

Chapter 3.

Classes

Topic: Ignore

The class interface is a set of C++ classes, including their public and protected data and methods (member functions), that an application can use directly to interact with ACIS. Developers may also derive their own classes from these classes to add application-specific functionality and data. Refer to the *3D ACIS Online Help User's Guide* for a description of the fields in the reference template.

pid_base

Class:	Persistent ID, SAT Save and Restore		
Purpose:	Defines the persistent identifier data (PID).		
Derivation:	pid_base : –		
SAT Identifier:	None		
Filename:	pid/pid_husk/pid/pid.hxx		
Description:	This class defines the persistent identifier data (PID). The ACIS PID consists of the following items: 1. An 8-character session user ID. 2. The time of the session start. 3. The index of this identifier within this session. 4. A copy index for this identifier.		
Limitations:	None		
References:	by PID	ATTRIB_PID	
Data:			
	None		
Constructor:			
	public: pid_base::pid_base ();		

C++ allocation constructor requests memory for this object but does not populate it.

```
public: pid_base::pid_base (
    const char* nm,           // session name
    const long tm,           // session time
    const int index,         // entity index
    const int copy_num       // copy number
        = 1
);
```

C++ initialize constructor requests memory for this object and populates it with the data supplied as arguments.

```
public: pid_base::pid_base (
    const pid_base*          // bit-wise copy
);
```

C++ copy constructor requests memory for this object and populates it with the data from the object supplied as an argument.

Destructor:

None

Methods:

```
public: const char* pid_base::get_base_name () const;
```

Returns the pid session user name for an ENTITY, or NULL if none exists.

```
public: int pid_base::get_copy_num () const;
```

Returns the copy number of the ENTITY's identifier, or zero if none exists.

```
public: int pid_base::get_index () const;
```

Returns the index of the eight character session user identifier, or zero if none exists.

```
public: long pid_base::get_time_val () const;
```

Returns the time of session start for the ENTITY's identifier, measured in seconds since January 1, 1970, or zero if none exists.

```
public: void pid_base::restore ();
```

Restores the pid_base from a file.

```
if (restore_version_number < CONSISTENT_VERSION)
    read_int          Number of characters
    for( i = 0; i < numchars; i++ )
        read_int      read in a character at a time
    else
        read_string    Read in the string
read_long            time value in seconds from
                    January 1, 1970
read_int             Entity index
read_int             copy number
```

```
public: void pid_base::save () const;
```

Saves the base name, time value, index, and copy number.

```
public: void pid_base::set_pid_base (
    const char* nm,          // session name
    const long tm,          // session time
    const int index,        // entity index
    const int copy_num      // copy number
    = 1
);
```

Sets the pid_base, with all the necessary information.

```
public: void pid_base::set_pid_base (
    const pid_base*          // bit-wise copy
);
```

Sets the pid_base to an existing one.

Related Fncs:

```
create_pid_base_name, get_base_name, get_copy_num, get_index,
get_next_current_pid, get_pid, get_time_val, pid_base_init, remove_pid,
set_next_current_pid
```