Published: 09/12/68

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.
q	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 = bounds (k,q,block);
```

Declaration of Arguments:

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

Description of Arguments:

<u>k</u>	is an index to the token vector indicating the point when the dimension attribute begins.
<u>q</u>	is a pointer to an attribute block in which the bounds will be recorded.
<u>block</u>	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:

<u>id</u>	is a pointer to the token table entry which represents the name.
<u>blk</u>	is a pointer to the block node in which the context was found.
<u>label</u>	is a pointer to the statement in which the context was found or is null.
<u>c</u>	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 = convert_int(s);
```

Declaration of Arguments:

```
dcl  s char(n),
     i 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:

<u>index</u>	is a pointer to the first element of the token vector.
<u>block</u>	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 = entry_attributes (k,q,block,recovery,error,caller);
```

Declaration of Arguments:

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

Description of Arguments:

<u>k</u>	is an index to the token vector which indicates the beginning of the attribute list.

<u>recovery</u>, <u>error</u>, and <u>caller</u> are used to provide context and recovery information.

<u>q</u>	is a pointer to an attribute block which will be used to record the attributes.
<u>block</u>	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 = file_attributes (k,q,);
```

Declaration of Arguments:

```
dcl  k fixed bin (15),
     q ptr;
```

Description of Arguments:

<u>k</u> is an index to the token vector which indicates the beginning of the attribute list.

<u>q</u> is a pointer to an attribute block which will be used to record the attributes.

function_attributes

Function of Entry:

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

Calling Sequence for Entry:

```
b = function_attributes (k,q,block,caller);
```

Declaration of Arguments:

```
dcl  (k,caller) fixed bin (15),
     (q,block) 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.
block	is a pointer to the current block node.
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:

<u>p</u> 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 = initial_list (k, q, block);
```

Declaration of Arguments:

```
dcl (q, block) ptr,
    k 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.
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:

```
dcl  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:

```
dcl  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 = refer_expression (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 _refer_ option.
_q_	is a pointer to the parse of the _refer_ option.
block	is a pointer to the current block node.
back	is a pointer to the owner of the parse of the _refer_ option.

reference

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:

<u>k</u>	is an index to the token vector which indicates the beginning of the reference.
<u>q</u>	is a pointer to the parse of the reference.
<u>block</u>	is a pointer to the current block node.
<u>back</u>	is a pointer to the node which owns the parse of the reference.