

Mit dem Programm Get\_Reply.PRO kann unter IDL [1] die interaktive Beantwortung einer gestellten Frage mit einer Prüfung der Gültigkeit der Antwort verbunden werden.

Durch die Anwendung dieses Funktions-Unterprogramms können Applikatio-

nen wesentlich benutzerfreundlicher gestaltet werden. So lassen sich z. B. Fehlerabbrüche, die sich aufgrund von falsch eingegebenen Antworten ergeben, meistens von vornherein vermeiden.

Das Programm steht im Internet zur Verfügung. Diese Beschreibung gibt den

Stand des Programms in der Version V 01.3 (= V 1.0 – Edit 3) wieder. Es wurde von der Abt. Wiss. Datenverarbeitung in der „Interactive Data Language“ IDL [1, 2] angefertigt und kann ggf. vom jeweiligen Anwender selbst modifiziert bzw. ergänzt werden.

## 1. Beschreibung:

Das Programm *Get\_Reply* fordert den Benutzer mit einer Frage oder einer Feststellung auf, eine Antwort (string) über das Tastenfeld einzugeben. Es wird dann geprüft, ob diese Antwort einer der im Aufruf von *Get\_Reply* vorgegebenen Möglichkeiten (*answers*) entspricht. Wenn dieses der Fall ist, dann wird die Antwort an das übergeordnete Programm zur weiteren Auswertung übergeben.

Entspricht die Antwort hingegen keiner der vorgegebenen Möglichkeiten, dann wird dieses dem Benutzer mit „Answer isn't allowed; try again.“ mitgeteilt und vom Benutzer erneut die Eingabe einer Antwort verlangt. Dieses wird solange wiederholt bis eine gültige Antwort vom Benutzer eingegeben worden ist.

Die Klein-/Großschreibung der Antwort spielt bei der Feststellung der Gültigkeit eine Rolle. Wird im Aufruf von *Get\_Reply* das Keyword /NOCASE angegeben, dann hat die Klein-/Großschreibung der Antwort keinen Einfluß auf die Prüfung.

## 2. Aufruf:

*Get\_Reply* ( *text*, *answers* [, *default* [, ...] ] )

### 2.1 Variablen (VAR):

*text* — Text (Frage oder Feststellung), der vor der Eingabe ausgegeben wird. *Get\_Reply* ergänzt diesen ggf. um den Default-Wert der Antwort in eckigen Klammern. Falls im Aufruf von *Get\_Reply* kein solcher Default-Wert angegeben wurde, dann wird an den Text „[-]“ angehängt. Außerdem wird von *Get\_Reply* an den Text als die Eingabeaufforderung (prompt) „ > “ angehängt.

*answers* — Die zugelassenen Antworten werden als Zeichenkette angegeben, wobei die einzelnen Möglichkeiten der Antworten durch „|“ getrennt werden müssen. Ein Beispiel: "EVP|AVE|ORI".

*default* — Diese Variable ist optional. Mit ihr kann ein Default-Wert für die Antwort vorgegeben werden. Diese wird immer dann verwendet, wenn der Benutzer nach der Aufforderung nur die „Return“-Taste betätigt.

### 2.2 Keywords:

Durch die Angabe von Keywords kann der Ablauf des Programms beeinflusst werden.

**/NOCASE** — Die Klein-/Großschreibung der Antwort wird bei der Feststellung der Gültigkeit ignoriert. Ohne die Angabe dieses Keywords spielt die Klein-/Großschreibung der Antwort bei der Feststellung der Gültigkeit eine Rolle.

**/INSERT** — Noch nicht implementiert!

Die vorgegebenen Antworten werden automatisch in dem auszugebenden Text (Frage bzw. Feststellung) in runden Klammern eingefügt.

## 2.3 Beispiel(e):

Die folgenden Beispiele von Aufrufen von *Get\_Reply* sind aus verschiedenen Applikationen entnommen.

```
a = Get_Reply(" Enter single (s) or diffs (d)", $
              "s|d", ans$)

t = Get_Reply(" Enter file type (EVP,AVE,ORI)", $
              "EVP|AVE|ORI", "EVP", /NoCase)

a = Get_Reply(" Do you want jump (j) or step (s)?", $
              "j|s", ans$)

commands = "add|sub|mul|div|pot"
a = Get_Reply(" Enter command", $
              commands, "ADD", /NoCase)
```

Hier noch ein Tip: Es hat sich als zweckmäßig erwiesen, für Variablen von Default-Werten den Namen der Originalvariablen um ein \$-Zeichen zu ergänzen (Im Beispiel ist das „ans\$“).

## 3. Erforderliche Programme:

Dieses Programm verwendet neben den Standard-Funktionen von IDL (u. a. sind dieses: Read, StrCompress, StrUppcase, Where) folgende IDL-Programme der Abt. Wiss. Datenverarbeitung, die in der IDL-Bibliothek des HAIRclusters zur Verfügung stehen.

- *F\_Element* (P 476/33): Extrahieren einer Teil-Zeichenkette (substring) aus einer Zeichenkette (string) [3].

## 4. Sonstige Hinweise:

- Zur Zeit Keine.

## 5. Literatur:

- [1] Research Systems, Inc.: IDL – Interactive Data Language. Version 2.1. Boulder (USA): RSI 1991 (Edition vom 2.4.1991). E-Mail: idl@boulder.colorado.edu.
- [2] Dittberner, K.-H.: Interactive Data Language IDL: Eine Einführung. FU Berlin (IfP): wdv-notes Nr. 85, 1988–1992.
- [3] Dittberner, K.-H.: IDL-Programm: F\_Element.PRO (P 476/33). FU Berlin (IfP): wdv-notes Nr. 105, 1990–1992.

## 6. Copyright-Notiz:

© 1990-1993 – K.-H. Dittberner c/o Freie Universität Berlin. All rights reserved. No part of this software package may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any form by any means without the written permission of Karl-Heinz Dittberner. Permission will be granted for non-profit redistribution.