
Project: Application initiated drag&drop

Revision:

A

Author: Björn Hagström

Revision Date:

20090419

<http://openamiga.org/?function=viewproject&projectid=49>

A proposal for implementing

Application initiated drag&drop

An OpenAmiga.org project

Project: Application initiated drag&drop

Revision: A

Author: Björn Hagström

Revision Date: 20090419

<http://openamiga.org/?function=viewproject&projectid=49>

Content

1	Application initiated drag&drop.....	3
1.1	Revision history.....	3
1.2	Introduction.....	4
1.3	Definitions.....	5
2	Project.....	5
2.1	Required goals.....	5
2.2	Optional future goals.....	5
2.3	Description.....	6
2.4	Example of use.....	7
2.5	Possible Caveats.....	8
2.6	Default icon images.....	8
3	Autodoc entry.....	11

Project: Application initiated drag&drop

Revision:

A

Author: Björn Hagström

Revision Date:

20090419

<http://openamiga.org/?function=viewproject&projectid=49>

1 Application initiated drag&drop

1.1 Revision history

Description	Author	Date	Revision
Document created	Björn Hagström	2009-03-26	PA1
Updated so that several Drop Clips can be sent at the same time.	Björn Hagström	2009-03-27	PA2
Added suggested filenames as requested by Chris Young. Added further information about gadget drops in optional future goals.	Björn Hagström	2009-04-14	PA3
Added default drag&drop images from Martin Merz	Björn Hagström	2009-04-18	PA4
Approved issue	Björn Hagström	2009-04-19	A

Project: Application initiated drag&drop

Revision:

A

Author: Björn Hagström

Revision Date:

20090419

<http://openamiga.org/?function=viewproject&projectid=49>

1.2 Introduction

Drag&Drop is currently available in OS4 in the form of workbench being able to send Appmessages to AppWindows and AppIcons when the user has drag&dropped one or more icons onto them.

However there is currently no way for applications to initiate Appmessages in a similar manner, which limits the current drag&drop functionality of OS4.

This document aims to describe a way for applications to initiate such events and how workbench.library could be updated to support it, as well as providing suggestions for possible future functionality that can be added later on.

Note that the main part of this document only describes the functionality in broad terms, and if questions surface while reading the descriptive parts then please refer to the autodoc entry in the last chapter for detailed information.

Also since not all internal workings of the Workbench is known to the author this suggestion should not be interpreted as an absolute solution, that is, details may be altered as needed while considering and/or implementing this feature.

This document is mainly intended for developers internal to Hyperion.

More information can be found at:

<http://openamiga.org/index.php?function=viewproject&projectid=49>

Information about the OpenAmiga.org organization can be found at

<http://openamiga.org>

Project: Application initiated drag&drop

Revision:

A

Author: Björn Hagström

Revision Date:

20090419

<http://openamiga.org/?function=viewproject&projectid=49>

1.3 Definitions

Drop Clip

A file containing data associated with a drag&drop event. It can either be a binary file with unknown contents or, like the clipboard function, be an IFF file.

2 Project

2.1 Required goals

- Being able to drag&drop arbitrary data between applications.
- Being able to create files by dragging into a workbench drawer window.
- Providing a safe way to identify the contents of a drag&drop event.
- Must fit into the current framework without breaking backward compatibility.

2.2 Optional future goals

These are desired features that may be added to the drag&drop functionality in the future but that does not have to be a part of the initial implementation.

Short term

Creating a set of default icon images that will be displayed while the drag is in progress. Applications would be able to tell workbench which default icon that should be used and workbench will load the appropriate icon from disk.

Long term

Updating 'Reaction'/gadgets to detect that a drag motion has started and call upon workbench to start the drag event. The purpose would be to allow the user to drag things like text between text gadgets and to reduce the need for developers to reinvent drag motion detection over and over.

An alternative could be to let individual gadgets register if a IDCMP message should be sent to the application each time a drag motion is detected over it, ie left mouse button is pressed on top of the

Project: Application initiated drag&drop

Revision:

A

Author: Björn Hagström

Revision Date:

20090419

<http://openamiga.org/?function=viewproject&projectid=49>

gadget and the mouse moves. This message should only be sent once while the left mouse button is pressed.

It would also be desirable to be able to identify the Gadget that a Drop is made on, automatically or either by defining gadget drop zones or the gadgets themselves registering their drop zones. The AppMessage could then contain two new fields.

```
ULONG am_GadgetID    /* Gadget ID relevant to window */
LONG  am_GadgetData  /* Gadget specific data, see below */
```

For a listbrowser `am_GadgetID` would contain its `GA_ID` and the `am_GadgetData` would contain the listbrowser node index that the drop was made on. To realize this the `ListBrowser` node would have a new `LBNA_Dropable` (BOOL) field, default FALSE. When set to TRUE `am_GadgetID` would contain the listbrowser node index and when set to FALSE it would be set to -1 to identify the `Listbrowser` itself.

2.3 Description

To implement this suggestion a lot of existing workbench functionality can be reused, but it also requires new additions to support it in full.

It is suggested that AppMessages are used in the same way as for workbench initiated drag&drop events but the current AppMessage structure is however a bit limited and should be extended to support a wider range of options.

New AppMessage fields:

```
struct Window *am_Win; /* the original window that sent the message, used */
                  /* as ID only, do not use for any other purpose! */
WORD  *am_DataTypes; /* type of data for each entry in am_ArgList */
ULONG  am_DragType;  /* application specific data */
STRPTR *am_Names;    /* Suggested filenames */
```

am_Win is an optional window identifier. Applications can use it to figure out if the event was initiated from one of its own windows.

am_DataTypes tells the application what type of data is pointed to by each entry in `am_ArgList`. It can be either in plain binary form or be packaged as an IFF structure with identifiable contents. The reason for providing a binary form is that some applications might want to let the user drag some data that

Project: Application initiated drag&drop

Revision:

A

Author: Björn Hagström

Revision Date:

20090419

<http://openamiga.org/?function=viewproject&projectid=49>

already has a specific type directly to an application that can read the data, such as a PDF file into a PDF reader. Dragging an IFF file into the receiving application would not be suitable in this case as it would need to be updated to handle it.

am_DragType can be used to define the type of drag event so that the receiver knows how to deal with it. Values below 1024 would be reserved for specific actions such as a Move, Copy or Gadget events.

Some reserved values:

- Move - The initiator envisioned that the dragged data should be moved
- Copy - The initiator envisioned that the dragged data should be copied
- Gadget - The initiator envisioned that the data would be copied into a gadget

am_Names is an optional array containing suggested filenames for each DropClip. If no names were provided by the originating process then it is set to NULL. Note that individual entries may be set to NULL as well. Workbench will not use these names when storing the DropClips in RAM: these are only intended as suggested filenames to be used by a receiving process when it in turn stores one or more DropClips to disk.

2.4 Example of use

To initiate a drag event an application would first need to identify that the user is trying to drag something in a window by checking the state of the left mouse button and the mouse movement.

Once identified the application would assemble the data it wants to send and then call a workbench function to start the event. Workbench takes over by acting much like as if the user had marked some icons and started moving them. First one or more Drop Clips will be created with the data the application wanted to send in RAM: much like using the clipboard. During dragging the workbench will display an icon defined by the calling application and once the left mouse button is released it will identify if it was released on top of a AppWindow/Appicon and dispatch an AppMessage to that AppWindow/AppIcon.

The receiving application checks the contents of the message and acts appropriately. In this case it will read the ArgList and open the file or files it points to (the Drop Clips). Once it calls Reply() the Drop Clips will be removed from storage.

Project: Application initiated drag&drop

Revision: A

Author: Björn Hagström

Revision Date: 20090419

<http://openamiga.org/?function=viewproject&projectid=49>

Note that applications not familiar with the new data in the AppMessage would not need to be updated. They will identify the AppMessage as either a list of files or an deiconify event just as before. If a drop is made in a window that is not an AppWindow it is discarded and the Drop Clips are removed.

2.5 Possible Caveats

Since the Drop Clips are intended to be removed once Reply() is called, Workbench might have to be updated depending on how it internally handles the Reply() call. The Drop Clips must be read/copied/moved before Reply() is called. The same applies to existing applications.

To counter-effect this the Drop Clip removal on Reply() could be skipped when implementing this, but with the risk of leaving the Drop Clip in ram: longer than necessary.

The autodoc entry mentions "struct IFFHandle *" as the pointer type used for drag&drop data. This may not be the appropriate way of indicating an IFF structure to the workbench and may be replaced with some other pointer type if needed.

2.6 Default icon images

(Possible short term future addition)

Instead of providing a DiskObject pointer the application can select among a set of default images. These would be defined in an enum with values below 1024. Workbench would load these as needed from a predefined location.

The images below are available in the OpenAmiga.org SVN.

DC_IMG_DEFAULT // Default image



DC_IMG_SINGLE // Generic single item



DC_IMG_MULTI // Generic multiple items



Project: Application initiated drag&drop

Revision:

A

Author: Björn Hagström

Revision Date:

20090419

<http://openamiga.org/?function=viewproject&projectid=49>

DC_IMG_OUTLINE // Empty square



DC_IMG_OUTLINE_M // Several empty squares



DC_IMG_FILE // A single file



DC_IMG_FILE_M // Several files



DC_IMG_DRAWER // A single drawer



DC_IMG_DRAWER_M // Several drawers



DC_IMG_DOCUMENT // A document



DC_IMG_DOCUMENT_M // Several documents



DC_IMG_IMAGE // An image



DC_IMG_IMAGE_M // Several images



DC_IMG_AUDIO // An audio clip



Project: Application initiated drag&drop

Revision:

A

Author: Björn Hagström

Revision Date:

20090419

<http://openamiga.org/?function=viewproject&projectid=49>

DC_IMG_AUDIO_M // Several audio clips



DC_IMG_VIDEO // A video clip



DC_IMG_VIDEO_M // Several video clips



DC_IMG_TEXT // A text clip



DC_IMG_TEXT_M // Several text clips



Project: Application initiated drag&drop

Revision: A

Author: Björn Hagström

Revision Date: 20090419

<http://openamiga.org/?function=viewproject&projectid=49>

3 Autodoc entry

New methods in workbench.library:

```
int32 CreateDragContextTagList(const struct TagItem *tags);
int32 CreateDragContextTags(uint32 tag1, ...);
```

Tells the Workbench that it should enter drag&drop mode. Once the left mouse button is released the Workbench will notify the AppWindow/AppIcon at that mouse position with an AppMessage, even if it's the same AppWindow/Appicon as the originating one. The call will not return to the calling application until the user releases the left mouse button.

When using DC_DropClips one or more Drop Clip files containing the supplied data will be created. The resulting Drop Clip files will be removed once the receiving application calls IDOS->Reply(). If no reply is sent the files will be overwritten the next time IWorkbench->CreateDragContextAttrs() is called. If the receiver is not an AppIcon/AppWindow then no file will be created and any previously existing files will be removed.

If you want to drag and drop from windows in the same application but neither want to use the DC_DropClips method nor use a list of file names then set DC_DropClips to NULL, DC_NumArgs to 0 and DC_ArgList to NULL and then simply use am_Win in the AppMessage to determine from which of your own windows the drag motion was generated and collect whatever data you need internally.

Tags:

DC_Icon (struct DiskObject *)

The icon image that will be used while dragging. If not specified then a default image will be used. Default NULL. A value below 1024 will be treated as NULL.

(Optional: Possible future addition, to create a common set of drag icon images for all applications to use. Icon files could be stored in sys:prefs/presets/icons/ or similar.

Default icon images:

```
enum
{
    DC_IMG_DEFAULT=0,    // Default image
    DC_IMG_SINGLE,      // Generic single item
    DC_IMG_MULTI,       // Generic multiple items
    DC_IMG_OUTLINE,     // Empty square
    DC_IMG_OUTLINE_M,   // Several empty squares
    DC_IMG_FILE,        // A single file
    DC_IMG_FILE_M,      // Several files
    DC_IMG_DRAWER,      // A single drawer
}
```

Project: Application initiated drag&drop

Revision:

A

Author: Björn Hagström

Revision Date:

20090419

<http://openamiga.org/?function=viewproject&projectid=49>

```
DC_IMG_DRAWER_M,    // Several drawers
DC_IMG_DOCUMENT,   // A document
DC_IMG_DOCUMENT_M, // Several documents
DC_IMG_IMAGE,      // An image
DC_IMG_IMAGE_M,    // Several images
DC_IMG_AUDIO,      // An audio clip
DC_IMG_AUDIO_M,    // Several audio clips
DC_IMG_VIDEO,      // A video clip
DC_IMG_VIDEO_M,    // Several video clips
DC_IMG_TEXT,       // A text clip
DC_IMG_TEXT_M,     // Several text clips
```

```
/* etc, undefined values below 1024 will be treated as DC_IMG_DEFAULT to
provide backward compability in case more types are added in the
future. */
```

```
};
```

DC_WindowID (struct Window *)

Window identifier, this identifier is available to the receiving application in the AppMessage.am_Win field. Default NULL.

DC_DropClips (struct DropClip **)

Pointer to an array of struct DropClip pointers. Each pointing to data that will be sent to the receiving application. Default NULL.

```
/* Struct DropClip contents */
```

```
struct DropClip
```

```
{
```

```
  APTR dc_Data;    /* Pointer to arbitrary data or struct IFFHandle */
```

```
  WORD dc_DataType; /* DC_DATATYPE_DROPCLIP or DC_DATATYPE_BINARY */
```

```
  ULONG dc_DataSize; /* Byte length of the data pointed to by dc_Data */
```

```
  STRPTR dc_Name, /* Suggested filename for the DropClip */
```

```
};
```

DC_DATATYPE_DROPCLIP

dc_Data will be interpreted as a struct IFFHandle * that points to an IFF structure containing the data that will be written to the Drop Clip file.

DC_DATATYPE_BINARY

dc_Data will be interpreted as a byte array of arbitrary data with no known internal structure. The data will be written as is to the Drop Clip file using the dc_DataSize to determine the length of the data.

dc_Name contains a suggested filename and can be set to NULL. This allows the original process to suggest a filename for each DropClip to the receiving process.

DC_NumArgs (LONG)

Number of Drop Clips in DC_DropClips.

Default 0.

Project: Application initiated drag&drop

Revision:

A

Author: Björn Hagström

Revision Date:

20090419

<http://openamiga.org/?function=viewproject&projectid=49>

DC_ArgList (struct WBArg *)

Contains valid filenames to be sent to the receiving appwindow/appicon. Note that if DC_DropClips is set then DC_ArgList will be ignored and the resulting AppMessage.am_ArgList will contain a list of paths pointing to the drop clip files generated by Workbench. "T:dropclip.1", "T:dropclip.2" etc.

When DC_ArgList is used but DC_DropClips is set to NULL the resulting AppMessage.am_DataTypes array will all be DC_DATATYPE_BINARY.

Default NULL.

DC_DragType (ULONG)

Application specific information. Can be used by the sending window and receiving window to communicate what kind of drag event this is. This is available to the receiving application in the AppMessage.am_DragType field. Use with care, remember that other applications may put any value in this field. DC_DragType enum contains predefined reserved values, using values not defined in this enum below 1024 is discouraged.

Default 0.

```
/* struct AppMessage contents */
struct AppMessage
{
    struct Message am_Message; /* standard message structure */
    UWORD am_Type; /* message type */
    ULONG am_UserData; /* application specific */
    ULONG am_ID; /* application definable ID */
    LONG am_NumArgs; /* # of elements in arglist */
    struct WBArg * am_ArgList; /* the arguments themselves */
    UWORD am_Version; /* will be >= AM_VERSION */
    UWORD am_Class; /* message class */
    WORD am_MouseX; /* mouse x position of event */
    WORD am_MouseY; /* mouse y position of event */
    ULONG am_Seconds; /* current system clock time */
    ULONG am_Micros; /* current system clock time */
    ULONG am_Reserved[8]; /* avoid recompilation */

    // From 54.xx, fields added to handle application generated drag events
    struct Window *am_Win; /* the original window that sent the message, used */
    /* as ID only, do not use for any other purpose! */
    WORD *am_DataTypes; /* type of data for each file pointed to */
    /* by am_ArgList */
    ULONG am_DragType; /* application specific data */
    STRPTR *am_Names; /* Suggested filenames */
}

#define DC_DROPCLIP "T:dropclip" /* The drop clip data will be stored */
/* here but with a sequence number after it*/
```

Project: Application initiated drag&drop

Revision:

A

Author: Björn Hagström

Revision Date:

20090419

<http://openamiga.org/?function=viewproject&projectid=49>

```
/* depending on its position in the */  
/* am_ArgList. */
```

```
/* DC_DataType enum */  
enum  
{  
    DC_DATATYPE_DROPCLIP=0,  
    DC_DATATYPE_BINARY  
};
```

```
/* DC_DragType enum */  
enum  
{  
    DC_DRAGTYPE_NOTYPE=0,  
    DC_DRAGTYPE_COPY,  
    DC_DRAGTYPE_MOVE,  
    DC_DRAGTYPE_GADGET  
};
```

Result:
Boolean, FALSE on failure.