NAME

union – union attributes

SYNOPSIS

```
// in sdl:

interface a; // forward
enum enumx { one,two,three,four,five,six };

union union_t switch (enumx tag) {
      case one:       int       u_integer;
      case two:       boolean   u_boolean;
      case three:     set<a>  u_set;
      case four:      ref<a>  u_ref;
      case five:      string    u_string;
};


interface my_obj {
public:
      attribute union_t _u;
};
```

LANGUAGE BINDING

```
struct union_t :
sdl_heap_base {
    union_t();
    union_t(const union_t & arg);
    const union_t & operator = (const union_t & );
    const  enum enumx tag ;

    void set_utag( enum enumx  _tg_val );
    void set_tag( enum enumx  _tg_val );

    enum enumx  get_utag() const ;
    enum enumx  get_tag() ;

    long    &set_u_integer() ;
    const  long   & get_u_integer() const ;

    boolean   & set_u_boolean() ;
    const  boolean   & get_u_boolean() const ;

    Set < Ref < a  >   >    &set_u_set() ;
    const  Set < Ref < a  >   >    &get_u_set() const ;

    Ref < a  >    &set_u_ref() ;
    const  Ref < a  >    &get_u_ref() const ;

    sdl_string   & set_u_string() ;
    const sdl_string   & get_u_string() const ;
};
```

DESCRIPTION

Unions declared in SDL result in tagged unions in the C++ language binding: a tag value and a union construct. The language binding for a union attribute contains methods for setting and reading the tag value, and a corresponding pair of methods for each arm of the union.

It is the responsibility of the application program to set the tag properly **before** setting the value of the union:

```
o.update()->set_tag(one);
o.update()->set_u_integer() = 1;
```

If the tag is incorrect, the **set_xxx()** and **get_xxx()** methods will fail an assertion, e.g.:

```
o.update()->set_tag(three); // should be "one"
o.update()->set_u_integer() = 1;
```

yields

```
union.h:60: failed assertion '_armi()==0'
```

(The values the the method **_armi()** returns are not the tag's enumeration values.)

VERSION

This manual page applies to Version 1.1 of the Shore software.

SPONSORSHIP

The Shore project is sponsored by the Advanced Research Project Agency, ARPA order number 018 (formerly 8230), monitored by the U.S. Army Research Laboratory under contract DAAB07-91-C-Q518.

SEE ALSO

intro(cxxlb), intro(oc), method(cxxlb), index(cxxlb), ref(cxxlb), set(cxxlb), and the **Shore Data Language Reference Manual**