used. */
  _d_dx_width_qualifier        = _d_ditch_backslope_width

  /* Ending point number WIDTH qualifier. Set to 999 if not used */
  _d_x_marked_point_qualifier  = 999

  /* Delta y HEIGHT qualifier. Set to 999 if not used. */
  _d_dy_height_qualifier       = 999

  /* Ending point number HEIGHT qualifier. Set to 999 if not used */
  _d_y_marked_point_qualifier  = 999

  /* Beginning marked point number */
  _d_p1                         = 504

  /* Ending marked point number */
  _d_p2                         = 505

  /* Temp point use for computations */
  _d_p3                         = 587

  /* Initialize internal variables and marked points */
  mark x[_d_p2] y[_d_p2]
  _d_end_delta_x = 0
  _d_end_delta_y = 0

  /* call the subroutine to establish the end point location */
  call pgl_chain_profile_length_qualifier
}
```

```

    }
end sub

sub pgl_chain_profile_length_qualifier
{
    _s_end_point_located          = ^no^
    _s_end_point_offset_computed = ^no^
    _s_end_point_elev_computed   = ^no^

    /* An end point profile name has been specified */
    if (not _s_end_point_profile_name = ^none^) then
    {
        /* An end point PGL Chain name and Profile name have been specified */
        if (not _s_end_point_chain_name = ^none^) then
        {
            /* Test to see if given chain name is at current station. Make sure user */
            /* did not specify a chain name that does not reside in the area of this */
            /* cross section pattern line. */
            mark _mp_tmp100
            _d_TempOffset = Cogo_OffsetFromChain(_mp_tmp100, _s_end_point_chain_name)

            /* Confirmed pattern station within specified chain name limits */
            if (_d_TempOffset < 100000000) then
            {
                /* Chain stationing and profile stationing are valid and user specified to use the */
                /* station of the specified ending chain. */
                if (chain _s_end_point_chain_name sta within prof = _s_end_point_profile_name) and
                    (_s_use_chain_stationing = ^yes^) then
                {
                    draw skip to chain _s_end_point_chain_name prof elev = _s_end_point_profile_name
                    mark x[_d_p2] y[_d_p2]
                    _s_end_point_elev_computed = ^yes^
                    _d_end_delta_y = (y[_d_p2] - y[_d_p1])

                    /* Lock offset from chain is user set */
                    if (_s_use_chain_offset = ^yes^) then
                    {
                        _s_end_point_offset_computed = ^yes^
                        _s_end_point_located          = ^yes^
                        _d_end_delta_x = (x[_d_p2] - x[_d_p1])
                    }
                }

                /* Chain stationing either not valid or not to be used --> set by user. */
                /* Test profile to determine if the given profile stationing falls within */
                /* the shape cluster baseline. */
            }
        }
    }
}

```

```
else if (sta within prof = _s_end_point_profile_name) and
  (_s_use_chain_stationing = ^no^) then
{
  draw skip trace in off=0 for dx=1000
  draw skip dx=1 dy=0 to chain _s_end_point_chain_name
  draw skip dx=0 dy=100
  draw skip dx=0 dy=-1 to prof elev = _s_end_point_profile_name
  mark x[_d_p2] y[_d_p2]
  _s_end_point_elev_computed = ^yes^
  _d_end_delta_y = (y[_d_p2] - y[_d_p1])
  /* Lock offset from chain is user set */
  if (_s_use_chain_offset = ^yes^) then
  {
    _s_end_point_offset_computed = ^yes^
    _s_end_point_located = ^yes^
    _d_end_delta_x = (x[_d_p2] - x[_d_p1])
  }
}

/* Only use the chain for an offset where the profile information is invalid */
else if (_s_use_chain_offset = ^yes^) then
{
  draw skip trace in off=0 for dx=1000
  draw skip dx=1 dy=0 to chain _s_end_point_chain_name
  mark x[_d_p2] y[_d_p2]
  _s_end_point_offset_computed = ^yes^
  _d_end_delta_x = (x[_d_p2] - x[_d_p1])
  mark x[_d_p3] y[_d_p3]
}

} /* Ends test for confirmed pattern station within specified chain name limits */

else
{
  /* Specified chain name is not within the limits of the current pattern line. */
  /* No purpose for this else statement other than for debugging to verify the failure */
  /* of the specified chain name. It is not reasonable to specify a chain name that does */
  /* not exist at the given pattern line station. */
}

} /* Ends test for end point chain name and profile name being specified */

/* Only an end point Profile name has been specified */
else if (sta within prof = _s_end_point_profile_name) then
{
  draw skip dx=0 dy=100
```

```
        draw skip dx=0 dy=-1 to prof elev = _s_end_point_profile_name
        mark x[_d_p2] y[_d_p2]
        _s_end_point_elev_computed = ^yes^
        _d_end_delta_y = (y[_d_p2] - y[_d_p1])
    } /* Ends test for only setting and end point Profile Name. */

    draw skip to x[_d_p1] y[_d_p1]

} /* Ends test for end point profile name being specified */

if (_s_end_point_located = ^no^) then
{
    draw skip to x[_d_p1] y[_d_p1]

    /* Establish offset by Slope, DX width qualifier or X Marked Point if specified */
    if (_s_end_point_offset_computed = ^no^) then
    {
        if (_s_end_point_elev_computed = ^yes^) and (_d_normal_slope_to_end_point < 999) then
        {
            draw skip slope = _d_normal_slope_to_end_point to y[_d_p2]
        }
        else if (_d_dx_width_qualifier < 999) then
        {
            draw skip dx=_d_dx_width_qualifier dy=0
        }
        else if (_d_x_marked_point_qualifier < 999) then
        {
            draw skip dx=1 dy=0 to x[_d_x_marked_point_qualifier]
        }
        else if (_d_normal_slope_to_end_point < 999) and (_d_dy_height_qualifier < 999) then
        {
            draw skip slope = _d_normal_slope_to_end_point for dy= _d_dy_height_qualifier
        }
        mark x[_d_p3] y[_d_p3]
        _s_end_point_offset_computed = ^yes^
        _d_end_delta_x = abs(x[_d_p3] - x[_d_p1])
        draw skip to x[_d_p1] y[_d_p1]
    }

    /* Establish elevation by Slope, DY height qualifier or Y Marked Point if specified */
    if (_s_end_point_elev_computed = ^no^) then
    {
        if (_s_end_point_offset_computed = ^yes^) and (_d_normal_slope_to_end_point < 999) then
        {
            draw skip slope = _d_normal_slope_to_end_point to x[_d_p3]
```

```
    }
    else if (_d_dy_height_qualifier < 999) then
    {
        draw skip dx=0 dy=_d_dy_height_qualifier
    }
    else if (_d_y_marked_point_qualifier < 999) then
    {
        draw skip dx=0 dy=100
        draw skip dx=0 dy=-1 to y[_d_y_marked_point_qualifier]
    }
    else if (_d_normal_slope_to_end_point < 999) and (_d_dx_width_qualifier < 999) then
    {
        draw skip slope = _d_normal_slope_to_end_point for dx= _d_dx_width_qualifier
    }
    mark x[_d_p3] y[_d_p3]
    _s_end_point_elev_computed = ^yes^
    _d_end_delta_y = (y[_d_p3] - y[_d_p1])
    draw skip to x[_d_p1] y[_d_p1]
}

} /* Ends test for end point located */

draw skip to x[_d_p1] y[_d_p1]

draw skip dx=_d_end_delta_x dy=_d_end_delta_y

/* Mark the final ending point location. */
mark x[_d_p2] y[_d_p2]

}
end sub
```