```
        void    sv_pos_b2u(SV* sv, I32* offsetp)
```

sv_pos_u2b

Converts the value pointed to by offsetp from a count of UTF-8 chars from the start of the string, to a count of the equivalent number of bytes; if lenp is non-zero, it does the same to lenp, but this time starting from the offset, rather than from the start of the string. Handles magic and type coercion.

```
        void    sv_pos_u2b(SV* sv, I32* offsetp, I32* lenp)
```

sv_pv

Use the SvPV_nolen macro instead

```
        char*   sv_pv(SV *sv)
```

sv_pvbyte

Use SvPVbyte_nolen instead.

```
        char*   sv_pvbyte(SV *sv)
```

sv_pvbyten

A private implementation of the SvPVbyte macro for compilers which can't cope with complex macro expressions. Always use the macro instead.

```
        char*   sv_pvbyten(SV *sv, STRLEN *len)
```

sv_pvbyten_force

A private implementation of the SvPVbytex_force macro for compilers which can't cope with complex macro expressions. Always use the macro instead.

```
        char*   sv_pvbyten_force(SV* sv, STRLEN* lp)
```

sv_pvn

A private implementation of the SvPV macro for compilers which can't cope with complex macro expressions. Always use the macro instead.

```
        char*   sv_pvn(SV *sv, STRLEN *len)
```

sv_pvn_force

Get a sensible string out of the SV somehow. A private implementation of the SvPV_force macro for compilers which can't cope with complex macro expressions. Always use the macro instead.

```
        char*   sv_pvn_force(SV* sv, STRLEN* lp)
```

sv_pvn_force_flags

Get a sensible string out of the SV somehow. If flags has SV_GMAGIC bit set, will mg_get on sv if appropriate, else not. sv_pvn_force and sv_pvn_force_nomg are implemented in terms of this function. You normally want to use the various wrapper macros instead: see SvPV_force and SvPV_force_nomg

```
        char*   sv_pvn_force_flags(SV* sv, STRLEN* lp, I32 flags)
```

sv_pvutf8

Use the SvPVutf8_nolen macro instead