New

The XSUB-writer's interface to the C `malloc` function.

```
        void    New(int id, void* ptr, int nitems, type)
```

Newc

The XSUB-writer's interface to the C `malloc` function, with cast.

```
        void    Newc(int id, void* ptr, int nitems, type, cast)
```

Newz

The XSUB-writer's interface to the C `malloc` function. The allocated memory is zeroed with `memzero`.

```
        void    Newz(int id, void* ptr, int nitems, type)
```

Poison

Fill up memory with a pattern (byte 0xAB over and over again) that hopefully catches attempts to access uninitialized memory.

```
        void    Poison(void* dest, int nitems, type)
```

Renew

The XSUB-writer's interface to the C `realloc` function.

```
        void    Renew(void* ptr, int nitems, type)
```

Renewc

The XSUB-writer's interface to the C `realloc` function, with cast.

```
        void    Renewc(void* ptr, int nitems, type, cast)
```

Safefree

The XSUB-writer's interface to the C `free` function.

```
        void    Safefree(void* ptr)
```

savepv

Perl's version of `strdup()`. Returns a pointer to a newly allocated string which is a duplicate of `pv`. The size of the string is determined by `strlen()`. The memory allocated for the new string can be freed with the `Safefree()` function.

```
        char*   savepv(const char* pv)
```

savepvn

Perl's version of what `strndup()` would be if it existed. Returns a pointer to a newly allocated string which is a duplicate of the first `len` bytes from `pv`. The memory allocated for the new string can be freed with the `Safefree()` function.

```
        char*   savepvn(const char* pv, I32 len)
```

savesharedpv

A version of `savepv()` which allocates the duplicate string in memory which is shared between threads.