```
        SvIOK_on
        SvNOK_on
        SvPOK_on
        SvROK_on
```

The particular macro you must use depends on which sv_set*v routine you called first. This is because every sv_set*v routine turns on only the bit for the particular type of data being set, and turns off all the rest.

For example, to create a new Perl variable called "dberror" that contains both the numeric and descriptive string error values, you could use the following code:

```
    extern int  dberror;
    extern char *dberror_list;

    SV* sv = get_sv("dberror", TRUE);
    sv_setiv(sv, (IV) dberror);
    sv_setpv(sv, dberror_list[dberror]);
    SvIOK_on(sv);
```

If the order of sv_setiv and sv_setpv had been reversed, then the macro SvPOK_on would need to be called instead of SvIOK_on.

70.2.16 Magic Variables

[This section still under construction. Ignore everything here. Post no bills. Everything not permitted is forbidden.]

Any SV may be magical, that is, it has special features that a normal SV does not have. These features are stored in the SV structure in a linked list of struct magic's, typedef'ed to MAGIC.

```
    struct magic {
        MAGIC*      mg_moremagic;
        MGVTBL*     mg_virtual;
        U16         mg_private;
        char        mg_type;
        U8          mg_flags;
        SV*         mg_obj;
        char*       mg_ptr;
        I32         mg_len;
    };
```

Note this is current as of patchlevel 0, and could change at any time.

70.2.17 Assigning Magic

Perl adds magic to an SV using the sv_magic function:

```
    void sv_magic(SV* sv, SV* obj, int how, const char* name, I32 namlen);
```

The sv argument is a pointer to the SV that is to acquire a new magical feature.

If sv is not already magical, Perl uses the SvUPGRADE macro to convert sv to type SVt_PVMG. Perl then continues by adding new magic to the beginning of the linked list of magical features. Any prior entry of the same type of magic is deleted. Note that this can be overridden, and multiple instances of the same type of magic can be associated with an SV.

The name and namlen arguments are used to associate a string with the magic, typically the name of a variable. namlen is stored in the mg_len field and if name is non-null and namlen >= 0 a malloc'd copy of the name is stored in mg_ptr field.