file mspm

(DRAFT)

THE CREATION OF CONTEXTUAL DECLARATIONS

R. A. Freiburghouse

Contexts which may possibly yield attributes or which may constitute a declaration are recognized by the parse. For each such context the parse calls a module which records in a data base the identifier and the context in which it occurred. This data base consists of a list of context nodes rooted in block nodes. Each block node contains a list of context nodes; each context node represents an identifier and contains a record of all the contexts in which that identifier was used in that block.

The first module of the declaration processor is the "context processor". This program scans the block structure working from the bottom to the top. For each block it scans the list of context nodes and for each context node it performs the actions shown in the attached flow chart. Note that the flow chart does not show what occurs when the "identifier" was previously declared as a result of contextual declaration processing. If the first declaration resulted from a "weak" context and the second resulted from a "strong" context, the first declaration will be overwritten by the new declaration. Weak contexts are those which result in "implicit" or "builtin" declarations. Strong contexts are those which result in "contextual" declarations as defined in the PL/I Language Specification.

Upon completion of the "context processor" the declarations produced by the parse from declare statements and the declarations produced by the "context processor" look alike and are processed by the declaration processor as described in BZ.8.04.

Identifiers which are not used in a "strong" context or do not appear on the left side of an assignment operator will be declared as "implicit declarations" when they are encountered during semantic translation. Similarly the default return values for function references are determined when the function reference is encountered during semantic translation.

Flow chart of the processing performed for each context node.