TECHNICAL REPORT

ISO/IEC TR 9789

First edition
1994-12-15

Information technology — Guidelines for the organization and representation of data elements for data interchange — Coding methods and principles

Technologies de l'information — Principes directeurs pour l'organisation et la représentation des éléments de données pour l'échange de données — Méthodes et principes de codage

Reference number
ISO/IEC TR 9789:1994(E)

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization.
National Bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC/JTC 1.

The main task of technical committees is to prepare International Standards. In exceptional circumstances a technical committee may propose the publication of a Technical report of one of the following types:

- type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

- type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility for an agreement on an International Standard;

- type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard; ("state of the art", for example).

Technical reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transferred into International Standard. Technical reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

ISO/IEC TR 9789, which is a Technical Report of type 3, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology* Subcommittee 14, *Data element principles.*

Introduction

This Technical Report is a guide to develop and implement coded representations.
Coding covers the way and the form in which data are expressed.

The increased use of data processing and electronic data interchange heavily relies on accurate, reliable, controllable and verifiable data recorded in data bases.
In formal communication and storage data are expressed in symbols (usually digits or letters), arithmetic numbers and descriptions, which should have a fixed stable meaning for every one involved and thereby be suitable for purposes of processing and communication.

This Technical Report presents the objectives of coding, the characteristics, advantages and disadvantages of different coding methods, a survey of the features of codes and guidelines for the design of codes.

This page intentionally left blank

Information technology -- Guidelines for the organization and representation of data elements for data interchange -- Coding methods and principles

1 Scope

This Technical Report provides general guidance on the manner on which data can be expressed by codes.

It describes the objectives of coding, the characteristics, advantages and disadvantages of different coding methods, the features of codes and gives guidelines for the design of codes.

This Technical Report is not directed toward any specific application area nor dependent on any design method for application systems or data interchange.

2 References

2.1 General references

ISO/IEC 646:1991, *Information technology -- ISO 7-bit coded character set for information interchange.*

ISO 2382-4:1987, *Information processing systems - Vocabulary - Part 04:Organization of data.*

ISO 2375:1985, *Data processing - Procedure for registration of escape sequences.*

ISO 7064:1983, *Data processing - Check character systems.*

ISO/IEC 11179-3:1994, *Information technology - Specification and standardization of data elements - Part 3: Basic attributes of data elements.*

2.2 Examples of applications of this Technical Report

ISO 9735:1988, *Electronic data interchange for administration, commerce and transport (EDIFACT) - Application level syntax.*

ISO 8601:1988, *Data elements and interchange formats - Information interchange - Representation of dates and times.*

ISO 3166:1993, *Codes for the representation of names of countries.*

3 Definitions

For the purpose of this Technical Report, the following definitions apply.

3.1 attribute: A characteristic of an object.

3.2 character set: A finite set of different characters that is complete for a given purpose.

Example: The international reference version of the character set of ISO 646.

3.3 code: A collection of rules that maps the elements of one set on to the elements of another set.

NOTES

1. The elements may be characters or character strings.
2. The first set is the *coded set* and the other set is the *code element set*.
3. An element of the code element set may be related to more than one element of the coded set but the reverse is not true.

3.4 code element: The result of applying a *code* to an element in a *coded set*.

Examples:

1 "CDG" as the representation of Paris Charles de Gaulle in the code for three-letter representation of airport names.

2 The seven binary digits representing the delete character in ISO 646.

3.5 code element set: The result of applying a *code* to all elements of a *coded set*.

Example: All the three-letter international representations of airport names.

3.6 code set: Synonym of **code element set.**

3.7 code value: Synonym of **code element.**

3.8 coded representation: Synonym of **code element**.

3.9 coded set: A set of elements which is mapped on to another set according to a *code*.

Example: A list of the names of airports which is mapped on to a corresponding set of three-letter abbreviations.

3.10 coding scheme: Synonym of **code.**

3.11 data code: See preferred term **code element**.

3.12 data element instance: An occurrence of a *data element type*.

3.13 data element type: A category of data which represents a concept and whose properties are expressed as a set of data element type attributes which permit it to support information interchange.

3.14 delimiter: One or more characters used to indicate the beginning or the end of a character string.

3.15 entity: Any concrete or abstract thing of interest, including associations among things.

3.16 field: A specified area on a data medium or in storage, used for a particular class of data elements.

3.17 identifier: One or more characters used to identify or name a data element type and possibly to indicate certain properties of that data element type.

3.18 key: An identifier within a set of data element types.

3.19 position: Any location in a *string* that may be occupied by an element and that is identified by a serial number.

3.20 string: A sequence of elements of the same nature, such as characters, considered as a whole.

3.21 table: An arrangement of data each item of which may be identified by means of arguments or keys.

4 Principles of coding

4.1 Information and coding

In daily life information is understood as facts of and propositions about all the concrete or abstract things of interest expressed by: data, messages and further particulars.
Information is necessary for the proper execution of any conceivable task be it in administration, commerce, transport, science, etc.
Accurate, objective and unambiguous information is a prerequisite in cases of computer based information systems and the data interchange between them.

In formal information systems data is expressed in symbols (usually digits or letters), arithmetic numbers and descriptions, which should have a fixed, stable meaning for every one involved and thereby be suitable for purposes of processing and communication.

Users, irrespective of their function or tasks, should be able to understand, interpret and handle their information correctly.
Information shared by different user groups or application systems has to have an agreed definition e.g. semantic meaning of a concept (connotation) and all instances of a concept (denotation) and an agreed representation.

Coding covers the way and the form in which data is expressed by codes.
It is necessary to make clear agreements on these representations.
An explanation of the representation forms and code elements is part of the specification of data.

4.2 Coding

By coding is understood the rule-based assignment of code elements to a named and defined set of elements in an orderly way.
Coding is mostly done by means of symbols, (usually digits or letters), resulting in a concise representation.

Example: The assignment of code element: "CDG" as the concise representation of airport name: "Paris Charles de Gaulle". This airport name belongs to the set of airport names maintained by the International Air Transport Association (IATA). IATA has set the rules for establishing the concise representations.

Coding is a necessary tool for information processing. Coding of information enables it to be recorded, interpreted, processed and transmitted by humans and/or by machines.
All kinds of information can be coded: information about products, persons, processes, documents, countries, currencies, packages, etc.

Before making agreements on the coding methodology, i.e. the representation of information concerning events, actions, concrete or abstract objects in the real world, it must be investigated which data are relevant for the intended application. Information analysis of the universe of discourse concerned has to determine the role of the data in the information structure thereof.
In doing this a clear distinction should be made between identification, classification and reference needs.

5 Coding objectives

Information about any abstract or concrete object, action, or event of interest, (its characteristics or attributes) can be coded. Before making agreements on the configuration of their representation, the coding rules, it is necessary to determine the objective of the coding effort. It is not enough to design an ordered short representation of certain data. First the information requirements must be clear.

The following requirements generally occur:

- identification
- classification
- key to further information.

When data modelling is applied for the specification of application systems or messages for use in data interchange, the objectives of the users in the application environment will determine the choice of the entities and attributes to be taken into account as well as their interrelationship. The methods to be used for identification, classification or referencing will depend on those objectives.

5.1 Identification

The purpose of identification is to distinguish elements of a set from each other.
To be able to do this it must first be determined which characteristics have to be taken into account. Based on the selected characteristics comparisons can be performed and it can be ascertained whether an element of the set is equal to another element or different from it. To which degree of detail characteristics have to be recorded to indicate similarity or difference of elements of the set, depends on the area of application for which identification is needed.

Example
For its stock control of stationery an organization wants to identify various types of sheets of paper. Sometimes it may be sufficient to distinguish the various formats: A3, A4 or A5. Depending on the utilization of the sheets of paper other characteristics may be added, such as thickness, weight, chemical composition. If no specific requirements have to be met the recording of the format may suffice. In other cases, where handling and processing of the sheets of paper are critical the necessary characteristics have to be mentioned.

So, identification can be defined as:
The systematic registration of characteristics of elements of a set in such a way that they can be distinguished from each other.
The characteristics to be distinguished are inalienably part of an object or concept of interest.
The extent of the details to be observed is dependent on the user's objectives and the area of application.
These criteria result in the design or selection of an identification system.

Examples of details to be distinguished:

In the real world a person has a family name and a given name, is born in a country, on a certain date, resides in a country, his eyes have a certain colour, etc.

Application 1: In a governmental application system one wants to distinguish the colours of eyes of citizens.

Application 2: In an application system for medical research project one wants to distinguish the colours of eyes of human beings.

In these applications the required degree of discrimination need not be identical.
In the medical research application, it may be required that more colours of eyes of human beings are to be distinguished than in an application of a governmental body registering the colours of eyes of citizens.

In practice this may result into a compromise when selecting an identification system. Often the choice of a system is then determined on basis of the wish to have a minimum number of identification

systems to accommodate a maximum number of functions.

The objectives and applications determine which intrinsic characteristics will be taken into consideration.

5.2 Classification

The purpose of classification is to group objects or concepts of interest into classes in accordance with predetermined characteristics based on which similarities can be ascertained.
Classification is often used to support decision making or to get insight on trends or developments, without having to examine each instance of a set separately.

So, classification can be defined as:
A systematic arrangement of elements in groups or categories based on the similarity of predetermined characteristics.
Classification is done by means of control characteristics, i.e. those characteristics which have been assigned or are related to an object or concept of interest.
These characteristics may be intrinsic or extrinsic.

Example of control characteristics:
Place of manufacturing of products, turnover speed, market sector, production process for a product.

The information requirements and the business policy are determinative for the choice of control characteristics.

An organization may choose to apply various classification systems for the same type of objects, dependent on different needs.

Example:
A product may be classified according to
- function on behalf of sales
- manufacturing process on behalf of production
- value on behalf of inventory control
- volume/weight on behalf of transport
- type on behalf of Customs or statistical requirements

5.3 Key to further information

A key is an identifier within a set of data element types. Within the context of an application or data interchange a key shall be unique. In many application systems a reference number is needed as key to further information. The key in itself can be meaningless, but it gives access to the data required.

Examples:
- an order number may give access to the party to whom the order was sent, on which date, and the goods or services ordered;
- an article number may be related to a description, its price, its production process, the place of manufacturing;
- a salary number may refer to an employee, his name and address, his birthdate, his rank, his salary.

Reference numbers may be identifying in one application area, and classifying in another.

6 Types of codes

This chapter provides a description of basic coding methods. It is intended to assist in selecting appropriate code structures based upon specific application requirements and the nature of the elements in the set to be coded. It also provides principles and criteria to be considered in assessing alternative code structures, and mentions advantages and disadvantages of each coding method.

The choice of code structures is fairly extensive. The following information, however, should help to select the best method.

6.1 Forms of codes

The coding methods discussed in this chapter are outlined by the following listing. The set of methods shown is not exhaustive but does include all the significant types.
Many code structures applied in practice are often combinations of these basic types.

Non-significant codes
Sequential
  incremental sequential
  group sequential
  arranged sequential (chronological, alphabetical)
Random

Significant codes
Mnemonic
Abbreviation based
Matrix
Hierarchical
Juxtaposition
Combination
Value addition

6.2 Sequential codes

6.2.1 Principle

Elements of a set to be coded are assigned a number taken sequentially from an ordered set of numbers. These numbers are mostly natural integer numbers (e.g. beginning with "1") but alphabetic characters may also be used, e.g. AAA, AAB, AAC ...

Sequential assignment of code elements may be based on lists built in various ways, for example:

1. The list of natural integer numbers limited to the number of possibilities wished to be available.
2. Lists of numbers arranged on the basis of an algorithm, e.g. only even numbers or multiples of 10.

6.2.2 Use

Sequential codes are generally used as self contained codes for identification or referencing purposes, or as part of a composite code, often in addition to a classifying code.

Remark
In a numerically defined field with a fixed number of positions leading zeroes must be used to fill the field up to the number of positions required, whenever appropriate.

Example: In a field defined as 3-numeric "one" is coded 001, and "fifteen" is coded 015.

6.2.3 Types of sequential codes

There are four types of sequential codes:
- incremental;
- pure incremental;
- group;
- arranged.

6.2.3.1 Incremental sequential codes

Elements of a set to be coded are assigned a code value determined by increasing the previously assigned code element by a predefined number, e.g. 1 (= pure incremental) or 2 in case of even numbers, or 10 if only multiples of 10 may be assigned).
With this method a code value does not express any meaning. Similar elements of the set are not grouped.
The rationale for assigning code values other than increasing by 1 can be the requirement to use the intermediate code values for subsequent modifications of the original coded set.

Advantages
- Fast assignment of code values.
- Conciseness.
- Easy validation of coded representations.

Disadvantages
- Classification or grouping of elements of a set is impossible by means of the coded representations.
- The maximum capacity is not fully used.

6.2.3.2 Pure incremental sequential codes

Elements of a set to be coded are assigned a code value determined by increasing the previously assigned code value by 1.
With this method a code value does not express any meaning. Similar elements of the set are not grouped.
In addition to the advantages mentioned in 6.2.3.1, another advantage is that the maximum capacity is used.

6.2.3.3 Group sequential codes

With this type of code ranges code values are assigned to categories of elements of the set which have something in common.
Before assigning a code value the category to which the element of the set belongs must be determined. It is then assigned the next higher code value belonging to the range of its category.

Example
Code for products in the area of oil refining.
In this code, each product is represented by a 4-digit number taken sequentially from a series that is characteristic of the grouping of products.

These groups of products, which have been determined in advance, and the corresponding ranges of code values are:

- gas 1000 - 2999
- petrol and fuels 3000 - 4999

Advantages
- Fast assignment of code values.
- Conciseness.
- Easy validation of coded representations.

Disadvantages
- The maximum capacity is not fully used.

Remark
This system must not be confused with hierarchical codes (see 6.7).
Group sequential coding must only be used if the categories are stable and there is no possibility that an element of a set may belong to different categories, now or in the foreseeable future.

6.2.3.4 Arranged sequential codes

This is not a pure sequential code, although the code values may give the impression.
This type can only successfully be used if all elements of a set are known beforehand and the set will not expand.
Before assigning code values the elements of the set are arranged on the basis of some characteristic, e.g. alphabetical order of name, chronological (for events, actions). The order thus obtained is expressed again by the code values which themselves have been chosen sequentially from an ordered list.

This coding scheme makes it possible to easily retrieve and sort data element instances. It is used in cases where it is impossible or difficult to obtain the required results by use of the non-coded data element instances, e.g. because of the presence of spaces separating words or resulting from digit adjustment.

Advantages
- Fast assignment of code values.
- Conciseness.
- Easy validation of coded representations.

Disadvantages
- The maximum capacity is not fully used.

Example
In France "Départements" have been coded numerically based on a list sorted on alphabetical sequence. This code may be an element in other codes, e.g. postal code, national identity number, vehicle registration number.

A drawback of this system is that it cannot accommodate extensions, which will probably arrive in future.
Taking the example above: "Département Corse" was assigned code value 20. When "Corse" was divided into "Haute Corse" and "Corse du Sud" one was obliged to use letters (2A and 2B, respectively) in a numeric code and thus created anomalies.

6.2.4 Conclusions

When selecting a sequential code the pure incremental type offers the best solution for purposes of flexibility, expendability and ease of assignment.

6.3 Random codes

6.3.1 Principle

Elements of a set are assigned a code value from a set of possible non-arranged code values or by means of an algorithm. There is no correlation whatever between the element of the set and its code value. This method differs from sequential coding in that the latter uses sequentially ordered ranges for code value assignment.
The purpose of this method is to render any interpretation impossible.

6.3.2 Advantages

- Easy and fast assignment of code values, possibly automated.
- Conciseness.
- Use of maximum capacity

6.3.3 Disadvantages

- Classification or grouping of elements of a set is impossible by means of the coded representations.
- Need for a pre-established list or an algorithm to generate random numbers, provided that any duplication of numbers is excluded.

6.3.4 Use

Random codes may be used as self contained codes for identification purposes or as part of a composite code where the other part(s) are based on other coding principles.

6.4 Mnemonic codes

6.4.1 Principle

Elements of a set are assigned a coded representation by taking one or more characters from their names.
The purpose of this method is to enhance the efficiency of user memorization of code values.

6.4.2 Advantages

- Ease of user memorization of code values, thus avoiding frequent consultation of code lists.
- Facilitation of the detection of errors.

6.4.3 Disadvantages

Coding closely depends on the way in which the elements of the set are originally expressed (language, measurement system, etc.). A risk exists when creating mnemonics which, in the user's language, has a meaning with no connection to the occurrences represented.

6.4.4 Use

Mnemonic codes can be usefully applied for limited identifying code sets which are rather stable and of which the names for elements of the set are commonly known in the user's environment.

Example 1
From: ISO 3166 Codes for the representation of names of countries

Entity name	Country code
Austria	AT
Canada	CA
France	FR
United States	US

Example 2
From: UN/ECE Recommendation 20 Codes for units of measurement

Unit name	Coded representations
Centilitre	CLT
Centimetre	CMT
Cubic centimetre	CMQ
Cubic foot	FTQ
Degree Celsius	CEL
Degree Fahrenheit	FAH
Gallon	GLI
Kilogram	KGM
Kilometre	KMT
Kilowatt	KWT
Metre	MTR
Minute	MIN
Piece	PCE
Square centimeter	CMK
Square foot	FTK

6.5 Abbreviation based codes

6.5.1 Principle

The essential characteristic of this code is the method of abbreviating the elements of the set denomination which is defined uniformly right from the start.

The purpose of this method is to generate a shorter representation directly from the original name of the element of the set. This method resembles, but is different from, a mnemonic code as regards its purpose.

6.5.2 Advantages

Easy compression of sometimes voluminous data extent.

6.5.3 Disadvantages

Uniqueness of the code value resulting from the abbreviation process cannot be guaranteed, without re-examination of all the code values after each added code value.

6.5.4 Use

Use of this type of code is recommended for large sets of elements, when it is required to reduce the size of the representations of the elements of the set.

Example

Telex answerback codes
The answerback code is a string of characters that is automatically transmitted by a telex terminal after it has received a control character designated as WRU (who are you?). By agreement, the length of the string is restricted to at most 14 characters.

These coded representations are of a variable number of characters, but must not exceed 14 characters. The codes are most frequently made up by the contraction of the name of the organization concerned. The coded representations do not necessarily involve the Telex calling number.

Organization	Call number	Answerback
Comptoir des viandes européennes 5, rue de la Corderie, RUNGIS	204 045	CEV 204 045F
Tuilerie générale du midi RN de Clermont, BEDARIEUX	490 695	TEGEMIDI BEDAR
BENDIX Automobile Division de DBA	611 138	DBABX 611138

6.6 Matrix codes

6.6.1 Principle

A matrix code is based on the existence of a double entry table. The values assigned to the rows and columns of this table are used to make up the coded representations of the elements of the set within this table at the relevant coordinates.
The purpose of this method is to assign meaningful code values to elements of the set which have a number of characteristics in common in different combinations.

6.6.2 Advantages

- Interpretability of code values.
- Easy assignment of code values.

6.6.3 Disadvantages

- Need for pre-establised tables, covering all the characteristics of the elements of the set to be coded.
- Difficult to accommodate new requirements such as new or modified characteristics, and new combinations.

6.6.4 Use

Matrix codes can be usefully applied for identification purposes of elements of a set where the characteristics are well structured and stable.

Example
The following table may be used to code light bulbs based on their characteristics. For instance, the standard clear 220/230 V bulb, power 60 W, B22 fitting with a coded representation 04 B01 may be interpreted as follows:

04: table number
B : line number corresponding to a power of 60 W
01: column number corresponding to B22 fitting

Table 04	Standard clear bulb 220/230 V			Standard clear bulb 250/260 V	
	Fitting 01	02	03	Fitting 04	05
40 W	A B22	E27	B22	E27	
60 W	B B22	E27	B22	E27	
75 W	C B22	E27	B22		
100 W	D B22	E27	B22	E27	
150 W	E B22	E27	E27		
200 W	F B22	E27	E10		

6.7 Hierarchical codes

6.7.1 Principle

A hierarchical code is based on the breakdown of a set of elements to be coded into successively and increasingly smaller groups.

Each group at a higher hierarchical level includes all the groups of its lower levels and only these groups. This type of code is based on the differences between the characteristics of the elements at each level.
The characteristics at each level must be mutually exclusive.

6.7.2 Advantages

- Ease of classification or grouping of elements.
- Enables reporting at higher aggregation levels.
- Interpretability of code values.

6.7.3 Disadvantages

- Limited use of theoretical capacity.
- Lack of flexibility due to the refinement principle (where a level may become overcrowded or higher levels cannot be rearranged without a total review of the code).
- The need to follow the sequence of code levels from the highest level downwards in order to assign a code value or interpret it.
- The complexity, dependent on the number of levels, risking the reintroduction of a characteristic that has already been used at a higher level.

6.7.4 Use

Hierarchical codes are generally used for classification purposes. The number of levels established depends on the information requirements. Hierarchical codes are less suitable for identification and referencing purposes.

Hierarchical codes are very suitable e.g. for statistical purposes, for reporting on goods movements, for classification of publications on subject. Both fixed and variable formats occur in practice. Fixed formats are more easily processed than variable ones.

Example 1

Fixed incremental format

From: Harmonized Commodity Description and Coding System (Harmonized System)

The harmonized system code format consists of 6 numeric positions, split up in interdependent segments of 2 positions.

37	Photographic and cinematographic goods
3705	Photographic plates and films exposed and developed, other than cinematograph film
370520	Microfilms

Example 2

Variable incremental format

From: Universal decimal classification (UDC)

The number of characters and segmentation of the coded representations are variable, so that the degree of detail of the description can be extended to the desired level.
The concept of "slope of roofs in architecture" is expressed by the coded representation 624.024.13.

```
624           Civil engineering
624.02      Construction components
624.024   Roofs, roofing
624.024.13  Roof slopes
```

6.8 Juxtaposition codes

6.8.1 Principle

Juxtaposition codes are composite codes consisting of segments providing the characteristics chosen to represent the elements of the set. These characteristics are independent of each other. The coded representations combined in this way may be of any type (sequential, mnemonic, random).

6.8.2 Advantages

- Ease of grouping of elements of the set is based on one or more characteristics expressed in the code values.
- Capacity related to the number of values which each characteristic can take.
- Interpretability of code values.

6.8.3 Disadvantages

- Requirement to include a great number of characteristics may result in a large number of characters per code value.
- Difficult to accommodate requirements for new characteristics.

6.8.4 Use

Juxtaposition codes are very suitable for classification of articles that have several characteristics in common.
Fields of application are e.g. traceability coding (which product, where and when manufactured) or group technology approaches for development and manufacturing.

Example

```
Coding of metals:
NNNN     NN     NNN
grade    shape dimension
```

The three types of characteristics (grade, shape and dimension) are, to a large extent, independent of each other.

6.9 Combination codes

6.9.1 Principle

Combination codes are composite codes consisting of segments providing the different characteristics of the elements of the set. These characteristics are dependent on each other and generally have a hierarchical relationship. In this sense they differ from juxtaposition codes.

6.9.2 Advantages

- Easy assignment of code values.
- Facilitates the allocation and maintenance of code values.
- Enables to a certain extent the interpretability of the code values.
- Facilitates the validation of code values.

6.9.3 Disadvantages

- Theoretical capacity may not be fully used.

6.9.4 Use

Combination codes are often applied for identification purposes covering a wide area of application.

Example of the European Article Number code

```
EAN code
nnnnnnnnnnnnn       complete EAN-product identification
nn                  country
 nnnnn          enterprise (affiliation number)
  nnnnn     product identification
            n   check digit
```

6.10 Value addition code

6.10.1 Principle

With this type of code only numbers representing powers of 2 are permissible and valid for the representation of elements of the set. This type of code can be applied when elements of the set have to be combined.
The coded representation of each occurrence is a power of 2:
1, 2, 4, 8, 16, 32, 64, 128, 256, 512, etc.

A table of values is drawn up by assigning a value of a power of 2 to each occurrence:

Occurrence	Value	Occurrence	Value
1st	1	6th	32
2nd	2	7th	64
3rd	4	8th	128
4th	8	9th	256
5th	16	10th	512

6.10.2 Advantages

Conciseness.

6.10.3 Disadvantages

Algorithm needed for assignment and interpretation.

6.10.4 Use

This method is suitable for small sets of elements, requiring many combinations, provided that their order is indifferent. By using the table it is possible to assign a unique coded representation to any combination by adding the values of all relevant elements.

Example

The combination of elements 1, 8, 4 and 9 will be coded 393, being the sum of 1, 128, 8 and 256.

With knowledge of the table of attributes of a code of this type, it is always possible to identify its elements because with the numbering chosen (powers of 2), 393 can only be the sum of 256 + 128 + 8 + 1.

7 Features of codes

7.1 General features

The general features of a code are the following:

- Uniqueness
- Expandability
- Conciseness
- Simplicity
- Versatility
- Suitability for sorting
- Stability
- Significance
- Size
- Structure and format
- Capacity

7.2 Uniqueness

Uniqueness is a basic characteristic of any code.
Uniqueness is obtained if one and only one element of the set corresponds with one code element and vice versa.

7.3 Expandibility

This means the adaptability of the code to accommodate requirements for new elements to be added at the right place in the coded set, without having to recode or restructure the code. The code should allow for extension of existing subdivisions as well as for addition of new subdivisions.
The planned duration of a code must take into account the number of elements of the set to be coded and the stability or dynamics of the environment governing the extension of the set of elements.

7.4 Conciseness

The number of characters of a coded representation governs the capacity of a code.
A coded representation should have the smallest number of characters required to code current and future elements of a set.

A short coded representation is advantageous for:
- data entry
- data exchange by electronic means or otherwise
- storage in data bases
- data processing.

7.5 Simplicity

The coding method should be easily understood and applied by users, also by less experienced ones. Conciseness and choice of the character set contribute to this as much as the clarity of the concepts chosen, definition of structures, the absence of exceptions and deviations, and the effectiveness of assignment procedures.

7.6 Versatility

The code should be applicable in different related application areas. This supports system integration.

7.7 Suitability for sorting

If it is required to generate or process data in a prescribed order the code should facilitate this requirement.
Suitability for sorting depends closely on the type of code chosen.
In composite codes (e.g. hierarchical codes) sorting and/or grouping possibilities of greater or lower complexity may be chosen depending on the nature of the elements of the set and the purpose of the system. These possibilities depend basically on the number of segments provided for in the code structure or the significance expressed in the code values.

7.8 Stability

A code is stable when it makes allowance for projected variations without its structure having to be modified. At the level of coded representations this concept is seen in terms of the stability of the unique elements that they represent.

Users need a code that is stable. The assignment of code values must be done with the slightest possible chance of modifications, with respect to the code values themselves as well as to the code structure.
Modifications are costly, give rise to errors, require much time and capacity and may harm the total system if they are not managed properly.
Coded representations that are deleted when elements disappear from the set, must not be re-used for other items.

7.9 Significance

A code is said to be significant if its coded representations express their meaning directly (e.g. mnemonic codes) or indirectly by reference to one or more tables (e.g. hierarchical codes, juxtaposition codes).
Significance relates also to the possibilities of sorting and grouping on the basis of the characteristics of elements of the set using their coded representations.
Significance is particularly important in case of classification purposes. For identification and reference purposes non-significant codes are recommended.

7.10 Size

Size is expressed as the number of positions of a coded representation.
Code values can be defined as having a fixed or a variable number of characters.

Code values of a variable number of characters have two major drawbacks:

1. The number of characters is unpredictable which may give rise to justification problems, when storing the code value in a data field which accommodates more characters than used for the code value.
   Data may either have to be "left justified" or "right justified" or to be aligned on a decimal sign.

   Left and right justification errors are common in data entry (field shift). The incorrect use of coded representations having a variable number of characters in fixed format data fields may lead to different representations (such as **12, *12* or 12**, where * represents SPACE) that are not recognized by data processing devices as being the same coded representation.

2. Errors due to omission or addition of characters cannot be easily detected either by humans or by machines.
   The uncontrolled addition of prefixes and/or suffixes to an original code value is a general and often occurring problem. As prefixes and suffixes mostly are of a variable nature and will not always be present, this will easily give rise to wrong interpretations and errors.

For these reasons code values with a fixed number of characters are recommended.

7.11 Structure and format

The format (or character structure) of a code value should preferably be all numeric or all alphabetic. Mixed formats (alphanumeric formats) can be used when particular character positions, e.g. the first or last ones, are always alphabetic or numeric. Random alphanumeric formats should not be used, because there is considerable risk for human errors.

The structure of a code defines:

- the number of positions or groups of positions making up the coded representation;
- the set of valid characters for each position.

Note: Spaces can be an integral part of a structure.

Input validation to detect syntax errors relate basically to the structure. Each position of the coded representations for each group of several positions could thus be defined as: alphabetic, numeric, alphanumeric, special.

7.12 Capacity

This is the number of coded representations which may be formed by the combination of all usable characters for each position, taking account of the chosen numeric base.

Examples

1. for 1 position, base 2, with binary characters: C = 2
2. for 1 position, base 10, with numeric decimal characters: C = 10
3. for 1 position, base 26, with alphabetic characters: C = 26

These capacities are theoretical. They presuppose the use of all combinations of all characters.
Restrictions initially made for practical or conceptual reasons reduce these theoretical capacities.

When calculating the capacity of a given code for covering all situations while maintaining code uniqueness, the following formula applies (assuming 24 alpha characters and 10 numeric digits are used, because the letter I and O should be avoided whenever possible):

C = (24A) (10N)
where
C = total available code combinations possible
A = number of alpha positions in the code
N = number of numeric positions in the code
(A + N, when combined, equal the total positions of the code).

Note: The above formula assumes that a given code position is either alpha or numeric never both. If a given position can have both alpha and numeric characters, the formula becomes:

C = (36)A+N or

C = (34)A+N when the letters "I" and "O" are not used.

The position(s) occupied by check character(s) should not be taken into account when calculating capacity.

In practice, the choice of capacity is the result of a compromise between:

1. forecasts for extending the system;
2. limitations in the number of characters comprising the coded representation;
3. ease of writing and using coded representations;
4. desired useful life of the system;
5. operating costs, etc.

Capacity should be considered as a limit: it is for information, as distinct from the number of coded representations existing in the code at any given time or the number of coded representations which have been assigned since the code was set up (some of them may have been deleted in the meantime).

8 Representation of code values

8.1 Character sets

A feature of all codes is that there is a method of representation of code values which makes use of different characters.

A code is said to be:

a. numeric if it uses digits only;
b. alphabetic if it uses letters only;
c. alphanumeric if it uses both digits and letters, possibly along with special characters.

Digits

The digits used are the Arabic numerals: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.

Letters

The alphabet used is the Latin alphabet.

When required, the whole alphabet of 26 letters may be used or only part of this alphabet. The use of national characters in coded representations is out of the scope of this technical report.

It is recommended to use only letters in a single case (upper case or lower case).

Special characters

Special characters may be used in coded representations to supplement alphabetic characters (&, $ ...); in this way, capacity is increased and families of characters are available which may be reserved for specific processing

The following characters should be avoided:

1. characters which do not belong to the invariant set of ISO 646 IRV;

2. characters which can be misinterpreted or incorrectly transcribed. Thus, spaces should be eliminated wherever possible.

   Example: 123 ABC should be written 123ABC because the space has no meaning and it could be omitted when it is transcribed.

3. characters which can be used as service characters in the framework of syntax for data interchange.

   Example: question mark (?), colon (:), plus sign (+), apostrophe (') are used as service characters in class A of ISO 9735 (EDIFACT). However, an escape possibility is provided for.

   NOTE: The use of an alternative character set is permitted by extensions of the characterer sets which are registered according to ISO 2375. If a character set, other than ISO 646 IRV is specified, the subset of the character set to be used for the coded representation shall be identified by the Issuing Organization.

8.2 Representation techniques

For the physical representation of code values a number of options is available. Which option will be chosen will depend on the application area and the devices available.

For manual processing preference is usually given to human readable coded representations. In this case code values will be presented in the traditional way by character symbols using Roman letters and Arabic numerics. This presentation will also be chosen for computer output, be it in paper documents and listings or on visual display units.

If requirements ask for processing by mechanical or electronic means other methods are available.
One of the most frequently used methods is automatic identification, of which the bar coding method is in widespread use. A number of bar coding symbologies exist.
The choice of the symbology should be carefully made. It depends on the application area, the user groups and the agreements made in certain environments or functions.

For numeric code values two symbologies are in general use:

- Symbology used in the retail trade for points of sale, developed by UPC (Universal Product Code) for the North American area and EAN (European Article Number) for the rest of the world.
  This symbology is mainly used to encode manufacturer and article identification numbers on consumer products.

- Interleaved 2 of 5.
  This symbology encodes numeric data in a high density format.

For alphanumeric code values the most popular symbology is:

- Code 3 of 9 (Code 39).
  This symbology encodes a full uppercase alphabetic and numeric character set, plus several other characters, in variable length bar code symbols.

Beside bar coding other automatic identification methods exist such as optical character reading (OCR) or magnetic stripes.
A new development is the smart card provided with an integrated circuit. A smart card can hold a big volume of data which in most cases can be processed.

9 Design of codes

This chapter is intended to support the design of codes. They may help to avoid the potentially expensive consequences of inadequately conceived and developed codes.

It should be noted that often requirements may be in conflict with each other. For example, if a coding structure is to have sufficient expendability for future needs, conciseness may have to be sacrificed to some degree. Hence, all aspects must be thoroughly considered and trade-offs be made to obtain optimum efficiency for the application area concerned.

9.1 General

9.1.1 Preparation

When a new code is required sufficient time and effort must be spent in preliminary study, definition and planning of the activity. Potential problems must be anticipated and all alternatives be carefully evaluated prior to implementation of the new code.

9.1.2 Use of existing codes

Existing codes should be used whenever possible. New codes should not be designed unless absolutely necessary. The preferences of the users should always be taken into consideration. It is advantageous to consider all coding schemes employed by the intended users of a new code.
Preference should always be given to the adoption of agreed international codes, such as those laid down in ISO standards, UN/ECE recommendations and directories and by international organizations.

9.1.3 Code significance

When properly used, significant codes provide a basis for additional information and tend to be easier and more reliable for human use than non-significant codes. However, caution must be exercised in the development of significant codes to assure that significant parts are

connected to stable entities. For example, a significant code for an organization should not be associated with the location of the organization when a change in location would result in a change in the code. Excessively significant codes can become unmanageable and lack expendability, and should thus be avoided.
For most identification purposes and for all referencing purposes the use of non-significant codes is recommended. See chapter 3 for the advantages and disadvantages of the various coding methods.

9.1.4 Multiple code set compatibility

In some instances more than one code or representation is necessary to meet system requirements. A single code is the ideal objective, but is not always the most practicable solution. Multiple codes, if needed should be translatable from one code to another, i.e. the items remain unchanged, only the codes are different.

9.1.5 Mnemonic codes

Mnemonic codes may be used to aid association and memorization, thus increasing human processing efficiency, provided they are not used for identification of very long, unstable lists of items. Mnemonic structures must be carefully chosen, however, to insure that flexibility is not sacrificed. Mnemonics should generally be avoided if the potential code set is very large, because the effectiveness of the mnemonic feature decreases as the number of items to be coded increases. Where mnemonic or otherwise meaningful coded representations cannot be provided for all elements of the set in the system, preference should be given to the elements having the highest use frequency.

9.1.6 Code naming

All independent code segments must be individually named with standard, unique, consistently applied labels.

9.1.7 Calculation of code capacity

When calculating the capacity of a given code for covering all situations while maintaining code uniqueness, the formula given in 7.12 applies.

9.2 Number of characters

9.2.1 Conciseness

Code values should consist of a minimum number of characters to conserve space and reduce data communication time, but at the same time be optimized in terms of the code users capabilities.

9.2.2 Fixed number of characters

A code of a fixed number of characters (e.g. always three characters, not one, two, or three) is more reliable and easier to use than a variable length code.

9.2.3 Segmentation

Code values having more than four alphabetic or five numeric characters should be divided into smaller segments for purposes of reliable recording, e.g. XXX-XXX-XXXX is more reliable than XXXXXXXXXX.

9.2.4 Potential expansion

The code structure should provide for adding new elements to the set without having to recode existing elements or extending the format of the coded representation.

9.3 Format of the coded representation

9.3.1 User considerations

Code components and segments should be formatted according to user needs for information, considering greatest ease of scanning for accuracy and completeness, and compactness of the data content.

9.3.2 Alphabetical versus numeric

Human recording of numeric code values is generally more reliable than that of alphabetic (all letters) or alphanumeric code values where no mnemonic characteristics exist. Controlled alphanumeric code values (i.e. where certain positions are always alphabetic or numeric) are more reliable than random alphanumeric code values. For example, AA999 (where the first two characters are always letters and the last three are always numbers) is a more reliable format than when letters or numbers can appear in any position.

9.3.3 Character grouping

In cases where the code is structured with both alpha and numeric characters, similar character types should be grouped and not dispersed throughout the coded representation. For example, fewer errors occur in a three character code where the structure is alpha-alpha-numeric (e.g. HW5) than in the sequence alpha-numeric-alpha (e.g. H5W).

9.3.4 Code position sequence

If a code divides an entire entity set into smaller groupings, the high-order positions should be broad, general categories; low-order positions should be the most selective and discriminating (including any suffixes). An example is the numeric representation of the date according to
ISO 8601 (YYMMDD). If a composite code is designed consisting of two or more independent codes, the individual code segment occupying the higher-order position shall be based on usage requirements and processing efficiency considerations.

9.3.5 Separation of code segments

Code segments should be separated by a hyphen (when displayed) or exist in complete separation (when stored and displayed) if the positions or segments are completely independent of each other and can stand alone (i.e. no other code is required for their meaning).

9.3.6 Check characters

Consideration should be given to the addition of an error-detecting character to avoid errors in recording. Employment of a self checking code prevents many unnecessary problems of posting data to the wrong record and providing misinformation.

9.4 Character sets

9.4.1 Special characters

Familiar characters should be used, and characters other than letters or numbers (such as the hyphen, period, space, asterisk, etc.) should be avoided in code structures (except for separating code segments, where a hyphen or space may be used). Single case letters only, i.e. ABC...Z or abc...z, are to be used in codes. Names and abbreviations may use both upper and lower case letters and other characters. The vocabulary for a given code system should contain the fewest possible character classes. Wherever possible, the character set used for data standards should conform to the ISO 7 Bit Coded Character Set for Information Processing Interchange (see ISO 646).

9.4.2 Visual similarities

When it is necessary to use an alphanumeric random code structure, characters that are easily perceived as, or confused with, other characters should be avoided. Some examples are: letter I vs. number 1, letter O vs. number zero, letter Z vs. number 2; letter G vs. number 6; letters B and S vs. number 8; and letters O and Q.

9.4.3 Acoustical similarities

Non-significant codes should avoid characters that can be confused when pronounced (acoustically homogeneous); for example the letters B, D, G, P, and T or the letters M and N.

9.4.4 Vowels

Avoid the use of vowels (A, E, I, O and U) in alpha codes or portions of codes having three or more consecutive alpha characters to preclude inadvertent formation of recognizable plain language words.

9.4.5 Collating considerations

Any specific character position should be either alphabetic or numeric in order to avoid collating sequence incompatibility.

9.5 Assignment conventions