

**NAME**

XtSetArg, XtMergeArgLists - set and merge ArgLists

**SYNTAX**

```
#include <X11/Intrinsic.h>
```

```
int XtSetArg(Arg arg, String name, XtArgVal value);
```

```
ArgList XtMergeArgLists(ArgList args1, Cardinal num_args1, ArgList args2, Cardinal num_args2);
```

**ARGUMENTS**

- arg* Specifies the name-value pair to set.
- args1* Specifies the first **ArgList**.
- args2* Specifies the second **ArgList**.
- num\_args1* Specifies the number of arguments in the first argument list.
- num\_args2* Specifies the number of arguments in the second argument list.
- name* Specifies the name of the resource.
- value* Specifies the value of the resource if it will fit in an **XtArgVal** or the address.

**DESCRIPTION**

The **XtSetArg** function is usually used in a highly stylized manner to minimize the probability of making a mistake; for example:

```
Arg args[20];
int n;

n = 0;
XtSetArg(args[n], XtNheight, 100);  n++;
XtSetArg(args[n], XtNwidth, 200);   n++;
XtSetValues(widget, args, n);
```

Alternatively, an application can statically declare the argument list and use **XtNumber**:

```
static Args args[] = {
```

```
    {XtNheight, (XtArgVal) 100},  
    {XtNwidth, (XtArgVal) 200},  
};  
XtSetValues(Widget, args, XtNumber(args));
```

Note that you should not use auto-increment or auto-decrement within the first argument to **XtSetArg**. **XtSetArg** can be implemented as a macro that dereferences the first argument twice.

The **XtMergeArgLists** function allocates enough storage to hold the combined **ArgList** structures and copies them into it. Note that it does not check for duplicate entries. When it is no longer needed, free the returned storage by using **XtFree**.

#### SEE ALSO

**XtOffset(3)**

*X Toolkit Intrinsic - C Language Interface*

*Xlib - C Language X Interface*