Identification

Declaration Parsing and Processing
R. Freiburghouse

(Note that the following are Abstracts, which should be replaced by a full description at a later time.)

attribute_set

Function of Entry:

Performs a parse of a list of attributes. The procedure is used only to parse PL/I declare statements.

Calling Sequence for Entry:

call attribute_set(k,q,block,recovery,error,caller);

Declaration of Arguments:

dcl (k,caller,recovery,error) fixed bin(15),
(q,block) ptr;

Description of Arguments:

block is a pointer to the current block node.

k is an index to the token vector which indicates the beginning of the attribute list.

recovery is an index to the token vector used for error recovery.

error is a flag used to indicate that a syntactic error was detected in the attribute list.

caller is an integer used to provide better context for error detection.

g is a pointer to the attribute block in which the attributes are recorded.
bounds

Function of Entry:
Performs a parse of the dimension attribute. If the parse is successful the procedure returns a value of "1"b. If unsuccessful it returns a value of "0"b.

Calling Sequence for Entry:
\[ b = \text{bounds}(k, q, \text{block}); \]

Declaration of Arguments:
\[ \text{dcl } k \text{ fixed bin (15)}, \]
\[ (q, \text{block}) \text{ ptr}; \]

Description of Arguments:
\[ k \quad \text{is an index to the token vector indicating the point when the dimension attribute begins.} \]
\[ q \quad \text{is a pointer to an attribute block in which the bounds will be recorded.} \]
\[ \text{block} \quad \text{is a pointer to the current block node.} \]
context

Function of Entry:

Record contextual information during the execution of the parse. Used only by the PL/I compiler.

Calling Sequence for Entry:

call context(id,blk,label,c);

Declaration of Arguments:

dcl (id,blk,label) ptr,
c fixed bin (15);

Description of Arguments:

id is a pointer to the token table entry which represents the name.

blk is a pointer to the block node in which the context was found.

label is a pointer to the statement in which the context was found or is null.

c is an integer between 1 and 13 which describes the context.
convert_int

Function of Entry:
   Convert the character string argument into a fixed point binary integer. Used only by the PL/I compiler.

Calling Sequence for Entry:
   \( i = \text{convert\_int}(s); \)

Declaration of Arguments:
   \[
   \text{dcl} \ s \ \text{char}(n), \\
   i \ \text{fixed bin}(31);
   \]

Description of Arguments:
   \( s \) is a non-varying character string, which consists only of digits 0-9.
declare_stmnt

Function of Entry:
Performs the parse of PL/I declare statements.

Calling Sequence for Entry:
call declare_stmnt(index,block);

Declaration of Arguments:
dcl index fixed bin(15),
    block ptr;

Description of Arguments:
index is a pointer to the first element of the
token vector.

block is a pointer to the current block node.
entry_attributes

Function of Entry:

Performs a parse of the attributes which may be given to an entry declaration. If successful it returns a "1"b, if unsuccessful it returns a "0"b.

Calling Sequence for Entry:

\[ b = \text{entry_attributes}(k,q,\text{block},\text{recovery},\text{error},\text{caller}); \]

Declaration of Arguments:

\[ \text{dc1 } (k,\text{error},\text{recovery},\text{caller}) \text{ fixed bin (15)}, \]
\[ (q,\text{block}) \text{ ptr}; \]

Description of Arguments:

- \( k \) is an index to the token vector which indicates the beginning of the attribute list.
- \( \text{recovery, error, and caller} \) are used to provide context and recovery information.
- \( q \) is a pointer to an attribute block which will be used to record the attributes.
- \( \text{block} \) is a pointer to the current block node.
file_attributes

Function of Entry:
Performs a parse of file attributes. If successful it returns a value of "1"b. If unsuccessful it returns a value of "0"b.

Calling Sequence for Entry:
\[ b = \text{file_attributes} (k, q,); \]

Declaration of Arguments:
\[
dcl \ k \ fixed \ bin \ (15), \\
q \ ptr;
\]

Description of Arguments:
\begin{itemize}
  \item \( k \) is an index to the token vector which indicates the beginning of the attribute list.
  \item \( q \) is a pointer to an attribute block which will be used to record the attributes.
\end{itemize}
function_attributes

Function of Entry:

Performs a parse of the attributes contained within a return attribute. If successful the procedure returns a value of "1"b. If unsuccessful it returns a "0"b.

Calling Sequence for Entry:

\[
b = \text{function_attributes} \left(k, q, \text{block}, \text{caller} \right);
\]

Declaration of Arguments:

\[
dcl \ (k, \text{caller}) \text{ fixed bin (15),}
(q, \text{block}) \text{ ptr};
\]

Description of Arguments:

- \(k\): is an index to the token vector which indicates the beginning of the attributes.
- \(q\): is a pointer to an attribute block in which the attributes will be recorded.
- \text{block}: is a pointer to the current block node.
- \text{caller}: is used to provide context information used in error detection and recovery.
initial_at

Function of Entry:
Allocates and initializes an attribute block.
Used only by the PL/I compiler.

Calling Sequence for Entry:
call initial_at(p);

Declaration of Arguments:
dcl p ptr;

Description of Arguments:
p is a pointer to the newly created attribute block
initial_list:

Function of Entry:

Performs a parse of the initial attribute.
If successful it returns a value of "1"b.
If unsuccessful it returns a value of "0"b.

Calling Sequence for Entry:

\[ b = \text{initial_list}(k, q, \text{block}); \]

Declaration of Arguments:

\[ \text{dcl} \ (q, \text{block}) \text{ ptr}, \]
\[ k \text{ fixed bin (15)}; \]

Description of Arguments:

\[ k \] is an index to the token vector which indicates the beginning of the attribute.
\[ q \] is a pointer to the parse of the attribute.
\[ \text{block} \] is a pointer to the current block node.
initial_symbol

Function of Entry:

Allocate and initialize a symbol table node.
Used only by the PL/I compiler.

Calling Sequence for Entry:

call initial_symbol (b, id, sym, t);

Declaration of Arguments:

dcl (b, id, sym) ptr,
    t fixed bin(15);

Description of Arguments:

b is a pointer to the block node in which the symbol table is to be created.

id is a pointer to the token table node representing the name to be declared.

sym is a pointer to the newly created symbol table node.

t is the type of declaration.
initialize

Function of Entry:

Initialize a set of tables used by the declare statement parse.

Calling Sequence for Entry:

call initialize (a, b, c);

Declaration of Arguments:

dc1 a(27) char(11),
b(27) fixed bin(15),
c(27) fixed bin(15);

Description of Arguments:

These arrays serve as driving tables for the parse of data attributes.
initialize_e

Function of Entry:

Initialize a set of tables used by the declare statement parse.

Calling Sequence for Entry:

call initialize_e (a, b, c);

Declaration of Arguments:

dcl a(8) char(11),
   b(8) fixed bin(15),
   c(8) fixed bin(15);

Description of Arguments:

These arrays serve as driving tables for the parse of entry attributes.
initialize_f

Function of Entry:

   Initialize a table used by the declare statement parse.

Calling Sequence for Entry:

   call initialize_f (a, b, c);

Declaration of Arguments:

   dc1 a(21) char(11),
       b(21) fixed bin(15),
       c(21) fixed bin(15);

Description of Arguments:

   The three arrays serve as driving tables for the parse of function attributes.
initialize_fa

Function of Entry:

Initialize a set of tables used by the declare statement parse.

Calling Sequence for Entry:

call initialize_fa (a, b, c);

Declaration of Arguments:

dcl a(18) char(11),
    b(18) fixed bin(15),
    c(18) fixed bin(15);

Description of Arguments:

The three arrays serve as driving tables for the parse of file attributes.
refer_expression

Function of Entry:

Performs a parse of the refer option.
If successful it returns a value of "1"b.
If unsuccessful it returns a value of "0"b.

Calling Sequence for Entry:

\[ b = \text{refer_expression}(k, q, \text{block}, \text{back}); \]

Declaration of Arguments:

\[ \text{dcl } (q, \text{block}, \text{back}) \text{ ptr, } \]
\[ k \text{ fixed bin(15);} \]

Description of Arguments:

\( k \) is an index to the token vector which indicates the beginning of the \text{refer} option.

\( q \) is a pointer to the parse of the \text{refer} option.

\( \text{block} \) is a pointer to the current block node.

\( \text{back} \) is a pointer to the owner of the parse of the \text{refer} option.
Function of Entry:

Performs a parse of PL/I references.
If successful it returns a value of "1"b.
If unsuccessful it returns a value of "0"b.

Calling Sequence for Entry:

b = reference (k, q, block, back);

Declaration of Arguments:

dcl (q, block, back) ptr,
     k fixed bin(15);

Description of Arguments:

k is an index to the token vector which indicates
the beginning of the reference.

g is a pointer to the parse of the reference.

block is a pointer to the current block node.

back is a pointer to the node which owns the
parse of the reference.