

## Chapter 20.

# Functions bs3\_curve Sa thru Zz

Topic: Ignore

## bs3\_curve\_same

Function: Spline Interface, Construction Geometry

Action: Determines whether two spline curves are the same.

Prototype: 

```
logical bs3_curve_same (  
    bs3_curve bs1,           // first curve  
    bs3_curve bs2,           // second curve  
    double tol                // parameter space  
        = 0.0                // tolerance for equal  
                                // control points  
);
```

Includes: 

```
#include "kernel/acis.hxx"  
#include "baseutil/logical.h"  
#include "kernel/spline/bs3_crv/bs3curve.hxx"  
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: This routine checks that the two curves share the same form, the same knot vectors, and the same control points (within tolerance). This routine is not to be used for coincidence testing, but only as a simple filter to discard obvious cases.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3\_crv/sp3crtn.hxx

Effect: System routine

## bs3\_curve\_save

Function: Spline Interface, Construction Geometry, SAT Save and Restore

Action: Saves a curve to a file.

**Prototype:**        `void bs3_curve_save (`  
                              `bs3_curve cur`                        `// given curve`  
                              `);`

**Includes:**        `#include "kernel/acis.hxx"`  
                              `#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
                              `#include "kernel/spline/bs3_crv/sp3crtm.hxx"`

**Description:**     Writes a representation of the parametric curve to some external medium, using routines `write_int`, `write_long`, `write_real`, and `write_string`, defined in ACIS file `kernutil/fileio/fileio.hxx`.

                             The overloaded `<<` operator acts like `bs3_curve_save`, but writes to a C++ style stream using stream operators `streams/streams.hxx`. The output format need not necessarily be the same as for `bs3_curve_save`, but it is strongly recommended that they be so.

`bs3_curve cur;`

**Errors:**            None

**Limitations:**     None

**Library:**          kernel

**Filename:**        `kern/kernel/spline/bs3_crv/sp3crtm.hxx`

**Effect:**           System routine

## bs3\_curve\_self\_int

**Function:**         Spline Interface, Construction Geometry

**Action:**           Determines self intersections in a curve.

**Prototype:**        `curve_curve_int* bs3_curve_self_int (`  
                              `bs3_curve& curv`                        `// given curve`  
                              `);`

**Includes:**        `#include "kernel/acis.hxx"`  
                              `#include "kernel/kernint/intcucu/intcucu.hxx"`  
                              `#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
                              `#include "kernel/spline/sg_bs3c/sps3crtm.hxx"`

**Description:**     Refer to Action.

**Errors:**           None

Limitations:     None  
Library:         kernel  
Filename:         kern/kernel/spline/sg\_bs3c/sps3crtm.hxx  
Effect:           System routine

## bs3\_curve\_set\_closed

Function:         Spline Interface, Construction Geometry  
Action:           Sets the form of a bs3\_curve to be closed.  
Prototype:        void bs3\_curve\_set\_closed (                  
                    bs3\_curve crv                               // given curve  
                    );  
Includes:          #include "kernel/acis.hxx"  
                    #include "kernel/spline/bs3\_crv/bs3curve.hxx"  
                    #include "kernel/spline/sg\_bs3c/sps3crtm.hxx"  
Description:       Refer to Action.  
Errors:           None  
Limitations:       None  
Library:           kernel  
Filename:           kern/kernel/spline/sg\_bs3c/sps3crtm.hxx  
Effect:           Changes model

## bs3\_curve\_set\_ctrlpt

Function:         Spline Interface, Construction Geometry  
Action:           Sets the position of one control point.  
Prototype:        void bs3\_curve\_set\_ctrlpt (                  
                    bs3\_curve curv,                           // bs3\_curve to modify  
                    int index,                                 // index of control point  
                    double\* pos,                                // xyz location copied  
  // into control point,  
  // size [3]  
                    double weight                               // weight which control  
  // point is assigned,  
  // only used if curv is  
  // rational  
                    );

Includes:        `#include "kernel/acis.hxx"`  
                  `#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
                  `#include "kernel/spline/bs3_crv/sp3crtm.hxx"`

Description:    Checks that `crv` has an index control point. If it does it copies the `xyz` values of `pos` into control point's data structure. When `crv` is rational it also copies the weight value into the control point's data structure.

Errors:         None

Limitations:    None

Library:        kernel

Filename:       kern/kernel/spline/bs3\_crv/sp3crtm.hxx

Effect:         System routine

## bs3\_curve\_set\_ctrlpts

Function:       Spline Interface, Construction Geometry

Action:         Sets the position of all control points.

Prototype:      `void bs3_curve_set_ctrlpts (`  
                  `bs3_curve crv,                    // tgt bs3_curve to`  
                  `// modify`  
                  `int cpt_count,                    // number of control`  
                  `// points`  
                  `double* pos,                    // [xyz0,xyz1..] to copy`  
                  `// sized:[3*cpt_count]`  
                  `double* weight                  // weights for each cpt.`  
                  `// only used if crv is`  
                  `// rational`  
                  `// sized:[cpt_count]`  
                  `);`

Includes:        `#include "kernel/acis.hxx"`  
                  `#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
                  `#include "kernel/spline/bs3_crv/sp3crtm.hxx"`

Description:    Iterates through all control points on `crv`. Copies the values of `pos` into the control points and if `crv` is rational also copies weight values.

Errors:         None

Limitations:    None

Library: kernel  
Filename: kern/kernel/spline/bs3\_crv/sp3crtm.hxx  
Effect: System routine

## bs3\_curve\_set\_fitol

Function: Spline Interface, Construction Geometry

Action: Sets the fit tolerance of a curve.

Prototype: 

```
double bs3_curve_set_fitol (  
    int npoints,           // number of points  
    SPAPosition const* pos, // array of object space  
                           // positions  
    double opt_tol         // starting value  
    = -1.0  
);
```

Includes: 

```
#include "kernel/acis.hxx"  
#include "baseutil/vector/position.hxx"  
#include "kernel/spline/bs3_crv/sp3crtm.hxx"
```

Description: Estimates a fit tolerance for interpolating the given set of initial points. It takes an initial value from the `fitol_curve_interp` option if this is positive, otherwise it starts with `SPAresfit`. Then it estimates the maximum curvature by fitting a circle to each consecutive set of three points, and, if necessary reduces the fit tolerance to an order of magnitude smaller than the corresponding minimum radius of curvature. As a sanity check, it never returns less than `SPAresabs`.

If `fit_tol` been set in the `fitol_curve_interp` option, that is used as a starting value, otherwise `opt_tol` is used if supplied. Otherwise the starting value is `SPAresfit`.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3\_crv/sp3crtm.hxx

Effect: System routine

## bs3\_curve\_set\_form

Function: Spline Interface, Construction Geometry

Action: Sets the form of a `bs3_curve`.

**Prototype:**        `void bs3_curve_set_form (`  
                              `bs3_curve bs3                        // given curve`  
                              `);`

**Includes:**        `#include "kernel/acis.hxx"`  
                              `#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
                              `#include "kernel/spline/sg_bs3c/sps3crtn.hxx"`

**Description:**     The form of the curve will be open (`bs3_curve_open_ends`), closed (`bs3_curve_closed_ends`), periodic (`bs3_curve_periodic_ends`), or unknown (`bs3_curve_unknown_ends`).

**Errors:**            None

**Limitations:**     None

**Library:**          kernel

**Filename:**        kern/kernel/spline/sg\_bs3c/sps3crtn.hxx

**Effect:**            System routine

## bs3\_curve\_set\_open

**Function:**           Spline Interface, Construction Geometry

**Action:**            Sets the form of a `bs3_curve` to be open.

**Prototype:**        `void bs3_curve_set_open (`  
                              `bs3_curve crv                        // given curve`  
                              `);`

**Includes:**        `#include "kernel/acis.hxx"`  
                              `#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
                              `#include "kernel/spline/sg_bs3c/sps3crtn.hxx"`

**Description:**     Refer to Action.

**Errors:**            None

**Limitations:**     None

**Library:**          kernel

**Filename:**        kern/kernel/spline/sg\_bs3c/sps3crtn.hxx

**Effect:**            System routine

## bs3\_curve\_set\_periodic

Function: Spline Interface, Construction Geometry

Action: Sets the form of a bs3\_curve to be periodic.

Prototype: 

```
void bs3_curve_set_periodic (  
    bs3_curve crv          // given curve  
);
```

Includes: 

```
#include "kernel/acis.hxx"  
#include "kernel/spline/bs3_crv/bs3curve.hxx"  
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: Refer to Action.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg\_bs3c/sps3crtn.hxx

Effect: System routine

## bs3\_curve\_shift

Function: Spline Interface, Construction Geometry

Action: Reparameterizes the given curve in place by adding the given shift value to its parameter values.

Prototype: 

```
void bs3_curve_shift (  
    double delta,          // parameter shift  
                          // desired  
    bs3_curve cur          // given curve  
);
```

Includes: 

```
#include "kernel/acis.hxx"  
#include "kernel/spline/bs3_crv/bs3curve.hxx"  
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: Refer to Action.

Errors: None

Limitations: None

Library: kernel  
Filename: kern/kernel/spline/bs3\_crv/sp3crtn.hxx  
Effect: System routine

## bs3\_curve\_span

Function: Spline Interface, Construction Geometry

Action: Creates a new curve that is the nth span of the given curve.

Prototype: 

```
bs3_curve bs3_curve_span (  
    int n,                // given span index  
    bs3_curve cur         // given curve  
);
```

Includes: 

```
#include "kernel/acis.hxx"  
#include "kernel/spline/bs3_crv/bs3curve.hxx"  
#include "kernel/spline/bs3_crv/sp3crtn.hxx"
```

Description: Creates a new curve that consists only of the nth simple span of the given curve. The knot vector of the new curve will have start and end multiplicities equal to the degree; therefore, the new curve will represent a single Bezier span. In the case of a rational curve, the weights associated with the start and end control points have not been normalized.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3\_crv/sp3crtn.hxx

Effect: System routine

## bs3\_curve\_span\_poly

Function: Spline Interface, Construction Geometry

Action: Gets the normalized rational polynomial for the indexed span of the given curve.

Prototype: 

```
rat_poly_vec bs3_curve_span_poly (  
    int n,                // given span index  
    bs3_curve cur         // given curve  
);
```



**Includes:** `#include "kernel/acis.hxx"`  
`#include "kernel/kernutil/poly/poly.hxx"`  
`#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
`#include "kernel/spline/bs3_crv/sp3crtn.hxx"`

**Description:** On the assumption that a `bs3_curve` is a piecewise rational polynomial vector function of its parameter, convert the `nth` span into a rational polynomial vector.

The parameter range of the polynomial is normalized to `[0, 1]`, to minimize numerical problems. The class `rat_poly_vec` is defined in `kernutil/poly/poly.hxx`, but can be treated roughly as four arrays of doubles, each being the coefficients of the polynomial in order of increasing degree.

**Errors:** None

**Limitations:** None

**Library:** kernel

**Filename:** kern/kernel/spline/bs3\_crv/sp3crtn.hxx

**Effect:** System routine

## bs3\_curve\_span\_range

**Function:** Spline Interface, Construction Geometry

**Action:** Gets the parameter bounds of the span specified.

**Prototype:**

```
SPAinterval bs3_curve_span_range (
    int n,                // span number
    bs3_curve cur         // given curve
);
```

**Includes:** `#include "kernel/acis.hxx"`  
`#include "baseutil/vector/interval.hxx"`  
`#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
`#include "kernel/spline/bs3_crv/sp3crtn.hxx"`

**Description:** Returns the parameter bounds of the span specified, counting from 0 to `bs3_curve_nspans - 1`, in increasing parameter order. If the integer index is out of range, this function returns an empty interval.

**Errors:** None

Limitations:     None

Library:         kernel

Filename:        kern/kernel/spline/bs3\_crv/sp3crtm.hxx

Effect:           System routine

## bs3\_curve\_split

Function:        Spline Interface, Construction Geometry

Action:           Splits a given 3D B-spline curve at a given parameter value.

Prototype:       bs3\_curve bs3\_curve\_split (

```

    bs3_curve& cur,           // given curve
    double param,             // given parameter
                                // value
    SPAposition const& split_pt // given position
    =*(SPAposition*)NULL_REF,
    SPAunit_vector const&      // given direction
    split_dir                  // of split
    =*(SPAunit_vector*)NULL_REF,
    SPAunit_vector const&      // given direction on
    high_split_dir             // "high" side
    =*(SPAunit_vector*)NULL_REF
);
```

Includes:        #include "kernel/acis.hxx"

```

#include "baseutil/vector/position.hxx"
#include "baseutil/vector/unitvec.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/bs3_crv/sp3crtm.hxx"
```

Description:     This function divides a given 3D B-spline curve at a given parameter value. If an object space point (which need not be exactly at the point defined by the curve and parameter, but is assumed to be close) is given, the appropriate end point of each resulting spline is shifted to lie exactly on the given position.

If a tangent direction is given as a fourth argument, an attempt is made to match tangent directions at the cut. If a fifth argument is also given, an attempt is made to match the fourth argument direction on the low parameter side of the split, and the fifth argument direction on the high parameter side of the split. This gives the function the ability to split non-G1 splines nicely.

If the B-spline is open, this function creates a new spline for the initial portion of the curve, and returns this as its value. If the B-spline is closed, it takes the portion before the given parameter value and tacks it onto the end, but marks the spline as open henceforth. In this case, the function returns NULL.

The second SPAunit\_vector gives the direction which the second (higher in parameter) curve is to have at the split. It only needs to be passed for curves that are not G1 at the split point (which is, and should be, very unusual). If omitted it is taken to be the same as the the first given direction.

The return value from this function is a curve representing the portion of the original curve before the split point, or NULL if the split point was at the start of the original curve. The given curve is modified to become the part of the curve subsequent to the split point, or is set to NULL if the split point was at the end.

|              |  |
|--------------|--|
| Errors:      | None                                   |
| Limitations: | None                                   |
| Library:     | kernel                                 |
| Filename:    | kern/kernel/spline/bs3_crv/sp3crtn.hxx |
| Effect:      | System routine                         |

## bs3\_curve\_start

|              |   |
|--------------|---|
| Function:    | Spline Interface, Construction Geometry   |
| Action:      | Gets the start point of the given spline curve.   |
| Prototype:   | <pre>SPAposition bs3_curve_start (<br/>    bs3_curve bs           // given curve<br/>);</pre>   |
| Includes:    | <pre>#include "kernel/acis.hxx"<br/>#include "baseutil/vector/position.hxx"<br/>#include "kernel/spline/bs3_crv/bs3curve.hxx"<br/>#include "kernel/spline/sg_bs3c/sps3crtn.hxx"</pre> |
| Description: | Refer to Action.  |
| Errors:      | None  |

Limitations:     None

Library:         kernel

Filename:        kern/kernel/spline/sg\_bs3c/sps3crtn.hxx

Effect:          System routine

## bs3\_curve\_start\_tangent

Function:         Spline Interface, Construction Geometry

Action:          Gets the normalized tangent to the given spline at the start.

Prototype:        SPAunit\_vector bs3\_curve\_start\_tangent (  
                                   bs3\_curve bs                         // given curve  
                                   );

Includes:         #include "kernel/acis.hxx"  
                   #include "baseutil/vector/unitvec.hxx"  
                   #include "kernel/spline/bs3\_crv/bs3curve.hxx"  
                   #include "kernel/spline/sg\_bs3c/sps3crtn.hxx"

Description:      Refer to Action.

Errors:           None

Limitations:     None

Library:         kernel

Filename:        kern/kernel/spline/sg\_bs3c/sps3crtn.hxx

Effect:          Read-only

## bs3\_curve\_subset

Function:         Spline Interface, Construction Geometry

Action:          Creates a curve that is a subset of a given one.

Prototype:        bs3\_curve bs3\_curve\_subset (  
                                   bs3\_curve old\_bs,                         // given\_curve  
                                   SPAinterval const& new\_range, // required bounds  
                                   double    = 0,                         // requested fit  
   // tolerance  
                                   double& actual\_fit                        // returned actual  
                                   =\*(double\*)NULL\_REF                       // fit tolerance used  
                                   );

**Includes:**        `#include "kernel/acis.hxx"`  
                   `#include "baseutil/vector/interval.hxx"`  
                   `#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
                   `#include "kernel/spline/bs3_crv/sp3crtm.hxx"`

**Description:**    Creates a curve that is a subset of a given one, being the overlap in parameter space of the given curve and a given interval. A curve periodic in one or both parameter directions is rolled around if need be to cover the required range. It is very unlikely that the tolerance arguments will be needed, but they are included for completeness.

**Errors:**            None

**Limitations:**    None

**Library:**         kernel

**Filename:**        kern/kernel/spline/bs3\_crv/sp3crtm.hxx

**Effect:**           System routine

## bs3\_curve\_tangent

**Function:**         Spline Interface, Construction Geometry

**Action:**           Determines the curve direction at the given parameter value.

**Prototype:**        `SPAunit_vector bs3_curve_tangent (`  
                               `double param,                    // given parameter value`  
                               `bs3_curve cur                   // given curve`  
                               `);`

**Includes:**        `#include "kernel/acis.hxx"`  
                   `#include "baseutil/vector/unitvec.hxx"`  
                   `#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
                   `#include "kernel/spline/bs3_crv/sp3crtm.hxx"`

**Description:**    Normally, this normalizes the result of `bs3_curve_deriv`.

**Errors:**            None

**Limitations:**    None

**Library:**         kernel

**Filename:**        kern/kernel/spline/bs3\_crv/sp3crtm.hxx

**Effect:**           System routine

# bs3\_curve\_tangent\_cone

Function: Spline Interface, Construction Geometry

Action: Creates a cone bounding the tangent directions of a curve.

Prototype: 

```
bs3_curve_tancone bs3_curve_tangent_cone (  
    bs3_curve cur,           // given curve  
    logical approx_OK       // make a quick  
        = FALSE             // approximation  
);
```

Includes: 

```
#include "kernel/acis.hxx"  
#include "baseutil/logical.h"  
#include "kernel/spline/bs3_crv/bs3curve.hxx"  
#include "kernel/spline/bs3_crv/sp3crtcn.hxx"
```

Description: A tangent cone is a cone that contains all the tangent vectors of a curve. If the curve is a straight line, the cone would be directed along the line and having a very small angle would contain all the tangent vectors of the line; in fact, the angle could be 0. All the tangent go in the same direction. If the curve was almost a straight line, the cone axis could approximate the direction of the curve somehow, and the cone's angle would need to be a little bigger. The more the curve curves, the bigger the cone angle. If the curve were a complete circle, the cone would need to encompass every direction (even though it lies in a plane), so the cone's half angle would be  $\pi$ . If the curve were a helix, the cone's half angle would be slightly less than  $\pi/2$ , depending on the pitch and radius of the helix.

If the logical approx argument is TRUE, a quick approximation is found. The resulting cone is guaranteed to be wholly within the one found with a FALSE argument, but is not guaranteed to be a true bound - the curve tangent could lie outside it at some parts of the curve.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3\_crv/sp3crtcn.hxx

Effect: System routine

# bs3\_curve\_tan\_pt\_crv

Function: Spline Interface, Construction Geometry

Action: Determines a position on a bs3\_curve where a line from a given position is tangent to the curve.

Prototype: 

```
logical bs3_curve_tan_pt_crv (  
    const SPAposition& point,    // input start point  
    bs3_curve crv,                // input curve  
    const SPAunit_vector& normal, // curve normal  
    double* t,                   // curve parameter  
    logical old_method           // solution method  
        = FALSE,  
    const SPAvector& xa          // x-axis  
        = (*(SPAvector*)NULL_REF),  
    const SPAvector& ya          // y-axis  
        = (*(SPAvector*)NULL_REF),  
    const SPAposition& guess_pt // input guess point  
        = (*(SPAposition*)NULL_REF  
    );
```

Includes: 

```
#include "kernel/acis.hxx"  
#include "baseutil/logical.h"  
#include "baseutil/vector/position.hxx"  
#include "baseutil/vector/unitvec.hxx"  
#include "baseutil/vector/vector.hxx"  
#include "kernel/spline/bs3_crv/bs3curve.hxx"  
#include "kernel/spline/sg_bs3c/sps3crtn.hxx"
```

Description: Determines a position on a bs3\_curve where a line from a given position is tangent to the curve. Uses a guess parameter value to aid the solution (especially where there are multiple solutions).

Returns TRUE if the point was computed; FALSE if no solution was found. FALSE is preferred as the solution method (do not use the old method). If old\_method is TRUE, then the x- and y-axis values are needed.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg\_bs3c/sps3crtn.hxx

Effect: System routine

## bs3\_curve\_term

Function: Spline Interface, Construction Geometry

Action: Flushes any retained storage in the underlying curve package.

Prototype: `void bs3_curve_term ();`

Includes: `#include "kernel/acis.hxx"`  
`#include "kernel/spline/bs3_crv/sp3crtm.hxx"`

Description: ACIS calls this routine; it should *not* be called externally.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3\_crv/sp3crtm.hxx

Effect: System routine

## bs3\_curve\_testpt

Function: Spline Interface, Construction Geometry

Action: Determines whether a point lies within tolerance of a three-dimensional B-spline curve.

Prototype: `logical bs3_curve_testpt (  
 SPAposition const& pos, // given point  
 double tol, // given tolerance  
 bs3_curve cur, // given curve  
 SPAparameter const& // guess value  
 param_guess // for parameter  
 =*(SPAparameter*)NULL_REF,  
 SPAparameter& param_actual// returned set to  
 // exact  
 =*(SPAparameter*)NULL_REF// parameter value  
);`

Includes: `#include "kernel/acis.hxx"`  
`#include "baseutil/logical.h"`  
`#include "baseutil/vector/param.hxx"`  
`#include "baseutil/vector/position.hxx"`  
`#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
`#include "kernel/spline/bs3_crv/sp3crtm.hxx"`



|              |  |
|--------------|--|
| Description: | Refer to Action.                       |
| Errors:      | None                                   |
| Limitations: | None                                   |
| Library:     | kernel                                 |
| Filename:    | kern/kernel/spline/bs3_crv/sp3crtn.hxx |
| Effect:      | System routine                         |

## bs3\_curve\_to\_array

Function: Spline Interface, Construction Geometry

Action: Gets the dimension, degree, rationality, control points, weights, and knots for a 3D B-spline curve.

Prototype:

```
void bs3_curve_to_array (
    bs3_curve bs,           // given curve
    int& dim,               // returned dimension
    int& deg,               // returned degree
    logical& rat,           // returned rational
    int& num_ctrlpts,       // returned number of
                           // control points
    SPAPosition*& ctrlpts,  // returned control
                           // points
    double*& weights,       // returned weights
    int& num_knots,         // returned number of
                           // knots
    double*& knots         // knots
);
```

Includes:

```
#include "kernel/acis.hxx"
#include "baseutil/logical.h"
#include "baseutil/vector/position.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sp3crtn.hxx"
```

Description: This function creates arrays of control points, weights, and knot points. It is up to the application to delete these arrays.

If the curve is rational (i.e., `rat = TRUE`), an array of weights is returned. Otherwise, the weight array is `NULL`. Uses only the *x*- and *y*-components of the positions returned as the control points.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg\_bs3c/sps3crtn.hxx

Effect: Read-only

## bs3\_curve\_to\_bs2\_curve

Function: Spline Interface, Construction Geometry

Action: Resolves the AG references in the skin code.

Prototype:

```
bs2_curve bs3_curve_to_bs2_curve (  
    bs3_curve in_cur          // given curve  
);
```

Includes:

```
#include "kernel/acis.hxx"  
#include "kernel/spline/bs2_crv/bs2curve.hxx"  
#include "kernel/spline/bs3_crv/bs3curve.hxx"  
#include "kernel/spline/sg_bs2c/sps2crtn.hxx"
```

Description: Not recommended for use. This routine simply transfers the spline data from the bs3\_curve structure to the bs2\_structure.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg\_bs2c/sps2crtn.hxx

Effect: System routine

## bs3\_curve\_trans

Function: Spline Interface, Construction Geometry, Transforms, Modifying Models

Action: Transforms the given curve in place.

Prototype:

```
void bs3_curve_trans (  
    bs3_curve cur,          // given curve  
    SPAttransf const& t     // given transform  
);
```

**Includes:** `#include "kernel/acis.hxx"`  
`#include "baseutil/vector/transf.hxx"`  
`#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
`#include "kernel/spline/bs3_crv/sp3crtm.hxx"`

**Description:** Simply transforms the curve's control points and scales the knot values. The parameterization is not affected by the transformation.

A SPAttransf consists of a 3 x 3 matrix with unit determinant, giving an affine transformation, an overall scaling factor and a translation vector. There are also three logical flags, relating to the matrix:

rotate indicates whether the matrix is anything other than the identity.

reflect indicates whether the determinant is -1.

shear is set if the matrix is not orthogonal.

**Errors:** None

**Limitations:** None

**Library:** kernel

**Filename:** kern/kernel/spline/bs3\_crv/sp3crtm.hxx

**Effect:** System routine

## bs3\_curve\_u\_param\_line

**Function:** Spline Interface, Construction Geometry

**Action:** Creates a curve along the  $u$  parameter of a parametric surface, with the  $v$  parameter held constant.

**Prototype:**

```
bs3_curve bs3_curve_u_param_line (
    bs3_surface surf,          // given surface
    double v                  // v parameter of surface
);
```

**Includes:** `#include "kernel/acis.hxx"`  
`#include "kernel/spline/bs3_crv/bs3curve.hxx"`  
`#include "kernel/spline/bs3_crv/sp3crtm.hxx"`  
`#include "kernel/spline/bs3_srf/bs3surf.hxx"`

**Description:** This line is one whose  $u$  parameter of the surface varies, while the  $v$  parameter is fixed. The parameterization of the resulting curve is undefined, though it must have the same sense as the surface  $u$  parameter, but would normally be the same as the surface  $u$  parameter. The curve will be open, closed or periodic according to whether the surface was open, closed or periodic in the  $u$  direction.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3\_crv/sp3crt.n.hxx

Effect: System routine

## bs3\_curve\_v\_param\_line

Function: Spline Interface, Construction Geometry

Action: Creates a curve along the  $v$  parameter of a parametric surface, with the  $u$  parameter held constant.

Prototype: 

```
bs3_curve bs3_curve_v_param_line (  
    bs3_surface surf,          // given surface  
    double u                  // u parameter of surface  
);
```

Includes: 

```
#include "kernel/acis.hxx"  
#include "kernel/spline/bs3_crv/bs3curve.hxx"  
#include "kernel/spline/bs3_crv/sp3crt.n.hxx"  
#include "kernel/spline/bs3_srf/bs3surf.hxx"
```

Description: Creates a curve along a  $v$ -parameter line (i.e., one with constant  $u$  parameter) of a spline surface. This is one that the  $v$ -parameter of the surface varies, while the  $u$ -parameter is fixed. The parameterization of the resulting curve is undefined, though it must have the same sense as the surface  $v$ -parameter, but would normally be the same as the surface  $v$ -parameter. The curve will be open, closed, or periodic according to whether the surface was open, closed, or periodic in the  $v$ -direction.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/bs3\_crv/sp3crt.n.hxx

Effect: System routine

# bs3\_curve\_weight

Function: Spline Interface, Construction Geometry

Action: Gets the weight value corresponding to a specific control point for a rational, 3D B-spline curve.

Prototype: 

```
logical bs3_curve_weight (
    bs3_curve bs,           // input curve
    int index,              // returned number
                           // of weights
    double& weight          // weight array
);
```

Includes: 

```
#include "kernel/acis.hxx"
#include "baseutil/logical.h"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtm.hxx"
```

Description: This function returns the weight associated with a control point of a rational 3D curve.

Errors: The index value must be valid and the curve must be rational.

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg\_bs3c/sps3crtm.hxx

Effect: System routine

# bs3\_curve\_weights

Function: Spline Interface, Construction Geometry

Action: Gets the number of weights and the values of the weights for a rational, 3D B-spline curve.

Prototype: 

```
void bs3_curve_weights (
    bs3_curve bs,           // input curve
    int& num_pts,           // returned number of
                           // weights
    double*& weights        // returned weight array
);
```

Includes: 

```
#include "kernel/acis.hxx"
#include "kernel/spline/bs3_crv/bs3curve.hxx"
#include "kernel/spline/sg_bs3c/sps3crtm.hxx"
```

Description: This function creates an array of weights for a curve. It the application's responsibility to delete this array.

Errors: None

Limitations: None

Library: kernel

Filename: kern/kernel/spline/sg\_bs3c/sps3crtm.hxx

Effect: System routine