NAME

MrmRegisterNames — Registers the values associated with the names referenced in UIL (for example, UIL callback function names or UIL identifier names)

SYNOPSIS

#include <Mrm/MrmPublic.h>
Cardinal MrmRegisterNames(
MrmRegisterArglist register_list,
MrmCount register_count);

DESCRIPTION

The **MrmRegisterNames** function registers a vector of names and associated values for access in MRM. The values can be callback functions, pointers to user-defined data, or any other values. The information provided is used to resolve symbolic references occurring in UID files to their run-time values. For callbacks, this information provides the procedure address required by the Motif Toolkit. For names used as identifiers in UIL, this information provides any run-time mapping the application needs.

This function is similar to **MrmRegisterNamesInHierarchy**, except that the scope of the names registered by **MrmRegisterNamesInHierarchy** is limited to the hierarchy specified in the call to that function, whereas the names registered by **MrmRegisterNames** have global scope. When MRM looks up a name, it first tries to find the name among those registered for the given hierarchy. If that lookup fails, it tries to find the name among those registered globally.

register_list

Specifies a list of name/value pairs for the names to be registered. Each name is a case-sensitive, NULL-terminated ASCII string. Each value is a 32-bit quantity, interpreted as a procedure address if the name is a callback function, and uninterpreted otherwise.

register_count

Specifies the number of entries in *register_list*.

The names in the list are case-sensitive. The list can be either ordered or unordered.

Callback functions registered through **MrmRegisterNames** can be either regular or creation callbacks. Regular callbacks have declarations determined by Motif Toolkit and user requirements. Creation callbacks have the same format as any other callback:

void CallBackProc(
Widget *widget_id,
Opaque tag,
XmAnyCallbackStruct *callback_data);

widget_id Specifies the widget ID associated with the widget performing the callback (as in any callback function).

tag Specifies the tag value (as in any callback function).

callback_data

Specifies a widget-specific data structure. This data structure has a minimum of two members: event and reason. The reason member is always set to **MrmCR_CREATE**.

Note that the widget name and parent are available from the widget record accessible through *widget_id*.

RETURN

This function returns one of the following status return constants:

MrmSUCCESS

The function executed successfully.

MrmFAILURE

The function failed.