

pUtilityD

COLLABORATORS

	<i>TITLE :</i> pUtilityD		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 29, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	pUtilityD	1
1.1	pUtilityD.doc	1
1.2	putility.library/UtilityAllgemeines()	1
1.3	putility.library/pOS_FindTagData()	1
1.4	putility.library/pOS_FindTagItem()	2
1.5	putility.library/pOS_MapTagItems()	3
1.6	putility.library/pOS_NextTagItem()	3

Chapter 1

pUtilityD

1.1 pUtilityD.doc

putility.library

```
pOS_FindTagData ()   pOS_FindTagItem ()   pOS_MapTagItems ()  
pOS_NextTagItem ()   UtilityAllgemeines ()
```

1.2 putility.library/UtilityAllgemeines()

STRUKTUREN

```
struct pOS_UtilityBase;  
struct pOS_TagItem;
```

INCLUDES

```
pUtil/TagItem.h  
pUtil/TxtTags.h  
pUtil/UtilBase.h  
proto/pUtil.h
```

BESCHREIBUNG

1.3 putility.library/pOS_FindTagData()

PROTOTYP

```
ULONG data = pOS_FindTagData  
(  
    pOS_UtilityBase *utilitybase,  
    ULONG tagtyp,  
    const pOS_TagItem *taglist,  
    ULONG default  
);
```

FUNKTION

Ermitteln von Tag-Daten einer Tag-Liste.

PARAMETER

utilitybase (_R_LB)
Zeiger auf pOS_UtilityBase
tagtyp (_R_D0)
Name von gesuchtem Tag
taglist (_R_A0)
Zeiger auf die Tag-Liste die durchsucht werden soll.
Der Zeiger darf NULL sein.
default (_R_D1)
Wert der zurückgegeben werden soll, wenn der Tag nicht
in der Liste vorhanden war.

ERGEBNIS

tag (_R_D0)
Tag-Wert des ersten gefundenen Tags oder default,
wenn kein passender Tag gefunden werden konnte.

SIEHE AUCH

<pUtil/TagItem.h>
struct pOS_TagItem
pOS_FindTagItem(), pOS_NextTagItem()

AMIGA FUNKTION

ULONG GetTagData(Tag, ULONG, struct TagItem *);

1.4 putility.library/pOS_FindTagItem()

PROTOTYP

```
pOS_TagItem *tag = pOS_FindTagItem  
(  
    pOS_UtilityBase *utilitybase,  
    ULONG tagtyp,  
    const pOS_TagItem *taglist  
);
```

FUNKTION

Suchen eines Tags in einer Tag-Liste.

PARAMETER

utilitybase (_R_LB)
Zeiger auf pOS_UtilityBase
tagtype (_R_D0)
Name von gesuchtem Tag
taglist (_R_A0)
Zeiger auf die Tag-Liste die durchsucht werden soll.
Der Zeiger darf NULL sein.

ERGEBNIS

tag (_R_D0)
Zeiger auf die erste ermittelte TagItem-Struktur;

NULL wenn der Tag nicht in der Liste vorhanden ist.

SIEHE AUCH

```
<pUtil/TagItem.h>
struct pOS_TagItem
pOS_NextTagItem()
pOS_FindTagData()
```

AMIGA FUNKTION

```
struct TagItem *FindTagItem(Tag, struct TagItem *);
```

1.5 putility.library/pOS_MapTagItems()

PROTOTYP

```
VOID pOS_MapTagItems
(
    pOS_UtilityBase *utilitybase,
    const pOS_TagItem *maplist,
    pOS_TagItem *taglist
);
```

FUNKTION

Konvertieren von Daten in einer Tag-Liste anhand einer zweiten Tag-Liste.
Die maplist wird komplett durchlaufen und der jeweils erste gleiche Tag, der in taglist gefunden wird, wird mit den Daten aus maplist gefüllt.

PARAMETER

```
utilitybase (_R_LB)
    Zeiger auf pOS_UtilityBase
maplist (_R_A0)
    Tag-Liste die die Veränderungen beschreibt
    ACHTUNG: darf nicht NULL sein
taglist (_R_A1)
    Tag-Liste die verändert werden soll
    ACHTUNG: darf nicht NULL sein
```

BEISPIEL

```
pOS_TagItem TagListe[]={ TAG1,1, TAG2,2, TAG3,3, TAG2,4, TAG_DONE };
pOS_TagItem MapListe[]={ TAG2,22, TAG_DONE };
pOS_MapTagItems (MapListe, TagListe);
TagListe danach: TAG1,1, TAG2,22, TAG3,3, TAG2,4, TAG_DONE
```

SIEHE AUCH

```
pOS_NextTagItem()
```

AMIGA FUNKTION

ACHTUNG: ist nicht identisch mit der AMIGA FUNKTION MapTags()

1.6 putility.library/pOS_NextTagItem()

PROTOTYP

```
pos_TagItem *tag = pos_NextTagItem
(
    pos_UtilityBase *utilitybase,
    const pos_TagItem **taglistrptr
);
```

FUNKTION

Durchläuft eine Tag-Liste und beachtet dabei die Standard-Tags.

TAG_DONE Kennzeichnet das Ende einer Tag-Liste.

TAG_END gleich wie TAG_DONE

TAG_IGNORE Ignoriert diesen Tag und fährt beim nächsten Tag mit der Bearbeitung fort.

TAG_MORE Ist eine Art Sprungmarke. In ti_Data befindet sich der Zeiger auf eine weitere (const struct pos_TagItem*) Liste. Nach der Bearbeitung von ti_Data wird NICHT mit dem folgenden Tag weitergearbeitet. Das Ende von ti_Data ist immer das Ende der gesamten Tag-Liste. Ist ti_Data==NULL, dann wird die Abarbeitung sofort beendet (TAG_END).

TAG_SKIP Überspringt folgende ti_Data Tags.
Z.B. {TAG_A,0, TAG_SKIP,2, TAG_B,1, TAG_C,3, TAG_D,4,...}
Ausgelesen werden: TAG_A, TAG_D
Befindet sich im Skip-Bereich ein TAG_END, so wird die Tag-Bearbeitung ordnungsgemäß beendet.
Wird im Skip-Bereich ein TAG_MORE entdeckt, wird der Skip abgebrochen und normal mit TAG_MORE weitergearbeitet.

PARAMETER

utilitybase (_R_LB)
Zeiger auf pos_UtilityBase

taglistptr (_R_A0)
Adresse des Zeigers auf ein Element der Tag-Liste; dieser Zeiger wird von der Funktion verändert und dient auch bei jedem folgenden Aufruf als Übergabeparameter.
Der Zeiger darf NULL sein.

ERGEBNIS

tag (_R_D0)
Zeiger auf die ermittelte TagItem-Struktur;
NULL am Ende der Tag-Liste.

SIEHE AUCH

```
<pUtil/TagItem.h>
struct pos_TagItem
pos_FindTagItem()
pos_FindTagData()
```

AMIGA FUNKTION

```
struct TagItem *NextTagItem(struct TagItem **);
```

END
