

## SETL Character Set - The Final Decisions

Robert B. K. Dewar  
Art Grand

This newsletter is a proposal for the final disposition of the remaining character set questions in SETL. It is in the form of a description of the handling of character sets as it might appear in the SETL manual.

## The Standard Character Set

The standard character set for SETL consists of the following upper case ASCII characters:

<	Less than
>	Greater than
(	Left parenthesis
)	Right parenthesis
'	Quote
.	Period
,	Comma
:	Colon
;	Semicolon
/	Slash
+	Plus
*	Asterisk
-	Minus
\$	Dollar sign
?	Question mark
_	Underline

All implementations of SETL will provide the standard character set in the form stated above, or in the case where one or more characters is simply not available, at least a one to one mapping of the standard character set. In the latter case, the mapping used is at the discretion of the implementation. If there is a standard translation from ASCII to the native character set of the machine in use, then it is usually advisable to follow it in implementing the one to one mapping. If there is a local implementation of PL1, further guidance may be obtained from it since many of the characters are identical in the two languages.

This means that programs which use only characters from the

standard set can be ported from one implementation to another with at most a one to one mapping of characters.

### Publication Character Set

The publication character set includes the following additional characters, which are alternates either for standard keywords or for sequences of special characters:

Character	Use
{ left brace	Left set bracket <<
} right brace	Right set bracket >>
[ left bracket	Left tuple bracket (/
] right bracket	Right tuple bracket /)
# number sign	<del>A</del>
" quote mark	.
vertical bar	ST <del>WHERE</del>
* division sign	/
x multiplication sign	*
∀ forall sign	<del>FOR</del> FORALL
∃ exists sign	EXISTS
∄ not exists sign	NOTEXISTS
∈ membership sign	IN
∉ non-membership sign	NOTIN
≠ not equal sign	/=
≥ greater or equals	>=
≤ less or equals	<=
¬ not	NOT
∧ and sign	AND
∨ or sign	OR

U	union	+
∩	intersection	*
⊆	subset	SUBSET
⊃	inclusion	INCS
←	left arrow	:=
↑	exponentiation	**
	<del>pair former</del>	

The publication set is particularly intended for communication of algorithms in print. Note that in certain cases, the publication characters only make sense for certain uses of the characters they replace. For example, the union sign, which is an alternate for + only makes sense with sets. Nevertheless it is formally permissible to write the union of two integers (which yields the integer sum) although it is to be hoped that such clearly undesirable dictions would be avoided. The reason for being permissive in this manner is to avoid any formal differences in semantics between programs written in the standard character set and in the publication character set.

Individual implementations may implement some or all of the publication characters, possibly substituting appropriate graphics which are available in the character set at hand. It is not the intent that implementations go to heroic efforts to implement the publication set by using peculiar characters. On the contrary, characters should only be made available if they make reasonable visual sense. Programs which make use of any of these implementation dependant publication characters may not be easily portable to other implementations (which do not provide the same selection of implemented characters). However, it is always possible to translate any publication characters into the standard character set equivalent without changing the meaning of the program.

### Implementation

In the SETL compiler there will be a table with one entry for each character in the standard set and one entry for each character in the publication set. For the characters in the standard set, an appropriate character code must be specified, preferably by using the numeric value in the character set being used (to avoid the appearance of peculiar graphics in the source as much as possible). For the publication character entries, either a numeric code is supplied in a similar manner, or an indication is made that the corresponding character is not supported.

### Other Related Points

It is proposed that stropping be abandoned altogether and that reserved word mode be standard. Upper and lower case letters will be treated as interchangeable in all contexts.

The discussion above implies the addition of two The keyword NOTEXISTS is added (with the same meaning as NOT EXISTS) so that the special character in the publication set has a clear and consistent translation into a single keyword.

### Note on CDC 6600 Implementation

The CDC 6600 implementation currently uses the not sign for the break (underline) character and underline for the IN operator. These two characters should be switched so that SETL is consistent with other languages available on the 6600.