Chapter 10.
Constant Definitions and #define

Constant definitions were often used during the evaluation of the data stored for a given save identifier. These are declared in #define statements in the file versions.hxx. This poses no problems for someone having ACIS. However, for those who do not have ACIS, the header declarations are listed below.

The versions.hxx file contains a list of the ACIS version numbers where the format of save files changed. It is used in individual ENTITY (and other) “restore” routines to allow old save files to be restored into new ACIS programs.

versions.hxx

```
// This file contains a list of the version numbers of Acis at
// which the format of ”save” files has changed. It is used in
// individual ENTITY (and other) ”restore” routines to allow old
// save files to be restored into new Acis programs.

// First declare the variables which contain the version numbers
// of the current save file being restored. There is a major and
// minor component, and a portmanteau number which combines the
// two, and is the value actually placed in the save file.

#if !defined( VERSIONS_HDR_DEF )
#define VERSIONS_HDR_DEF

#include ”kernel/dcl_kern.h”

extern DECL_KERN int restore_major_version;
extern DECL_KERN int restore_minor_version;
extern DECL_KERN int restore_version_number;  // Combined version
                                              // number.
```

```
// Now the same for the current ”save” file. This is normally set
// to the version number of this version of Acis, but can be set
// backwards by the application to cause backwards–compatible save
// files to be produced (provided the data structure is itself
// backwards–compatible).

extern DECL_KERN int save_major_version;
extern DECL_KERN int save_minor_version;
extern DECL_KERN int save_version_number;     // Combined version
                                              // number.

// Macros to define the way the portmanteau version number is
// obtained from the major and minor versions.

#define PORTMANTEAU( maj, min ) (100 * maj + min)
#define MAJOR_VERSION( port ) (port / 100)
#define MINOR_VERSION( port ) (port % 100)

// Release version numbers.

// The version at which the start and end parameter values of
// edges were removed from the save file, and recomputed when
// required.

const int PARAM_MAJOR = 1;
const int PARAM_MINOR = 1;
const int PARAM_VERSION = PORTMANTEAU( PARAM_MAJOR, PARAM_MINOR );

// The version at which LUMPs were introduced, to subdivide bodies
// into disjoint connected regions.

const int LUMP_MAJOR = 1;
const int LUMP_MINOR = 1;
const int LUMP_VERSION = PORTMANTEAU( LUMP_MAJOR, LUMP_MINOR );

// The version at which curve types were first output to the save
// file as character strings instead of integers.

const int CURVE_MAJOR = 1;
const int CURVE_MINOR = 3;
const int CURVE_VERSION = PORTMANTEAU( CURVE_MAJOR, CURVE_MINOR );
```

```
// The version at which surface types were first output to the
// save file as character strings instead of integers.

const int SURFACE_MAJOR = 1;
const int SURFACE_MINOR = 3;
const int SURFACE_VERSION =
    PORTMANTEAU( SURFACE_MAJOR, SURFACE_MINOR );

// The version at which subtypes of intcurves were defined
// requiring classification in the save file.

const int INTCURVE_MAJOR = 1;
const int INTCURVE_MINOR = 3;
const int INTCURVE_VERSION =
    PORTMANTEAU( INTCURVE_MAJOR, INTCURVE_MINOR );

// The version at which subtypes of splines were defined requiring
// classification in the save file.

const int SPLINE_MAJOR = 1;
const int SPLINE_MINOR = 3;
const int SPLINE_VERSION = PORTMANTEAU( SPLINE_MAJOR, SPLINE_MINOR
);

// The version at which pointers were first distinguished from
// integers in text files by a dollar sign.

const int DOLLAR_MAJOR = 1;
const int DOLLAR_MINOR = 3;
const int DOLLAR_VERSION = PORTMANTEAU( DOLLAR_MAJOR,
    DOLLAR_MINOR );

// The version at which sharable objects like int_cur and spl_sur
// were first surrounded by curly braces, and could be shared in
// the save file.

const int SHARABLE_MAJOR = 1;
const int SHARABLE_MINOR = 4;
const int SHARABLE_VERSION = PORTMANTEAU( SHARABLE_MAJOR,
    SHARABLE_MINOR );

// The version at which blend attributes lost their ”form” integer
// value (which was always zero, and so redundant).

const int BLEND_MAJOR = 1;
const int BLEND_MINOR = 5;
const int BLEND_VERSION = PORTMANTEAU( BLEND_MAJOR, BLEND_MINOR );
```

```
// The version at which parameter curves may have the defining
// parameter curve and surface in either slot 1 or 2.

const int PARCUR_MAJOR = 1;
const int PARCUR_MINOR = 5;
const int PARCUR_VERSION = PORTMANTEAU( PARCUR_MAJOR,
    PARCUR_MINOR );

// The version at which pcurves may have different subtypes, and
// so need a type code.

const int PCURVE_MAJOR = 1;
const int PCURVE_MINOR = 5;
const int PCURVE_VERSION = PORTMANTEAU( PCURVE_MAJOR,
    PCURVE_MINOR );

// The version at which faces may be double-sided, either free
// sheets or embedded in material.

const int TWOSIDE_MAJOR = 1;
const int TWOSIDE_MINOR = 5;
const int TWOSIDE_VERSION = PORTMANTEAU( TWOSIDE_MAJOR,
    TWOSIDE_MINOR );

// The version at which logical values are written to the save
// file as "T" or "F" instead of as integers.

const int LOGIO_MAJOR = 1;
const int LOGIO_MINOR = 5;
const int LOGIO_VERSION = PORTMANTEAU( LOGIO_MAJOR, LOGIO_MINOR );

// The version at which coedges were first expected to be sorted
// clockwise around the edge.

const int SORTCOED_MAJOR = 1;
const int SORTCOED_MINOR = 5;
const int SORTCOED_VERSION = PORTMANTEAU( SORTCOED_MAJOR,
    SORTCOED_MINOR );

// The version at which intervals could be (semi-)infinite, and so
// require different save file format.

const int INFINT_MAJOR = 1;
const int INFINT_MINOR = 6;
const int INFINT_VERSION = PORTMANTEAU( INFINT_MAJOR,
    INFINT_MINOR );
```

```
// The version at which curves have a subset range to bound them
// within their natural range.

const int BNDCUR_MAJOR = 1;
const int BNDCUR_MINOR = 6;
const int BNDCUR_VERSION = PORTMANTEAU( BNDCUR_MAJOR,
    BNDCUR_MINOR );

// The version at which surfaces have a subset range to bound them
// within their natural range.

const int BNDSUR_MAJOR = 1;
const int BNDSUR_MINOR = 6;
const int BNDSUR_VERSION = PORTMANTEAU( BNDSUR_MAJOR,
    BNDSUR_MINOR );

// The version at which multiple entities can be saved into a
// single save file unit (as opposed to multiple self-contained
// sections of a file).

const int MULTSAV_MAJOR = 1;
const int MULTSAV_MINOR = 5;
const int MULTSAV_VERSION = PORTMANTEAU( MULTSAV_MAJOR,
    MULTSAV_MINOR );

// The version in which save and restore started using a
// FileInterface object to do the io.

const int FILEINTERFACE_MAJOR = 1;
const int FILEINTERFACE_MINOR = 6;
const int FILEINTERFACE_VERSION =
    PORTMANTEAU( FILEINTERFACE_MAJOR, FILEINTERFACE_MINOR );

// The version in which "Wire Booleans" were implemented,
// requiring extensions to the data structure.

const int WIREBOOL_MAJOR = 1;
const int WIREBOOL_MINOR = 7;
const int WIREBOOL_VERSION = PORTMANTEAU( WIREBOOL_MAJOR,
    WIREBOOL_MINOR );

// The version in which "3D eye refinements" were implemented.

const int THREEDEYE_REF_MAJOR = 1;
const int THREEDEYE_REF_MINOR = 7;
const int THREEDEYE_REF_VERSION =
    PORTMANTEAU(THREEDEYE_REF_MAJOR,THREEDEYE_REF_MINOR);
```

```
// Version in which the History Manager was introduced

const int HISTORY_MAJOR = 1;
const int HISTORY_MINOR = 7;
const int HISTORY_VERSION =
    PORTMANTEAU( HISTORY_MAJOR, HISTORY_MINOR);

// The version in which ”Safe Ranges” for intcurves were
// implemented, requiring an extra field in the data structure.

const int SAFERANGE_MAJOR = 1;
const int SAFERANGE_MINOR = 7;
const int SAFERANGE_VERSION = PORTMANTEAU( SAFERANGE_MAJOR,
    SAFERANGE_MINOR );

// The version which introduced angled cross curves at the ends of
// face–face blends, requiring extra fields in ATTRIB_FFBLEND.

const int ANG_XCUR_MAJOR = 1;
const int ANG_XCUR_MINOR = 7;
const int ANG_XCUR_VERSION = PORTMANTEAU( ANG_XCUR_MAJOR,
    ANG_XCUR_MINOR );

// The version at which advanced blending facilities were first
// made available.

const int ADV_BL_MAJOR = 1;
const int ADV_BL_MINOR = 8;
const int ADV_BL_VERSION = PORTMANTEAU( ADV_BL_MAJOR,
    ADV_BL_MINOR );

// The version where the Intergraph spline library was first
// introduced.

//const int IGRSPLINE_MAJOR = 2;
//const int IGRSPLINE_MINOR = 0;
//const int IGRSPLINE_VERSION =
// PORTMANTEAU( IGRSPLINE_MAJOR, IGRSPLINE_MINOR );

// The version where the save/restore of logicals, enums was made
// consistent

const int CONSISTENT_MAJOR = 2;
const int CONSISTENT_MINOR = 0;
const int CONSISTENT_VERSION = PORTMANTEAU(CONSISTENT_MAJOR,
    CONSISTENT_MINOR);
```

```
// The version where the additional header information was added

const int FILEINFO_MAJOR = 2;
const int FILEINFO_MINOR = 0;
const int FILEINFO_VERSION = PORTMANTEAU(FILEINFO_MAJOR,
    FILEINFO_MINOR);


// The version where the mesh classes were added

const int MESH_MAJOR = 2;
const int MESH_MINOR = 0;
const int MESH_VERSION = PORTMANTEAU(MESH_MAJOR, MESH_MINOR);


// The version where extended curves and surfaces were introduced.

const int EXT_CU_SF_MAJOR = 2;
const int EXT_CU_SF_MINOR = 1;
const int EXT_CU_SF_VERSION = PORTMANTEAU(EXT_CU_SF_MAJOR,
    EXT_CU_SF_MINOR);


// The version where coedges were given sense enumeration strings.

const int COEDGE_SENSE_MAJOR = 2;
const int COEDGE_SENSE_MINOR = 2;
const int COEDGE_SENSE_VERSION =
    PORTMANTEAU(COEDGE_SENSE_MAJOR, COEDGE_SENSE_MINOR);


// The version at which Skin/Loft surfaces have a subset range to
// bound them within their natural range.

const int ARCWISE_SKIN_MAJOR = 2;
const int ARCWISE_SKIN_MINOR = 2;
const int ARCWISE_SKIN_VERSION =
    PORTMANTEAU( ARCWISE_SKIN_MAJOR, ARCWISE_SKIN_MINOR );


// The version in which adv_var_blend attributes output a logical
// specifying if two radii functions are used

const int ADV_VAR_BLEND_TWO_RADII_MAJOR = 2;
const int ADV_VAR_BLEND_TWO_RADII_MINOR = 2;
const int ADV_VAR_BLEND_TWO_RADII_VERSION =
    PORTMANTEAU( ADV_VAR_BLEND_TWO_RADII_MAJOR,
    ADV_VAR_BLEND_TWO_RADII_MINOR);
```

```
// The version in which laws first where added to ACIS

const int LAW_MAJOR = 2;
const int LAW_MINOR = 2;
const int LAW_VERSION = PORTMANTEAU( LAW_MAJOR, LAW_MINOR );


// The version in which reflection of offset_spl_surs was handled

const int OFFSET_REV_MAJOR = 2;
const int OFFSET_REV_MINOR = 2;
const int OFFSET_REV_VERSION = PORTMANTEAU( OFFSET_REV_MAJOR,
    OFFSET_REV_MINOR);


// The version in which discontinuity information was stored in
// int_curs and spl_surs

const int DISCONTINUITY_MAJOR = 3;
const int DISCONTINUITY_MINOR = 0;
const int DISCONTINUITY_VERSION =
    PORTMANTEAU( DISCONTINUITY_MAJOR, DISCONTINUITY_MINOR);


// The version in which taper_spl_surs were split into derived
// classes

const int TAPER_MAJOR = 3;
const int TAPER_MINOR = 0;
const int TAPER_VERSION =
    PORTMANTEAU( TAPER_MAJOR, TAPER_MINOR);


// The version in which net surface was added to ACIS

const int NET_SPL_MAJOR = 3;
const int NET_SPL_MINOR = 0;
const int NET_SPL_VERSION = PORTMANTEAU( NET_SPL_MAJOR,
    NET_SPL_MINOR );


// The version in which law curves and surfaces where added to
// ACIS

const int LAW_SPL_MAJOR = 4;
const int LAW_SPL_MINOR = 0;
const int LAW_SPL_VERSION = PORTMANTEAU( LAW_SPL_MAJOR,
    LAW_SPL_MINOR );
```

```
// The version at which cones have a new member, representing the
// scaling factor of the u parameter.

const int CONE_SCALING_MAJOR = 4;
const int CONE_SCALING_MINOR = 0;
const int CONE_SCALING_VERSION =
    PORTMANTEAU( CONE_SCALING_MAJOR, CONE_SCALING_MINOR );

// The version in which laws in lofts where added to ACIS

const int LOFT_LAW_MAJOR = 4;
const int LOFT_LAW_MINOR = 0;
const int LOFT_LAW_VERSION = PORTMANTEAU( LOFT_LAW_MAJOR,
    LOFT_LAW_MINOR );

// The version in which refinement ”min_u_grid_lines and
// min_v_grid_lines” were added.

const int REF_MIN_UV_GRID_MAJOR = 4;
const int REF_MIN_UV_GRID_MINOR = 0;
const int REF_MIN_UV_GRID_VERSION =
    PORTMANTEAU(REF_MIN_UV_GRID_MAJOR,REF_MIN_UV_GRID_MINOR);

// The version at which vertex blend attributes were given a new
// member describing the method of auto setback calculation, if
// any.

const int VBLEND_AUTO_MAJOR = 4;
const int VBLEND_AUTO_MINOR = 0;
const int VBLEND_AUTO_VERSION =
    PORTMANTEAU(VBLEND_AUTO_MAJOR,VBLEND_AUTO_MINOR);

// The version at which the true rolling–ball envelope surface was
// introduced for var_blend_spl_sur’s.

const int BL_ENV_SF_MAJOR = 4;
const int BL_ENV_SF_MINOR = 0;
const int BL_ENV_SF_VERSION =
    PORTMANTEAU(BL_ENV_SF_MAJOR,BL_ENV_SF_MINOR);

#endif
```