

GIPSY_Primer

Introduction to GIPSY, Vers. 1.01, 1988
W. Kynast

Creation of PPL-program Interpreter- and Preprocessor-Mode

Definition of Inputstream

The INCLUDE command is used to switch the input stream from one file module) to another. E.g.

%INCLUDE A.PPL input is taken from file A.PPL
%INCLUDE A input is taken from file A.PPL
%INCLUDE A(B) input is taken from module B from library A
%INCLUDE (B) input is taken from module B from the default library

Symbols

Before the input is parsed, a symbol replacement takes place. Symbols start with an exclamation point and are terminated by a dot. They are defined by the statement DEFINE.

E.g. %DEFINE alpha="pferd"
 %name= "Blument!alpha.e"

Symbol replacement is not done within comments.

Program structure

Data types and operations

All variables may be sequences without a specific declaration. E.g. % B = 2.5,5.14,1024

BIT bit variables, length 0 to 32767, constants are written like in PL/I
FIXED integer numbers
FLOAT floating point numbers (single precision)
CHARACTER strings
TABLE table elements are accessed by a key rather than by an index

Program control

Aggregates

Declarations and expressions

Procedures

Builtins

All names of the GIPSY builtin functions start like G\$.... Some are similar to PL/I builtin functions. Note however that nearly all can be applied to sequences as well as to scalars. Look in the chapter "Keywords" for more details.

I

Keywords, Reference summary

In the following all keywords of GIPSY are explained. Note, that those words cannot be used as variable names.

DECLARE

Declare GIPSY variables. It is not mandatory to declare variables in GIPSY, however in some cases it is convenient to let the attributes check by the computer. Following things are checked:

PARAMETER the declared variable is a parameter passed from the calling routine

VARIABLE declaration of variable

FILE file declaration
PROCEDURE GIPSY and PL/I entries can be declared here.

label	INCLUDE	DEFINE	PROCEDURE	ENDPROC	UNQUOTE	IF	ELSE
ON	FOR	WHILE	SELECT	WHEN	OTHERWISE	END	ENDSELECT
DIRECTORY	ERASE	LEAVE	SKIP	RETURN	EXECUTE	PLICALL	
COMMAND	MACROEXECUTE	FUNCTION	EXECUTE	MESSAGE	SPECIFY		
STEP	HALT	GO	FLOW				

ASSERT

A logic expression is checked and in case of not being true it is branched to error routines.

INTERPRET	TESTON	PROMPT	PUT	OPEN	READ	WRITE	PARAMETER
VARIABLE	INITIALIZE	FILE	SWITCH	DEFAULT	PROMPT	REQUIRED	
XNAME	EXPRESSION	NUMERIC	ENTRY	TITLE	PLIENTRY	ERROR	
DEBUG	OUTPUT	INPUT	BREAK	STEP			

G\$SORT(f) sorts the sequence f in increasing order

G\$HIGH(f) returns the biggest element of f

G\$LOW(f) returns the smallest element of f

G\$INDEX(e,f) returns the numbers of position (index) where e occurs in sequence f.

E.g. **G\$INDEX(3,(3,2,3,1,3))** returns 1,3,5

G\$INDEX("a","aa","x","a") returns 3

G\$POSITION(s1,s2) returns the position(s) of string s1 in string s2. E.g. **G\$POSITION("a","xaxbaa")** returns 2,5,6

G\$POSITION("A","xaxbaa") returns **NULLSEQUENCE('FIXED')**

G\$DATATYPE(expr) returns the data type of the specified expression

expr. e.g. **G\$DATATYPE((1,2,3))** returns "FIXED"

G\$ELEMENTS(expr) returns the number of elements in expr (=1 if scalar, >1 if sequence).

E.g. **G\$ELEMENTS((1,4,7))** returns 3

G\$ELEMENTS("emil") returns 1

G\$LENGTH(s) returns the length(s) of the character string(s) in s.

E.g. **G\$LENGTH("otto")** returns 4

G\$LENGTH("otto","emil","udo") returns 4,4,3

G\$NULLSEQUENCE(t) returns a null sequence of the specified type t.

E.g. **G\$NULLSEQUENCE("FIXED")**

G\$UPPERCASE("Klaus") returns "KLAUS"

G\$LOWERCASE("hANS") returns "hans"

G\$SUM(s) sums up all elements of the sequence s.

E.g. **G\$SUM((1,2,3))** returns 6

G\$SUM(3) returns 3

G\$SUM((1.5,2.5)) returns 4.0

G\$SUM(('1'B,'1'B)) returns 2

G\$RANDOM(n) returns a sequence of n random numbers

G\$EXP exponential function, like PL/I

G\$SQRT square root, like PL/I

G\$ABS absolute value, like PL/I

G\$SIGN signum, like PL/I

G\$FLOOR(x) returns largest integer not exceeding x, like PL/I

G\$CEIL(x) returns smallest integer greater or equal to x, like PL/I

G\$TRUNC(x) truncates x to **G\$FLOOR(x)** if x>0, to **G\$CEIL(x)** if x<0, like PL/I

Following functions are defined and can also be applied to sequences of BIT, FIXED, and FLOAT:

G\$SIN	G\$SIND	G\$SINH	G\$ASIN	G\$ASIND	G\$ASINH	G\$COS
	G\$COSD	G\$COSH	G\$ACOS	G\$ACOSD	G\$ACOSH	G\$TAN
	G\$TAND	G\$TANH	G\$ATAND	G\$ATANH		

G\$ATAN , may also have two arguments: G\$ATAN(x1,x2) returns the arcus tangens of x1/x2 .

G\$LOG(f,b) returns the logarithm of f to base b.

E.g. G\$LOG(10,(10,2)) returns 1.0,3.32193

G\$LOG10 same as G\$LOG(s,10)

G\$MOD((1,2,3),3) returns 1,2,0

G\$MOD(3.2,0.5) returns 0.2

G\$MIN(f1,f2,...,fn) returns minimum of sequences,

E.g. G\$MIN((1,2,3),2,(3,2,1)) returns 1,2,1

G\$MAX similar to G\$MIN

G\$TRIM cuts off specified character(s)(default is blank) from right and left ,

E.g. G\$TRIM(" OTTO ") returns "OTTO"

G\$TRIM(" OTTO ", "O") returns "TT"

G\$TRIM(("aabbcc", "kdjflsacccc", "basdc"), "ba", "cc") returns ("aabb", "kdjflsa", "sdc")

G\$CONCAT(("abc", "pq", "xyz")) returns "abcpqxyz"

G\$DATE returns the current date,

G\$DATE() returns "85-09-24"

G\$DATE("SHORT") returns "85-09-24"

G\$DATE("KURZ") returns "24.09.85"

G\$DATE("LONG") returns "TUESDAY, 24 SEP 85"

G\$DATE("LANG") returns "DONNERSTAG, 24 SEPTEMBER 1985 "

G\$TIME returns current time in hh:mm:ss