# pLocaleD

| COLLABORATORS | | | |
|---|---|---|---|
| | *TITLE* :<br><br>pLocaleD | | |
| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
| WRITTEN BY | | March 29, 2025 | |

| REVISION HISTORY | | | |
|---|---|---|---|
| NUMBER | DATE | DESCRIPTION | NAME |
| | | | |

# Contents

# Chapter 1

# pLocaleD

## 1.1 pLocaleD.doc

```
                              plocale.library

FormatDate()          FormatDateStr()      GetCatalogStr()
GetLocaleStr()        LocaleAllgemeines()  LocaleStruktur()
pOS_CloseCatalog()    pOS_CloseLocale()    pOS_ConvToLower()
pOS_ConvToUpper()     pOS_FormatDate()     pOS_FormatDateStr()
pOS_GetCatalogStr()   pOS_GetLocaleStr()   pOS_IsXXXX()
pOS_LocaleToLower()   pOS_LocaleToUpper()  pOS_OpenCatalog()
pOS_OpenCatalogA()    pOS_OpenLocale()     pOS_StrnCmp()
pOS_StrniCmp()
```

## 1.2 plocale.library/LocaleAllgemeines()

```
pOS Locale.library ist in der Lage, normale Amiga-Kataloge
zu verarbeiten. Es können also weiterhin die bisher ver-
wendeten Tools benutzt werden.

Neue Katalogcompiler sollten einen weiteren IFF-Chunk
generieren und ausfüllen. Dieser sollte die Versionsnummer
des Kataloges enthalten. Die Chunk-ID ist 'NVER'. Die
Versionnummer ist als UWORD zu sichern.

    Chunky-ID: 'NVER'
    Länge    : 2
    Inhalt   : UWORD Versionnummer des Kataloges


Zu einem späteren Zeitpunkt wird es zusätzlich ein Klartext-
Katalogformat geben.
```

## 1.3 plocale.library/LocaleStruktur()

```
Die Locale-Struktur ist öffentlich und darf gelesen werden.
Sie beinhaltet folgende Elemente:

const CHAR* loc_LocaleName
    Der Name.

const CHAR* loc_LanguageName
    Die Sprache, welche mit dieser Locale verbunden ist.

const CHAR* loc_PrefLanguages[10]
    Bevorzugte Sprachen.

ULONG loc_Flags
    Zur Zeit immer 0.

ULONG loc_CodeSet
    Zur Zeit immer 0.

const CHAR* loc_CountryCode
    Internationale Länderabkürzung (z.B. "D" für Deutschland).

ULONG loc_TelephoneCode
    Internationale Vorwahl.

LONG loc_GMTOffset
    Abstand zur GMT.

UBYTE loc_MeasuringSystem
    Das Maßsystem (enum pOS_MeasuringSystemConstants).

UBYTE loc_CalendarType.
    Die Art des Kalenders (enum pOS_CalendarTypeConstants).

const CHAR* loc_DateTimeFormat
    Datum & Zeit-Format-Angabe.

const CHAR* loc_DateFormat
    Datums-Format-Angabe.

const CHAR* loc_TimeFormat
    Zeit-Format-Angabe.

const CHAR* loc_ShortDateTimeFormat
    Abgekürzte Datum & Zeit-Format-Angabe.

const CHAR* loc_ShortDateFormat
    Abgekürzte Datums-Format-Angabe.

const CHAR* loc_ShortTimeFormat
    Abgekürzte Zeit-Format-Angabe.

const CHAR* loc_DecimalPoint
    Zeichen zum Trennen von Kommastellen.

const CHAR* loc_GroupSeparator
    Zeichen zum Trennen von Zahlengruppen vor dem Komma.
```

```
    const CHAR* loc_FracGroupSeparator
        Zeichen zum Trennen von Zahlengruppen nach dem Komma.

    const CHAR* loc_MonDecimalPoint
        Zeichen zum Trennen von Kommastellen bei Geldangaben.

    const CHAR* loc_MonGroupSeparator
        Zeichen zum Trennen von Zahlengruppen vor dem Komma bei
        Geldangaben.

    const CHAR* loc_MonFracGroupSeparator
        Zeichen zum Trennen von Zahlengruppen nach dem Komma bei
        Geldangaben.

    const CHAR* loc_CurrencySymbol
        Die Abkürzung der Währung im Landesformat.

    const CHAR* loc_SmallCurrencySymbol
        Die Abkürzung des Kleingeldes im Landesformat.

    const CHAR* loc_IntCurrencySymbol
        Die internationale Abkürzung der Währung nach ISO 4217.
```

## 1.4 plocale.library/pOS_CloseCatalog()

```
NAME
    pOS_CloseCatalog -- Schließen eines Katalogs.

SYNOPSIS
    pOS_CloseCatalog(catalog);
                     _R_A0

    VOID pOS_CloseCatalog(__ARID__ pOS_Catalog *);

FUNCTION
    Der Katalog wird geschlossen und der OpenCounter erniedrigt.
    Falls niemand den Katalog weiter benutzt, wird er aus dem
    Speicher entfernt.

INPUTS
    catalog - Zeiger auf den zu schliessenden Katalog.

SEE ALSO
    pOS_OpenCatalog(), pOS_GetCatalogStr()
```

## 1.5 plocale.library/pOS_CloseLocale()

```
NAME
    pOS_CloseLocale -- Schließen einer Locale-Beschreibung.
```

```
SYNOPSIS
     pOS_CloseLocale(locale);
                     _R_A0

     VOID pOS_CloseLocale(__ARID__ pOS_Locale *);

FUNCTION
     Schließt die Locale-Beschreibung und die zugehörige Sprach-
     beschreibung.

INPUTS
     locale - Zeiger auf das zu schließende Locale.

SEE ALSO
     pOS_OpenLocale()
```

## 1.6 plocale.library/pOS_FormatDate()

```
NAME
     pOS_FormatDate -- Erzeugen einer Datum/Zeit-Angabe aus der
       vorgegebenen Formatierung.

SYNOPSIS
     pOS_FormatDate(locale,fmtTemplate,date ,putCharFunc);
                    _R_A0  _R_A1       _R_A2 _R_A3

     VOID FormatDate(pOS_Locale*,const CHAR*,pOS_DateStamp*,
                     pOS_Callback*);

FUNCTION
     Aus der vorgegebenen Formatierung wird ein Datum/Zeit-Angabe
     erzeugt, wobei jeder einzelne Buchstabe an die Callback-
     Funktion übergeben wird.

INPUTS
     locale - die zur Formatierung zu verwendene Locale oder
              NULL für die System-Default-Locale.
     fmtTemplate - Ein mit NULL abgeschlossener String, welcher
                   das Format beschreibt. Die Formatanweisung
                   ist identisch der printf()-Formatanweisung.
                   Formatkodes beginnen mit einem %-Zeichen gefolgt
                   vom einem Formatkommando. Folgende Kommandos
                   werden von pOS_FormatDate verstanden:

            %a - abgekürzter Wochentags-Name
            %A - Wochentags-Name
            %b - Monats-Name
            %B - abgekürzter Monats-Name
            %c - Formatierung aus 'Locale->loc_ShortDateTimeFormat'
            %C - Formatierung aus 'Locale->loc_DateTimeFormat'
            %d - Tag mit führender 0
            %D - Formatierung aus 'Locale->loc_DateFormat'
            %e - Tag mit führendem ' '
```

```
            %H - Stunde (24h) mit führender 0
            %I - Stunde (12h) mit führender 0
            %m - Monat mit führender 0
            %M - Minute mit führender 0
            %n - Zeilenvorschub
            %p - AM oder PM Strings
            %q - Stunde (24)
            %Q - Stunde (12)
            %S - Sekunde mit führender 0
            %t - Tabulator-Zeichen
            %T - Formatierung aus 'Locale->loc_TimeFormat'
            %x - Formatierung aus 'Locale->loc_ShortDateFormat'
            %X - Formatierung aus 'Locale->loc_ShortTimeFormat'
            %y - Jahreszahl (zweistellig) mit führender 0
            %Y - Jahreszahl (vierstellig) mit führender 0
            %% - %

     date - das zu formatierende Datum
     putCharFunc - eine Callback-Funktion, welche für jedes
            auszugebenes Zeichen aufgerufen wird. Folgende
            Argumente werden übergeben:

               _R_A0  pOS_Callback*
               _R_A1  pOS_FormatDateCBData*



SEE ALSO
     pOS_FormatDateStr(),loctest.c
```

## 1.7 plocale.library/pOS_FormatDateStr()

```
NAME
     pOS_FormatDate -- Erzeugen einer Datum/Zeit-String aus der
       vorgegebenen Formatierung.

SYNOPSIS
     count = pOS_FormatDateStr(locale,fmtTemplate,date,
     _R_D0                     _R_A0  _R_A1      _R_A2
                          buffer,size);
                          _R_A3 _R_D0

     ULONG FormatDateStr(pOS_Locale*,const CHAR*,pOS_DateStamp*,
                          CHAR*,ULONG);

FUNCTION
     Aus der vorgegebenen Formatierung wird ein Datum/Zeit-Angabe
     erzeugt. Die Ausgabe erfolgt in den übergeben Speicherbereich.

INPUTS
     locale - die zur Formatierung zu verwendene Locale oder
              NULL für die System-Default-Locale.
     fmtTemplate - Ein mit NULL abgeschlossener String, welcher
                   das Format beschreibt. Die Formatanweisung
```

```
                    ist identisch der printf()-Formatanweisung.
                    Formatkodes beginnen mit einem %-Zeichen gefolgt
                    vom einem Formatkommando. Folgende Kommandos
                    werden von pOS_FormatDate verstanden:

            %a - abgekürzter Wochentags-Name
            %A - Wochentags-Name
            %b - Monats-Name
            %B - abgekürzter Monats-Name
            %c - Formatierung aus 'Locale->loc_ShortDateTimeFormat'
            %C - Formatierung aus 'Locale->loc_DateTimeFormat'
            %d - Tag mit führender 0
            %D - Formatierung aus 'Locale->loc_DateFormat'
            %e - Tag mit führendem ' '
            %H - Stunde (24h) mit führender 0
            %I - Stunde (12h) mit führender 0
            %m - Monat mit führender 0
            %M - Minute mit führender 0
            %n - Zeilenvorschub
            %p - AM oder PM Strings
            %q - Stunde (24)
            %Q - Stunde (12)
            %S - Sekunde mit führender 0
            %t - Tabulator-Zeichen
            %T - Formatierung aus 'Locale->loc_TimeFormat'
            %x - Formatierung aus 'Locale->loc_ShortDateFormat'
            %X - Formatierung aus 'Locale->loc_ShortTimeFormat'
            %y - Jahreszahl (zweistellig) mit führender 0
            %Y - Jahreszahl (vierstellig) mit führender 0
            %% - %

     date   - das zu formatierende Datum
     buffer - Speicherbereich in welchen die Stringausgabe erfolgen
              soll
     size   - Größe des Buffers. Es werden maximal size-1 Zeichen
              und ein NULL-Byte geschrieben.

RESULT
     count - Anzahl der geschriebenen Zeichen (ohne NULL-Byte).

SEE ALSO
     pOS_FormatDate(),loctest.c
```

## 1.8 plocale.library/pOS_GetCatalogStr()

```
NAME
     pOS_GetCatalogStr -- Ermittelt einen String aus dem angegebenen
        Katalog.

SYNOPSIS
     string = pOS_GetCatalogStr(catalog,stringNum,defaultString);
     _R_D0                      _R_A0   _R_D0     _R_A1

     const CHAR* GetCatalogStr(pOS_Catalog*,ULONG,const CHAR*);
```

```
FUNCTION
     Es wird der angeforderte String aus dem Katalog zurückgegeben.
     Ist der String nicht vorhanden und es existiert ein Default-
     String-Array, wird versucht, den String von dort zu ermitteln.
     Kann der String auch dort nicht gefunden werden bzw. ist der
     Zeiger auf den Katalog NULL, so wird der defaultString zurück-
     gegeben.

INPUTS
     catalog - ein Zeiger auf einen zuvor geöffneten Katalog oder NULL
     stringNum - die Message-ID innerhalb des Kataloges
     defaultString - ein String der zurückgegeben wird, falls der
           gesuchte String nicht im Katalog ist bzw. der Katalog NULL
           ist

RESULT
     string - Der String aus dem Katalog bzw. der defaultString. Der
           String ist solange gültig, wie der Katalog geöffnet ist.

SEE ALSO
     pOS_OpenCatalog(), pOS_CloseCatalog()
```

## 1.9 plocale.library/pOS_GetLocaleStr()

```
NAME
     pOS_GetLocaleStr -- ermittelt einen Standart-String aus
          dem Locale

SYNOPSIS
     string = pOS_GetLocaleStr(locale,stringNum);
     _R_D0                     _R_A0  _R_D0

     const CHAR* GetLocaleStr(pOS_Locale*,ULONG);

FUNCTION
     Ermittelt einen String aus der geöffneten Locale.

INPUTS
     locale - geöffnete Locale oder NULL für System-Default-Locale
     stringNum - Die Nummer des Strings. Siehe <plocale/locale.h>
               für vordefinierte Konstanten.

RESULT
     string - Zeiger auf einen NULL-abgeschlossenen String. Der
           Strings ist nur bis zum Zeitpunkt des Schließens der
           Locale gültig.

SEE ALSO
     pOS_OpenLocale(), pOS_CloseLocale(), <plocale/locale.h>,
     loctest.c
```

## 1.10 plocale.library/pOS_IsXXXX()

```
NAME
     pOS_IsXXXX -- ermitteln der Zeichentypes.

SYNOPSIS
     state = pOS_IsXXXX(locale,character);
     _R_D0              _R_A0  _R_D0

     BOOL pOS_IsXXXX(pOS_Locale*,ULONG);

FUNCTION
     Ermittelt den Typ eines Zeichens. Dabei werden auf die
     Besonderheiten verschiedenen Sprachen rücksicht genommen.

     IsAlNum() - alphanumerisches Zeichen
     IsAlpha() - alphabetisches Zeichen
     IsCntrl() - Steuerzeichen
     IsDigit() - Dezimalzeichen
     IsGraph() - darstellbares Zeichen
     IsLower() - Kleinbuchstabe
     IsPrint() - Freizeichen
     IsPunct() - Punktuations-Zeichen
     IsSpace() - Textformtierungszeichen
     IsUpper() - Großbuchstabe
     IsXDigit() - Hexadezimales Zeichen

INPUTS
     locale - zu verwendene Locale oder NULL für System-Default
     character - das zu zestende Zeichen

RESULT
     state - TRUE wenn das Zeichen der Testanforderung entspricht
```

## 1.11 plocale.library/pOS_LocaleToLower()

```
NAME
     pOS_ConvToLower -- konvertiert einen Buchstaben in dem
              entsprechenden Kleinbuchstaben

SYNOPSIS
     char = pOS_ConvToLower(locale,character);
     _R_D0                  _R_A0  _R_D0

     ULONG pOS_ConvToLower(pOS_Locale*,ULONG);

FUNCTION
     Diese Funktion wandelt Großbuchstaben in Kleinbuchstaben. Dabei
     wird auf sprachliche Besonderheiten rücksicht genommen.

INPUTS
     locale - zu verwendene Locale oder NULL für System-Default
```

```
     character - das zu konvertierende Zeichen

RESULT
     char - das möglicherweise konvertierte Zeichen
```

## 1.12 plocale.library/pOS_LocaleToUpper()

```
NAME
     pOS_ConvToUpper -- konvertiert einen Buchstaben in dem
              entsprechenden Großbuchstaben

SYNOPSIS
     char = pOS_ConvToUpper(locale,character);
     _R_D0                  _R_A0  _R_D0

     ULONG pOS_ConvToUpper(pOS_Locale*,ULONG);

FUNCTION
     Kleinbuchstaben werden in die entsprechenden Großbuchstaben
     gewandelt. Dabei wird auf sprachliche Besonderheiten
     rücksicht genommen.

INPUTS
     locale - zu verwendene Locale oder NULL für System-Default
     character - das zu konvertierende Zeichen

RESULT
     char - das möglicherweise konvertierte Zeichen
```

## 1.13 plocale.library/pOS_OpenCatalog()

```
NAME
     pOS_OpenCatalogA -- öffnet einen Katalog
     pOS_OpenCatalog  -- entsprechende Funktion mit Stackübergabe

SYNOPSIS
     catalog = pOS_OpenCatalogA(locale,name,tagList);
     _R_D0                      _R_A0 _R_A1 _R_A2

     __ARID__ pOS_Catalog *pOS_OpenCatalogA(pOS_Locale*,
               const CHAR*,pOS_TagItem);

     catalog = pOS_OpenCatalog(locale,name,firstTag, ...);

     __ARID__ pOS_Catalog *pOS_OpenCatalog(pOS_Locale*,
               const CHAR*,ULONG, ...);

FUNCTION
     Der angegebene Katalog wird geöffnet. Ist dieser noch nicht
     im Speicher vorhanden, so wird er nachgeladen. Dabei wird
```

an folgenden Plätzen gesucht:

```
CURRENTDIR:Catalogs/Name_der_Sprache/Katalog_Name
PROGDIR:Catalogs/Name_der_Sprache/Katalog_Name
LOCALE:Catalogs/Name_der_Sprache/Katalog_Name
```

Die Sprache richtet sich nach der Locale. Sollte der Katalog in dieser Sprache nicht vorliegen, so wird versucht, ihn in einer der anderen bevorzugten Sprachen zu laden.

Verschiedene Programme können sich einen Katalog teilen. Dieser Katalog wird dann auch nur einmal geladen. Dabei darf dann allerdings keine Default-String-Array per LOCTAG_DefStrings übergeben werden.

INPUTS

locale - zu verwendene Locale oder NULL für System-Default
name - der Name des Kataloges. Dieser sollte immer dem Namen der Applikation entsprechen und mit ".catalog" enden.
tagList - Zeiger auf eine Tagliste mit zusätzlichen Parametern

TAGS

LOCTAG_BuiltInLanguage (const CHAR*)
- Dieser Tag gibt die Sprache der eingebauten Strings an. Ist diese mit der zu öffnenden identisch, so wird der Katalog nicht geladen und es wird bei pOS_GetCatalogStr() immer der Strings aus den Default-String-Array bzw. der übergebene Default-Strings zurückgegeben.

LOCTAG_BuiltInCodeSet (ULONG)
- Dieser Tag ist für die Erweiterung des Codesets vorgesehen und sollte zur Zeit nicht verwendet bzw. auf 0 gesetzt werden.

LOCTAG_Language (const CHAR*)
- Expliziete Angabe der zu verwendenen Sprache. Kann der Katalog nicht in dieser geladen werden, so wird versucht, ihn in einer der bevorzugten Sprachen zu laden.
Dieser Tag sollte normalerweise nicht verwendet werden, da er die User-Voreinstellungen überschreibt.

LOCTAG_Version (UWORD)
- Version des zu ladenen Kataloges. Die Version muß genau mit der des Kataloges übereinstimmen.
Ist dieser Tag nicht gesetzt bzw. 0, wird jede gefundene Katalogversion geladen.

LOCTAG_DefStrings (const pOS_CatalogItem*)
- Hier kann ein Array mit Default-Strings übergeben werden. Diese werden dann benutzt, wenn kein entsprechender String im Katalog gefunden werden kann.
Das Array ist mit einem Eintrag, dessen Strings-Zeiger gleich NULL ist, zu beenden.
Wird dieser Tag benutzt, so kann der Katalog nicht von mehreren Programmen gleichzeitig benutzt werden, ohne ihn nochmal nachzuladen.

```
RESULT
     catalog - Zeiger auf den geöffneten Katalog oder NULL im
          Fehlerfall.

NOTE
     Sollte kein entsprechender Katalog gefunden werden, so wird
     totzdem ein Zeiger zurückgegeben. Am Struktureintrag
     "cat_NumStrings" ist ersichtlich, wieviele Strings geladen
     werden konnten.

     NULL wird nur zurückgegeben, wenn ein fataler Fehler, wie
     Speichermangel, aufgetreten ist. Aber auch dann kann ohne
     Probleme mit dem Programmablauf fortgefahren werden, da
     pOS_GetCatalogStr() immer den DefaultStrings zurückgibt, wenn
     der Zeiger auf den Katalog NULL ist.

SEE ALSO
     pOS_CloseCatalog(), pOS_GetCatalogStr()
```

## 1.14 plocale.library/pOS_OpenLocale()

```
NAME
     pOS_OpenLocale -- öffnet eine Locale

SYNOPSIS
     locale = pOS_OpenLocale(name);
     _R_D0                   _R_A0

     __ARID__ pOS_Locale *pOS_OpenLocale(const CHAR*);

FUNCTION
     Es wird die angegebene Locale geöffnet. Dabei wird der Treiber
     der Sprache und die Länderbeschreibung geladen.

INPUTS
     name - der Name der zu ladenen Locale oder NULL für die
            System-Default-Locale.
            Programme sollten immer die System-Default-Locale
            verwenden.

RESULT
     locale - Zeiger auf die initialisierte Locale-Struktur oder NULL
              im Fehlerfall.

              Wenn als Name ein NULL-Zeiger verwendet wurde, ist
              der zurückgegebene Zeiger immer gültig.

SEE ALSO
     pOS_CloseLocale()
```

## 1.15 plocale.library/pOS_StrnCmp()

```
NAME
     pOS_StrnCmp -- Stringvergleich mit Unterscheidung auf
         Groß- und Kleinschreibung.

SYNOPSIS
     result = pOS_StrnCmp(locale,string1,string2,length);
     _R_D0                _R_A0  _R_A1   _R_A2   _R_D0

     SLONG pOS_StrnCmp(pOS_Locale*,const CHAR*,const CHAR*,SLONG);

FUNCTION
     Vergleicht beide Strings. Dabei wird Groß- und Kleinschreibung
     unterschieden. Außerdem wird auf sprachlcihe Besonderheiten
     geachtet.

INPUTS
     locale - zu verwendene Locale oder NULL für System-Default
     string1 - NULL-abgeschlossender String
     string2 - NULL-abgeschlossender String
     length - die maximale Anzahl zu vergleichernder Zeichen oder
              -1, um bis zum Ende eines der beiden Strings zu
              Vergleichen.

RESULT
     result - Beziehung zwischen String1 und String2
                    <0 bedeutet String1 < String2
                    =0 bedeutet String1 = String2
                    >0 bedeutet String1 > String2

SEE ALSO
     pOS_StrniCmp()
```

## 1.16 plocale.library/pOS_StrniCmp()

```
NAME
     pOS_StrniCmp -- Stringvergleich ohne Unterscheidung auf
         Groß- und Kleinschreibung.

SYNOPSIS
     result = pOS_StrniCmp(locale,string1,string2,length);
     _R_D0                 _R_A0  _R_A1   _R_A2   _R_D0

     SLONG pOS_StrniCmp(pOS_Locale*,const CHAR*,const CHAR*,SLONG);

FUNCTION
     Vergleicht beide Strings. Dabei wird NICHT zwischen
     Groß- und Kleinschreibung unterschieden.
     Auf sprachlcihe Besonderheiten wird geachtet.

INPUTS
```

```
     locale - zu verwendene Locale oder NULL für System-Default
     string1 - NULL-abgeschlossender String
     string2 - NULL-abgeschlossender String
     length - die maximale Anzahl zu vergleichernder Zeichen oder
              -1, um bis zum Ende eines der beiden Strings zu
              Vergleichen.

RESULT
     result - Beziehung zwischen String1 und String2
                    <0 bedeutet String1 < String2
                    =0 bedeutet String1 = String2
                    >0 bedeutet String1 > String2

SEE ALSO
     pOS_StrnCmp()
```