

19. SYSTEM CLASSES

19.0 Permissible limitations

Either of the following limitations may be imposed on the use of system classes:

- a. A system class may be used as prefix on one single block level only.
- b. There will be as many incarnated copies of a system class as the number of different block levels where the system class has been used as prefix.

Limitation a implies that the dynamic validity of a parameter which is a system class may be completely checked at compile time for ordinary procedures.

The following restrictions may be imposed on the use of system classes as prefixes (p = permitted, n = not permitted):

	block	class
SIMULATION	p	p
process	n	p
MAIN PROGRAM	n	n not user accessible
SIMSET	p	p
linkage	n	p
link	n	p
head	n	p
BASICIO	p	p
FILE	n	n not user accessible
infile	(p)	(p)
outfile	(p)	(p)
directfile	(p)	(p)
printfile	(p)	(p)

19.1 Implementation of the class "SIMSET"

The class SIMSET and its associated procedures are assumed to be coded in assembly language.

1. For computers with a word size permitting two references in one word, the links used for set inclusion, SUC and PRED, would be conveniently put into the first word following the (MDP,PP) word in the object. The store collapse must know about this special form of packing, as both links must be followed and updated.
2. The validity of the parameter to a procedure declared within the class SIMSET must be checked by the compiler.
3. The procedures in the class SIMSET must be called by the user by remote referencing. The ref expression pointing to the linkage, link or head object must be considered as an implicit parameter to the runtime subroutines.

A set should be implemented as an off-line item.

The head must be an ordinary object with an MDP and a PP. The links SUC and PRED may be packed into one word provided that the compiler, the procedure parameter mechanism and the store collapse take this into account. An actual prototype need not be provided for the class head if the runtime system and the compiler is suitably written.

Prototypes for the classes linkage, link and head and for the procedures suc, pred, out, follow, precede, into, first, last, empty, cardinal and clear are not required if the compiler and the runtime system reflect this fact.

19.2 Implementation of the class "SIMULATION"

The main differences between various implementations of the class SIMULATION would probably be found in the organisation of the sequencing set (SQS).

Three logical alternative approaches are:

1. Linear list (ordered on time)
2. Unbalanced tree
3. Balanced tree

The reader should refer to relevant literature on list processing for details.