supermodel_class

COLLABORATORS

	TITLE : supermodel_class	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		January 23, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

supermodel_class

1.1 supermodel_class.doc

```
SM_IsMemberOf()
--datasheet--()
--datasheet-GlueFunc--()
--datasheet-supericclass--()
History()
ProcessTagList()
SM_NewSuperIC()
SM_NewSuperModel()
SM_SendGlueAttrs()
SMTAG_AddTag()
SMTAG_AddTags()
SMTAG_AllocTags()
SMTAG_ClearNumTags()
SMTAG_ClearTags()
SMTAG_FreeTags()
SMTAG_RemTag()
SMTAG_TagDone()
SMTAG_TagMore()
```

1.2 extras.lib/SM_IsMemberOf

```
NAME
    extras.lib/SM_IsMemberOf -- Check if Object belongs to a Class

SYNOPSIS
    memberof = SM_IsMemberOf(Object, ClassPtr, ClassID)

    BOOL SM_IsMemberOf(Object *, Class *, STRPTR);

FUNCTION


INPUTS

RESULT
```

```
   EXAMPLE

   NOTES
         Stolen from someone...

         Here's some IsMemberOf() code I whipped up quickly.  Might be nice i
f we all
         posted useful little BOOPSI snippets like this... maybe even collect
ed them
         together on a web site.  (I volunteer NOT to maintain this site :)

   BUGS

   SEE ALSO
```

1.3 supermodel.class/--datasheet--

```
   NAME
       supermodel.class -- Model and IC class

   SUPERCLASS
       modelclass

   DESCRIPTION


   METHODS


   ATTRIBUTES
       ICA_ attributes supported.
       SMA_AddMember - calls OM_ADDMEMBER
       SMA_RemMember - calls OM_REMMEMBER
       SMA_GlueFunc - Glue code, does tag mapping, etc.  See example.
           see --datasheet-GlueFunc--
       SMA_GlueFuncA6 - If your glue code is in a library, set this to
           your Library base.
       SMA_GlueFuncUserData -
       SMA_CacheStringTag - cache string data

   NOTES

   BUGS

   SEE ALSO
```

1.4 supermodel.class/--datasheet-GlueFunc--upermodel.class/--datasheet-GlueFunc--

```
   NAME
       GlueFunc
```

```
SYNOPSIS
    rv GlueFunc(GlueData, TagList, [userdata], [a6])
    d0          a0        a1       a2          a6

    ULONG GlueFunc(struct smGlueData *, struct TagItem *, APTR, APTR);

FUNCTION
    GlueFunc is a function you provide to modify a tag list before
    being sent to the members and ICA_TARGET of the modelclass.

    You may modify TagList and use the SMTAG_functions to modify
    TagList, however, don't free TagList.

    Call SM_SendGlueAttrsA() to send your new attributes back to the
    model class for notifiaction.

NOTES
    There is room for 50 tags in TagList

BUGS

SEE ALSO
```

1.5 supermodel.class/--datasheet-supericclass--I.class/--datasheet-supericclass--

```
NAME
    supermodel.class -- IC class

SUPERCLASS
    icclass

DESCRIPTION


METHODS


ATTRIBUTES
    ICA_ attributes supported.
    ICA_TARGET - targeted object.
    ICA_MAP - MapList.
    SICA_Model - Model object pointer, set by model object.

    SICA_InMap - TagMap, maps tags sent from Model object to Target object
.
    SICA_OutMap - TagMap, maps tags sent from Target object to Model.
        Both must be allocated with CloneTagItems(), and are given to the
        ic objects

NOTES

BUGS
```

```
SEE ALSO
```

1.6 supermodel.class/History

```
To Do

44.1 -
    * GlueCode was freeing memory twice.
    * Added SMA_CacheStringTag - Anytime this tag is passed through
      the model class, the string supplied is cached, and ti_Data
      is updated to point to the cache.
```

1.7 supermodel.class/ProcessTagList

```
NAME
    ProcessTagList -- Macro to process a taglist

SYNOPSIS
    ProcessTagList(TagList, Tag, TState)

    TState=TagList;
    while(Tag=NextTagItem(&TState))

EXAMPLE
    void SomeFunc(struct TagItem *TagList)
    {
      struct TagItem *tag, *tstate;

      ProcessTagList(TagList,tag,tstate)
      {
        seitch(tag->ti_Tag)
        {
          case GA_Left:
            ...
            break;
          etc...
        }
      }
    }
```

1.8 supermodel.class/SM_NewSuperIC

```
NAME
    SM_NewSuperIC -- Allocare SuperIC object

SYNOPSIS
```

```
FUNCTION


INPUTS

RESULT

EXAMPLE

NOTES

BUGS

SEE ALSO
```

1.9 supermodel.class/SM_NewSuperModel

```
NAME
    SM_NewSuperModel -- Allocate SuperModel object

SYNOPSIS
    model = SM_NewSuperModel( Tag1, Data1, TAg2, ...)

    Object *SM_NewSuperModel(Tag Tags, ...);

FUNCTION
    Allocate model object.

INPUTS
    Tags

RESULT

EXAMPLE

NOTES
    DisposeObject() returned model when done.

    Before targeted objects (ie gadgets) are freed you must either:
    1. Dispose() the SuperModel object, which also Dispose()s all
        SuperIC objects.
    2. SetAttr() ICA_TARGET to NULL on every SuperIC object.
    SuperIC objects need to clear the ICA_MAP and ICA_TARGET settings
    of it's targetted object.  If the target object nolonger exists,
    expect bad things to happen.

BUGS

SEE ALSO
```

1.10 supermodel.class/SM_SendGlueAttrs

```
NAME
    SM_SendGlueAttrs -- Send attributes from GlueFunc (SMA_GlueFunc)

SYNOPSIS
    unknown = SM_SendGlueAttrs(GlueData, TagList)
    d0                         a0        a1

    ULONG SM_SendGlueAttrs(struct smGlueData *, struct TagItem *);

FUNCTION
    This function sends TagList back to the modelclass for notification
    of other objects.

    ONLY to be called from inside a GlueFunction.

INPUTS
    GlueData -
    TagList -

RESULT

EXAMPLE

NOTES

BUGS

SEE ALSO
```

1.11 supermodel.class/SMTAG_AddTag

```
NAME
    SMTAG_AddTag -- Add a tag toa taglist.

SYNOPSIS
    ok = SMTAG_AddTag(TagList, Tag, Data)

    BOOL SMTAG_AddTag(struct TagItem *, ULONG, ULONG);

FUNCTION
    Add a tag pair to a taglist created with SMTAG_AllocTags()

INPUTS
    TagList - TagList created with SMTAG_AllocTags()
    Tag - ti_Tag value
    Data - ti_Data value

RESULT
    non zero if the tag was added.
    failure can be due to under sized taglist.

EXAMPLE
```

```
    see SMTAG_AllocTags()

NOTES
    Don't SMTAG_AddTag TAG_IGNORE, TAG_DONE, TAG_MORE, TAG_SKIP.
    This function will only effect the specified TagList, and
    not any other lists referenced by TAG_MORE.
    This function overwrites existing same tags.

SEE ALSO
    SMTAG_AllocTags()
```

1.12 supermodel.class/SMTAG_AddTags

```
NAME
    SMTAG_AddTags -- Add a taglist to a taglist.

SYNOPSIS
    ok = SMTAG_AddTags(TagList, Tag, Data)

    BOOL SMTAG_AddTags(struct TagItem *, ULONG, ULONG);

FUNCTION
    Add a taglist to a taglist created with SMTAG_AllocTags()

INPUTS
    TagList - TagList created with SMTAG_AllocTags()
    NewTags - Tags to add to TagList

RESULT
    non zero if the tag was added.
    failure can be due to under sized taglist.

EXAMPLE
    see SMTAG_AllocTags()

NOTES
    This function will only effect the specified TagList, and
    not any other lists referenced by TAG_MORE.
    This function overwrites existing same tags.

SEE ALSO
    SMTAG_AllocTags()
```

1.13 supermodel.class/SMTAG_AllocTags

```
NAME
    SMTAG_AllocTags -- Allocate blank Tag List

SYNOPSIS
    taglist = SMTAG_AllocTags(TagCount)

    struct TagItem *SMTAG_AllocTags(ULONG);
```

```
FUNCTION
    Allocate tag space for use with other SMTAG_? functions.

INPUTS
    TagCount - Number of blank tags to allocate.

RESULT
    An empty tag space ending with TAG_DONE, or NULL.

EXAMPLE

NOTES

BUGS

SEE ALSO
```

1.14 supermodel.class/SMTAG_ClearNumTags

```
NAME
    SMTAG_ClearTags -- Clear a TagList

SYNOPSIS
    void SMTAG_ClearTags(TagList, TagCount)

    SMTAG_ClearTags(struct TagItem *, ULONG);

FUNCTION
    Clears the TagList of all data.

INPUTS
    TagList - Allocated with SMTAG_AllocTags()
    TagCount - Number of blank tags to allocate.

EXAMPLE
    see SMTAG_AllocTags()

NOTES
    This function is called by SMTAG_AllocTags(), so
    the taglist is cleared when allocated.
    This function will only effect the specified TagList, and
    not any other lists referenced by TAG_MORE.

BUGS

SEE ALSO
    see SMTAG_AllocTags()
```

1.15 supermodel.class/SMTAG_ClearTags

```
NAME
    SMTAG_ClearTags -- Clear a TagList

SYNOPSIS
    void SMTAG_ClearTags(TagList)

    SMTAG_ClearTags(struct TagItem *);

FUNCTION
    Clears the TagList of all data.

INPUTS
    TagList - Allocated with SMTAG_AllocTags()

EXAMPLE
    see SMTAG_AllocTags()

NOTES
    This function will only effect the specified TagList, and
    not any other lists referenced by TAG_MORE.

BUGS

SEE ALSO
    see SMTAG_AllocTags()
```

1.16 supermodel.class/SMTAG_FreeTags

```
NAME
    SMTAG_FreeTags -- Clear a TagList

SYNOPSIS
    void SMTAG_FreeTags(TagList)

    SMTAG_FreeTags(struct TagItem *);

FUNCTION
    Frees the TagList.

INPUTS
    TagList - Allocated with SMTAG_AllocTags().

EXAMPLE
    see SMTAG_AllocTags()

SEE ALSO
    see SMTAG_AllocTags()
```

1.17 supermodel.class/SMTAG_RemTag

```
NAME
```

```
    SMTAG_RemTag -- Removea a tag to a taglist.

SYNOPSIS
    ok = SMTAG_RemTag(TagList, Tag)

    BOOL SMTAG_RemTag(struct TagItem *, ULONG);

FUNCTION
    Find and remove a tag from a taglist.

INPUTS
    TagList - TagList created with SMTAG_AllocTags()
    Tag - ti_Tag value

RESULT
    non zero if the tag was found and removed.

EXAMPLE
    see SMTAG_AllocTags()

NOTES
    Don't SMTAG_AddTag TAG_IGNORE, TAG_DONE, TAG_MORE, TAG_SKIP.
    This function will only effect the specified TagList, and
    not any other lists referenced by TAG_MORE.

SEE ALSO
    SMTAG_AllocTags()
```

1.18 supermodel.class/SMTAG_TagDone

```
NAME
    SMTAG_TagDone -- End the TagList with TagDone

SYNOPSIS
    void SMTAG_FreeTags(TagList)

    SMTAG_FreeTags(struct TagItem *);

FUNCTION
    Ends the taglist with TAG_MORE and link the list to MoreTags

INPUTS
    TagList - Allocated with SMTAG_AllocTags().

EXAMPLE
    see SMTAG_AllocTags()

SEE ALSO
    see SMTAG_AllocTags()
```

1.19 supermodel.class/SMTAG_TagMore

```
NAME
    SMTAG_TagMore -- End the TagList with TagMore

SYNOPSIS
    void SMTAG_FreeTags(TagList, MoreTags)

    SMTAG_FreeTags(struct TagItem *, struct TagItem *);

FUNCTION
    Ends the taglist with TAG_MORE and link the list to MoreTags

INPUTS
    TagList - Allocated with SMTAG_AllocTags().
    MoreTags - Tags to link

EXAMPLE
    see SMTAG_AllocTags()

SEE ALSO
    see SMTAG_AllocTags()
```