07a33da8-0

Deryk B Robosson

COLLABORATORS			
	TITLE : 07a33da8-0		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Deryk B Robosson	November 23, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 07a33da8-0 1

1.1 AFGadget 1

1.2 Gadget Object Information 2

1.3 AFGadget 2

1.4 ~AFGadget 2

1.5 Create 3

1.6 FillGadgetStruct 3

1.7 AddGadget 4

1.8 RemoveGadget 4

1.9 OnGadgetDown 5

1.10 OnGadgetUp 5

1.11 OnPaint 6

1.12 Misc Members 6

1.13 Includes 6

1.14 Source 7

1.15 History 9

1.16 Distribution 9

1.17 m_pgadget 10

1.18 m_pwindow 10

1.19 m_flags 11

Chapter 1

07a33da8-0

1.1 AFGadget

```
                      **********************************

                        Gadtet C++ Object

                        Intuition Gadget Wrapper

                        AFrame Version 1.0
                        Gadget Object Version 1.0

                        (c) 1995,1996 Jeffry A Worth
                                       Deryk B Robosson

                      **********************************
```

TABLE OF CONTENTS

```
  Gadget~Object~Information

  Methods:
    AFGadget
    ~AFGadget

    Create
    FillGadgetStruct
    AddGadget
    RemoveGadget

  Events:
    OnGadgetDown
    OnGadgetUp
    OnPaint

    Misc~Members

    Includes
    Source
    History
    Distribution
```

1.2 Gadget Object Information

```
AFGadget/Object Information

  The Gadget Object was written to provide a C++ Object Class interface for the  ←
     Intuition Gadget.  Gadget Management
with this class makes things much easier for the programmer.
```

1.3 AFGadget

```
AFGadget/AFGadget

  NAME AFGadget()

  DESCRIPTION
    Default class constructor.  Adds the gadget from the gadget list
    and window assigned to.  Modify this only if you wish each class to have
    the modifications

  INPUTS
    none

  RESULT
    none

  BUGS
    none known

  SEE ALSO
    ~AFGadget
```

1.4 ~AFGadget

```
AFGadget/~AFGadget

  NAME ~AFGadget()

  DESCRIPTION
    Default class destructor.  Removes the gadget from the gadget list
    and window assigned to.  Modify this only if you wish each class to have
    the modifications

  INPUTS
    none

  RESULT
    none
  BUGS
    none known

  SEE ALSO
    AFGadget
```

1.5 Create

```
AFGadget/Create

  NAME Create()

    void = Create(AFWindow* pwindow, AFRect *rect, ULONG id)

  DESCRIPTION
    Creates a gadget in pwindow with size and coordinates rect with GadID of id

  INPUTS
    pwindow - pointer to window to receive the status bar
    rect    - AFRect struct which shows size and position of gadget
    id      - gadget id

  RESULT
    none

  BUGS
    none known

  SEE ALSO
    AFWindow
    AFRect
```

1.6 FillGadgetStruct

```
AFGadget/FillGadgetStruct

  NAME FillGadgetStruct()

    void = FillGadgetStruct(LPGadget psgadget)

  DESCRIPTION
    Fills the gadget struct when AFGadget::Create is called with default
    values.  This function is used internally only

  INPUTS
    psgadget - pointer to the gadget struct to be filled

  RESULT
    none

  BUGS
    none known
  SEE ALSO
    LPGadget
```

1.7 AddGadget

```
AFGadget/AddGadget

  NAME AddGadget()

    void = AddGadget()

  DESCRIPTION
    Adds the gadget to the gadget list

  NOTE
    This function is used internally only

  INPUTS
    none

  RESULT
    none

  BUGS
    none known

  SEE ALSO
    RemoveGadget
```

1.8 RemoveGadget

```
AFGadget/RemoveGadget

  NAME RemoveGadget()

    void = RemoveGadget()

  DESCRIPTION
    Removes the gadget from the gadget list

  NOTE
    This function is used internally only

  INPUTS
    none

  RESULT
    none

  BUGS
    none known
  SEE ALSO
    AddGadget
```

1.9 OnGadgetDown

```
AFGadget/OnGadgetDown

  NAME OnGadgetDown()

    void = OnGadgetDown(LPIntuiMessage)

  DESCRIPTION
    This function calls the gadgets OnGadgetDown function.  You may override this
    function for your own purposes.

  INPUTS
    none

  RESULT
    none

  BUGS
    none known

  SEE ALSO
    OnGadgetUp
```

1.10 OnGadgetUp

```
AFGadget/OnGadgetUp

  NAME OnGadgetUp()

    void = OnGadgetUp(LPIntuiMessage)

  DESCRIPTION
    This function calls the gadgets OnGadgetUp function.  You may override this
    function for your own purposes.

  INPUTS
    none

  RESULT
    none

  BUGS
    none known

  SEE ALSO
    OnGadgetDown
```

1.11 OnPaint

```
AFGadget/OnPaint

  NAME OnPaint()

    void = OnPaint()

  DESCRIPTION
    This function calls the gadgets OnPaint function.  You may override this
    function for your own purposes.

  INPUTS
    none

  RESULT
    none

  BUGS
    none known

  SEE ALSO
```

1.12 Misc Members

```
AFGadget/Misc Members

    m_pgadget
    m_pwindow
    m_flags
```

1.13 Includes

```
  //////////////////////////////////////////////////////////////////////////////
  // gadget.hpp
  //
  // Jeffry A Worth
  // November 10, 1995
  //////////////////////////////////////////////////////////////////////////////

  #ifndef __GADGET_HPP__
  #define __GADGET_HPP__

  //////////////////////////////////////////////////////////////////////////////
  // INCLUDES
  #include "aframe:include/aframe.hpp"
  #include "aframe:include/window.hpp"
  #include "aframe:include/rect.hpp"

  //////////////////////////////////////////////////////////////////////////////
  // Definitions
```

```
#define AFGADGET_OWNERSTRUCT 0x80000000

///////////////////////////////////////////////////////////////////////////////
// Gadget Class

class AFGadget : public AFObject
{
public:
  AFGadget();
  ~AFGadget();

  virtual void DestroyObject();
  virtual char *ObjectType() { return "Gadget"; };

  virtual void Create(AFWindow* pwindow, AFRect *rect,ULONG id);
  virtual void Create(AFWindow* pwindow, LPGadget psgadget);
  virtual void FillGadgetStruct(LPGadget psgadget);
  virtual void AddGadget();
  virtual void RemoveGadget();

  // Events
  virtual void OnGadgetDown(LPIntuiMessage) { return; };
  virtual void OnGadgetUp(LPIntuiMessage) { return; };
  virtual void OnPaint() { return; };

  LPGadget m_pgadget;
  AFWindow *m_pwindow;
  ULONG m_flags;
};

///////////////////////////////////////////////////////////////////////////////
#endif // __GADGET_HPP__
```

1.14 Source

```
///////////////////////////////////////////////////////////////////////////////
// gadget.cpp
//
// Jeffry A Worth
// Nov 9, 1995
///////////////////////////////////////////////////////////////////////////////

///////////////////////////////////////////////////////////////////////////////
// INCLUDES
#include "aframe:include/gadget.hpp"

///////////////////////////////////////////////////////////////////////////////
//

AFGadget::AFGadget()
{
  m_pgadget = NULL;
  m_flags = NULL;
}
```

```
AFGadget::~AFGadget()
{
  // When the Window object is destroyed, all AFGadgets get their DestroyObject
  // method called.  To remove the CGadget from the window without destroying
  // the window, the user must call the DestroyObject method.
}

void AFGadget::DestroyObject()
{
  if(m_pgadget && (!(m_flags & AFGADGET_OWNERSTRUCT)) ) {
    RemoveGadget();
    delete m_pgadget;
    m_pgadget = NULL;
    m_flags = NULL;
  }
}

void AFGadget::Create(AFWindow* pwindow, AFRect *rect, ULONG id)
{
  m_pwindow = pwindow;
  m_pgadget = new struct Gadget;
  m_pgadget->NextGadget = NULL;
  m_pgadget->LeftEdge = rect->TopLeft()->m_x;
  m_pgadget->TopEdge = rect->TopLeft()->m_y;
  m_pgadget->Width = rect->Width();
  m_pgadget->Height = rect->Height();
  m_pgadget->GadgetID = id;
  FillGadgetStruct(m_pgadget);
  m_pgadget->UserData = this;

  // Gadget Tracking - Not implemented yet!
  //m_pwindow->m_pgadgets = new AFNode(m_pgadget,m_pwindow->m_pgadgets);

  AddGadget();
  return;
}

void AFGadget::Create(AFWindow* pwindow, LPGadget psgadget)
{
  // Set up OWNERSTRUCT Gadget
  m_pgadget = psgadget;
  m_pwindow = pwindow;
  m_flags |= AFGADGET_OWNERSTRUCT;
  psgadget->UserData = this;

  // Gadget Tracking - Not implemented yet!
  // m_pwindow->m_pgadgets = new CNode(m_pgadget,m_pwindow->m_pgadgets);

  AddGadget();
}

void AFGadget::FillGadgetStruct(LPGadget psgadget)
{
  psgadget->Flags = GFLG_GADGHCOMP;
  psgadget->Activation = GACT_RELVERIFY | GACT_IMMEDIATE;
  psgadget->GadgetType = GTYP_BOOLGADGET;
  psgadget->GadgetRender = NULL;
```

```
    psgadget->SelectRender = NULL;
    psgadget->GadgetText = NULL;
    psgadget->MutualExclude = NULL;
    psgadget->SpecialInfo = NULL;
    return;
  }

  void AFGadget::AddGadget()
  {
    ::AddGadget(m_pwindow->m_pWindow,m_pgadget,-1);
  }

  void AFGadget::RemoveGadget()
  {
    ::RemoveGadget(m_pwindow->m_pWindow,m_pgadget);
  }
```

1.15 History

```
                     ***HISTORY***

 ***********************************************************
  Gadget Object v1.0

  Created November 10, 1995
  Release November 11, 1995

   - Created all function class objects.
 ***********************************************************
```

1.16 Distribution

Distribution

The programs and files in this distribution are freely distributable, but are also Copyright (c) Jeff Worth and Deryk Robosson. They may be freely distributed as long as no more than a nominal fee is charged to cover time and copying costs. AFrame is distributed as non-crippled shareware, it is fully functional.

Commercial Distribution

Commercial usage is allowed if the following conditions are met:

```
    a) You state in your documentation that your program uses aframe.library
       and that AFrame is Copyright (c) Jeff Worth and Deryk Robosson.

    b) You send us a copy of your finished product(s) using aframe.library.
```

If these conditions are met you are allowed to include the Kickstart 2.0 or higher version of aframe.library and the installation script(s) with your commercial product.

Freely Distibutable Products

All of the files copyrighted by the authors must remain unmodified. None of these files may be distributed on its
own, the entire package must be distributed as one whole. 'demo.cpp' is full public domain and can be used in any
way you like.

There is one exception to the above. If you plan to release a freely distributable program (either public domain,
freeware or shareware), you may include 'libs/aframe.library', the installation scripts (with icon), the
documentation (with icons) with your distribution.

If you include AFrame with a crippled shareware program I'd like to ask you to send us a full working version.

Whether your program is freely distributable or commercial, you must state in your documentation that your program
uses aframe.library and that AFrame is Copyright (c) Jeff Worth and Deryk Robosson

1.17 m_pgadget

```
AFGadget/m_pgadget

  MEMBER TYPE
   LPGadget

  DESCRIPTION
   Used to maintain a pointer to the gadget object

  SEE ALSO
   LPGadget
```

1.18 m_pwindow

```
AFGadget/m_pwindow

  MEMBER TYPE
   AFWindow

  DESCRIPTION
   Used to maintain a pointer to the parent window in which the
   gadget object resides

  SEE ALSO
   AFWindow
```

1.19 m_flags

```
AFGadget/m_flags

  MEMBER TYPE
   ULONG

  DESCRIPTION
   Contains the flags for each gadget class object

  SEE ALSO
```