From: J. W. van Wingen
To: SC22/WG11
Subject: Comments on CD 11404
Status: To be reviewed by WG11 in July 1993

The DIS 11404 is a very interesting document, and I would have liked to have the time to study it thoroughly. My comments are restricted to what I discovered in topics regarding characters and strings. It may appear that some finer point escaped me, or that I did not look to the right place for an explanation.

4.1 The distinction between characters and "marks" (which I would see as "meta-characters" is very adequate. Should not be quotation-mark quote-mark, to conform with Table 4.1, like apostrophe-mark adding "mark" to the "Type" column, (hyphen mark => hyphen-mark)?

If we look now to 7.1 we understand why there are two different marks for the meta-quote: the quote-mark and the apostrophe-mark. It takes some effort (or a magnifying glass) to distinguish "’" from ’"’. (The term hyphen may cause confusion. A hyphen-minus is with SC2 the name for a "-", but the "_" is called "low line".) In 7.3.3 we now see that a character-literal is delimited by apostrophes, and a string-literal by quotes. This makes one wonder what the answer is to the old question: How to quote a quote? Or in this context: How to quote an apostrophe? I cannot find anything in 7.1 or 8.1.4 that forbids the character-literal ’’’, yet it is ambiguous.

In 10.1.4 the quote is excluded from the string-literal, and it has to be written like !quote! apparently, which is not very convenient, and excludes the usual way of writing it "" within a string. Even that makes counting elements in a string difficult. Therefore Snobol has two sets of quotes, "’" and ’"’, just the same as the meta-quote has in this DIS.

Using the same character for open-quote and close-quote is not done in some programming languages, like ALGOL 60, which have inner-strings. Quoting a close-quote is then a problem, which can be avoided by taking These three character quotes can be split, and the strings containing them can be concatenated. With this method it is possible to write a self-reproducing program. I am just wondering if that is also possible with the 11404 rules.

I am not quite happy with the relation character / octet. Many compilers do not restrict the contents of strings to visible characters only. In fact octets are often manipulated as if they were characters and shown as such. (C does the reverse.) The relation may be implementation-dependent, but if a coded character set standard is adopted for writing the program, it is fixed. Now that several SC2 standards (but not ASCII or ISO 646 and 8859-1) have many octets to which no character is assigned, it presents an extra burden to compiler writers to check for prohibited octets, and a nuisance to users. Therefore a datatype "octet-string" would very useful indeed. In practice it could become indistinguishable from a character string. It would be like a picture by M C Escher, where you start with birds and ends with fishes. Anyway these things exist, and the whole of the IBM OS/MVS softwarewould be unthinkable without octet strings. It is just that octets are ordered units, the bitpatterns are not of interest usually. It is the mapping of characters to octets that makes coded characters an ordered set, in contrast to characters of a repertoire which are still unordered.

Annex A presents a very strange selection of standards. The fundamental ISO 2022 is even left out, and the never implemented, nor approved DIS 6862 included. A recent list is appended.

INTERNATIONAL STANDARDS FOR CHARACTER CODES AND RELATED SUBJECTS
Version 3.4 of 1992-12-01
Johan W van Wingen

DIS: Draft International Standard, not yet approved by ISO
CD: Committee Draft (formerly DP : Draft Proposal)
(standards marked with 1993 are approved, but awaiting publication)

ISO 646:1991	ISO 7-bit coded character set for information interchange
ISO 9036:1987	Arabic 7-bit coded character set for information interchange
ISO 2022:1986	ISO 7-bit and 8-bit coded character sets - Code extension techniques (under revision)
ISO 6937:1993	Coded graphic character set for text communication - Latin alphabet
ISO 4873:1991	8-bit code for information interchange - Structure and rules for implementation
ISO 8859	8-bit single byte coded graphic character sets, in Parts:
ISO 8859-1:1987	Latin alphabet no. 1
ISO 8859-2:1987	Latin alphabet no. 2
ISO 8859-3:1988	Latin alphabet no. 3
ISO 8859-4:1988	Latin alphabet no. 4
ISO 8859-5:1988	Latin/Cyrillic alphabet
ISO 8859-6:1987	Latin/Arabic alphabet
ISO 8859-7:1987	Latin/Greek alphabet
ISO 8859-8:1988	Latin/Hebrew alphabet
ISO 8859-9:1989	Latin alphabet no. 5
ISO 8859-10:1993	Latin alphabet no. 6
ISO 10367:1991	Repertoire of standardized coded graphic character sets for use in 8-bit codes
ISO 10646:1993	Multiple-octet coded character set
ISO 6429:1993	Control functions for 7-bit and 8-bit coded character sets
ISO 10538:1991	Control functions for text communication
ISO 2047:1975	Graphical representations for the control characters of the 7-bit coded character set
ISO 2375:1985	Procedure for the registration of escape sequences
ISO 7350:1991	Text communication - registration of graphic character subrepertoires
ISO 5426:1983	Extension of the Latin alphabet coded character set for bibliographic information interchange
ISO 5427:1983	Extension of the Cyrillic alphabet coded character set for bibliographic information interchange
ISO 5428:1984	Greek alphabet coded character set for bibliographic information interchange
ISO 6438:1984	African coded character set for bibliographic information interchange
ISO 6861 DIS	Cyrillic alphabet coded character sets for Slavonic languages for bibliographic information interchange
ISO 6862 DIS	Mathematical coded character set for bibliographic information interchange
ISO 8957 CD	Hebrew coded character set for bibliographic information interchange
ISO 10585 DIS	Georgian coded character set for bibliographic information interchange
ISO 10586 DIS	Armenian coded character set for bibliographic information interchange
ISO 10754 DIS	Extension of the Cyrillic alphabet coded character set for non-Slavic languages for bibliographic information interchange
ISO 6630 ?	Bibliographic control functions
ISO 8884:1988	Keyboards for Multiple Latin-alphabet Languages: Layout and Operation
ISO 9995	Keyboard Layouts for Text and Office Systems, in Parts:
ISO 9995-1 DIS	General Principles Governing Keyboard Layouts
ISO 9995-2 DIS	Alphanumeric Section
ISO 9995-3 DIS	Common Secondary Layout of the Alphanumeric Zone of the Alphanumeric Section

ISO 9995-4 DIS	Numeric Section
ISO 9995-5 DIS	Editing Section
ISO 9995-6 DIS	Function Section
ISO 9995-7 DIS	Symbols Used to Represent Functions
ISO 9995-8 DIS	Allocation of Letters to the Keys of a Numeric Keyboard
ISO 9541	Font Information Interchange, in Parts:
ISO 9541-1:1991	Architecture
ISO 9541-2:1991	Interchange Format
ISO 9541-3 DIS	Glyph Shape Representation
ISO 9541-4 CD	Application-specific requirements
ISO 10036:1991	Procedure for registration of glyph and glyph collection identifiers

Correspondence between ISO and ECMA standards

ISO	ECMA	Registration number of escape sequence (ISO 2375)
8859/1	94	100
8859/2	94	101
8859/3	94	109
8859/4	94	110
8859/5	113	111
8859/6	114	127
8859/7	118	126
8859/8	121	138
8859/9	128	148
8859/10	144	157