5. THE FORMAT STATEMENT

5.1. Introduction

The FORMAT statement defines an additional statement to be added to one of the defined classes. A Format Class Dictionary can be thought of as an extendable PHRASE dictionary. In place of all alternatives having to be defined together, we are able to define alternatives as they are required. Due to this similarity the routines required for the implementation of the FORMAT statement are the same as those required for the implementation of the PHRASE statement.

5.2. FORMAT

The Master Routine recognizes the phrase FORMAT and enters the Format Routine (213). This reads in the whole of the Format Statement, converts the Phrase Identifiers using Routine 230 and analyzes the statement with respect to the following built in syntax.

PHRASE [FORMAT BODY] = [PI] = [FORMAT PHRASE] [SERIAL NO] [EOS]

where

PHRASE [SERIAL NO] = [,] [N], [,] [158], NIL

and

152	PI
158	FORMAT PHRASE

These two Phrases have been used in compiling the PHRASE statement. The second alternative to the [SERIAL NO] is not documented as far as I can see and probably not implemented. Once the Analysis has been completed correctly, a check is made that the Format Class Dictionary to be altered is in the Chain Store.

If a Serial Number is given, then this is allocated to the Format Routine. If not, then the next free position in the INDEX is allocated.

A count is made of the number of Phrase Identifiers in the FORMAT and this is compared with the last word of the dictionary which is always adjusted to contain the maximum number of Phrase Identifiers +1 in any Format.

As for the Phrase Statement, the routine 225 is used to merge the new entry into the dictionary.