

# **AmigaGuide**

**COLLABORATORS**

	<i>TITLE :</i> AmigaGuide		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 12, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>AmigaGuide</b>	<b>1</b>
1.1	binhaltdemo . . . . .	1

## Chapter 1

# AmigaGuide

### 1.1 binhaltdemo

```
/*-----*/
/* Programm: BInhalt List          */
/*-----*/
/* Funktion: Gibt eine Fileliste mit Hilfe */
/*           der AmBoSData.Library aus    */
/*-----*/

/*----- Standart Includes -----*/

#include <exec/exec.h>
#include <utility/tagitem.h>
#include <dos/dos.h>

#include <clib/exec_protos.h>
#include <clib/dos_protos.h>
#include <pragmas/exec_pragmas.h>
#include <pragmas/dos_pragmas.h>
/*-----*/

/*----- AmBoSDataLibrary Includes -----*/

#include <clib/AmBoSData_protos.h>
#include <libraries/AmBoSTags.h>

/*-----*/

/* für die Dos.Library          */
/* kommt von der c.o ;-) bei SAS */

extern struct Library *DOSBase;

/* AmBoSDataLibrary öffnen */

struct Library *ADL_Open(void)
{
    AmBoSDataBase = OpenLibrary("ambosdata.library", 0L);
    return (AmBoSDataBase);
}
```

```

/* AmBoSDataLibrary schließen */

void ADL_Close(void)
{
    if (AmBoSDataBase)
        CloseLibrary (AmBoSDataBase);
}
/*-----*/
/* Funktion: Besorgt das nächste Brett */
/*-----*/
/* Parameter:                               */
/*   BrettHandle das aktuellen Brettes*/
/* Rückgabe :                               */
/*   Das neue Brett oder NULL, für        */
/*   Ende oder Fehler                      */
/*-----*/

APTR GetNextBrett (APTR BrettHandle)
{
    APTR Temp;          /* Für das neue Brett */
    ULONG Art;         /* Für den Brett Typ */

    if (! BrettHandle) /* Haben wir denn was übergeben */
        return (NULL); /* Nein, dann zurück      */

    /* Brett Type holen */

    if (AGetAttrTag (BrettHandle, BRETT_BrettArt, &Art, TAG_DONE) != TRUE)
        return (NULL); /* Keinen bekommen, dann Zurück */

    if ((Art == BRETTART_Head) || (Art == BRETTART_ExternHead))
    {
        /*-----*/
        /* Ist das Brett ein Kopf, dann Verzweigen */
        /*-----*/

        Temp = (APTR) ADataSetTag (DATASET_Sub, BrettHandle, TAG_DONE);
        if (Temp)
        {
            /* Neues Brett zurück, Vorher altes freigeben */

            ADataSetTag (DATASET_Free, BrettHandle, TAG_DONE);
            return (Temp);
        }
    }
    /*-----*/
    /* Ist es wohl doch nur ein normales Brett */
    /*-----*/

    Temp = (APTR) ADataSetTag (DATASET_Next, BrettHandle, TAG_DONE);
    if (Temp)
    {
        /* Neues Brett zurück, altes freigeben */

        ADataSetTag (DATASET_Free, BrettHandle, TAG_DONE);
        return (Temp);
    }
}

```

---

```

}

/* Mhmm, war wohl das letzte */
/*-----*/
/* Versuchen wir mal ein höheres zu erhalten */
/*-----*/
while(1)
{
    /* Oberbrett holen, altes freigeben */

    BrettHandle = (APTR)ADatasetTag(DATASET_Parent, BrettHandle,
                                    DATASET_Free, BrettHandle, TAG_DONE);

    /* Haben wir den Kopf erhalten ? */

    if(! BrettHandle)
        return(NULL); /* Nein, dann sind wir wohl am Ende */
    /*-----*/
    /* Doch wir haben einen, dann nächstes Brett holen */
    /*-----*/
    Temp = (APTR)ADatasetTag(DATASET_Next, BrettHandle, TAG_DONE);
    if(Temp)
    {
        /* Das neue Brett zurück, altes löschen */

        ADatasetTag(DATASET_Free, BrettHandle, TAG_DONE);
        return(Temp);
    }
} /*:-):-):-):-):-):-):-):-):-):-)*/
return(NULL); /* Wir laufen wohl auf einem PC */
} /*:-):-):-):-):-):-):-):-):-):-)*/

/*-----*/
/* Funktion: Ausgabe des Pfads & Namen */
/*-----*/
/* Parameter: */
/*          Pfad, Namen, Art */
/* Rückgabe : */
/*          Keine */
/*-----*/

void PrintFile(char *Pfad, char *Name, ULONG art)
{
    char temp[44];
    short len = 0, len1 = 0;
    char leer[] = " ";

    while((len1 <= 40) && (Pfad[len1] != 0))
    {
        temp[len1]=Pfad[len1];
        len1++;
    }
    if(len1 < 40)
    {
        temp[len1] = '/';
        temp[len1+1] = 0;
    }
}

```

```
    Fputs(Output(),temp);
}
else
{

    temp[18]=0;
    Fputs(Output(),temp);
    Fputs(Output(),"...");
    len1 = 20;
    len = len1;

    while(Pfad[len] != 0)
    {
        len++;
    }
    len = len - 17;
    if(len < len1)
    {
        len = len1;
    }

    Fputs(Output(), &Pfad[len]);
    Fputs(Output(), "/");
}

len = 0;
while((len < 25) && (Name[len] != 0))
{
    temp[len] = Name[len];
    len++;
}
temp[len] = 0;
Fputs(Output(),Name);
Fputs(Output(),&leer[len]);

if(art == 2)
    Fputs(Output()," Nachricht");
if(art == 3)
    Fputs(Output()," Bin-File");

Fputs(Output(),"\n");
}

/*-----*/
/* Funktion: Hauptteil der Demo          */
/*-----*/
/* Parameter:                             */
/*      Keine                             */
/* Rückgabe :                             */
/*      Immer NULL                        */
/*-----*/

int main(void)
{
    APTR Brett;      /* Für das Brett          */
    APTR File;      /* Für den Brettinhalt */
}
```

```

ULONG Art;          /* Für den Type          */
char text[500];     /* Für den Netzpfad      */
char ftext[40];     /* Für BInhalt-Namen    */

/*-----*/
/* Eigene Routine, öffnet die AmBoSData.Library */
/*-----*/
if(!ADL_Open())
{
    /* Das öffnen ist fehlgeschlagen */
    Fputs(Output(),(char *)"Ohne Ambos wird das nichts ! :-)\n");
    return (NULL);
}

/*-----*/
/*          Hole erstes Brett          */
/*-----*/
Brett = ADataSetTag(DATASET_First,DATATYPE_Brett,TAG_DONE);
if(! Brett)
{
    /* Mhmm, das hat wohl nicht geklappt */
    Fputs(Output(),(char *)"Gibt's in dieser Box Bretter ? ;-(\n");
}

while(Brett != NULL)
{
    /*-----*/
    /*      Hole Netzpfad, Text und Brett-Type      */
    /*-----*/
    if (AGetAttrTag(Brett,BRETT_NetzPfad,text,BRETT_BrettArt,&Art,TAG_DONE) == TRUE)
    {
        /*-----*/
        /*      Puuuuhhh, das hätte ja schonmal geklappt      */
        /*-----*/
        if((Art != BRETTART_Head) && (Art != BRETTART_ExternHead))
        {
            /*-----*/
            /* Wenn es ein normales Brett ist, dann ausgeben*/
            /*-----*/

            /* BrettInhalt Objekt holen */
            File = ADataSetTag(DATASET_Sub,Brett,TAG_DONE);
            while(File != NULL)
            {
                /* Hat geklappt, dann Name und Art besorgen */
                if (AGetAttrTag(File,BRETTINHALT_Name,ftext,BRETTINHALT_Art,&Art,TAG_DONE) == TRUE)
                {
                    /* auch OK , dann ausgeben */
                    PrintFile(text,ftext,Art);
                }
                /* das Nächste File, und das letzte freigeben */
                File = ADataSetTag(DATASET_Next,File,DATASET_Free,File,TAG_DONE);
            }

            if(CheckSignal(SIGBREAKF_CTRL_C)) /* CTRL&C Checken */
            {
                ADataSetTag(DATASET_Free,File,TAG_DONE);
            }
        }
    }
}

```



```
    ADataSetTag(DATASET_Free, Brett, TAG_DONE);
    ADL_Close();
    Fputs(Output(), "\nTschau, muß das schon sein ?\n");
    return (NULL);
}
}
if (CheckSignal(SIGBREAKF_CTRL_C) /* CTRL&C Checken */
{
    ADataSetTag(DATASET_Free, Brett, TAG_DONE);
    ADL_Close();
    Fputs(Output(), "\nTschau, muß das schon sein ?\n");
    return (NULL);
}
}
}
/*-----*/
/* Jetzt wollen wir mal das nächste Brett holen */
/*-----*/
Brett = GetNextBrett(Brett);
}
ADL_Close(); /* Und diese schoene Routine schliesst die
              AmBoSDataLibrary wieder */
Fputs(Output(), (char *) "\nDas wars denn !\n");
return (NULL);
}
```

Hauptmenü

Wollen wir mal das Demo Starten ?

Bitte klicken Sie auf Zurück, um Weiter zu machen !