GESMES/TS

(Generic statistical message/time series)

The time series data exchange message

User Guide

(Release 3.00, 28 Feb 2003)

TABLE OF CONTENTS

TABLES

BOXES

1. INTRODUCTION

This is the seventh in a series of releases of GESMES/CB based on GESMES (an acronym for Generic Statistical Message)—a system for the electronic exchange of data and metadata. Whereas the previous user guides outlined releases of GESMES/CB (for use mainly by the central banking community), this guide describes GESMES/TS (time series), the latest release 3.0, for use by a larger statistical community. It will explain the GESMES/TS data model, define the syntax of this EDIFACT version, and guide the user to develop the necessary applications to use GESMES/TS.

You do not need to be familiar with the previous releases of GESMES/CB to use this guide. We'll point out in this introduction how beginners, or users already familiar with GESMES/TS, may use it and how the guide is organised. However, for those of you who would prefer a more general knowledge of GESMES, we recommend that you refer to the original documentation (*GESMES Version 2.1: Guidance to Users* and *GESMES Version 2.1: Reference Guide)*, issued on Sep. 14, 1997.[1]

"You" in this guide refers to an agency that collects and/or sends data—a center or a partner—whereas the term "we" refers to the authors of this guide—the Bank for International Settlements (BIS), the European Central Bank (ECB), Eurostat (the Statistical Office of the European Communities), and the International Monetary Fund (IMF).

HOW IS THIS GUIDE ORGANISED?

After the introduction (section 1), information will appear as follows:

- section 2—definition of GESMES/TS, generic GESMES, and EDIFACT;
- section 3—GESMES/TS data model (the common language and framework to describe statistical data exchanges);
- section 4—tasks you can perform using GESMES/TS;
- section 5—what a GESMES/TS message is and how to build one;
- section 6—the guide-within-a-guide (a *segment-by-segment reference guide*, showing all alternative variants for each segment that can appear in a message);
- section 7—delete messages;
- section 8—message acknowledgements (not supported);
- section 9—text in uncoded attributes: rules and considerations; and
- section 10—complete examples of messages to help you with practical implementation of a message.

[1] available on http://www.gesmes.org/

-

IF I'M ALREADY FAMILIAR WITH GESMES/TS, HOW DO I USE THIS GUIDE?

If you are familiar with GESMES/TS, you will want to refer only to section 6—the technical reference guide-within-a-guide. You may also want to scan the quick summary at the end of this introduction about the differences between this release 3.0 and the previous Release 2.0.

IF I'M A BEGINNER IN GESMES, HOW DO I USE THIS GUIDE?

Beginners will find it most helpful to read the guide in the following sequence:

1) sections 2, 3 and 4;

2) then Appendix I—it's important now to review EDIFACT syntax ("electronic data interchange for administration, commerce, and transport");

3) section 5;

4) section 6.1, including Box 1, which helps you refer to information provided in the rest of section 6—the reference guide;

5) *skip sections 7–9 for now;*

6) section 10—this gives you complete examples of messages, which you can understand fully, segment-by-segment, if you refer simultaneously to the segment descriptions in section 6; and

7) finally, the remaining sections.

WHAT DO I NEED TO KNOW, IN A NUTSHELL, TO USE GESMES/TS?

You must be familiar with three topics:

- **GESMES/TS data model** (explained in section 3).
- **Statistical-structure definitions** (used in the data model and explained throughout the guide). These definitions are the basis of GESMES/TS. They consist of three items—key family definitions, statistical concepts, and code lists of values.
- **Some basic EDIFACT knowledge** (explained in Appendix I). Although you do not need to have previous EDIFACT background or experience to fully understand the EDIFACT version of GESMES/TS, the knowledge of some basic EDIFACT principles is a prerequisite.

WHY IS IT IMPORTANT TO KNOW STATISTICAL STRUCTURE DEFINITIONS, INCLUDING CODE LISTS?

The central agencies already using GESMES/TS (a central agency is an institution that administers exchanges of statistical information with its partners) circulate **statistical structure definitions**. For

example, the ECB circulates its structural definitions to its partners—the European System of Central Banks (ESCB). Similarly, the BIS disseminates to its partners the structural definitions for its data exchanges. When an institution desires to create new structural definitions, we strongly recommend that it refer to the existing definition files of central institutions--to avoid creating new concepts or code lists (see Box 1, section 6.1 of this guide).

An agency's use of structural definitions is eased by its use of common **code lists**. The ECB, BIS, and the other user communities of GESMES/TS—such as Eurostat—have agreed to use common codes. For example, central administrators use common codes for the statistical concepts "unit," "unit multiplier," "observation status" (a piece of information "explaining" in a coded format, the status of a single observation), and "observation confidentiality." Similarly, most central agencies specify the identifier for the "frequency" as "FREQ." The aim is for GESMES/TS "messages" to look as similar as possible—not only syntactically but also semantically.

HOW MUCH KNOWLEDGE OF EDIFACT DO I NEED?

In general, if you wish to process a GESMES message directly from an application (rather than use a commercial EDIFACT translator), you should give particular attention to the EDIFACT truncation rules, which are built into and implied in the "fixed" rules of GESMES/TS.

If you used the previous GESMES and EDIFACT guides, you will have knowledge of the syntax tools used to "describe" the contents of a message based on the GESMES/TS data model. However, if you are new to GESMES/TS, you should refer to Appendix I in this guide.

An XML version of GESMES/TS, to be used in parallel to the EDIFACT version, is being developed. This will enable data-exchange partners to take advantage of XML-based tools and technologies (including XML-EDIFACT converters) if they wish.

Brief Glossary of terms

time series: a time-ordered vector of observations.

statistical concept: a statistical characteristic of a time series or an observation.

dimension: a statistical concept used to refer to and identify a time series (e.g., a concept indicating a certain economic activity or a concept identifying a geographical reference area).

attribute: a statistical concept providing qualitative information about a time series or an observation (e.g., concepts such as units, magnitude, currency of denomination, titles, and methodological comments could be used as attributes in the context of an agreed data exchange).

code list: a predefined list from which some statistical concepts (coded concepts) take their values.

key family: in order to exchange statistical information, a central institution has to agree with its partners which statistical concepts are necessary for identifying these particular series (concepts used as dimensions) and which statistical concepts to be used as attributes. These definitions form the key family structure definition (or simply "key family").

structural definitions: the concepts, key families, and code lists defined by a center institution (usually for the exchange of statistical information with its partners). Such an institution acts as a "structural definition maintenance agency."

metadata: any kind of statistical information beyond pure observations (e.g., attributes, structures, code lists).

statistical message: a predefined and agreed way of syntactically representing sets of statistical data, attributes, and structural metadata which need to be exchanged between partners.

EDIFACT: Electronic Data Interchange For Administration, Commerce, and Transport.

GESMES: a United Nations standard (EDIFACT message), allowing partner institutions to exchange statistical multidimensional arrays of data in a generic but standardized way.

XML: eXtensible Markup Language.

GESMES/TS RELEASES

GESMES/CB–Release 1.0–March 6, 1997
GESMES/CB–Release 1.1–April 30, 1997
GESMES/CB–Release 1.2–May 31, 1997
GESMES/CB–Release 1.3–October 30, 1997
GESMES/CB–Release 1.4–February 8, 1999
GESMES/CB–Release 2.0–July 30, 2000
GESMES/TS–Release 3.0–February 28, 2003

WHAT'S NEW IN THIS GESMES/TS RELEASE 3.0 VIS-À-VIS THE PREVIOUS RELEASE 2.0?

Although these will be explained in detail ahead, briefly the new features in this release are:

- A new name for the message—GESMES/TS—and a more complete description of the "frequency" dimension. These two changes affect the syntax of (existing) segments as follows:

  What's the new name? The name has been changed from GESMES/CB to GESMES/TS. The name reflects the adoption of the message by the larger statistical community, including the IMF and the OECD. Accordingly, you will need to adjust the string providing the message identification in the "*application-reference*" element (UNB segment).

  What's the more complete identification of the "frequency" dimension"? To unambiguously identify the frequency dimension, we have enriched the parameters used in the SCD segment with one additional value. This change affects the "reading" or "writing" structural definition messages in the applications. We have also introduced a rule, "frequency" has to be the first dimension.

- An increase of the maximum expected length for code values from 12 to 18 characters.
- The greater importance of the data set identifier in the data model.

  You will now be able, for example, to exchange versions of time series. To enable you to do this, we have replaced in several places in section 3 (Data Model) the reference to "key family" with the concept "data set." We have now oriented the key family concept only to the *structure* of the exchanged data (and attributes). And we have oriented a data set as a *time-series container*. Users can now identify times series by a data set identifier and the series key (whereas before they were key family identifier and a series key).

  As you could before, you can still attach attributes at the levels of observation, time series, and sibling group (which is a group of time series). But now you can also attach attributes at the data set level (before: key family). You can still perform deletions at the levels of observation, time series, and sibling groups. But now you can also delete at data set level (before: key family).

  This change clearly distinguishes the structure of the data (key family) from the exchanged data and attributes (actual data and attributes of a data set). Now it is possible for you to exchange one or more variations (vintages) of data and attribute values as *distinct* data sets, using a common key family definition.

  Although the data model acquires more genericity to respond to corresponding new requirements, you will not have to change the syntax of the message. Of course, if you desire to exchange data set

versions using this new role of GESMES/TS, you may need to adjust your internal reception or reporting systems.

As long as you do not use the additional role, your internal applications will continue to be fully compatible. For instance, you might specify only that there is one data set to be exchanged in a given key family. Or (less restrictive) you might exclude in advance the possibility of exchanging more than one version of a time series in a key family. Then your data model holds, and your processing of messages may continue as you were instructed in the previous guide release.

An additional feature, introduced here, allows you to inform other institutions about the list of "data sets" that are relevant in your data exchanges. This is explained in Appendix IV. Moreover, Appendix V discusses how important is for institutions administering data exchanges to provide as complete as possible documentation to their partners about the data sets they exchange, illustrating this requirement with examples of duplicates in data exchange.

- An adjustment to some of the textual descriptions in sections 6.6.8 (attribute attachment level), 6.7.2 (data set), 6.9.4 (relation within the data set for attributes) and 7 (deletions). The adjustment results from the upgraded role of the data set in the data model (some descriptions were consistent already in Release 2.0).
- A new paragraph in the "frequently asked questions" section. This paragraph reviews the meanings of "mandatory" and "conditional" in different contexts.
- The correction of several typos and inaccuracies.
  For instance, now the examples show the "frequency" in the position that is usually specified (first position in the key structure). And we have corrected examples that used non-permissible characters in code values or identifiers. We dropped some duplicate or redundant terms; for instance, we replaced "mnemonic" everywhere by "identifier."
- Other corrections in more detail are:

  1) *sections 6.4.2, 6.4.4, 6.5.2, and 6.6.2*: We present the format specifications more precisely.

  2) *section 6.4.5*: We now show the expression "free-text" in the last paragraph five times instead of the previous erroneous six; and

  3) *section 6.6.4*: We have simplified the text referring to the three categories of dimensions and array elements.

2. WHAT IS GESMES/TS? HOW DID IT EVOLVE FROM THE GENERIC GESMES AND THE EDIFACT STANDARDS?

2.1 Definition

GESMES/TS is a time-series data exchange "message" designed to allow you to exchange statistical information with other organisations, independently of technical platforms. A message is the predefined and agreed syntactical way for partners to exchange sets of statistical data, attributes, and structural metadata.

This data-exchange message implements a **time-series data exchange model** (GESMES/TS data model), which allows you to exchange and identify time series through a multidimensional key and various associated metadata. It uses a generic GESMES profile and, for the version described in this guide, the EDIFACT syntax.

Whereas the full GESMES message is a generic statistical data model—with flexible syntax to describe virtually any statistical data model—GESMES/TS has a fixed syntax. This allows you and your partner institutions to design and build applications you need to "read" and "write" GESMES/TS messages, while avoiding intermediate files and special translators.

Owing to this fixed syntax, in most cases you will find the rules in GESMES/TS to be stronger and more restrictive than those in generic GESMES. However, we could in the future enhance and further generalize the TS message, if this is needed.

GESMES/TS offers several features:

- you can easily adapt GESMES/TS to any economic domain; it flexibly covers all types of economic and statistical data;
- its techniques of representation are modern: it has conceptually 'clean' multidimensional keys;
- it's efficient: you do not have to unnecessarily repeat information;
- the series keys are not restricted in length;
- you can easily introduce new attributes;
- the message now has various levels of attributes (observation, time series, and 'higher');
- you can exchange "rich" metadata, organised flexibly and efficiently;
- the message is suitable for a two-way exchange of time series, that is, reporting and disseminating;

- in *paperless* fashion, you as a center can disseminate a whole statistical database: you can electronically disseminate data, metadata, definitions, key structures, and code lists from your center to other institutions;
- you can implement the message stage-by-stage: you could start with the essential parts (carrying the administrative and the numeric data) and later use the whole message;
- the message is easy to implement; you should not have to purchase special software;
- the message is consistent with international standards (EDIFACT);
- a wider range of groups can use it: national central banks (NCBs) and national statistical institutes (NSIs), BIS, IMF, OECD, Eurostat and ECB; and
- it's consistent with the long-term goals of international institutions that exchange statistical data.

2.2. How did generic GESMES and the EDIFACT standards originate?

The first messages of electronic data exchange were developed for a commercial environment, which needed to exchange rather simple and static object types like invoices and purchase orders. In time, new sectors joined the industry, requiring more complicated and varied data structures for statistics. And it became evident that more generic message types would need to be created to prevent an explosion in the number of messages.

Working within the international UN/EDIFACT standards body, a group of European statistical organisations developed GESMES--Generic Statistical Message. They gave GESMES all the features required to exchange multidimensional arrays of data, including time series and metadata (such as attributes and footnotes). And they standardized it, giving it a structure that could be understood and processed by software applications without human intervention.

GESMES holds the advantage of being—in preference to a proprietary data format—an internationally agreed standard that is both open and fully functional. It is not tied to the constraints and formats of one particular application. It allows you—a data-sending or collecting institution—to use the data set structure you prefer. In particular it supports the exchange of:

- metadata;
- multidimensional arrays;
- time series; and
- administrative data.

In 1995, the UN accepted GESMES as UN/EDIFACT Status 1 messages and first published it in the UN/D95A directory. Eurostat, which has led the development of statistical UN/EDIFACT messages, is implementing GESMES into the data flows with the reporting competent national authorities. It is also promoting the use of the messages by other international organisations and other sectors.

A description of the EDIFACT syntax is provided in Appendix I. For those of you unfamiliar with the EDIFACT terminology and syntax, **it is strongly recommended that you familiarise yourselves with this Appendix** (especially when reading the sections following section 4).

Before we explain in sections 4 and beyond how to use GESMES/TS, we'll describe in section 3 the data model—summarizing several terms and helpful pointers. This is intended to be only an overview. More detail will follow.

3. THE GESMES/TS DATA MODEL

3.1. Concepts, definitions, properties and rules

This section provides a common language and a common conceptual framework to describe statistical data exchanges.

1. A **period** is a *time reference*.

2. An **observation** is the value, at a particular period, of a particular variable (sometimes called the economic phenomenon).

3. To be useful, an observation must have more information relating to it than just a value and an associated period. Information about observations is called **metadata**.

4. The characteristics of observations that make up the metadata are known as **statistical concepts** (e.g. reporting country). Each statistical concept is either *coded* or *uncoded*.

   - A **coded statistical concept** takes values from a **code list** of valid values. For example, a coded statistical concept called 'reporting country' might be created, taking its values from the list of ISO list of country codes. A code list may supply the values of more than one statistical concept.
   - An **uncoded statistical concept** takes its values as free form text (e.g. time series title).

5. A **time series** is a time-ordered vector of observations.

6. If a time series has a constant time interval between its observations, this interval determines the **frequency** of the time series.

7. **Data exchange context** is the framework in which two or more partners agree to:

   - exchange one or more identified sets of data and related attributes ("**exchanged time series**"; **ETS**).
   - use one or more key families to serve this requirement;
   - possibly, on some business and implementation agreements.

8. **Structural definitions maintenance agency**: An institution that devises key families.

9. The ETS is a collection of **data sets**.

10. Each data set takes its structure from exactly one **key family**.

11. Each data set is uniquely identified within an ETS by a **data set identifier**

12. Each key family **links** exactly one code list to each coded statistical concept used in that key family.[1]

13. Each key family is uniquely identified by a structural definitions maintenance agency using a unique **key family identifier**.

14. Each key family has a **key structure**, namely an ordered set of coded statistical concepts whose combination of values uniquely *identifies* each time series within a data set.

    - The coded statistical concepts assigned as members of a key family's key structure are called the **dimensions** of the key family.
    - No key family is permitted to assign a particular coded statistical concept as a dimension more than once.
    - **Frequency** must be assigned as a dimension in every key family and it has to be the first dimension.[2]
    - Every time series takes a value for every dimension of the key family to which the series belongs.
    - The meaning attached to the value of one dimension is not permitted to depend upon the values of any other dimensions.
    - The list of values uniquely identifying a time series within a data set is called the **key** of the time series.
    - Within the ETS a time series is uniquely identified by a data set identifier combined with the time series key.
    - Within a data set an observation is identified by a time series key combined with a time period.
    - Only coded statistical concepts are permitted to be dimensions of a key structure.

15. Within a data set, a set of time series whose keys differ only in the value taken by the frequency dimension is called a **sibling group**. Within an ETS a sibling group is uniquely identified by a data set identifier combined with the sibling group key.

16. In addition to the dimensions, each key family assigns a set of statistical concepts that *qualify* the observations within the key family. The members of this set of statistical concepts are called the **attributes** of the key family.

---

[1] Consequence: Code lists are valid across all data sets that are based on the same key family. It is not possible, for example, to change the description of a code value between data sets following the same key family.

[2] See also paragraph on "Positioning the frequency dimension" in Section 3.4.

- No key family is permitted to assign a particular statistical concept as an attribute more than once.

- No statistical concept is permitted to be assigned as both an attribute and a dimension of the same key family.

- Each key family has a property for each of its attributes that determines whether:
  - the attribute takes an independent value for each *observation* in the data set
  - the attribute takes an independent value for each *time series* in the data set
  - the attribute takes an independent value for each *sibling group* in the data set
  - the attribute takes a single value for the entire *data set.*

  This property uniquely identifies the **attachment level** of the attribute for the key family.

- Within a given key family, each attribute is considered either *mandatory* or *conditional.*
  - a **mandatory attribute** is an attribute which must take a value, otherwise the corresponding observation(s), which it refers to, is (are) not considered meaningful enough (e.g. the "status" of an observation or the units to which a whole time series is expressed).
  - a **conditional attribute** is permitted to take empty values within the key family.

17. Each key family has the following properties:

- **Identifier:** It provides a unique identification within the set of key families specified by a structural definitions maintenance agency.

- **Name:** It is also unique.

- **Description:**# Description of the purpose and domain covered by the key family.

18. Each data set has the following properties:

- **Identifier**: It provides a unique identification within an ETS.

- **Description:** Description of the purpose and domain covered by the data set (supported in the data set lists exchange message).

- **key family**: Key family describing the structure of the data set

Therefore, in a given data exchange context, the partners need to agree and establish the **list of data set identifiers** (data set list) that is expected to be relevant in their data exchanges.

19. Each statistical concept has the following properties:

- **Identifier**: It provides a unique identification within the set of statistical concepts specified by a structural definitions maintenance agency.
- **Name:** It is also unique.
- **Description:**# Description of the meaning and purpose of the statistical concept.
- **Representation:** [*coded* | *uncoded*]

20. In addition to the properties listed just above, each uncoded statistical concept has the following properties:

    - **text length:** The maximum number of characters in the text values of the concept

21. Each code list has the following properties:

    - **Identifier:** It provides a unique identification within the set of code lists specified by a structural definitions maintenance agency.
    - **Name:** It is also unique.
    - **Description:**# Description of the purpose of the code list.
    - **Code value length:** Either an exact or a maximum number of characters and a type (i.e. numeric or alphanumeric) must be specified.

22. Each code in a code list has the following properties:

    - **Value:** Unique identifier within the code list.
    - **Description:** It uniquely describes the code value.

The "description" is not supported in the EDIFACT implementation of the data model.

3.2. Data types

- Identifiers are:
  - Maximum 18 characters;
  - Any of A..Z (upper case alphabetic), 0..9 (numeric), _ (underbar);
  - The first character is alphabetic.
- Names are:
  - Maximum 70 characters.
  - From ISO 8859-1 character set;[1]
- Descriptions are:
  - Maximum 350 characters;
  - From ISO8859-1 character set.
- Code values are:
  - Maximum 18 characters;
  - Any of `A..Z` (upper case alphabetic), `0..9` (numeric), _ (underscore), `/` (solidus slash), `=` (equal sign), `-` (hyphen);

    However, code values providing values to a dimension must use only the following characters:
    A..Z (upper case alphabetic), 0..9 (numeric), _ (underscore)
- Observation values are:
  - Decimal numeric values (signed only if they are negative);
  - The maximum number of significant figures is:
    ⇒ 15 for a positive number;
    ⇒ 14 for a positive decimal or a negative integer;
    ⇒ 13 for a negative decimal.
  - Scientific notation may be used.
- Uncoded statistical concept text values are:
  - Maximum 1050 characters;
  - From ISO8859-1 character set.
- Time series keys:

  In principle, the maximum permissible length of time series keys used in a data exchange does not need to be restricted. However, for working purposes, an effort is made to limit the maximum length to 35 characters; in this length, also one (separator) position is included between all successive dimension values;[2] this means that the maximum length allowed for a pure series key (concatenation of dimension values) can be less than 35 characters.

[1] It includes accented characters.

[2] e.g. the series key `f:xxx:yy:zz:rr:ss:k:jj:d:zz:ll:hh:e` uses the maximum permissible length (35 characters).

3.3. Guidelines

3.3.1. Central institutions devise structural definitions for statistical data exchanges

"Central" institutions are the organisations to which partner institutions "report" statistics. The central institutions use these statistics either to compile aggregates and/or to assemble and make them available uniformly (e.g., on-line or on a CD-ROM or through file transfers). Therefore, central institutions both receive data from other institutions and also, usually, "disseminate" data to individuals and/or institutions for end-use.

What are examples of central institutions? Eurostat (the statistical office of the European Communities), as a central institution, collects data from national statistical institutes (NSIs) and other sources. And it disseminates data to NSIs, other partners, and to the public. Within a country, an NSI or a national central bank (NCB) plays, of course, a role of central institution, because it collects data from other entities and disseminates statistical information to end users.

Why is the role of central institution important? In GESMES/TS the role of central institution is important, because it is the particular agency that usually has devised the underlying structural definitions on which every statistical message is based. These definitions may be statistical concepts, code lists, and key families. Such an institution plays the role of the reference "structural definitions maintenance agency" for the corresponding messages that are exchanged.

Is the presence of central institutions always necessary for a data exchange? The descriptions above assume data exchange arrangements in the context of a "hub model"; however, GESMES/TS can be used by any institutions (e.g. between two partners) who can devise and use a commonly agreed set of structural definitions (see also next paragraph); so, a central hub does not need to be always present.

What about structural information devised by a third institution? Two institutions could exchange data using/referring to structural information devised by a third institution. For example, a central bank could report balance of payments data to Eurostat/BOP, using structural information that is technically administered by the ECB or the IMF. (In this case, the ECB or the IMF would be the "structural definitions maintenance agency" for this message.) In other words, central institutions can play a double role in collecting and further disseminating statistics and in devising structural definitions for use in data exchanges. In the same way, a central bank and statistical institute of a country can exchange between them balance of payments statistical data using a new key family they could agree on or using a key family devised by the IMF or Eurostat/BoP.

3.3.2. As a central institution, how do I devise structural definitions?

We suggest that you use the following guidelines when you build structural definitions. And we advise that you consider these guidelines particularly when devising new key family definitions. (Again, this is a brief summary. We'll go into more detail in sections 4 and beyond.)

- ***Avoid dimensions that are not appropriate for all the time series in the key family.*** Again, a dimension is a statistical concept used to identify a time series. If some dimensions are not appropriate for some series, then consider moving these series to a new key family in which you drop these dimensions from the key structure.[1]

- ***Avoid composite dimensions.*** Each dimension should correspond to a single characteristic of the data, not to a combination of characteristics.

- ***Avoid creating a new code list where one already exists.*** We highly recommend that you make structural definitions and code lists consistent with internationally agreed standard methodologies, wherever they exist (e.g., the *System of National Accounts* 1993; *Balance of Payments Manual*, fifth edition; *Monetary and Financial Statistics Manual*; *Government Finance Statistics Manual*, etc.). When you set up a new data exchange, we suggest the following order of priority when you consider code lists:
  - international code lists (e.g. ISO, UN, IMF level);
  - international code lists supplemented by other international (e.g. BIS, OECD) and/or European institutions (e.g. Eurostat, ECB);
  - standardized lists used already by international institutions (e.g. UN, IMF, BIS, OECD, Eurostat);
  - new code lists agreed between two international or European institutions (e.g. ECB, BIS, Eurostat),
  - new specific code lists (e.g. within the Eurostat, the BIS or the ESCB data exchange circuits).

  You can use the same code list for several statistical concepts, within a key family or across key families.

- ***Keep the following in mind when you define a new key family.*** As a structural definitions maintenance agency, you must specify the following items when defining a new key family:

  1. Key family identification:
     - key family identifier

[1] If it is decided not to create a separate key family then, for the set of time series for which the dimension is not relevant, a value such as "non-applicable", "non-defined" "all" or "total" has to be assigned to this dimension.

- key family name

2. A list of coded statistical concepts assigned as dimensions of the key family. For each dimension, you need a:
   - statistical concept identifier
   - statistical concept name
   - ordinal number of the dimension in the key structure
   - code list identifier

   The "frequency" has to be the first dimension.

3. A list of statistical concepts assigned as attributes for the key family. For each attribute, you need a:
   - statistical concept identifier
   - statistical concept name
   - code list identifier if the concept is coded
   - assignment status: mandatory or conditional
   - attachment level
   - maximum text length for the uncoded concepts
   - maximum code length for the coded concepts

4. A list of the code lists used in the key family. For each list, you need a:
   - code list identifier
   - code list name
   - code values and descriptions

- ***Agree on the following when defining data sets.*** Two (or more) partners performing data exchanges in a certain context need to agree on:
  - the list of data set identifiers they will be using;
  - for each data set:
    - its content and description
    - the relevant key family definition

- ***Define the mandatory attributes.*** Once you have decided the key structure of a key family, then you must define the set of mandatory attributes of this key family. In general, some statistical concepts are necessary across all key families to qualify the contained information. These are:
  - Reference area
  - Frequency (always a dimension)
  - Descriptive title (see also comment below)
  - Collection (e.g., end of period, averaged, or summed over period)
  - Unit (e.g., currency of denomination)
  - Unit multiplier (e.g., expressed in millions)
  - Availability (which institutions can a series become available to)
  - Decimals (i.e., number of decimal digits used in a time series)
  - Observation status (e.g., estimate, provisional, *normal*)

  Therefore, those concepts that are not dimensions within a key family have to be present in that key family as mandatory attributes. Moreover, you may consider additional attributes as mandatory when

you define a specific key family. In key families, where by definition the title or *title complement* is identical to the concatenation of the names relating to the code values of the dimensions of the key, you may omit these two attributes from the key family definition.

3.3.3. As a central institution, how do I set and maintain attribute values?

- *Static properties*
  - Upon creating a series, you, as the sender, must provide to the receiver the values for all mandatory attributes. If values for conditional attributes are also available, you should also provide them. Whereas initially you may provide this information by means other than GESMES/TS messages (e.g., paper, telephone), we expect that, over time, partner institutions will be in a position to provide this information in GESMES/TS format.
  - As the center, you may agree with your data-exchange partners on special procedures for authorising the setting of attributes' initial values.
  - The center that administers the exchanged data set will exclusively set and maintain attribute values at data set level.
- *Communication of changes to the center*
  - Following the creation of a series, you, as the sender, do not have to report the attribute values again, as long as they do not change.
  - Whenever attribute values for a series (or sibling group) change, you (the reporting institution) should report either all attribute values again (this is the recommended option) or only the attribute values that have changed. This applies both to the mandatory and the conditional attributes. For example, if a previously reported value for a conditional attribute is no longer valid, you must report this to the center.
  - A center may agree with its data exchange partners on special procedures for authorising modifications in the attribute values.

What are the fixed observation-level attributes—"observation status," "observation confidentiality," "observation pre-break"?

- The observation-level attribute "observation status" is part of the fixed syntax of the ARR segment, which contains the arrayed data (the key, the numeric values, and their corresponding status flags used for reporting observations). The observation status is a piece of information "explaining," in a code, the status of a single observation (e.g., estimate, provisional, *normal*). Whenever you exchange an observation, you must also exchange the corresponding observation status attached to the observation, regardless of whether it has changed or not since the previous data exchange.
- If the "observation status" changes and the observation remains unchanged, you should report both components.
- Some key families contain definitions of the observation-level attributes "observation confidentiality" and "observation pre-break" as well. Observation confidentiality provides information about the confidentiality of an observation, and observation pre-break is a numeric observation or a missing value indicator. When key families contain these attributes, the following rule applies: if you receive an observation with an observation-status attribute only

attached, this means that the associated observation confidentiality and pre-break observation attributes either never existed or from now on, do not have a value for this observation.[1]

3.4. Best practices

When devising key families and administering the exchange of data sets

Well designed key families facilitate the data exchange and allow to efficiently respond to additional requirements in future. As an institution devising key families and administering the exchange of data sets, you will need to go through several considerations:

When new **key families** need to be defined a special care should be given in choosing the most appropriate dimensions not only for the moment when the data exchange requirement is expressed, but possibly also foreseeing extensions of this requirement in future. This would help to minimise cases in which a key family could become incapable after some time to serve the exchange of a more extended data set as compared to the one initially agreed. As an example, **the exchange of data versions** could be performed using a dimension (allowing different code values for this dimension to reflect the different "versions") or using the "data set" mechanism. The partners should evaluate in this case the alternative options and choose the most appropriate. In general, a good key family design is a key element for easily responding to additional requirements in future.

When you decide with your partners **the exchange of a new data set**, you should provide them with all necessary information. This will probably include your agreements on:

- the telecommunication means to be used;
- the key family to be used;
- the list of series to be included in the data set to be exchanged;
- a concrete plan about the implementation (including possibly a plan for testing);
- supplementary elements such as clarifications whether the new data set is a "version" of another data set already exchanged and what this would imply in practical terms; also, clarifications may be needed on

[1] However, this logic does not apply to the observation comment attribute. If it is not received in an interchange and if it had previously existed, that previously received value should be still kept in receiver's database (the "updates and revisions" principle applies).

whether the exchanged series and attributes through two (or more) data sets involve the exchange of duplicate information and in which context. You can find more details on this point in Appendix V.

The new data set should be incorporated in the list of data sets exchanged between two institutions. This list is normally useful in configuring the reception and data extraction systems and it can be maintained using GESMES/TS (more details on this can be found in Appendix IV) or other means (e.g. document, text files).

As a center administering data exchanges you should avoid to rename data sets once they are used in production, as this would imply a need for changes in your partners' systems or difficulties in administering time series which had been exchanged in the past using other data set names.

The definition of new key families or the introduction of new data sets in data exchange should always allow sufficient time for the implementation by partner institutions.

In general, the change management of key families and data sets should be always done very cautiously taking into account the potential implications for partners.

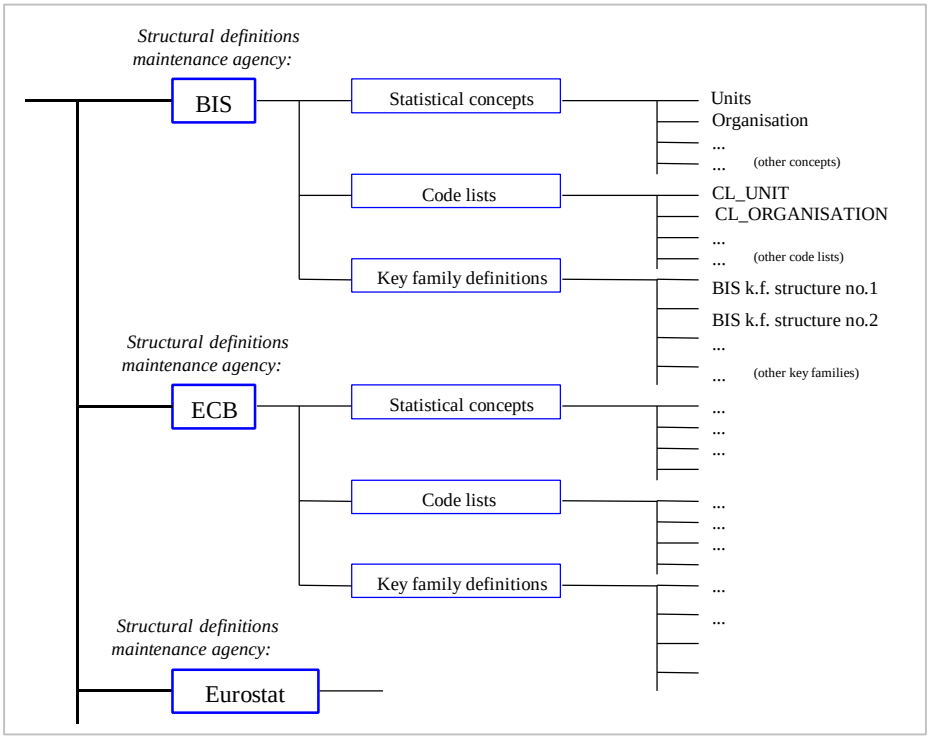
When more than one central institution is involved in a data exchange

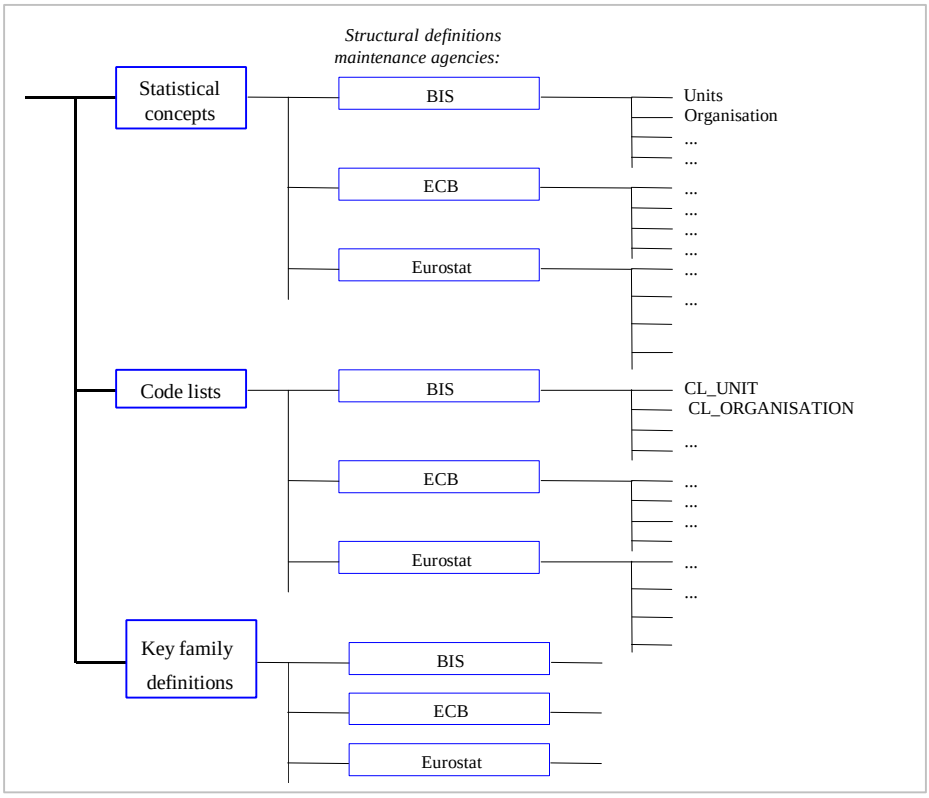
In the paragraphs above discussing the role of central institutions, we mentioned that a central institution usually administers the exchange of data sets based on the structural definitions it devises. However, in other cases, a data exchange could use a third institution's structural definitions. In this case, if you are a central institution administering this data set(s), you should take care (possibly in cooperation with the corresponding structural definitions maintenance agency) that you make all the involved data-exchange partners aware of the necessary structural definitions. In addition, you should properly maintain, and if necessary, appropriately update the corresponding GESMES/TS structural definition messages.

In principle, each partner institution should design its system in such a way that it can cope with an environment in which more than one structural definition maintenance agency can exist. That is, GESMES/TS permits partner institutions to design generic data-exchange systems that can take into account the role of their central institution in devising structural definitions.

For example, the following graphs describe alternative ways to organise structural definitions, assuming the existence of three central institutions (e.g., BIS, ECB, Eurostat). In practice, you could envisage more central institutions and, therefore, more central branches in the tree, including possibly even the home

institution (e.g., a central bank or statistical institute), if the home institution plays a role in "devising" structural definitions within a user community.





Positioning the "frequency" dimension

The key family definition unambiguously identifies the position of the “frequency” dimension. Moreover, if you are an institution devising key families, you have to assign the first position in the key structure to this dimension. However, if you develop new applications and a new system, it is recommended not to rely on this position of the frequency and to use instead the identification of the frequency as foreseen in the ASI group of segments.

Easily identifying the agency that defined the key family

In order to immediately recognize the structural definition maintenance agency that defined a key family, most central institutions that devise structural definitions use the first characters of the key family identifiers to identify their institution (e.g., BIS_MACRO, EUROSTAT_BOP_01, ECB_BOP1, etc.).

Easily identifying the institution that administers the data set

In order to immediately recognize the institution administering a data set, many central institutions prefer to use the first characters of the data set identifiers to identify their institution (e.g., BIS_MACRO, ECB_BOP1, ECB_BOP1T, etc.). This helps also to avoid defining the same data identifier in two different data exchange contexts (which may anyway overlap with respect some partners) to denote two different data sets.

What are some “frequency”-related special issues?

Special frequencies. The issue of data collected at special (regular or irregular) intervals at a lower-than-daily frequency (e.g., 24 or 36 or 48 observations per year, on irregular days during the year) is not extensively discussed here, but the data model and the message are compatible with such data exchange requirements.

Tick data. The issue of data collected at irregular intervals at a higher-than-daily frequency (e.g., tick-by-tick data) is not extensively discussed here either. However, for data exchange purposes, such series can already be exchanged in the GESMES/TS format by using the option to send observations with the associated time stamp.

4. USING GESMES/TS, WHAT TASKS CAN I PERFORM?

You can GESMES/TS to build a 'two-way' statistical data exchange. For instance, such an exchange could take place between a national central bank (NCB) or a national statistical institute (NSI) and a central institution (e.g., BIS, IMF, Eurostat, ECB, OECD) or between any two institutions, as long as you have made known to each other the semantics of the exchanged information (structural definitions, that is, statistical concepts, code lists, key families). Of course, an NCB or an NSI could act as a "center" vis-à-vis other institutions by devising appropriate structural definitions for the data exchanges it is going to administer.

We'll discuss the tasks before showing you in section 5 how to build them into GESMES/TS messages. Again, a message is the predefined and agreed syntactical way for partners to exchange sets of statistical data, attributes, and structural metadata.

Fundamentally, the tasks performed via GESMES/TS messages are 1) update data, 2) update data and attributes, 3) delete data, 4) exchange structural definitions, and 5) exchange data set lists.
As a data-exchange partner, you can gradually build up to using all the GESMES/TS message functions (depending on the task you wish to perform). You'll start with only the update-data message and build up, in the longer term, to exchanging structural definitions and data set lists.

Here, briefly, are the tasks that you can perform:

1. ***Update data:*** You can exchange ***only*** observations (including the *mandatory* attribute "observation status").
2. ***Update data and attributes:*** In addition to exchanging observations and observation status, you can also exchange coded and uncoded *attributes*, linked to the data on a data set, sibling-group, time-series, and observation level.
3. ***Delete data:***
    - send delete messages of observations and observation status;
    - in addition to observations and observation status, send delete messages for coded and uncoded attributes, linked to the data on a data set, sibling-group, time-series, and observation level.

and:

4. ***Exchange structural definitions:*** Exchange code lists, lists of statistical concepts, and key family definitions.
5. ***Exchange data set lists.***

4.2. What do I need to know about being a reporting institution or a central institution before I begin performing these tasks?

a) As an institution reporting to a central institution

You may use the ***data-update message*** to report ***only*** **data (no attributes) to a central institution** (there is a minimum essential requirement).

Over time, you will develop the capability to also provide coded and uncoded attributes relating to the series you report, that is, you eventually will also use the ***data and attribute update message***.

When you use the *data-update* message only, remember this caveat: If you erroneously report observations **and** attributes, you will also have to send a ***data (and attribute) delete message*** (the delete messages are in fact very similar to the messages used for updating).

b) As a central institution performing regular dissemination

As a central institution, you may disseminate selected output or a *complete database*[1] (e.g., on a CD-ROM) or regularly provide *net updates and revisions*:

b.1. A center institution sending complete databases

For a complete database, you would send in GESMES/TS format:

- ***data and attribute update message(s)*** with the observation values and the values of the coded and uncoded attributes;
- ***a structural-definition message***, which gives information about the statistical concepts, code lists, and key families used in the data message(s); and
- ***possibly, the list of data sets used.***

You cannot include delete messages, because they provide a complete replacement of the previous copy of the database.

b.2. A center institution sending updates and revisions

When you as a center disseminate updates and revisions, you can enable receiving institutions to receive only the information on observations and attributes that has changed in the database since a specific date. (You can disseminate new series [or sibling groups] that you have added to the database. You can also send deletions at the sibling-group, series, and observation level.) In order to take full advantage of this option, the receiving institutions have, of course, to be able to interpret *data and attribute update* and *delete messages*.

[1] A complete database: the whole ETS (including data, attributes, structural definitions) or a consistent subset of it.

The use of the different messages[1] of GESMES/TS is summarised in the following table. The table also indicates the 'activity' (i.e., *write* or *read*) that you, as a partner institution or a center, perform in the message.

Table 1. Activities and message types

Activity	Message type	Partner	Center
'Simple' reporting to a center	• Data update message	W	R
	• Data delete message	W	R
'Advanced' reporting to a center, including information on coded and uncoded attributes	• Data/attribute update message	W	R
	• Data/attribute delete message	W	R
Full database dissemination from a center	• Structural definitions message	R	W
	• Data/attribute delete message	R	W
Update/revision output from a center, ad hoc or scheduled	• Data/attribute update message	R	W
	• Data/attribute delete message	R	W

5. TO BEGIN USING GESMES/TS, HOW DO I BUILD MESSAGES? HOW ARE THEY STRUCTURED?

This section will give you a brief, but global, overview of messages (which consist of segments and sections) of a GESMES/TS data *interchange* (file). This overview will start to familiarize you with pre-defined segments. Later in section 6, we will provide more detail about segments.

A GESMES/TS interchange (file) may contain several *messages*.

Each one of those messages comprises a sequence of segments, reflecting in each one a small logical section of the message. Each segment starts with a unique 3-character (alphabetic upper-case) identifier (segment tag) and ends with a segment terminator. Segment identifiers that begin with UN are called *service segments* and are defined as a part of the EDIFACT syntax [ISO 9735]. (Other segments are called *user-data segments*, and you will find them defined in the Segments Directory of the UN Trade Data Interchange Directory [UNTDID].)[2]

A collection of segments is a "group," and a collection of segments and groups compose a "section."

[1] Actually, these are sub-messages of GESMES/TS; each one of them reflects a selection of appropriate sections of GESMES/TS in order to serve the desired activities and to support the corresponding functionalities. In this table the exchange of lists of data sets is not considered.

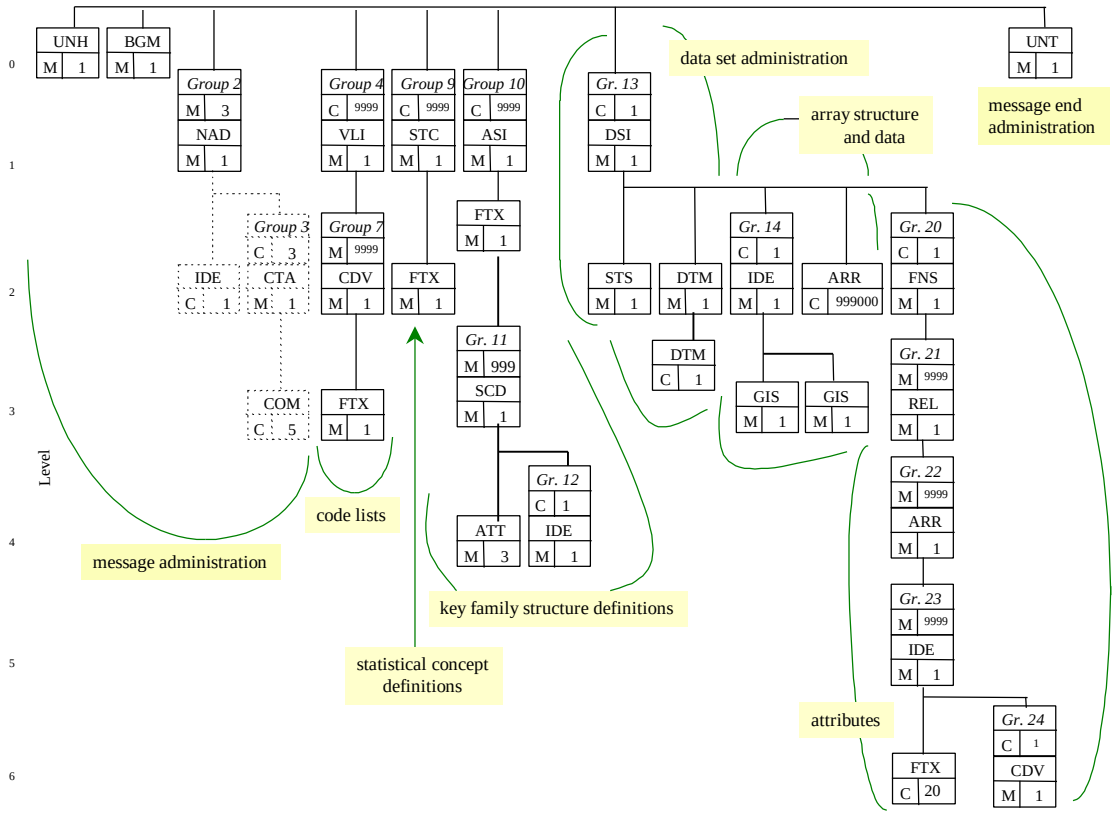
[2] http://www.unece.org/trade/untdid/d99b/trmd/gesmes_c.htm

5.1. WHAT IS A BRANCHING DIAGRAM?

The most common way to present the contents of a GESMES message is to use its *branching diagram*. A branching diagram shows the *segments* that you include in the message (via their segment tags), their *status* (mandatory or conditional), and the maximum number of *occurrences* that are allowed.

Because the branching diagram describes a message and not the entire interchange file, it does not show an interchange header and trailer. (An interchange header and trailer identify the interchange, the sender, receiver, date and time of preparation, and other optional data. More on that later.)

Below is the branching diagram of a GESMES/TS message, including all its segments grouped in sections. After the diagram, we'll show you what it means. Appendix I also elaborates on how to interpret a typical branching diagram.



Note: The "conditional" status of the first segment of each group (e.g., VLI in Group 4, DSI in Group 13, FNS in Group 20) would turn into "mandatory" if you had to use the corresponding facility.

5.2. What are the first steps in building the possible tasks into message sections?

In the list below, we group the segments of the previous diagram into categories called **sections**. Each one of them performs a logical task. (In this example, we include an interchange header and trailer—numbered lines 1 and 10. In the numbered lines 2-9, we present the logical sections of the main body of a message.) Do not be concerned for the moment what UNA, UNB, etc. mean. We'll get to that later, or you can look now at Appendix I for the definitions.

1. Interchange administration (UNA and UNB segments)
2. Message administration (UNH, BGM, NAD+Z02, NAD+MR, NAD+MS segments)
3. Code lists (VLI, CDV, FTX segments)
4. Statistical concept definition (STC, FTX segments)
5. Key family definition (ASI, FTX, SCD, ATT, IDE segments)
6. Data set administration (DSI, STS, DTM segments)
7. Array structure (IDE, GIS, GIS segments)
8. Data (ARR segment)
9. Attributes (FNS, REL, ARR, IDE, FTX, CDV segments)
10. End-of-message administration (UNT segment)

<at this point, just after the "end of message administration", more *messages [repetitions of the sequence of items 2-9]* can be present in the same interchange>

11. End of interchange administration (UNZ segment)

The table below shows which of the sections above *are necessary* (and their order) *in a message,* according to the task that you desire to perform with a message.[1]

Table 2. Data and Attribute Update Messages

Type of message/data exchange:	***ABSOLUTELY ESSENTIAL*** **Data update message**	**Attribute update message**	**Data and attribute update message**
Section	*(ordered) sections which must be present*		
Interchange administration segments	①	①	①
Message administration segments	②	②	②
Code lists			
Statistical concepts			
Key families			
Data set administration segments	③	③	③
Array structure segments	④	④	④
Data segment(s)	⑤		⑤
Attributes		⑤	⑥
End of message administration	⑥	⑥	⑦
End of interchange administration	⑦	⑦	

[1] In this table the exchange of lists of data sets is not considered.

Example 1—Data Update message

As you see in the table above, when you use a **Data Update message**, you ***must*** have the sections (1) interchange administration, (2) message administration segments, (3) data set administration segments, (4) array structure, (5) data segment(s), (6) end of message administration, and (7) end of interchange administration.

Example 2—Data and Attribute Update message

As you see in the table above, when you use a **Data and Attribute Update message** (contains attributes), you ***must*** also include the *attributes group* (just before the *end of message administration*).

The table below shows what sections are necessary (and their order) in a message, when your purpose is to indicate to the receiving institution that it should delete objects:

Table 3. Data and Attribute Delete Messages

Type of message/data exchange:	Data delete	Attribute delete	Data and Attribute delete message
Section	*(ordered) sections that must be present*		
Interchange administration segments	①	①	①
Message administration segments	②	②	②
Code lists			
Statistical concepts			
Key families			
Data set administration segments	③	③	③
Array structure	④	④	④
Data segment(s)	⑤		⑤
Attributes		⑤	⑥
End of message administration	⑥	⑥	⑦
End of interchange administration	⑦	⑦	

The list of segments indicated in this table is in principle the same as the previous one. But when you delete attributes, you do not use all segments included in the *attributes* group (for further information see the *Delete Messages* section ahead in this guide).

The table below shows which of the sections above are necessary (and their order) in a message, if you are a center that wants to disseminate structural messages:

Table 4. Structural messages

Type of message/data exchange:	Code lists	statistical concept definition	key family definition
Section	*(ordered) groups of segments which must be present*		
Interchange administration segments	①	①	①
Message administration segments	②	②	②
Code lists	③		
Statistical concepts		③	
Key families			③
Data set administration segments			
Array structure			
Data segment(s)			
Attributes			
End of message administration	④	④	④
End of interchange administration	⑤	⑤	⑤

Important observation:

You can observe in the tables above, that, in fact, to build the message to perform the task you want, you do not have to determine the type of message you will need. Instead, the presence (or not) of the specific sections (and the setting of some parameters) will exclusively determine what type of message it is (e.g., data update or attribute update or data and attribute update or data delete, etc.).

In the next subsections of this section 5, we will further illustrate each type of message (e.g., data and attribute update) and present how each segment-within-a-group behaves.

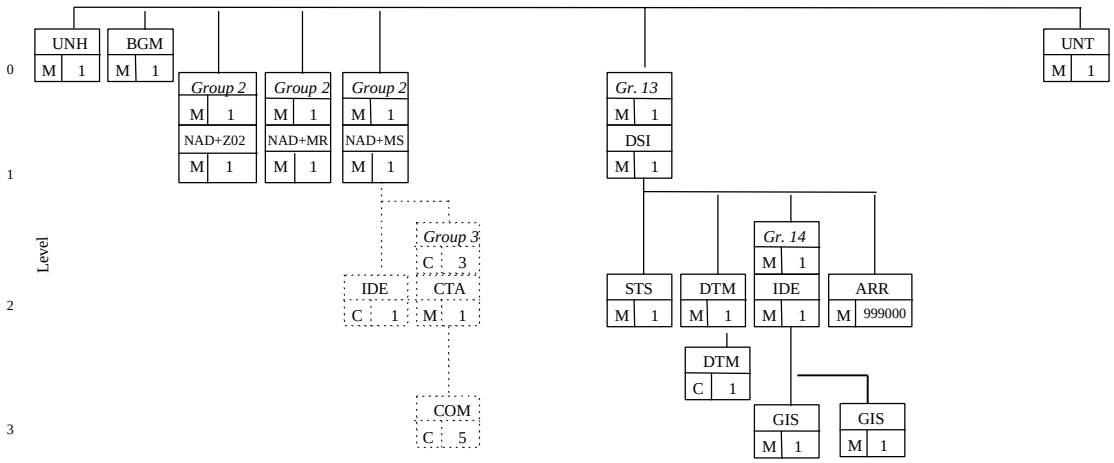
Because the branching diagrams presented in the rest of this section describe specific *message types*, they reflect subsets of the full message. Moreover, this section emphasizes how to support desired activities and to serve specific roles. Therefore, the indicated segment status (mandatory/conditional) might be different here for some segments (as opposed to the full message) to indicate what you really need per activity and desired tasks.

When you peruse the following tables of this section, you might find it convenient also to refer (for each segment) to the analytical presentation of the *segments* and *sections* of *messages* of the segment-by-segment reference guide in the next section, section 6. In this way, the details per segment presented in the segment-by-segment reference guide will enlighten the very brief (but global) overview given in this section. This would be the most efficient way to study the whole guide.

5.3. The basic GESMES/TS message — the Data-Update Message

When your goal is to exchange *observations* (and their associated *array cell attributes*) **only**, then the complete branching diagram collapses to the following simpler one. This diagram describes the absolutely essential parts of the message that you and every institution should be able to manage (write and read) when you first begin implementing GESMES/TS messages.

This diagram presents all segments (apart from the ones in dotted lines) here as mandatory because, indeed, they are all necessary in order to serve this particular use. In other words, you will not use the segments belonging to the structural sections (code lists, concepts and key family definitions) and to the attribute section.



Note: We present the three NAD segments (NAD+Z02, NAD+MR, NAD+MS) here separately for reasons of clarity (more on this later).

The following table briefly explains the segments that we used (and needed) above in a simple message. This is to begin to ease you into viewing GESMES/TS structure. Detail will be provided in section 6.

Table 5. Data update message

Sections and segments of the message	Purpose
interchange administration	
UNA*service-characters*	defines the service characters
UNB+*syntax-identifier*:*syntax-version*+*sender-identification*+ *receiver-identification*+*date-of-preparation*:*time-of-preparation* +*interchange-reference*++*application-reference(++++test indicator)* '	interchange header
message administration	
UNH+*message-reference-number*+*message-type*:*message-type-version*:*message-type-release*:*controlling-agency* '	message identification
BGM+*message-function* '	message function (=74 for data/attr.)
NAD+*Z02*+*organisation-id* '	code list maintenance agency
NAD+*MR*+*organisation-id* '	receiver identification
NAD+*MS*+*organisation-id* '	sender identification
IDE+*10*+*message-identity* '	message identity (conditional)
data set administration	
DSI+*data set-identifier* '	data set identifier
STS+3+*status-code* '	update/replace (*status-code*=7)
DTM+*date-time-type:date-time*:*date-time-format* '	data set preparation date and time
DTM+*date-time-type:date-time*:*date-time-format* '	(and -conditional- reporting period)
array structure and data	
IDE+*5*+*identifier* '	key family identifier
GIS+*AR3* '	method used to send data in the ARR segment
GIS+1:::*symbol-used-for-missing-values* '	symbol used for missing values
ARR++*key:period:date-format:observation:obs-status:obs-confid.* '	contains the numeric data; two methods can be used:
ARR++*key:period:date-format:observation: obs-status:obs-confid.* '	- a single observation per ARR segment...
...	
etc.	
or:	
ARR++*key:start_date-end_date:time_range_format:observation: obs-status:obs-confid.+ observation: obs-status:obs-confid.+... +observation: obs-status:obs-confid.* '	- or a time range with the corresponding observations;
ARR++*key:start_date-end_date:time_range_format:observation:obs-status:obs-confid.+ observation:obs-status:obs-confid.+... +observation: obs-status:obs-confid.* '	the ARR segment can appear up to 999000 times.
...	
etc.	
end of message administration (message trailer)	
UNT+*number-of-segments*+*message-reference-number* '	end of message control data
end of interchange administration (interchange trailer)	
UNZ+*number-of-messages*+*interchange-reference* '	end of interchange control data

Example:

Let's assume the National Bank of Belgium (NBB) is sending the time series M:BE:PROD:GN:NS and Q:BE:PROD:GN:NS (belonging to a key family called ECB_TESTPRICES) to the ECB.

For the monthly time series M:BE:PROD:GN:NS, the NBB reports the following observations (together with their "status"):

Sep95	Oct95	Nov95	Dec95	Jan96	Feb96	Mar96
99.10 A	98.10 A	98.40 A	99.50 A	100.00 A	99.20 A	99.80 E C

The flags A (="normal") and E (="estimate") are values for the Observation Status attribute, which is attached next to each observation. The flag C (=confidential) is a value for the (conditional) Observation Confidentiality attribute, which you can attach next to the observation status to provide information about the confidentiality of an observation. For more details about the use of these code lists, please refer to Box 1 (in section 6.1) and to Appendix II, presenting the corresponding code lists.

For the quarterly series Q:BE:PROD:GN:NS, the NBB wants to send the following data:

95q4	96q1
98.67 A	99.67 A

Using GESMES/TS, the NBB must send these data in the following file:

BE2=National Bank of Belgium, 4F0 = ECB *(example codes for organisations involved)*

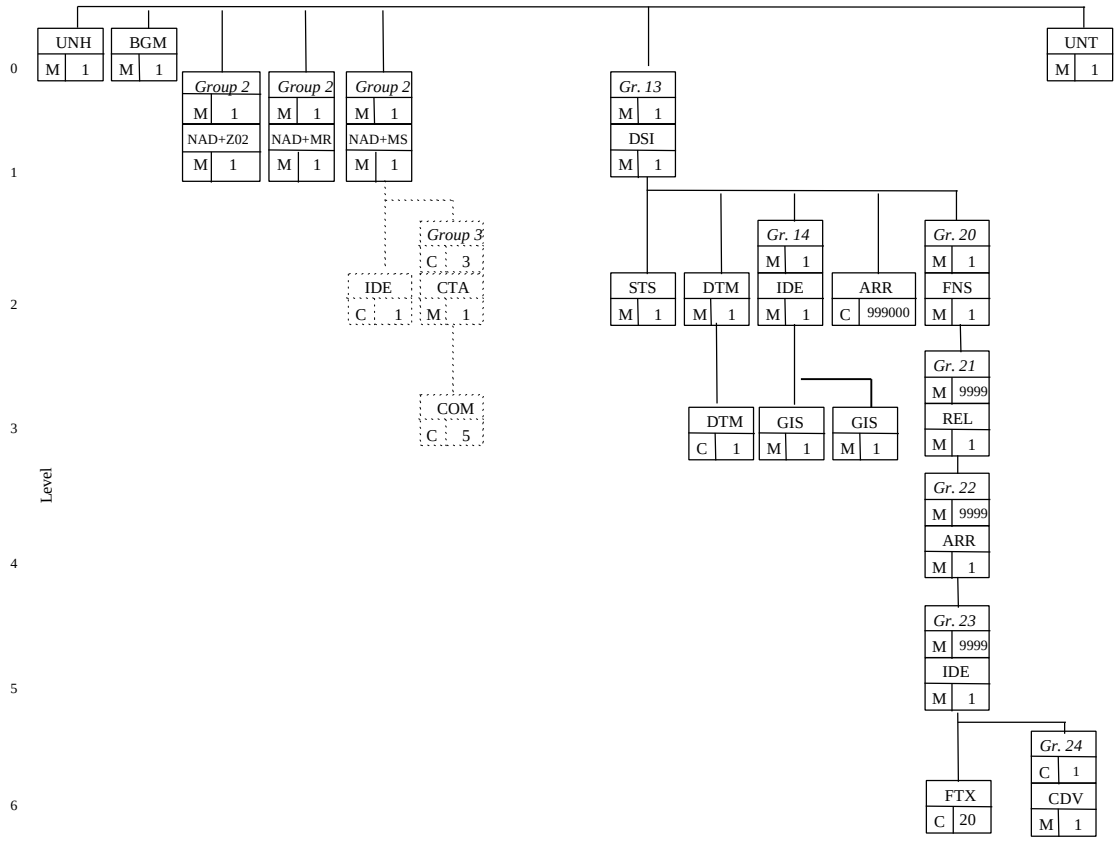
```
UNA:+.? '
UNB+UNOC:3+BE2+4F0+970525:1539+IREF000001++GESMES/TS'
UNH+MREF000001+GESMES:2:1:E6'
BGM+74'
NAD+Z02+ECB'
NAD+MR+4F0'
NAD+MS+BE2'
DSI+ECB_TESTPRICES'
STS+3+7'
DTM+242:199705251539:203'
IDE+5+ECB_TESTPRICES'
GIS+AR3'
GIS+1:::-'
ARR++M:BE:PROD:GN:NS:199509199603:710:99.10:A+98.10:A+98.40:A+99.50:A+100.00:A+
99.20:A+99.80:E:C'
ARR++Q:BE:PROD:GN:NS:1995419961:708:98.67:A+99.67:A'
UNT+14+MREF000001'
UNZ+1+IREF000001'
```

This example underlines the *non-fixed* elements, and you will see that, apart from the actual data, they provide mainly administrative information (e.g., BE2=central bank of Belgium, 4F0=ECB, ECB_TESTPRICES=key family identifier). For more explanations about the use of these segments, refer to section 6 of this document.

5.4. The Data and Attribute Update Message

When you want to report data, you can send—apart from observations—additional information that helps the users of the exchanged time series better understand their nature or their special characteristics (title, methodology, etc.). These pieces of information are called *attributes*, and you can send them either coded or uncoded, according to their type. You exchange them using the FNS segment and the segments that follow it. FNS stands for "footnote set" and is used to begin an attribute section.

The branching diagram presented in the previous paragraphs is now enlarged, including also the set of segments that carry the attributes:



Notes:

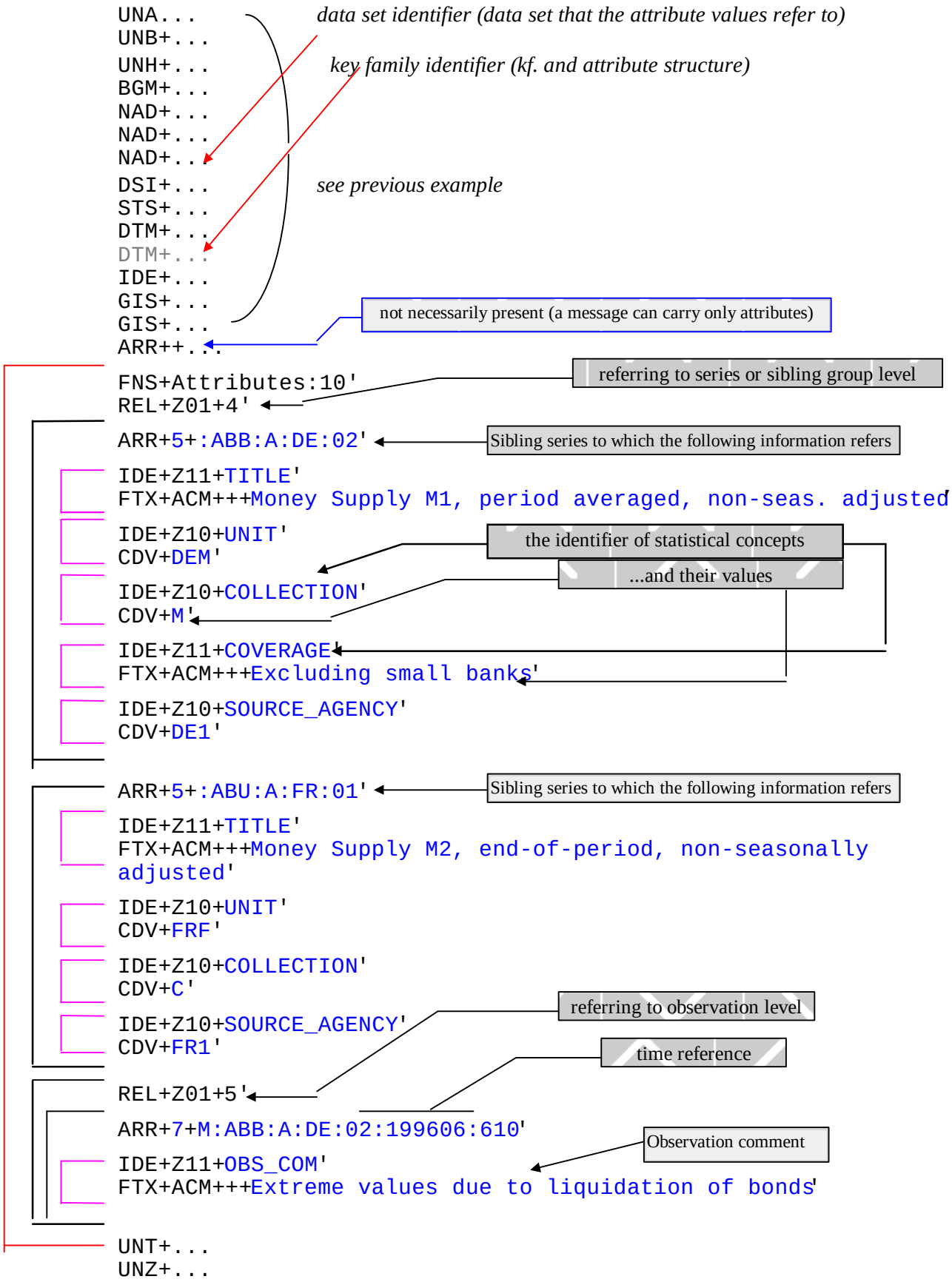
- We'll start to talk about "groups," which are collections of sections. Here, you will see that for any "data and attribute update" message, Group 13 is mandatory. Without it, it is impossible to exchange observations and/or attributes.
- For this type of message, Group 14 is also mandatory. For reporting observations (with their associated observation flags), it provides information about the structure in the ARR segment following this group. Again, an ARR segment is the array of data.
- If you need to exchange both observations and attributes, then, apart from the essential Groups 13 and 14, the ARR segment that follows Group 14 and Group 20 must be present.
- If you need to use the message only to report attributes, then Groups 13, 14 and 20 must be present. (The ARR segment that follows immediately after Group 14 is the only segment that you do not need in this case.)

The following table highlights the use of segments when you report both data and attributes.

Table 6. Data and attribute update message

Sections and segments of the message		Purpose
UNA UNB+... UNH+... BGM+... NAD+... NAD+... NAD+... DSI+... STS+... DTM+... DTM+... IDE+... GIS+... GIS+... ARR++...	*update/replace status*	*as described in the previous table (see DATA UPDATE MESSAGE)*
Attributes		
FNS+*general-attribute-identifier*:*identity-number-type* '		[segment introducing attributes]
REL+*Z01*+*array-scope* '		attribute scope
ARR+*last-dimension-position*+*key-dimension*:*key-dimension*:*key-dimension-key*:*dimension etc.* '		dimension/key pointer
IDE+*Z10*+*coded-attribute-identifier* ' CDV+*coded-attribute-value* '		attribute identifier (Z10=coded attr.) attribute code value
IDE+*Z10*+*coded-attribute-identifier* ' CDV+*coded-attribute-value* ' etc.		attribute identifier attribute code value
IDE+*Z11*+*uncoded-attribute-identifier* ' FTX+ACM+++*text* '		attribute identifier (Z11=uncoded) attribute text
IDE+*Z11*+*uncoded-attribute-identifier* ' FTX+ACM+++*text* ' etc.		attribute identifier attribute text
ARR+*last-dimension-position*+*key-dimension*:*key-dimension*:*key-dimension-key*:*dimension etc.* '		a new dimension/key pointer can follow
etc.		new sets of segments 'IDE/CDV' and IDE/FTX can follow
UNT+... UNZ+...		*as described in the previous table*

Example:


UNA...
UNB+...
UNH+...
BGM+...
NAD+...
NAD+...
NAD+...
DSI+...
STS+...
DTM+...
DTM+...
IDE+...
GIS+...
GIS+...
ARR++...
data set identifier (data set that the attribute values refer to)
key family identifier (kf. and attribute structure)
see previous example
not necessarily present (a message can carry only attributes)
FNS+Attributes:10'
REL+Z01+4'
referring to series or sibling group level
ARR+5+:ABB:A:DE:02'
Sibling series to which the following information refers
IDE+Z11+TITLE'
FTX+ACM+++Money Supply M1, period averaged, non-seas. adjusted'
IDE+Z10+UNIT'
CDV+DEM'
IDE+Z10+COLLECTION'
CDV+M'
the identifier of statistical concepts
...and their values
IDE+Z11+COVERAGE'
FTX+ACM+++Excluding small banks'
IDE+Z10+SOURCE_AGENCY'
CDV+DE1'
ARR+5+:ABU:A:FR:01'
Sibling series to which the following information refers
IDE+Z11+TITLE'
FTX+ACM+++Money Supply M2, end-of-period, non-seasonally adjusted'
IDE+Z10+UNIT'
CDV+FRF'
IDE+Z10+COLLECTION'
CDV+C'
IDE+Z10+SOURCE_AGENCY'
CDV+FR1'
referring to observation level
time reference
REL+Z01+5'
ARR+7+M:ABB:A:DE:02:199606:610'
IDE+Z11+OBS_COM'
FTX+ACM+++Extreme values due to liquidation of bonds'
Observation comment
UNT+...
UNZ+...

For further explanations about the use of the attribute-related segments, you can refer to the segment-by-segment reference guide section of this guide.

5.5. The Data and Attribute DELETE Message

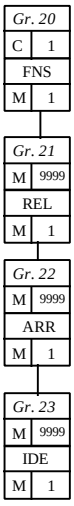
For the delete message, you also use the structure presented in the previous paragraph to delete objects that you have already made available to the receiving institution in the past. However, in addition, you must flag appropriately the message using a different parameter in the STS (or "status") segment.

- **Deletion of data**:
  use the ARR segment references that follow Group 14 simply to point to (existing, previously reported, via an interchange in the past) observations that should be deleted. (The ARR should not contain observations.)

  Example 1:
  The segments
  ```
  ...
  STS+3+6'
  ...
  ARR++M:BE:PROD:GN:NS:199509199603:710'
  ...
  ```
  inform the receiving institution that it should delete the monthly observations of the series `M:BE:PROD:GN:NS` from Sep95 to Mar96.

  Example 2:
  The segments
  ```
  ...
  STS+3+6'
  ...
  ARR++:BE:PROD:GN:NS'
  ...
  ```
  inform the receiving institution that it should delete the sibling group (assuming that the first dimension is the frequency) `:BE:PROD:GN:NS` and all associated attributes at this and at lower levels.

- **Deletion of attributes**:
  do not use FTX ("free text") and CDV ("code value definition") after Group 23 (see diagram on the right) and use the ARR segment (Group 22) exclusively to point to the attributes that have to be deleted.

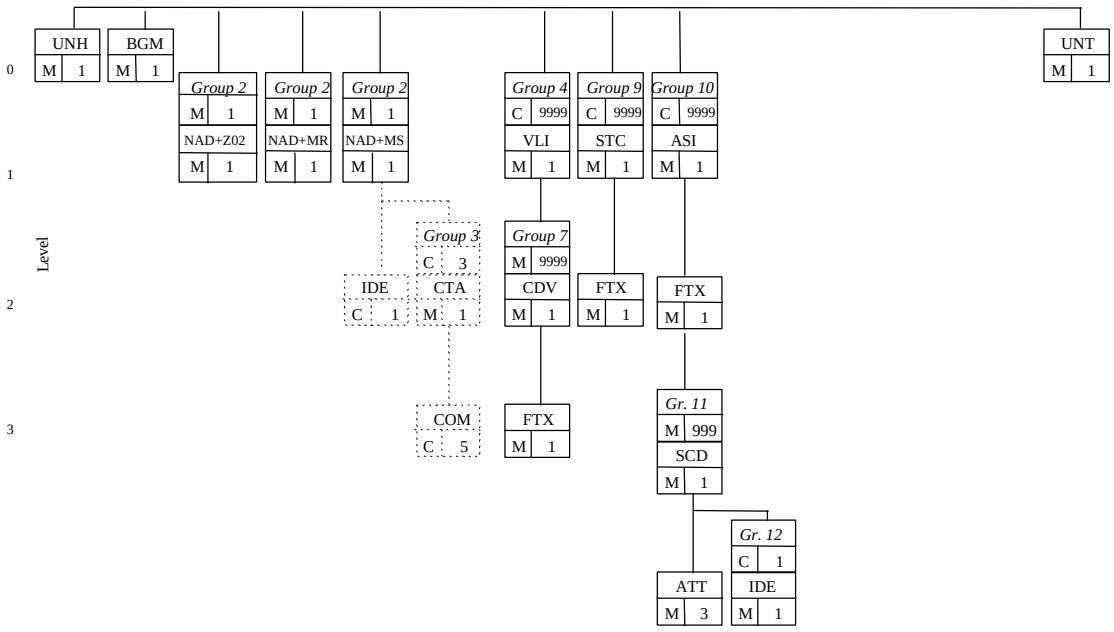
  

  Example:
  The segments
  ```
  ...
  STS+3+6'
  ...
  FNS+Attributes:10'
  REL+Z01+4'
  ARR+5+:ABB:A:DE:02'
  IDE+Z11+COVERAGE'
  ```
  inform the receiving institution that it should delete the attribute referring to the *Coverage* of the sibling group `:ABB:A:DE:02`.

*A **more extensive and detailed discussion on issues related to deletions** is presented as a separate section (see page 153).*

5.6. The Structural Definition Message

The structural definition message is produced by a center institution to communicate to its partner(s) in electronic form information about key families it maintains. Thus, if you are a center that disseminates all structural data (concepts, key family definitions, and code lists) using GESMES/TS, your statistical data exchange system can become completely paperless. In such a structural message you would include only structural (and administrative) segments, as in the message below:



A structural message can contain either one of the three sections (i.e., code lists, statistical concept definitions, or key families) or two of them or all three. Normally, you would use all three sections (in one or more messages) to disseminate to your partners in an electronic form all structural definitions they need.

5.6.1. What segments do I use in this type of message to send code lists?

The segments for code lists are VLI (for value list identification), CDV (for code value identification) , and FTX (for free text). They let you disseminate code lists for coded statistical concepts, and you use them either as dimensions in a specific key family or as coded attributes.

5.6.2. What segments do I use in this message to send statistical-concept definitions?

The segments are STC (for statistical concept) and FTX. You use them to provide the link between a statistical concept identifier and its actual name.

5.6.3. What segments do I use in this message to send key family definitions?

The segments for the structural definition of a specific key family serve several purposes. You use them to:

- assign a key family identifier and a description to a key family;
- define the statistical concepts used as dimensions in the key structure (information which is provided: position of each concept in the key, code value length, and relevant code list);
- define the exact structure of the ARR segment (which contains the numeric values of the message) for this particular key family; and
- define the coded and uncoded attributes used in the key family (concept identifier, field length, usage status, and attachment level; also, relevant code list in the case of coded attributes).

You can use a special GESMES/TS message to exchange the list of data sets (data set identifiers, description, and key family from which they take their structure) used in a specific data exchange context. This is presented in Appendix IV: Exchanging Lists of Data Sets.

Table 7. Structural data exchange message

Sections and segments of the message	Purpose
UNA UNB+... UNH+... BGM+... NAD+... NAD+... NAD+...	BGM +73' *should be written now (no data/attr., only definitions)* *as before*
code lists	
VLI+*code-list-id*+++*code-list-name* '	Identifier of the code list
CDV+*code-value* ' FTX+ACM+++*code-value-description* '	code value text description
CDV+*code-value* ' FTX+ACM+++*code-value-description* ' *etc.*	code value text description
VLI+*code-list-id*+++*code-list-name* '	Identifier of next code list
CDV+*code-value* ' FTX+ACM+++*code-value-description* '	code value text description
CDV+*code-value* ' FTX+ACM+++*code-value-description* ' *etc.*	code value text description
(etc. ...for other code lists)	
statistical concept definition	
STC+*concept-identifier* ' FTX+ACM+++*concept-name* '	statistical concept identifier concept name
STC+*concept-identifier* ' FTX+ACM+++*concept-name* '	statistical concept identifier concept name
etc.	
key family definition	
ASI+*key-family-identifier* ' FTX+ACM+++*key-family-name* '	key family identifier key family name
(i) key structure (dimensions)	
SCD+*n*+*concept-identifier*++++:*dimension-position-in-the-key* ' ATT+3+5+:::AN*field-length* ' IDE+1+*code-list-id* '	*n*=13 if dimension is FREQ, *n* =4 otherwise corresponding code list
SCD+4+*concept-identifier*++++:*dimension-position-in-the-key* ' ATT+3+5+:::AN*field-length* ' IDE+1+*code-list-id* '	 corresponding code list
etc. (for all key family dimensions)	

(ii) time, time format, observation value, coded attributes at obs. level	*(other ARR elements:)*
SCD+1+TIME_PERIOD++++:*position-in-the-ARR-structure*' ATT+3+5+:::AN35' SCD+1+TIME_FORMAT++++:*position-in-the-ARR-structure*' ATT+3+5+:::AN3' SCD+3+OBS_VALUE++++:*position-in-the-ARR-structure*' ATT+3+5+:::AN15' SCD+3+OBS_STATUS++++:*position-in-the-ARR-structure*' ATT+3+5+:::AN1' ATT+3+35+2:USS' ATT+3+32+5:ALV' IDE+1+CL_OBS_STATUS'	*time period* follows series key in the ARR structure then, *time format* follows then, the *observation* follows the *observation status* follows and: - it is one character long - it is mandatory (=2) - attached at the obs. level (=5) - it takes values from the CL_OBS_STATUS code list
(iii) key family attributes *[a. for coded attributes]* **SCD+Z09+*concept-identifier*'** ATT+3+5+:::AN*field-length*' (or: "AN..*max-field-length*" for variable length) ATT+3+35+*usage-status-code*:USS' ATT+3+32+*attachment-level-code*:ALV' IDE+1+*code-list-id*' **SCD+Z09+*concept-identifier*'** ATT+3+5+:::AN*field-length*' (or: "AN..*max-field-length*" for variable length) ATT+3+35+*usage-status-code*:USS' ATT+3+32+*attachment-level-code*:ALV' IDE+1+*code-list-id*' *etc. (for all coded attributes of the key family)*	**statistical concept identifier** (max) length of the attribute value mandatory or conditional status attachment level corresponding code list **statistical concept identifier** (max) length of the attribute value mandatory or conditional status attachment level corresponding code list
[b. for uncoded attributes] SCD+Z09+*concept-identifier*' ATT+3+5+:::AN..*max-field-length*' ATT+3+35+*usage-status-code*:USS' ATT+3+32+*attachment-level-code*:ALV' SCD+Z09+*concept-identifier*' ATT+3+5+:::AN..*max-field-length*' ATT+3+35+*usage-status-code*:USS' ATT+3+32+*attachment-level-code*:ALV' *etc. (for all uncoded attributes of the key family)*	statistical concept identifier max. length of the attribute value mandatory or conditional status attachment level statistical concept identifier max. length of the attribute value mandatory or conditional status attachment level
ASI+...etc. (key family definitions may follow for additional key families)	
UNT+... UNZ+...	*(as discussed in the previous tables)*

Example:

```
UNA:+.? '
UNB+UNOC:3+4F0+BE2+970525:1539+IREF000001++GESMES/TS'
UNH+MREF000001+GESMES:2:1:E6'
BGM+73'
NAD+Z02+ECB'
NAD+MR+BE2'
NAD+MS+4F0'

VLI+CL_FREQ+++Frequency'
CDV+A'
FTX+ACM+++Annual'
CDV+Q'
FTX+ACM+++Quarterly'
CDV+M'
FTX+ACM+++Monthly'
CDV+D'
FTX+ACM+++Daily'

VLI+CL _AREA+++Country or area'
CDV+BE'
FTX+ACM+++Belgium'
CDV+DK'
FTX+ACM+++Denmark'
...
CDV+US'
FTX+ACM+++United States'

VLI+CL_BOP_ITEM+++IMF Bal. of payments component'
CDV+100'
FTX+ACM+++Current account; goods'
CDV+200'
FTX+ACM+++Current account; services'
CDV+300'
FTX+ACM+++Factor income'
...
CDV+995'
FTX+ACM+++Financial account'
CDV+998'
FTX+ACM+++Errors and omissions'

VLI+CL_BOP_DATA_TYPE+++Type of data'
CDV+1'
FTX+ACM+++Stock at the beginning of the period'
CDV+2'
FTX+ACM+++Credit flow'
CDV+3'
FTX+ACM+++Debit flow'
CDV+4'
FTX+ACM+++Net flow'
CDV+5'
FTX+ACM+++Price valuation adjustment'
CDV+6'
FTX+ACM+++Exchange rate adjustment'
CDV+7'
FTX+ACM+++Reclassification, other adjustments and other revaluations'
CDV+8'
FTX+ACM+++Stock at the end of the period'

VLI+CL_UNIT+++Unit'
```

Disseminating code lists

(1= Stock at the beginning of the period)

```
CDV+BEF'
FTX+ACM+++Belgian franc'
CDV+DEM'
FTX+ACM+++Deutsche Mark'
...
CDV+USD'
FTX+ACM+++US dollar'

VLI+CL_UNIT_MULT+++Unit multiplier'
CDV+6'
FTX+ACM+++Millions'
CDV+9'
FTX+ACM+++Billions'     etc.
```

Disseminating statistical concept definitions

```
STC+FREQ'
FTX+ACM+++Frequency'
STC+REF_AREA'
FTX+ACM+++Reference country or area'
STC+BOP_ITEM'
FTX+ACM+++IMF breakdown; balance of payments items (standard component)'
STC+DATA_TYPE'
FTX+ACM+++Type of data'
STC+TITLE'
FTX+ACM+++Title'
STC+UNIT'
FTX+ACM+++Unit'
STC+UNIT_MULT'
FTX+ACM+++Unit multiplier'

STC+TIME_PERIOD'
FTX+ACM+++Time period or time range'
STC+TIME_FORMAT'
FTX+ACM+++Time format'
STC+OBS_VALUE'
FTX+ACM+++Observation value'
STC+OBS_STATUS'
FTX+ACM+++Observation status'

STC+OBS_CONF'
FTX+ACM+++Observation confidentiality'
```

Disseminating a key family's definition

```
ASI+BAL_OF_PAYM_TEST'
FTX+ACM+++Bal. of payments k.f. for testing'

SCD+13+FREQ++++:1'          First dimension  is "frequency"
ATT+3+5+:::AN1'                and it is one character long;
IDE+1+CL_FREQ'                 corresponding code list: CL _FREQ
SCD+4+REF_AREA++++:2'       Second dimension of the series is "reference area"
ATT+3+5+:::AN2'                and it is two characters long;
IDE+1+CL_AREA'                 corresponding code list: CL_AREA
SCD+4+BOP_ITEM++++:3'    Third dimension is "IMF BoP standard component"
ATT+3+5+:::AN3'                and it is three characters long;
IDE+1+CL_BOP_ITEM'             corresponding code list: CL_BOP_ITEM
SCD+4+DATA_TYPE++++:4'      Fourth dimension is "Type of data/position of transaction"
ATT+3+5+:::AN1'                and it is one character long;
```

```
IDE+1+CL_BOP_DATA_TYPE'          corresponding code list: CL_BOP_DATA_TYPE
```

Other component elements in the ARR structure

```
SCD+1+TIME_PERIOD++++:5'         Fifth component element is time
ATT+3+5+:::AN..35'               and it is a descr. up to 35 char. long;

SCD+1+ TIME_FORMAT++++:6'        Sixth component element is the time format
ATT+3+5+:::AN3'                  and it is a number 3 char. long;

SCD+3+OBS_VALUE++++:7'           Seventh component element is the observation
ATT+3+5+:::N15'                  and it is a 15 char. long numeric field;
                                                    -- observation attributes --
SCD+3+OBS_STATUS++++:8'          Eighth component element is the observation status,
ATT+3+5+:::AN1'                  it is one character long,
ATT+3+35+2:USS'                  it is mandatory (usage status=2),
ATT+3+32+5:ALV'                  it is attached at the observation level (attachment level=5) and
IDE+1+CL_OBS_STATUS'             it takes its values from the CL_OBS_STATUS code list.

SCD+3+OBS_CONF++++:9'            Ninth component element is the observation confidentiality flag
ATT+3+5+:::AN1'                  it is one character long,
ATT+3+35+1:USS'                  it is conditional (usage status=1),
ATT+3+32+5:ALV'                  it is attached at the observation level (attachment level=5) and
IDE+1+CL_OBS_CONF'               it takes its values from the CL_OBS_CONF code list.
```

Disseminating same key family's definition of attributes

```
SCD+Z09+TITLE'                   "Title" is an attribute (=Z09) of the key family
ATT+3+5+:::AN..70'                 it can be up to 70 characters long
ATT+3+35+2:USS'                  it is mandatory (usage status=2) and
ATT+3+32+9:ALV'                  it is attached at the sibling group level (attachment level=9)
SCD+Z09+UNIT'                    "Unit" is an attribute of this key family
ATT+3+5+:::AN..12'                 and it can be up to 12 characters long,
ATT+3+35+2:USS'                  it is mandatory (usage status=2),
ATT+3+32+9:ALV'                  it is attached at the sibling group level (attachment level=9) and
IDE+1+CL_UNIT'                   it takes its values from the CL_UNIT code list

SCD+Z09+UNIT_MULT'               "Unit multiplier" is an attribute of the key family,
ATT+3+5+:::AN..2'                it can be 1 or 2 positions long,
ATT+3+35+2:USS'                  it is mandatory (usage status=2),
ATT+3+32+9:ALV'                  it is attached at the sibling group level (attachment level=9) and
IDE+1+CL_UNIT_MULT'              it takes its values from the CL_UNIT_MULT code list

UNT+135+MREF000001'
UNZ+1+IREF000001'
```

The example given above shows how a center (e.g., the ECB, coded as 4F0) can disseminate structural data to a central bank (e.g., the National Bank of Belgium, coded as BE2). Disseminating the concept definition *(segments starting with STC)* and the code lists *(groups starting with VLI)*, you provide the receiver with some basic metadata. Then, you give the dimensions *(first group of segments starting with ASI)*, and the receiver can form the key structure of the example key family *(BAL_OF_PAYM_TEST)*:

FREQ : REF_AREA : BOP_ITEM : DATA_TYPE

Dimension value length in characters: *1* *2* *3* *1*

This is the structure of the key of the time series of the key family. For instance a specific time series could be the monthly series (FREQ=*M*) referring to the *goods* (BOP_ITEM=*100*) *balance* (type:net=*4*) of Belgium (REF_AREA=*BE*); its key would be: M:BE:100:4

Additional information is provided by the attributes TITLE (uncoded), UNIT (coded) and UNIT_MULT (coded) of the key family, which for this series might take the values:

TITLE=*Merchandise trade balance*, UNIT=*BEF*, UNIT_MULT=*9*

6. SEGMENT-BY-SEGMENT REFERENCE GUIDE

6.1. How is this segment-by-segment guide organised?

This reference guide presents segments within separate sections like the ones presented in Table 2 in section 5 of this guide. As we discussed earlier, each section reflects a type of service (e.g., message administration).

All sections in this reference guide are organised as follows:

Scope
This defines the scope of the task.

GESMES segments used
This shows the branching diagram of GESMES with the relevant segments highlighted.

Features and guidelines for usage
This explains how to use a segment to implement a task.

For each segment

- Data-element sequence

This shows the sequence of data elements in the segment.

- data element usage rules

This is a table that defines:

⇒ the ***usage rules for the data element***, including its status:

mandatory: *the data element must be present;*

conditional: *the presence of the data element is conditional; the rules are explained in this table;*

optional: *the data element is not processed and, if sent, is for documentary purposes only.*

⇒ the ***format of the data element—****from the point of view of what the* ***GESMES/TS syntax allows.***

The notation used is:

an=alphanumeric,

n=numeric, and

".." = no more than.

Thus, an example of the format of a data element would be:

an4=four character (exact) alphanumeric, or

an..4=up to four characters alphanumeric.

This is useful for syntax checking vis-à-vis the general GESMES rules.

⇒ the ***valid codes of data elements and their meaning (and/or concrete format specifications) in the GESMES/TS context.*** Whenever the description of a segment refers to *code lists*, these lists are in the relevant documents or files provided by the structural definitions maintenance agency. (see "Code lists" paragraph further below).

- content and general usage rules of the segment

This describes the role of the segment and rules in the context of GESMES/TS.

- dependencies: This describes dependencies with other segments, apart from the ones within the group.

Code lists

Some of the presented segments refer to "code lists." Indeed, some "coded" elements need to take appropriate values from predefined code lists. We give these either as *concrete (mandatory for use)* code lists or as *recommendable* code lists or as *example* code lists (see Box 1 below).

BOX 1. CODE LISTS IN GESMES/TS

GESMES/TS has a fixed syntax without, of course, imposing specific code lists for the exchanged coded objects.

In general, the need for pre-defined code lists arises always in the context of a concrete data exchange agreement and implementation. More specifically, the code lists used in GESMES/TS could be distinguished in the following three categories:

- **Code lists mandatory for use by all GESMES/TS users:** these are the code lists which are very closely related to *core concepts* of EDIFACT (e.g. codes for period formats), GESMES (e.g. 73=definitions, 74=data in the BGM segment) and GESMES/TS (e.g. 7=update, 6=delete in the STS segment, 9="sibling" group of series as used in the 3rd occurrence of the ATT segment of the SCD group). A more complete presentation is provided on the corresponding Appendix discussing the use of EDIFACT code lists in GESMES/TS.
- **Highly recommendable code lists** are the code lists which have been commonly agreed between the largest part of the GESMES/TS community (e.g. the BIS, the ECB, Eurostat/BoP and all their world-wide partners): in order to make GESMES/TS messages to "look" as similar as possible, not only from a syntactical point of view but also "semantically", *there has been close co-operation between centres and agreements have been made on the use of a minimum set of commonly agreed code lists*. Like that, if some partners did not have the resources to build fully automated and generic systems processing the metadata found on the incoming messages according to pre-stored *alternative* sets of structural metadata, they could still "read" most of the qualitative information provided on the received GESMES/TS messages interpreting them in a similar way. That is why, for example, across most GESMES/TS partners the information about the "units", the "unit multiplier", the receiving or sending "organisation" is exchanged using the same code lists, regardless of which is the structural definition maintenance agency. The highly recommendable code lists for common use are the following (further information can be provided by one of the centres administrating GESMES/TS data exchanges):
  1. "*unit multiplier"* (e.g. millions) of the units of the series
  2. "*units*" of the series
  3. "*frequency*" of the series (e.g. daily, annual, quarterly)
  4. "*observation status*" (a mandatory flag attached to each observation)
  5. "*observation confidentiality*" (an optional flag attached to some observations)
  6. "*adjustment*" (if such adjustment has been performed)
  7. identification of partner *organisations*

  etc.
- **Code lists decided by each structural definitions maintenance agency and its partners:** these are code lists that are used in specific statistical data exchange needs. Usually, these code lists give values to the "dimensions" of the keys of the exchanged data (e.g. a code list for "balance sheet" items). Nevertheless, in order to promote harmonisation and to reduce "mapping" costs, all centre institutions are trying to use internationally accepted code lists and they are putting growing effort to have more and more code lists agreed, even in the context of "specific" implementations.

Section presentation one-by-one, including segment types and groups:

6.2. Interchange administration section

Scope

The administration data at the level of the interchange identifies the syntax version and character set used, the sender and receiver, and the date and time of the preparation of the interchange.

The Interchange Administration ***must*** be present in any interchange GESMES/TS file.

GESMES segments used

This comprises the UNA (a service segment that specifies syntax character) and UNB (indicates interchange header) segments. These segments are not a part of the message structure. They consist instead of the envelope header for the interchange, which can contain many GESMES messages.

Features and guidelines for use

You can send one or more GESMES messages in one interchange.

Interchange administration: segment structure

UNA*service-characters*

UNB+*syntax-identifier*:*syntax-version*+*sender-identification*+ *receiver-identification*+*date-of-preparation*:*time-of-preparation* +*interchange-reference*++*application-reference*'

6.2.1. Segment Type: UNA

Segment Type Name: *Syntax Character Specification*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

UNA*syntax-characters*

Data Element Rules

local name	usage rules	Format	code values
Syntax-characters	mandatory	an6 *(including the segment terminator)*	*In GESMES/TS these characters have to be:* `:+.? '`

Content and general usage rules

UNA specifies the data element separator and segment terminator characters that you use. The character repertoire for GESMES/TS is UNOC (see relevant discussion in the description of the next segment). To use the recommended service character set with the UNOC character repertoire, you will find it necessary to specify the service characters in the UNA segment as detailed below. In GESMES/TS the UNA segment has a fixed syntax and is written always as: `UNA:+.? '`

(=UNA followed by colon, plus sign, dot, question mark, space and single quote. The single quote is the character corresponding to the ANSI/ASCII character 39). The UNA statement, as it is given above, means:

- the *component separator* used is : (colon), the *data element separator* is + (plus sign), the *decimal sign* is . (dot), the *release indicator* is ? (question mark), space is reserved for possible future use and ' (single quote; ASCII/ANSI 39) is used as the *segment terminator*.

Your reading and writing applications should have these characters parameterised (at least the service characters: colon, plus sign, release indicator). For the proper use of the release character, see Appendix I on the EDIFACT syntax and the special section in this guide on the text contained in uncoded attributes. (The release character is not needed before the point, because the point is not an EDIFACT service character).

Example

It has to be written always as: `UNA:+.? '`

Attention!
(1) There is a space between ? and '.
(2) In GESMES/TS only the point (.) is used for the decimal notation.
(3) The last character (') is part of the segment (and segment terminator by "coincidence").

6.2.2. Segment Type: UNB

Segment Type Name: *Interchange header*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

UNB+*syntax-identifier*:*syntax-version*+*sender-identification*+*receiver-identification*+*date-of-preparation*:*time-of-preparation*+*interchange-reference*++*application-reference*++++*test-indicator*'

Data Element Usage Rules

local name	usage rules	format	code values
syntax-identifier	mandatory this identifies the character set used in the interchange	an4	UNOC - (ISO 8859-1) this supports acce-nted characters and the text can be in any European language except Greek
syntax-version	mandatory this is 3, which is the latest version of the syntax	n1	3
Sender-identification	mandatory it identifies the sender.	an..35	Central institution administering the data exchange provides an appropriate code list for identifying partner organisations.
Receiver-identification	Mandatory it identifies the receiver	an..35	As above.
Date-of-preparation	Mandatory this is the date of preparation of the interchange the format is YYMMDD	n6	e.g. 970525 (Attn! when 2000, then YY will be 00)
Time-of-preparation	Mandatory this is the local time of preparation of the interchange the format is HHMM	n4	e.g. 0950
Interchange-reference-number	mandatory a unique reference which identifies the interchange, incl. a serial incremental number (see also usage rules below)	an..14	in GESMES/TS the format is: IREFnnnnnn e.g. IREF000001
Application-reference	mandatory	an..14	GESMES/TS
Test indicator	conditional if found, then the contents should not be used to update or modify databases, as the interchange serves only testing purposes	n1	1

Content and general usage rules

This segment is the envelope header for one or more messages and contains administration and routing information that a message transfer service can use. *UNOC:3* is the syntax identifier referring to the character set used in the interchange (in GESMES/TS it has been set equal to UNOC:3, which allows the use of upper- and lower-case characters in the message, including the accented Latin characters).

The *date-of-preparation* and *time-of-preparation* (fixed format for both together: `YYMMDD:hhmm`) refers to the date and the local time that the system of the sender produced the interchange file.

The *interchange reference number* is a serial (incremental by one) integer number that the sender produces. It is strongly recommended, if you are an institution sending a file to a specific receiver, to increment this counter by one (keeping in each sending institution different counters for each receiving institution). This is the only safe way to detect not only duplicate interchanges but also missing interchanges sent to a center. The number part (full format: IREFnnnnnn) is six characters long, starting with 000001, 000002, ...etc. The value IREF999999 will be succeeded by IREF000000 and then by IREF000001 (starting again from the beginning). A *test interchange* should also be flagged with an interchange reference number incremented by one, vis-à-vis the previous interchange towards the same receiving institution.

Important note. When a file contains live data, which are supposed to be used to update receivers' live databases, the UNB segment ends with the element "GESMES/TS". However, for files that a receiver should not use to update live databases (e.g., test files), the sender must terminate the segment with the test indicator component (++++1).

Dependencies

The same *interchange reference number* has to appear also at the end of the message, in the UNZ segment.

Example 1 - exchange of live data

```
UNB+UNOC:3+BE2+4F0+950825:1236+IREF000006++GESMES/TS'
```

In this example, we have taken the values BE2 (=National Bank of Belgium) and 4F0 (=European Central Bank) from a code list called CL_ORGANISATION (commonly agreed and administered by the ECB, Eurostat/BOP, and the BIS; other data exchange circuits may need to define and use a different code list). The interchange file was produced on August 25, 1995 (at 12:36), and it is the sixth interchange of this institution via which it is sending data to the ECB.

Example 2 - exchange of test data

```
UNB+UNOC:3+BE2+4F0+950825:1236+IREF000007++GESMES/TS++++1'
```

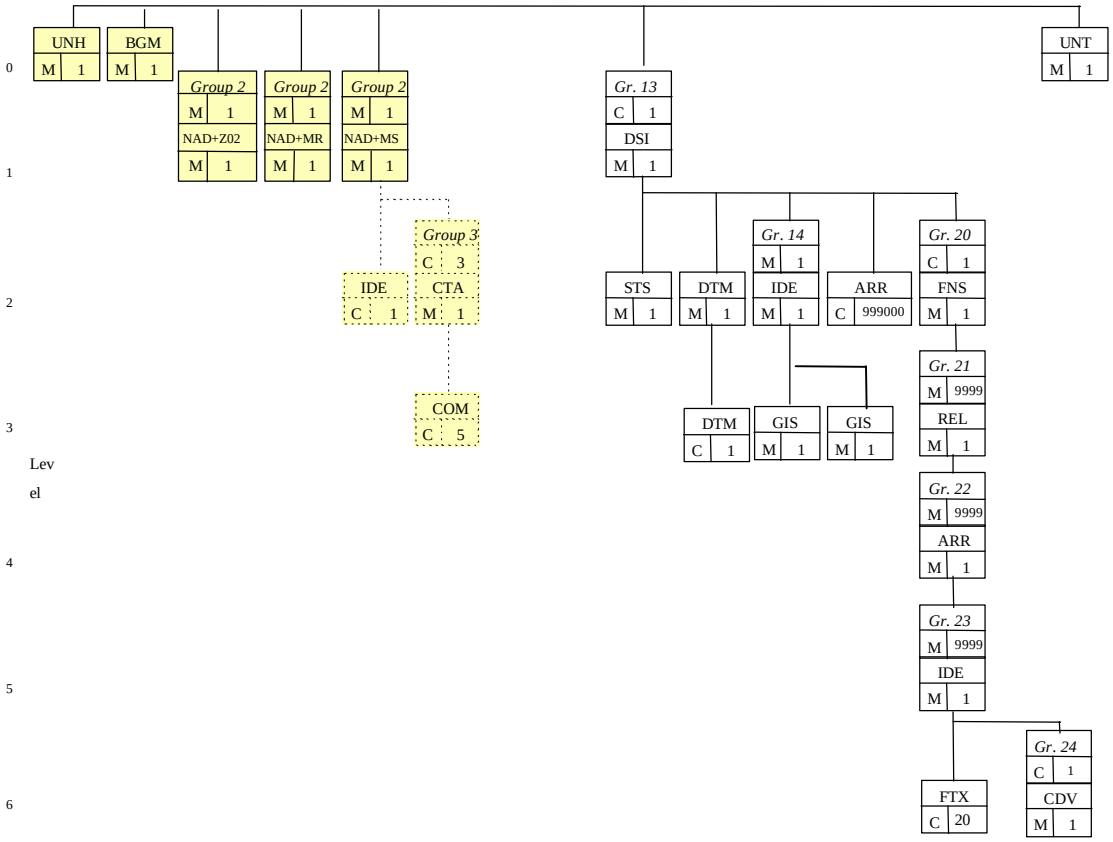
As before, but, owing to the presence of the test indicator (++++1), the receiver should not use the contents of the interchange to update or to modify databases, because this interchange serves only testing purposes.

6.3. Message administration section

Scope

This contains the administration data for the message (shaded boxes in the following branching diagram).

GESMES segments used



Features and guidelines for use

UNH ("header" of message) identifies the message type and version.

BGM ("beginning" of message) identifies the message purpose, either as containing a data set (or references to a data set) or as containing structural data.

NAD ("name and address"): three occurrences—one for identifying the code list maintenance agency (NAD+Z02), one for identifying the receiver (NAD+MR), and the third for identifying the sender (NAD+MS). The last one may trigger an IDE ("identity") segment and/or Group 3 (CTA and possibly COM). The IDE segment provides a message identity (assigned by the sender), and Group 3, which can be

repeated up to three times, provides detailed communication information about the sender. (If you as a sending institution use this, then you choose your one, two, or three most relevant contact people.)

- The Message Administration Group is necessary in all types of messages (the three NAD segments are mandatory).
- An interchange can contain more than one message; therefore, every message within the interchange should carry a Message Administration Group.

Message administration section: segment structure

UNH+*message-reference-number+message-type:message-type-version:message-type-release:controlling-agency***'**

BGM+*message-name***'**

NAD+Z02+*organisation-id***'** (code list maintenance agency identifier)

NAD+MR+*organisation-id***'** (receiver identifier)

NAD+MS+*organisation-id***'** (sender identifier)

IDE+10+*message-id*' (message identification provided by the sender)

it can be repeated up to three times

CTA+*contact-function+contact-id*:*contact-name*' (contact information)

it can be repeated up to five times

COM+*communication-number*:*communication-channel*' (com. number and type of channel)

6.3.1. Segment Type: UNH

Segment Type Name: *Message Header*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

UNH+*message-reference-number*+*message-type*:*message-type-version*:*message-type-release*:*controlling-agency*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
message-reference-number	mandatory reference that must be unique within the interchange using a serial (incremental by one) integer number produced by the sender (see also rules below).	an..14	*format:* MREFnnnnnn
message-type	Mandatory	an..6	GESMES
message-type-version	Mandatory	an..3	2
message-type-release	Mandatory	an..3	1
controlling agency	Mandatory	an..2	E6

Content and General Usage Rules

This segment starts the message and contains data that identify the message type and a unique reference that you can use to validate the integrity of the message. In GESMES/TS the only non-fixed element is the *message reference number*.

The message reference number (format: MREFnnnnnn) is a serial (incremental by one) integer number that you as a sender produce for each message in the current interchange (an interchange can include more than one message). The part of the pure number (nnnnnn) is six characters long, starting with MREF000001, MREF000002, ...etc. You can include no more than 999,999 messages in one interchange. The first message reference number should be 000001.

Dependencies

The same message reference number has to appear also at the end of the message, in the UNT segment.

Example

```
UNH+MREF000001+GESMES:2:1:E6'
```

6.3.2. Segment Type: BGM

Segment Type Name: *Beginning of Message*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

BGM+*message-name*'

Data Element usage Rules

local name	usage rules	Format	GESMES/TS code values
message-name	Mandatory **73** - statistical definitions this is used if the message contains only statistical definitions (e.g. structure definitions, code lists) and there are no data sets in the message (i.e. no occurrence of the DSI group) **74** - statistical data this is used to indicate that data and/or attributes are sent in the message or that the message contains delete references; therefore, the message contains the DSI group (no occurrence of the VLI, SCD and ASI groups). **DSL** - data set list the message contains only a list of data set identifiers and the key family definitions that describe their structure (only one occurrence of the VLI group and no occurrence of the SCD, ASI and DSI groups).	an..3	73 - statistical definitions 74 - statistical data DSL - data set list

Content and General Usage Rules

This segment identifies the function of the message.

Dependencies

If you have set the *message-name* to 74 (see table above), then the data set administration section has to be present, and you should not include structural sections (e.g., code lists, key family definition) in the message. The reading applications should expect numeric data and/or attributes in this message. On the other hand, if the *message-name* has been set to 73, the data set administration section should *not* be

present. And the reading application should conclude that the message contains structural definitions (lists of concepts or code lists or key family definitions).

If the *message-name* has been set to DSL then the message will contain only a data set list.

Example 1

```
BGM+74'
```

This message contains or refers to data (observations) and/or attributes.

Example 2

```
BGM+73'
```

This message contains only code lists and/or statistical concept definitions and/or key family definitions.

Example 3

```
BGM+DSL'
```

This message contains a data set list.

6.3.3. **Segment Group:** **Group 2 (NAD)**

Max. Number of Occurrences: *3*

Status: *Mandatory*

Content and General Usage Rules

The NAD group is used to identify:

(1) the organisation maintaining the code lists and the key family definitions,

(2) the receiving organisation, and

(3) the sending organisation (and, possibly, more information provided by the sender).

Each piece of information is provided by a simple NAD segment (NAD+Z02+..., NAD+MR+... and NAD+MS+... respectively). The third one (NAD+MS+...) might be followed by an IDE segment and/or up to three occurrences of Group 3 (CTA-COM), which give further information related to the sender.

6.3.4. Segment Type: NAD First occurrence - ref. to code lists administration agency

Segment Type Name: *Name and Address*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

NAD+*party-type*+*organisation-id*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
Party-type	mandatory Z02 indicates that this NAD segment refers to the party maintaining the default code lists	an..3	Z02 (fixed) Maintenance agency for code lists and key family definitions
Organisation-id	mandatory the code list maintenance agency id is given here	an..35	Examples: BIS ECB EUROSTAT IMF OECD

Content and General Rules

Use this to identify the organisation maintaining the code lists and key family definition used in the message. The sending institution chooses appropriately this parameter in accordance with the guidelines of the center whose statistical concepts, code lists, and structures are used in the message. For a more extensive discussion on the practical role and the use of the "maintenance agency" concept, you are advised to refer to the paragraph on central institutions.

Example 1

```
NAD+Z02+EUROSTAT'
```

This is how you should write the segment if the message uses/refers to structural definitions administered by Eurostat.

Example 2

```
NAD+Z02+BIS'
```

Example 3

```
NAD+Z02+ECB'
```

6.3.5. **Segment Type:** **NAD** Second occurrence - reference to the message receiver

Segment Type Name: *Name and Address*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

NAD+*party-type+organisation-id*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
Party-type	mandatory *party-type* is MR	an..3	MR - message receiver
Organisation-id	mandatory message receiver	an..35	A code value for identifying the receiving organisation; it is provided by the statistical center administering the data exchange.

Content and General Rules

Use this to identify the receiving organisation.

Example

NAD+MR+4F0'

Here, the message is supposed to be sent to the European Central Bank (here coded as 4F0).

6.3.6. Segment Type: NAD Third occurrence; reference to the message sender

Segment Type Name: *Name and Address*

Max. Number of Occurrences: *1*

Status: *Mandatory (trigger)*

Data Element Sequence

NAD+*party-type*+*organisation-id*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
party-type	mandatory *party-type* is MS	an..3	MS - message sender
organisation-id	mandatory message sender	an..35	A code value for identifying the sending organisation; the underlying code list is provided by the statistical center administering the data exchange.

Content and General Rules

Use this to identify the sending organisation.

Example

NAD+MS+BE2'

The National Bank of Belgium (here coded as BE2) is sending this message.

6.3.7. Segment Type: IDE

Segment Type Name: *Message identity*

Max. Number of Occurrences: *1*

Status: *Conditional*

Data Element Sequence

IDE+*object-type*+*message-identity*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
object-type	mandatory *object-type* is 10	an..3	10 - message context
message-identity	mandatory message identity assigned by the sender of the message (see also rules below)	an..35 (see also rules below)	free text *E.g. Regular daily update, Regular monthly reporting, etc.*

Content and General Rules

You as the sender provide the message identity. It is free text and, in general, can be up to 35 characters long (e.g., "daily update", "monthly tape").

Example 1

```
IDE+10+Quarterly BoP reporting'
```

You have identified this message as "quarterly BOP reporting."

6.3.8. Segment Group: Group 3 (CTA-COM)

Max. Number of Occurrences: *3*

Status: *Conditional*

Content and General Usage Rules

You use each occurrence of this group to provide details about a contact that is relevant to the sending organisation identified in the preceding NAD+MS segment. These details include the name of the contact and, optionally, contact numbers such as telephone, fax, e-mail.

This group is conditional. Certain data exchange circuits might find it useful and could agree to regularly use it in the exchanged messages.

Because you can repeat Group 3 three times, if you as a sender used it, you can choose the (maximum) three most relevant people whose details could be useful to the receiver.

6.3.9. Segment Type: CTA

Segment Type Name: *Contact information*

Max. Number of Occurrences: *1*

Status: *Mandatory (trigger)*

Data Element Sequence

CTA+*contact-function*+*contact-id*:*contact-name*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
contact-function	Mandatory It identifies the function of the person (on the sending side) whose name follows	an..3	CC - Responsible person for information production CP - Responsible person for computer data processing CF - Head of unit for information production CE - Head of unit for computer data processing
contact-id	Optional the identity of the contact (dept. id) as known in sender's side	an..17	e.g. BOP, M&B, EDP, ICSD etc.
contact-name	Mandatory the name of the contact person	an..35	e.g. John Smith

Content and General Rules

Use this to identify a contact name at your institution if you are the sender.

Example 1

```
CTA+CC+:Mr John Smith'
```

Mr. John Smith is the person responsible for compiling and producing the data at the sending institution.

Example 2

```
CTA+CP+IS/BoP:Mr John Smith'
```

Mr. John Smith is the person responsible for data computer processing at the IS/BOP unit in the sending institution.

6.3.10. Segment Type: COM

Segment Type Name: *Communication number and type of channel*

Max. Number of Occurrences: *5*

Status: *Conditional*

Data Element Sequence

COM+*communication-number*:*communication-channel*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
communication-number	mandatory Telephone or fax number or e-mail address etc.	an..512	e.g. 0049 69 1344 0
communication-channel	mandatory	an..3	EM - e-mail TE - telephone FX - fax XF - X.400

Content and General Rules

Use this to provide a contact number for the person indicated in the preceding CTA segment and to identify the type of this number (communication channel).

Example 1

```
COM+0049 69 13440:TE'
```

Example 2

Combining with previous segments:

```
NAD+MS+4F0'

IDE+10+ECB-monthly disseminated aggregated data'

CTA+CC+Money and Banking:Mr John Smith'
COM+0049 69 13440:TE'
COM+0049 69 13446000:FX'
COM+jsmith@test.com:EM'

CTA+CP+IS/M&B:Mr Klaus Roberts'
COM+0049 69 1344888:TE'
```

This example shows the use of the whole (conditional) group following NAD+MS. The ECB (=4F0) provides information about the message contents (IDE segment) and people to contact (CTA/COM segments).

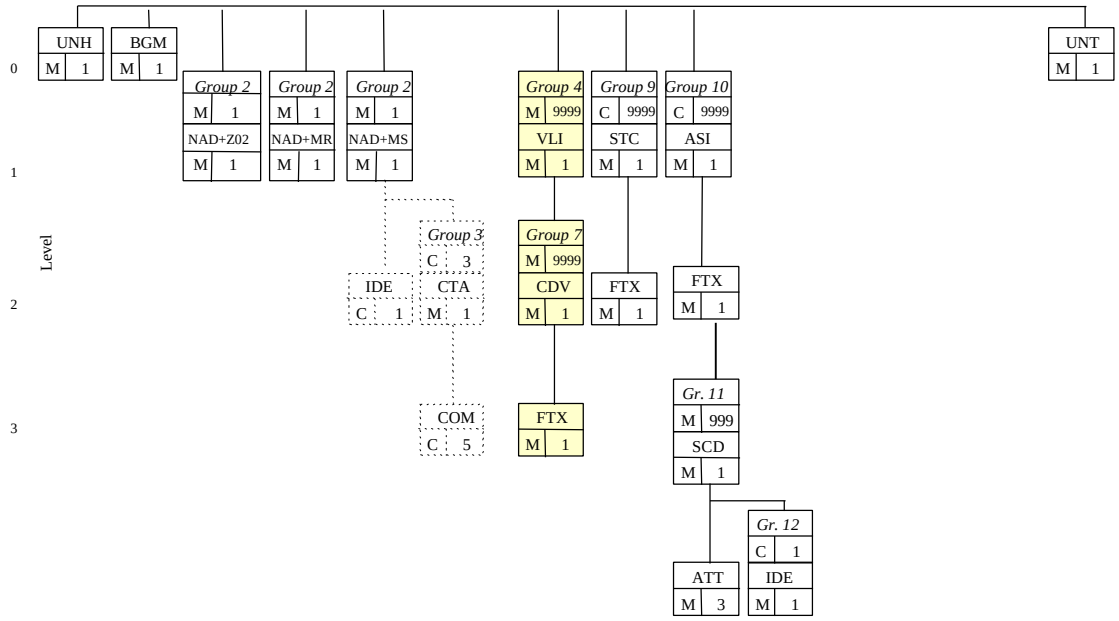
6.4. Code-list section

Scope

When you as a center disseminate the code lists, this enables the receiving institutions to build tables with the possible values for all coded statistical concepts (dimensions used in key families and coded attributes). Any of the data sets (sent in other GESMES/TS messages) can refer to the code lists.

GESMES segments used

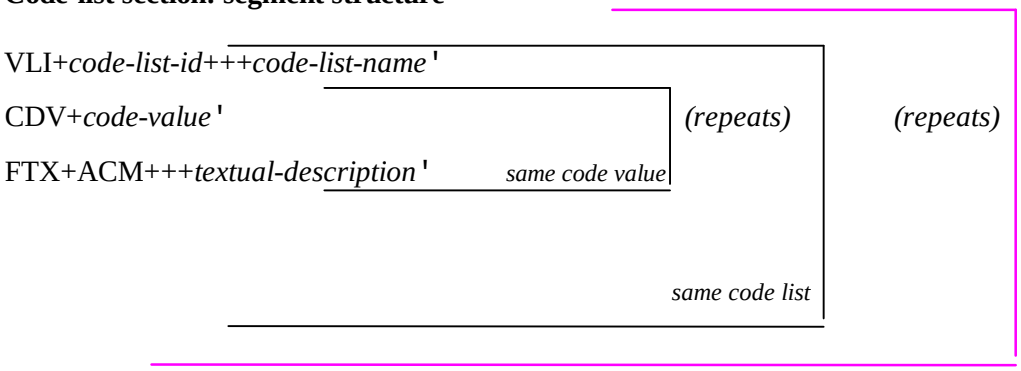
In GESMES/TS, you cannot mix data and attributes (DSI or "data set id" Group) in the same message with the groups of segments carrying the structural definitions (i.e., code lists–VLI, statistical concepts–STC, key family definitions–ASI). Therefore, the following branching diagram presents a typical GESMES/TS structural message carrying code lists, statistical concept definitions, and key family definitions. The shaded boxes are the ones used for the exchange of code lists, which is discussed in this section. This is why we have flagged Group 4 here as mandatory.



Features and guidelines for use

You send code lists in Group 4. You must give each list a unique identifier and send it in the VLI ("value list id") segment. You send the individual code values in the CDV segment (one code in each occurrence of a CDV segment) and give their textual description in the FTX segment (one textual description for each occurrence of a CDV segment).

Code-list section: segment structure



Please note that we foresee a slightly different use of the VLI group for the exchange of data set lists (a special type of message that is described in the corresponding Appendix IV).

6.4.1. **Segment Group:** **Group 4 (VLI-Group 7)**

Max. Number of Occurrences: *9999*

Status: *Conditional*

Content and General Usage Rules

A group of segments containing a code list. The receiver can store it on a structural definitions database (see also alternative ways of organizing structural definitions in section 3.4.1.).

6.4.2. Segment Type: VLI

Segment Type Name: *Value List Identification*

Max. Number of Occurrences: *1*

Status: *Mandatory (trigger)*

Data Element Sequence

VLI+*code-list-id*+++*code-list-name*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
code-list-id	mandatory the identity of the code list	an..18	Code list identifier as provided by a center institution, e.g.: CL_UNIT
code-list-name	mandatory the name of the code list	an..70	Short description (name) of a code list provided by a center institution, e.g.: Units code list

Content and General Usage Rules

Use this segment to provide a unique identifier and a name for a code list.

Example 1

VLI+CL_ADJUSTMENT+++Adjustment code list'

We provide a code list called CL_ADJUSTMENT via the segments that follow (CDV, FTX).

Example 2

VLI+CL_ORGANISATION+++Organisation code list'

We provide a code list called CL_ORGANISATION via the segments which follow (CDV, FTX).

6.4.3. Segment Group: Group 7 (CDV-FTX)

Max. Number of Occurrences: *9999*

Status: *Mandatory*

Content and General Usage Rules

A group of segments containing the code values in a code list and their descriptions.

The CDV segment occurs once for each code, and the FTX segment once for each CDV.

6.4.4. Segment Type: CDV

Segment Type Name: *Code Value*

Max. Number of Occurrences: *1*

Status: *Mandatory (trigger)*

Data Element Sequence

CDV+*code-value*'

Data Element Usage Rules

local name	usage rules	Format	GESMES/TS code values
code-value	mandatory the code value	an..18	A code value from a code list provided by a center institution.

Content and General Rules

Use this to provide a code value.

Example

CDV+DK2'

This is a code value from the CL_ORGANISATION code list (mentioned in a previous example when demonstrating VLI).

6.4.5. Segment Type: FTX

Segment Type Name: *Free Text (code value textual description)*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

FTX+*text-subject*+++*code-value-description*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
Text-subject	Mandatory in Gesmes/TS: "statistical description"	an..3	ACM (=statistical description)
Code-value-description	Mandatory the text of the code value	an..350	The textual description of the meaning of a code value. Expression for maximum length: text:text:text:text:text where each "text" can be up to 70 characters long.

Content and General Rules

Use this to describe a code value.

You use each occurrence of this segment to describe a code value. It can be up to (total) 350 characters long. However, if the description is longer than 70 characters (upper limit for a single component element), then you need to split the *text* in (the maximum) 5 component element-texts (each one of them no longer than 70 characters and separated among them with a *component separator*).

The reading applications should read the components of the received text and consider that they form a string that is the result of the added sub-strings. The writing applications should break the text into pieces of 70 characters or shorter. It does not matter whether a word is broken between two component elements or not (see also the Appendix I on the EDIFACT syntax).

Briefly, you can give the code value description as *free-text*:*free-text*:*free-text*:*free-text*:*free-text*' where *free-text* can be up to 70 characters long and can repeat up to five times. See also section 6.9.9.

Example

FTX+ACM+++Danmarks Nationalbank'

This description explains the code value shown in the previous example demonstrating CDV.

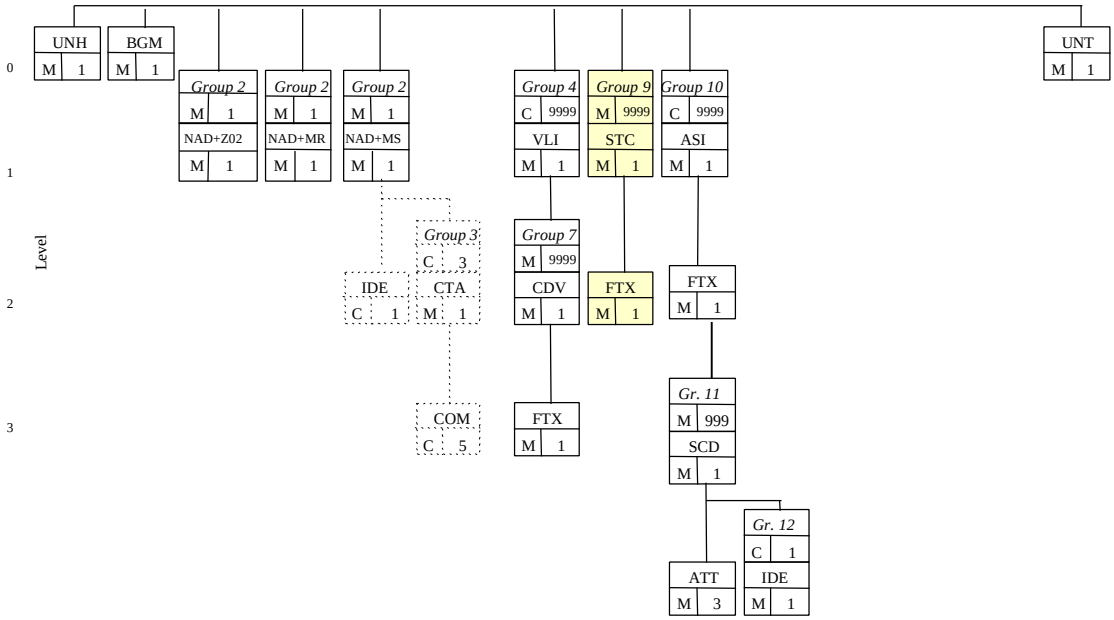
6.5. Statistical concept definition section

Scope

This provides a list of the statistical concepts, together with their names, used by the partners in a given data exchange context.

GESMES segments used

In the following GESMES/TS branching diagram, we use the shaded boxes in defining the concepts. Group 9



is shown here as mandatory, because without it, you cannot disseminate statistical concept definitions.

Features and guidelines for use

You use a pair of segments; the first segment (STC) provides the identifier of the statistical concept and the second one (FTX) its name.

Statistical concept definition section: segment structure

STC+*concept-identifier*'
FTX+ACM+++*concept name*' *repeats*

6.5.1. **Segment Group:** **Group 9 (STC-FTX)**

Max. Number of Occurrences: *9999*

Status: *Conditional*

Content and General Usage Rules

A pair of segments describing a statistical concept.

6.5.2. Segment Type: STC

Segment Type Name: *Statistical concept*

Max. Number of Occurrences: *1*

Status: *Mandatory (trigger)*

Data Element Sequence

STC+*concept-identifier*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
concept-identifier	mandatory the identifier of the statistical concept	an..18	A statistical concept identifier as provided by a center institution. e.g.: UNIT

Content and General Usage Rules

This segment identifies the statistical concept.

Example

STC+SOURCE_AGENCY'

The *name* of a concept, whose identifier is SOURCE_AGENCY, will be given in the FTX segment that follows.

6.5.3. Segment Type: FTX

Segment Type Name: *Free Text (name of the concept)*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

FTX+*text-subject*+++*code-label*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
text-subject	mandatory in GESMES/TS: "statistical description"	an..3	ACM (=stat.description)
code-label	mandatory name of the statistical concept	an..70	A name for the concept specified on the previously given STC segment.

Content and General Rules

Use this to provide the name of the concept stated in the previous segment (STC).

Example

FTX+ACM+++Source agency'

This is the name of the concept shown in the previous example (see description for STC).

6.6. Key family definition section

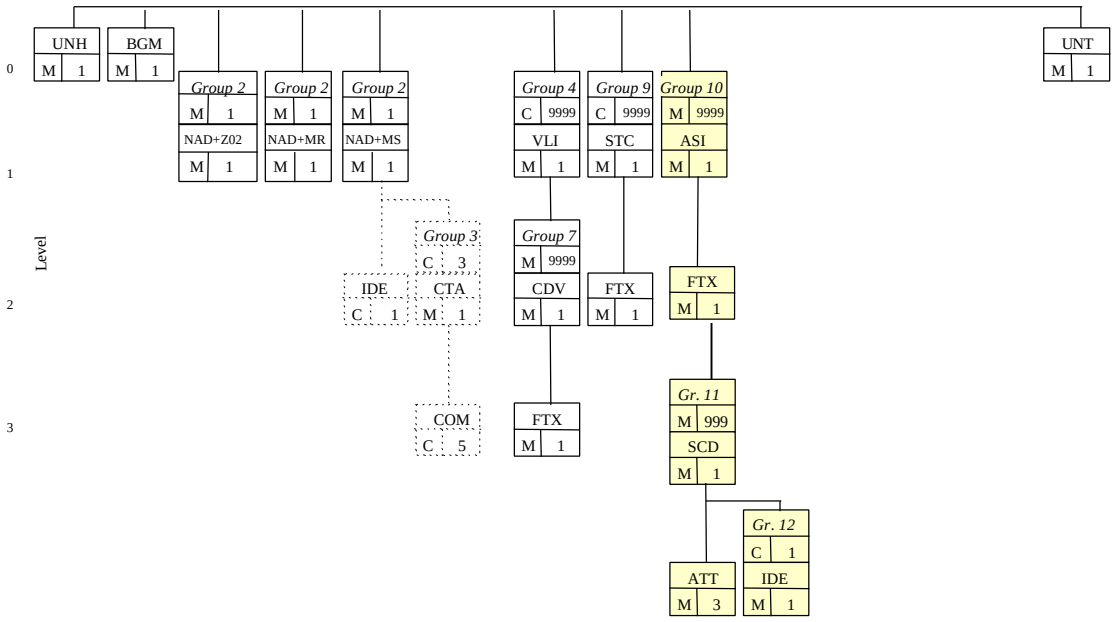
Scope

This section allows you as an organisation *receiving* the corresponding structural message to interpret key family definitions (which dimensions and at which positions) and the corresponding lists of relevant attributes.

Remember that by now your central organisation should have already informed you (e.g., via Group 8, "exchanged" concepts) as a partner institution about some first structural components (i.e., code lists, statistical concepts). In other words, you should by now know each concept i.d. used in this key family definition. Otherwise, you will not be able to interpret the key family definitions.

GESMES segments used

When the parties exchange key family definitions, they must use Group 10 (the shaded boxes show the segments involved); that is why this group is shown here as mandatory:

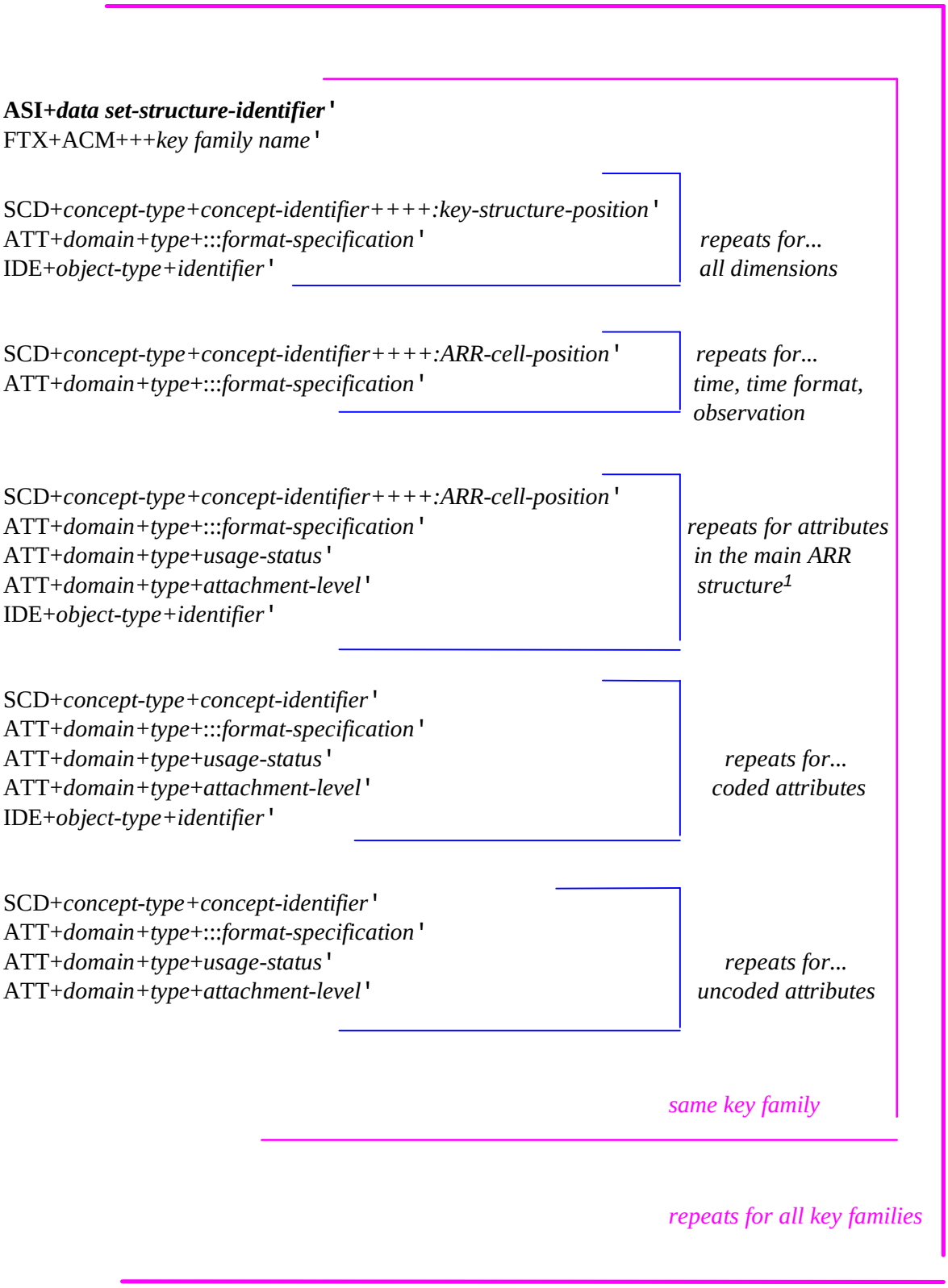


Features and guidelines for use

A key family definition is reflected in one occurrence of the ASI segment. An FTX segment provides the name of the key family. You identify each structure component as an occurrence of the SCD segment, in terms of the statistical concept identifier (and the position of the dimension in the key structure if the concept is a dimension). In general, a complete key family definition comprises:

- the identification of all dimensions of this key family and also of all the other relevant observation component elements. (This structural information is needed for the interpretation of the ARR segment following Group 14—not shown here—which holds the data to be exchanged.)
- the identification of all the attributes (mandatory and non-mandatory) that are in use within this key family (SCD segment), the definition of their status, and their attachment level;
- the description of the format of the values of the statistical concepts (ATT segment); and
- the identification of the corresponding code list (IDE) if the statistical concept is coded.

Key family definition section: segment structure



[1] However, the observation pre-break value is not coded and, therefore, no IDE follows for it.

6.6.1. **Segment Group:** **Group 10 (ASI-Group 11)**

Max. Number of Occurrences: *9999*

Status: *Conditional*

Content and General Usage Rules

Use each occurrence of this group to define a key family (its dimensions and attributes). The key family may be new, or it may already exist on the receiver's database, in which case the definition in this group should replace the definition in the existing structural definitions database.

6.6.2. Segment Type: ASI

Segment Type Name: *Array Structure Identification*

Max. Number of Occurrences: *1*

Status: *Mandatory (trigger)*

Content and General Usage Rules

Use this to identify a key family.

Data Element Sequence

ASI+*data set-structure-identifier*'

Data Element Usage Rules

Local name	usage rules	format	GESMES/TS code values
Data set-structure-identifier	Mandatory	an..18	a key family identifier

Content and General Rules

Use this to identify a key family.

Example

```
ASI+ESCB_BAL_OF_PAYM01'
```

6.6.3. Segment Type: FTX

Segment Type Name: *Free Text (name of a key family)*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

FTX+*text-subject*+++*code-label*'

Data Element Usage Rules

local name	Usage rules	Format	GESMES/TS code values
text-subject	Mandatory in GESMES/TS: "statistical description"	an..3	ACM (=statistical description)
code-label	Mandatory name of the statistical concept	an..70	The name (short description) of the key family specified on the previously given ASI segment.

Content and General Rules

Use this to provide the name of the key family stated in the previously given segment (ASI).

Example

FTX+ACM+++ECB kf for BoP data'

This is the name of the key family shown in the previous example (see description for ASI).

***6.6.4.* Segment Group:** **Group 11 (SCD-ATT-Group 12)**

Max. Number of Occurrences: *999*

Status: *Conditional*

Content and General Usage Rules

This segment group identifies the statistical concepts of the key family and defines them as either dimensions or attributes. Dimensions must be coded, while attributes may be or not.
In addition, it provides information required for interpreting the ARR segment following group 14:

- time dimensions: "time period" ("To which periods do the data in the ARR segment relate?") and "time format" ("Do the data in the ARR segment relate to a single period or to a range and what is the frequency?")
- array cells: observation value, the observation-level attribute observation status (mandatory), and the observation-level attributes observation confidentiality and observation pre-break (both conditional).

We elaborate on these categories below:

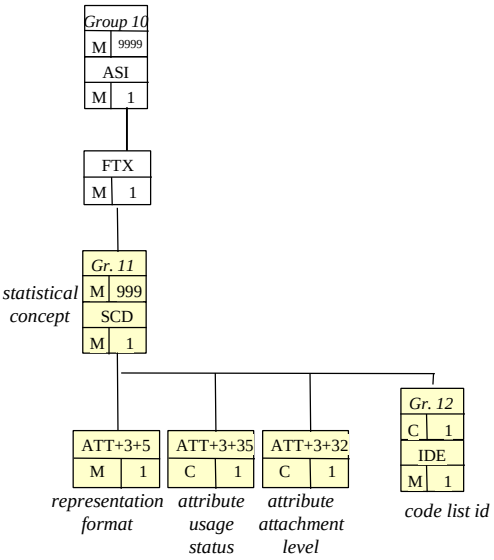
Dimension: For each dimension of the key family, (1) its position in the key structure, (2) its representation (length of the code values and whether they are numeric or alphanumeric), and (3) the name of the relevant code list are provided.

Time period, time format, observation value, and "array" attributes: These are defined by their representation and their position in the array structure. Moreover, for the three observation-level attributes ("array cell attributes": observation status, observation confidentiality, observation pre-break value) and also the attachment level, the segment provides their usage status and (but not for the "observation pre-break") the relevant code lists.

Attribute definitions: All attributes for the key family are defined with the information on

- whether they are coded or uncoded
- their usage status (mandatory or conditional)
- their attachment level (data set, sibling series, time series or observation)
- representation: field length (exact or maximum length) and the information whether they are numeric or alphanumeric
- relevant code list (for coded attributes) .

You use the ATT segment for three different purposes. (That is why in the branching diagram we show that it can be repeated up to three times.) Thus, for reasons of clarity in the following pages, we present the ATT segment separately, for each one of its three uses. The diagram below provides a “zoom” view on the group:



Segment structure

SCD+concept type+concept identifier++++:position in key (or array structure)'

ATT+domain+type+:::format specification' (*for the representation*)

ATT+domain+type+code:code list qualifier' (*for the usage status of attributes only*)

ATT+domain+type+code:code list qualifier' (*for the attachment level of attributes only*)

IDE+object type+identifier' (*code list for dimensions, coded attributes*)

Note

We have derived the examples for the various segments to use in the key family definition section from the example key family defined in the next paragraph. We have built up the examples in such a way that you can, with each added segment, retain the overall view of the whole group. We then present the complete section with required segments following the presentation of the IDE segment (group 12).

6.6.5. Segment Type: SCD

Segment Type Name: *Structure Component Definition*

Max. Number of Occurrences: *1*

Status: *Mandatory (trigger)*

Data Element Sequence

SCD+*concept-type*+*concept-identifier*++++*:position-in-key-structure*'

Data Element Usage Rules

local name	usage rules	Format	GESMES/TS code values
concept-type	Mandatory	an..3	1 - time 3 - array cell 13 - dimension "frequency" in key family's structure 4 - dimension in key family's structure (other than "frequency") Z09 - attribute
concept-identifier	Mandatory the identifier of the statistical concept	an..18	Provided by center institution.
position-in-key-structure	Conditional this gives the position of the statistical concept (as a dimension) in the key family structure (e.g. a value of 1 is the first position), if the concept is a dimension or element of the ARR structure.	an..6	1 - first position 2 - second position ... etc.

Content and General Rules

Use this 1) to identify the statistical concepts in the key family and their type (dimension or attribute) and 2) in the case of dimensions, to define their position in the key structure. In addition, apart from the dimensions, it defines the array cell position in the ARR segments of the supplementary elements (i.e., time dimension, time format, observation, observation status [and confidentiality and observation pre-break, if used in the key family]).

Example

Let's consider the definition of the following (example) key family:

Statistical Concept	Identifier	alphanum.(an)	concept role	Position in the key / *array struct. position*
DIMENSIONS:				
Frequency	FREQ	an1	dimension	**1**
Reporting/ref. country	REF_AREA	an2	dimension	**2**
IMF Bal.of Paym. comp.	BOP_ITEM	an3	dimension	**3**
Type of data	DATA_TYPE	an1	dimension	**4**
Time period	TIME_PERIOD	an..35	*time dimension*	**5**
Time format	TIME_FORMAT	an3	*time dimension*	**6**
Observation	OBS_VALUE	an..15	*observation*	**7**
Observation status	OBS_STATUS	an1	*array cell*	**8**
Obs. confidentiality	OBS_CONF	an1	*array cell*	**9**
Pre-break value	OBS_PRE_BREAK	an..15	*array cell*	**10**
ATTRIBUTES:				
Title	TITLE	an..70	attribute	-
Unit	UNIT	an..12	attribute	-
Unit multiplier	UNIT_MULT	an..2	attribute	-

(In this key family we have assumed that the observation confidentiality and observation pre-break have been included in the array structure of the key family; this might not be the case for other key families.)

The structure definition segments for this key family would be:

SCD+**13**+FREQ++++:1'... *(corresponding ATT and IDE segments should follow)* ...
SCD+**4**+REF_AREA++++:2'
... *(corresponding ATT and IDE segments should follow)* ...

SCD+**4**+BOP_ITEM++++:3'
... *(corresponding ATT and IDE segments should follow)* ...
SCD+**4**+DATA_TYPE++++:4'
... *(corresponding ATT and IDE segments should follow)* ...

Other component elements (time dimension and array cells) that are present in the ARR segments for this particular key family are:

SCD+**1**+TIME_PERIOD++++:5'
(corresponding ATT segment should follow)

SCD+**1**+TIME_FORMAT++++:6'
...*(corresponding ATT segment should follow)*...

SCD+**3**+OBS_VALUE++++:7'
...*(corresponding ATT segment should follow)*...

SCD+**3**+OBS_STATUS++++:8'

. . .*(corresponding ATT and IDE segments should follow)*. . .

SCD+**3**+OBS_CONF++++:9'
. . .*(corresponding ATT and IDE segments should follow)*. . .

SCD+**3**+OBS_PRE_BREAK++++:10'
. . .*(corresponding ATT segments should follow)*. . .

Here are the attributes used in this key family (**note that only *concept-type* and *concept-identifier* are used here**):

SCD+Z09+TITLE'
. . .*(corresponding ATT segments should follow)*. . .

SCD+Z09+UNIT'
. . . *(corresponding ATT and IDE segments should follow)*. . .

SCD+Z09+UNIT_MULT'
. . .*(corresponding ATT and IDE segments should follow)*. . .

6.6.6. Segment Type: **ATT** First occurrence - ref. to representation format

Segment Type Name: *Attribute*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence
ATT+*domain*+*type*+:::*format-specification*'

Data Element Usage Rules

Local name	usage rules	format	GESMES/TS code values
Domain	Mandatory	an..3	3 - related to the key family definition
Type	Mandatory	an..3	5 - representation
Format-specification	mandatory this gives the format (numeric or alphanumeric and the length) of the code values (AN*n*=exact, AN..*n*=up to *n* character long).	an..35	e.g. AN2 (=alphanumeric, exactly 2 char. long) AN..3 (=up to 3 characters long) etc.

Content and General Rules

Use this one to inform about the format of the values taken by all the statistical concepts used in the key family.

Example

Consider again the previous example:

Statistical Concept	Identifier	alphanum. (an) or numeric (n)	Concept role	Position in the key / *array struct. position*
Frequency	FREQ	an1	Dimension	1
Reporting/ref. country	REF_AREA	an2	Dimension	2
IMF Bal.of Paym. comp.	BOP_ITEM	an3	Dimension	3
Type of data	DATA_TYPE	an1	Dimension	4
Time period	TIME_PERIOD	an..35	*time dimension*	5
Time format	TIME_FORMAT	an3	*time dimension*	6
Observation	OBS_VALUE	an..15	*observation*	7
Observation status	OBS_STATUS	an1	*array cell*	8
Obs. confidentiality	OBS_CONF	an1	*array cell*	9
Pre-break value	OBS_PRE_BREAK	an..15	*array cell*	10
Title	TITLE	an..70	Attribute	-
Units	UNIT	an..12	Attribute	-
Unit multiplier	UNIT_MULT	an..2	Attribute	-

The structure definition segments (SCD and the representation definition in ATT) would now look like this:

```
SCD+13+FREQ++++:1'
ATT+3+5+:::AN1'
....                                    IDE segment to follow
SCD+4+REF_AREA++++:2'
ATT+3+5+:::AN2'
....                                    IDE segment to follow
SCD+4+BOP_ITEM++++:3'
ATT+3+5+:::AN3'
....                                    IDE segment to follow
SCD+4+DATA_TYPE++++:4'
ATT+3+5+:::AN1'
....                                    IDE segment to follow
SCD+1+TIME_PERIOD++++:5'
ATT+3+5+:::AN..35'                      No further segments relating to TIME_PERIOD
SCD+1+TIME_FORMAT++++:6'
ATT+3+5+:::AN3'                         No further segments relating to TIME_FORMAT
SCD+3+OBS_VALUE++++:7'
ATT+3+5+:::AN..15'                      No further segments relating to OBS_VALUE
SCD+3+OBS_STATUS++++:8'
ATT+3+5+:::AN1'
....                                    Other ATT and IDE segments to follow
SCD+3+OBS_CONF++++:9'
ATT+3+5+:::AN1'
....                                    Other ATT  and IDE segments to follow
SCD+3+OBS_PRE_BREAK++++:10'
ATT+3+5+:::AN..15'
....                                    Other ATT segments to follow
SCD+Z09+TITLE'
ATT+3+5+:::AN..70'
....                                    Other ATT segment to follow
SCD+Z09+UNIT'
ATT+3+5+:::AN..12'
....                                    Other ATT  and IDE segments to follow
SCD+Z09+UNIT_MULT'
ATT+3+5+:::AN..2'
....                                    Other ATT and IDE segments to follow
```

6.6.7. Segment Type: ATT Second occurrence - ref. to usage status

Segment Type Name: *Attribute*

Max. Number of Occurrences: *1*

Status: *Conditional*

Data Element Sequence

ATT+*domain*+*type*+*code*:*code-list-qualifier*'

Data Element Usage Rules

Local name	usage rules	format	GESMES/TS code values
Domain	mandatory	an..3	3 - related to the key family definition
Type	mandatory	an..3	35 - usage status
Code	mandatory this gives the status of the attribute; if it is mandatory, then it should be expected to have always a value	an..3	1 - conditional 2 - mandatory
Code-list-qualifier	mandatory this gives the code list from which the previous values are taken	an..3	USS – usage status

Content and General Rules

You must use this for all attributes (and only for the attributes). This applies to the attributes that you exchange as array cell attributes (attached next to the observation) and to the attributes that you exchange using the FNS group. (So, it does not apply to the definition of the observation value, which is mandatory)

Example

Taking again the previous example, the structure-definition segments (SCD and the representation and usage status definition in ATT) relating to the attributes only would now look like this:

```
SCD+3+OBS_STATUS++++:8'
ATT+3+5+:::AN1'
ATT+3+35+2:USS'          Usage status for OBS_STATUS is mandatory
....                     Other ATT and IDE segments to follow
SCD+3+OBS_CONF++++:9'
ATT+3+5+:::AN1'
ATT+3+35+1:USS'          Usage status for OBS_CONF is conditional
....                     Other ATT  and IDE segments to follow
SCD+3+OBS_PRE_BREAK++++:10'
ATT+3+5+:::AN..15'
ATT+3+35+1:USS'          Usage status for OBS_PRE_BREAK is conditional
```

`....`	Other ATT segment to follow
`SCD+Z09+TITLE'`	
`ATT+3+5+:::AN..70'`	
`ATT+3+35+2:USS'`	**Usage status for `TITLE` is mandatory**
`....`	Other ATT segment to follow
`SCD+Z09+UNIT'`	
`ATT+3+5+:::AN..12'`	
`ATT+3+35+2:USS'`	**Usage status for `UNIT` is mandatory**
`....`	Other ATT and IDE segments to follow
`SCD+Z09+UNIT_MULT'`	
`ATT+3+5+:::AN..2'`	
`ATT+3+35+2:USS'`	**Usage status for `UNIT_MULT` is mandatory**
`....`	Other ATT and IDE segments to follow

6.6.8. Segment Type: **ATT** Third occurrence - ref. to the attachment level

Segment Type Name: *Attribute*

Max. Number of Occurrences: *1*

Status: *Conditional*

Data Element Sequence

ATT+*domain*+*type*+*code*:*code-list-qualifier*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
Domain	Mandatory	an..3	3 - related to the key family definition
Type	Mandatory	an..3	32 - object link type (attachment level)
Code	Mandatory this gives the attachment level of the attribute	an..3	1 - data set 4 - time series 5 - observation 9 - sibling group
code-list-qualifier	Mandatory this gives the code list from which the previous values are taken	an..3	ALV – attachment level

Content and General Rules

You must use this for all attributes (and only for the attributes). This applies to the attributes that you exchange as array cell attributes (attached next to the observation) and to the attributes that you exchange using the FNS group. (So, it does not apply to the definition of the observation value)

Example

Using again the previous example, we assume that in this key family, you define:

- the "observation status," the "observation confidentiality," and the "observation pre-break" at the observation level;
- the "title" at the sibling group level, and
- the "unit" and "unit multiplier" at the time series level.

The structure definition segments (SCD and the representation, usage status, and attachment level definition in ATT) relating to the attributes only would now look like the following (dimensions and the time dimensions do not carry an attachment level).

```
SCD+3+OBS_STATUS++++:8'
ATT+3+5+:::AN1'
ATT+3+35+2:USS'
ATT+3+32+5:ALV'          Attachment of OBS_STATUS is at the observation level
....                     IDE segment (identifying relevant code list) to follow
```

```
SCD+3+OBS_CONF++++:9'
ATT+3+5+:::AN1'
ATT+3+35+1:USS'
ATT+3+32+5:ALV'
....
```

Attachment of OBS_CONF is at the observation level

IDE segment to follow

```
SCD+3+OBS_PRE_BREAK++++:10'
ATT+3+5+:::AN..15'
ATT+3+35+1:USS'
ATT+3+32+5:ALV'
```

Attachment of OBS_PRE_BREAK is at the obs. level

```
SCD+Z09+TITLE'
ATT+3+5+:::AN..70'
ATT+3+35+2:USS'
ATT+3+32+9:ALV'
```

Attachment of TITLE is at the sibling level

```
SCD+Z09+UNIT'
ATT+3+5+:::AN..4'
ATT+3+35+2:USS'
ATT+3+32+4:ALV'
....
```

Attachment of UNIT is at the time series level

IDE segment to follow

```
SCD+Z09+UNIT_MULT'
ATT+3+5+:::AN..2'
ATT+3+35+2:USS'
ATT+3+32+4:ALV'
....
```

Attachment of UNIT_MULT is at the time series level

IDE segment to follow

6.6.9. Segment Group: **Group 12 (IDE)**

Max. Number of Occurrences: *1*

Status: *Conditional*

Content and General Usage Rules

Use this group to identify the code list used by a coded statistical concept.

6.6.10. Segment Type: IDE

Segment Type Name: *Identity*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

IDE+*object-type+identifier'*

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
object-type	mandatory	an..3	1 - code list
Identifier	mandatory the identifier of the code list	an..18	Provided by center institution.

Content and General Rules

This sequence contains the identifier of the code list of the statistical concept specified in the preceding SCD segment if this concept is coded.

Example 1

```
IDE+1+CL_FREQ'
```

Note that this IDE should be preceded by a relevant SCD segment (and one or three ATT segments depending on whether the concept is a dimension or an attribute). For example, to introduce the previous segment, the following segments could precede:

```
SCD+13+FREQ++++:1'
ATT+3+5+:::AN1'
```

Obviously, you will not use the IDE segment if the related statistical concept is uncoded.

Example 2

Referring to the example presented in the previous pages, the complete structure definition section (SCD and the representation, usage status, and attachment level definition in ATT and conditional IDE segment) for dimensions, the time dimension, array cells, and attributes would now look like this:

```
SCD+13+FREQ++++:1'
ATT+3+5+:::AN1'
IDE+1+CL_FREQ'
SCD+4+REF_AREA++++:2'          Concept: REF_ AREA (=reference area)
ATT+3+5+:::AN2'
```

IDE+1+CL_AREA_EE' **Code list CL_AREA_EE is used for REF_AREA**
SCD+4+BOP_ITEM++++:3'
ATT+3+5+:::AN3'
IDE+1+CL_BOP_ITEM'
SCD+4+DATA_TYPE++++:4'
ATT+3+5+:::AN1'
IDE+1+CL_BOP_DATA_TYPE'
SCD+1+TIME_PERIOD++++:5'
ATT+3+5+:::AN..35'
SCD+1+TIME_FORMAT++++:6'
ATT+3+5+:::AN3'
SCD+3+OBS_VALUE++++:7'
ATT+3+5+:::AN..15'
SCD+3+OBS_STATUS++++:8'
ATT+3+5+:::AN1'
ATT+3+35+2:USS'
ATT+3+32+5:ALV'
IDE+1+CL_OBS_STATUS'
SCD+3+OBS_CONF++++:9'
ATT+3+5+:::AN1'
ATT+3+35+1:USS'
ATT+3+32+5:ALV'
IDE+1+CL_OBS_CONF'
SCD+3+OBS_PRE_BREAK++++:10'
ATT+3+5+:::AN..15'
ATT+3+35+1:USS'
ATT+3+32+5:ALV'
SCD+Z09+TITLE'
ATT+3+5+:::AN..70'
ATT+3+35+2:USS' No IDE segment for TITLE, as it is uncoded
ATT+3+32+9:ALV'
SCD+Z09+UNIT'
ATT+3+5+:::AN..4'
ATT+3+35+2:USS'
ATT+3+32+4:ALV'
IDE+1+CL_UNIT'
SCD+Z09+UNIT_MULT'
ATT+3+5+:::AN..2'
ATT+3+35+2:USS'
ATT+3+32+4:ALV'
IDE+1+CL_UNIT_MULT'

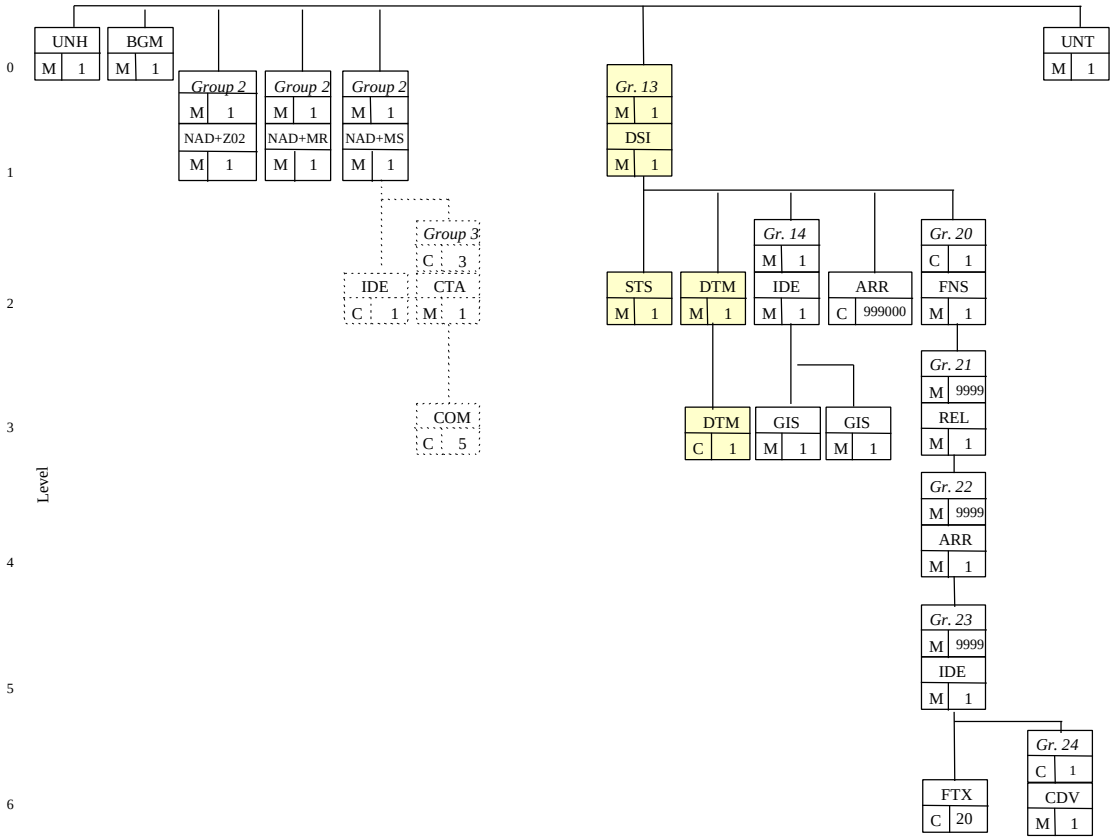
6.7. Data set administration section

Scope

The data set administration section includes the identification of the data set, the action that you must perform (update/replace or delete), and the included data and attribute values extraction date and time.

GESMES segments used

In this branching diagram, the relevant boxes appear shaded, and Group 13 is shown as mandatory.



Guidelines for use of segments

The data set identifier is indicated in the DSI segment. It is always present in messages carrying data and/or attributes (or containing instructions for deletions of data and/or attributes).

You do not use this group in "structural messages" (containing, e.g., key family definitions or code lists). Thus, if the sender uses Group 13, then the message should contain also data (or references to delete data) and/or attributes (or references for deleting attributes).

In the context of GESMES/TS, it has been agreed that:

- an update or delete message can contain data or references to **only one key family** and to **one data set**. Therefore, **the DSI segment can appear only once in a message**;
- **if a message contains Group 13, it should not contain any structural definitions** (Groups 4 to 12 should not be present).

Data set administration section: segment structure

[If you use group 13:]

DSI+*data set-identifier*'	(data set identification: *mandatory*)
STS+*status-type*+*status-code*'	(status of the message: *mandatory*)
DTM+*date-time-type*:*date-time*:*date-time-format*'	(1st DTM - data set extraction time stamp: *mandatory*)
DTM+*date-time-type*:*date-time*:*date-time-format*'	(2nd DTM - reporting period: *conditional*)

***6.7.1.* Segment Group:** **Group 13 (DSI-STS-DTM)**

Max. Number of Occurrences: *1*

Status: *Conditional*

Content and General Usage Rules

In GESMES/TS, you are allowed only one occurrence of this group in a message. Group 13 comprises the segments:

- DSI (data set identification)
- STS (status)
- DTM (first occurrence: date time of data set extraction)
- DTM (second occurrence: reporting period; conditional segment)

When you use this group in a message, this message should also contain:

- Sufficient information (i.e., group 14) for the ARR structure identification
- and
  - observations and/or attributes

    or
  - in case of a "delete message," references to observations and/or attributes to be deleted

6.7.2. Segment Type: DSI

Segment Type Name: *Data Set Identification*

Max. Number of Occurrences: *1*

Status: *Mandatory (trigger)*

Data Element Sequence

DSI+*data set-identifier*'

Data Element Usage Rules

local name	Usage rules	Format	GESMES/TS code values
Data set-identifier	Mandatory	an..18	The data set identifier is specified and agreed between data exchange partners in the context of a data exchange agreement.

Content and General Usage Rules

The data set identifier (DSI) describes the specific statistical context that the included data and/or attribute values belong to. The center administering the data exchange provides the guidelines concerning the use of the data set (see also Box 2 on next page).

Implementation remark

We recommend that you as a partner institution build your database systems and filters with "data set" intelligence. In other words, base your design (when for example you receive data files) on the DSI identifier of the received message.

The same applies for the extraction systems: it is better to base the development of the "export" routines on the concept of *a data set (in general, a subset of series belonging to the same key family) that needs to be extracted* and not on the extraction of all series belonging to the same key family.

One or more data sets (the DSI segment identifies the data set) can follow the structural definitions for the same key family (the key family identifier on the IDE segment has provided this information). See also discussion in Box 2.

Example 1

```
DSI+ECB_BOP1'
```

The reported data belong to the `ECB_BOP1` data set.

Example 2

```
DSI+BIS_MACRO'
```

The reported data in the message belong to the BIS_MACRO data set.

Example 3

```
DSI+EUROSTAT_BOP_Y1'
```

The reported data belong to the EUROSTAT_BOP_Y1 data set.

BOX 2. DATA SET IDENTIFICATION AND DISTINCTION OF SPECIAL REPORTS AND TARGET DATABASES

In GESMES/TS the structure of the data is provided by the key family definition. A *data set* consists of series based on the same key family definition. And a data set *may* comprise *all* time series following the same key family definition; however, in general, data exchange arrangements between institutions may also stipulate the "grouping" of sibling groups and time series into different data sets as indicated in the examples below. Note that such different data sets or "groups" may be mutually exclusive or not (allowing in the latter case also the exchange of data and attribute vintages).

For reasons of simplicity and if no special distinction is needed, a common identifier might be used in the DSI segment (data set identifier) and the IDE segment (key family identifier in Group IDE-GIS-GIS), but this is not mandatory. In general, data exchange partners have to agree on the exact definition of each data set.

Here are some examples of exchanging multiple data sets (i.e. many data set identifiers involved) based on the same key family (i.e. using a single key family identifier):

- The receiver might prefer to receive data sets organised into tables or "reports" (a different message for each "report", e.g. the monthly report with monthly Balance of payments data and the quarterly report with quarterly Balance of payments data. This is an example for mutually exclusive data sets on the time series level.
- Two parties may agree to use the same key family for more than one data exchange projects: e.g. a key family definition may be relevant for two or three economic sub-domains and a "domain" distinction might be needed so that the receiver can forward the data sets to different databases or production units if required.
- There may be a need to exchange vintages of time series and/or attributes compiled using different methodologies, timings or sources.
- A data flow based on a key family is regularly performed. However, based on the same key family definition, an experimental (pilot) data exchange project may be agreed for some time including additional sibling groups. In this case, most probably, the partners would prefer not to mix in the same receiving databases the data flows serving the new pilot data exchange with the ones belonging to the regular production arrangements.
- Quarterly and monthly data sets may be compiled and updated in different timings, leading therefore to non-comparable datasets that may need to be distinguished.

To illustrate the case, let us assume a key family called «MACRO_VARS_KF» which has been defined in such a way that it is used for both national accounts data and public finance data. The receiving institution has two separate production Units, one for national accounts and one for public finance data, and each one of them would like to receive only the message for which it is responsible. We assume that there is a group of reporting series needed to both areas and, thus, there is no way to split them into distinct groups basing the selection addressed to each unit on a "time series key" criterion (the two data sets are not mutually exclusive). The agreement could be to use two different messages as follows:

Message 1

```
…
DSI+MACRO_NAT_ACCS'
…
IDE+5+MACRO_VARS_KF'
…
```

Message 2

```
…
DSI+MACRO_PUB_FIN'
…
IDE+5+MACRO_VARS_KF'
…
```

Thus, the receiving system should recognise and distinguish the incoming messages, forwarding them possibly to the database of the corresponding responsible Unit.

For a discussion on some additional aspects see Appendix V.

6.7.3. Segment Type: STS

Segment Type Name: *Status Report*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

STS+*status-type*+*status-code*'

Data Element Usage Rules

Local name	usage rules	format	GESMES/TS code values
status-type	mandatory this specifies the type of *status-code* used	an..3	3 - data contents
status-code	this specifies the status identified by *status-type*	an..3	7 - update or replace 6 - delete

Content and General Usage Rules

This segment specifies the action that the receiver is expected to perform on the contents of the message. As in GESMES/TS, only one DSI (one reference to a data set) is allowed per message. **A message can be used to instruct the receiver either to perform deletions or to update a receiving database (these two actions cannot be mixed in the same message)**. The action declared by this status code affects the whole message:

- If you have written the segment as **STS+3+7'** then the (update) message should contain either data (in the ARR segment following Group 14) or attributes (in Groups 20-24) or both.
- If you have written the segment as **STS+3+6'** then the (delete) message should contain either references to delete data (in the ARR segment following Group 14) or references to delete attributes (in Groups 20 to 23) or both.

Examples

- STS+3+7'

The receiver should use the contents of the message to update his/her databases and/or to replace previously existing values (assuming that the test indicator of the interchange was not used).

- STS+3+6'

The references in the ARR segments of the message are pointers for deletions that the receiver should perform in his/her databases (of course the receiving end could implement adequate security and control mechanisms, if desirable, in order to check or to prevent fully automated deletions).

***6.7.4.* Segment Type:** **DTM** (1st occurrence: data set extraction time stamp)

Segment Type Name: *Date/Time/Period*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

DTM+*date-time-type*:*date-time*:*date-time-format*'

Data Element Usage Rules

Local name	usage rules	format	GESMES/TS code values
date-time-type	Mandatory	an..3	242 - data set preparation date
date-time	mandatory date and local time of the extraction of the data set; its format is defined by *date-time-format*	an..35	format (in GESMES/TS): CCYYMMDDhhmm
date-time-format	Mandatory	an..3	203 (=format: CCYYMMDDhhmm)

Content and General Rules

In general, use DTM to present dates/time and periods. Use its first occurrence in GESMES/TS to present the data set "preparation date/time" (=242). The date format used is always the same (CCYYMMDDhhmm, which is coded as 203 in EDIFACT). As a sending institution, you should provide in this segment the date/time you extracted the data set reported in the message from the internal database system (local date and time of the sender).

If your previous STS segment indicated that the message is to be used for deletions, then this date/time should be the date/time that you prepared the message.

Example

```
DTM+242:199702241345:203'
```

You as the sender extracted the reported data set from your database at 13:45 (local time) on February 24, 1997.

***6.7.5.* Segment Type:** **DTM** (2nd occurrence: reporting period)

Segment Type Name: *Date/Time/Period*

Max. Number of Occurrences: *1*

Status: *Conditional*

Data Element Sequence

DTM+*date-time-type*:*date-time*:*date-time-format*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
date-time-type	mandatory	an..3	Z02 - reporting period
date-time	mandatory	an..35	Period or range with format depending on the value of the following element.
date-time-format	mandatory	an..3	• for specific periods: 102 - for CCYYMMDD 602 - for CCYY 604 - for CCYYS 608 - for CCYYQ 610 - for CCYYMM 616 - for CCYYWW for period • ranges: 711 - for CCYYMMDD-CCYYMMDD 702 - for CCYY-CCYY 704 - for CCYYS-CCYYS 708 - for CCYYQ-CCYYQ 710 - for CCYYMM-CCYYMM 716 - for CCYYWW-CCYYWW *where:* CC - century YY - year S - half-year (1,2) Q - quarter (1,2,3,4) MM - month in range 01-12 WW - week within a calendar year in the range 1-53 DD - day in month 01-31

Content and General Rules

The second occurrence of DTM is conditional. Use it to declare the reporting period (if relevant and/or requested by a center institution). Of course, if you use this segment, the period or period range you give should be consistent with the contents (data) written in the message.

Example 1

The following segment says that the message contains data for 1992 (single period):

```
DTM+Z02:1992:602'
```

Example 2

The following segment says that the message contains data from 1992 to 1993 (period range):

```
DTM+Z02:19921993:702'
```

Example 3

The following segment says that the message contains data from 1995q1 to 1995q4 (period range):

```
DTM+Z02:1995119954:708'
```

.

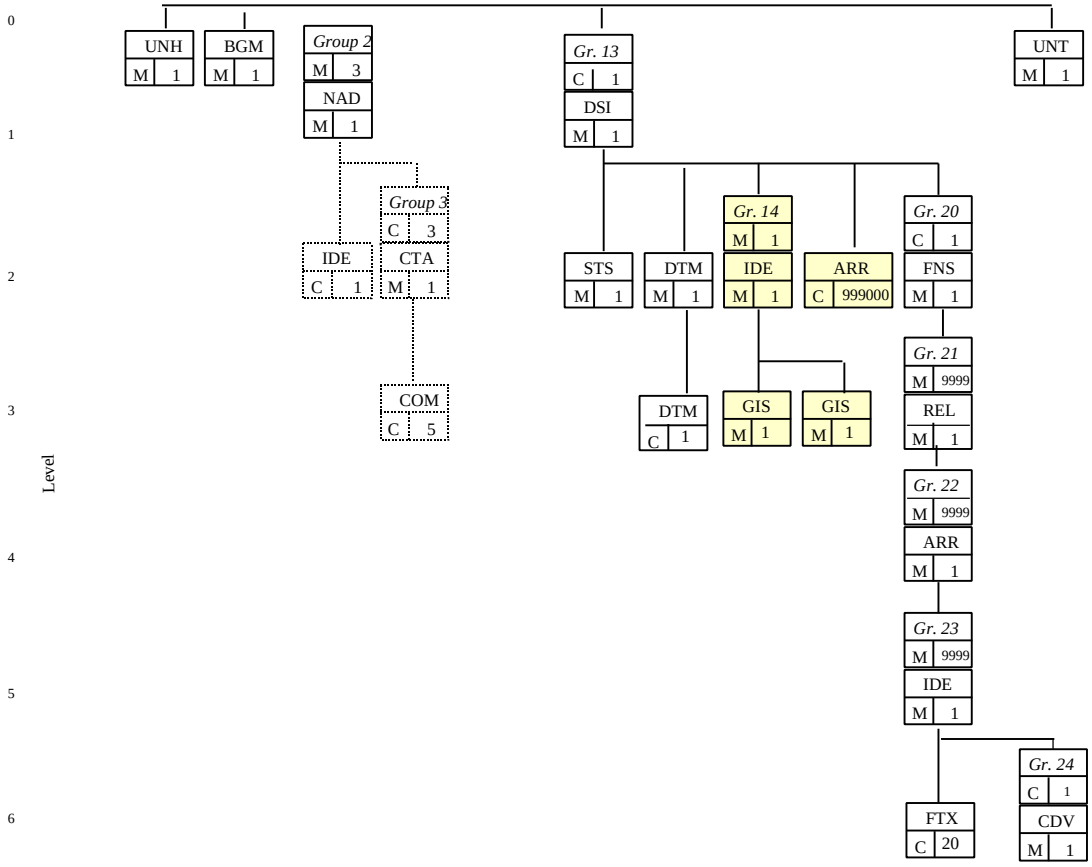
6.8. Array structure and data section

Scope

The array data identify the data set structure (IDE or identity), indicate the method used to place data values in the ARR segment (first occurrence of GIS), indicate the character used for the missing values (second occurrence of GIS), and present the data values (or references for deletions) in the ARR segment.

GESMES Segment Usage

In this branching diagram, the shaded boxes (Group 14 and ARR) are the ones related to the *array structure* and the contained *data* (or references). We show Group 14 and the following ARR segment here as mandatory to indicate that the *array structure and data section* is present.



Features and guidelines for use

The IDE (identity) segment provides the data set structure, defining the expected sequence of data values in the ARR segment. GESMES/TS indicates this by the *key family identifier*.

The GIS segment (first occurrence) indicates which one of the four alternative ways of placing data in the ARR segment has been used *(in GESMES/TS it is always the same: AR3)*. Use the second occurrence of GIS to indicate the missing value character used. In GESMES/TS the ARR segment contains the key, the observations and their corresponding status flags.

Array structure and data section: segment structure

array structure

IDE+*object-type+identity-number'*

GIS+*processing-indicator:list-qualifier:code-maintenance-agency:processing-value'*

(GIS has to be present twice)

array data

ARR++*array-cell-data:array-cell-data:array-cell-data:array-cell-data:array-cell-data:array-cell-data:etc.'*

(The sequence just above is the generic expression; for further information, see p. 123 and the pages that follow.)

6.8.1. Segment Group: Group 14 (IDE-GIS-GIS)

Max. Number of Occurrences: *1*

Status: *Mandatory*

Content and General Usage Rules

Group 14 comprises the segments:

- IDE - array structure identification
- GIS - message processing indicator
- GIS - missing value indicator

This group *must be present* in all messages containing or referring to data and/or attributes. (This includes the case of a pure attribute update or a delete message, in which case the ARR segment, which follows Group 14, would *not* be present).

6.8.2. Segment Type: IDE

Segment Type Name: *Identity*

Max. Number of Occurrences: *1*

Status: *Mandatory (trigger)*

Data Element Sequence

IDE+*object-type*+*identity-number*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
object-type	Mandatory	an..3	5 - data set structure
identity-number	Mandatory the identity of the data set structure	an..35	the key family identifier

Content and General Rules

Use this segment to indicate the structure of the data set (e.g., how many dimensions, which ones, and in which order). The key-family identifier provides this information.

.

Example 1

```
IDE+5+ECB_BOP1'
```

Example 2

```
IDE+5+BIS_MACRO'
```

6.8.3. Segment Type: GIS (first occurrence)

Segment Type Name: *General Indicator*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

GIS+*processing-indicator*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
processing indicator	mandatory	an..3	AR3 - ARR usage method 3

Content and General Usage Rules

Use this segment (first occurrence) to define how you use the data elements in the ARR segment. *In GESMES/TS it is always AR3.*

Example

```
GIS+AR3'
```

In GESMES/TS, you must write the segment in this fixed format.

6.8.4. Segment Type: GIS (second occurrence)

Segment Type Name: *General Indicator*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

GIS+*processing-indicator*::: *processing-value*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
processing indicator	Mandatory	an..3	1 - no data available
processing-value	mandatory	an..17	- dash; the symbol for missing values in GESMES/TS

Content and General Usage Rules

Use this segment (second occurrence) in GESMES/TS to define which character you are using for the missing values in the ARR segment. The reading applications should read this character, parameterise it, and, based on this, recognize the missing values in the ARR segment.

Example

```
GIS+1:::-'
```

In GESMES/TS you must write this segment in this fixed format.

6.8.5. Segment Type: ARR

Segment Type Name: *Array Information*

Max. Number of Occurrences: *999000*

Status: *Conditional*

Data Element Sequence in GESMES/TS

- *"Single observation" technique:*

ARR++*series-key:period:time-format:observation:observation-status:obs.-confidentiality: obs-pre-break*'

ARR++*series-key:period:time-format:observation:observation-status:obs.-confidentiality: obs-pre-break*'

(repeats)

- *"Time range" technique:*

ARR++*series-key:period-range:time-format:observation:observation-status:obs.-confidentiality:obs-pre-break+*
observation:observation-status:obs.-confidentiality:obs-pre-break +
etc.
observation:observation-status:obs.-confidentiality:obs-pre-break'

ARR++*series-key:period-range:time-format:observation:observation-status:obs.-confidentiality:obs-pre-break +*
observation:observation-status:obs.-confidentiality:obs-pre-break+
etc.
observation:observation-status:obs.-confidentiality:obs-pre-break'

(repeats)

Data Element Usage Rules

Local name	usage rules	Format	GESMES/TS code values
Series-key	mandatory the key of the reported time series; the rules are detailed below	component elements separated by component separators	dimension values of the time series key delimited by component element separators
Period (or period-range)	mandatory the period (or period range) to which the reported observation(s) in this ARR segment correspond; the rules are detailed below	an..35	period or range of periods written in one of the foreseen formats. See Table 8. period and period range format codes
Time-format	mandatory the rules are detailed below	an3	the code list for these values is presented in the table below
Observation	mandatory for updating [not used in "delete messages"] the rules are detailed below	an..15	a numeric observation or a missing value indicator

Observation-status	mandatory for updating [not used in "delete messages"] the rules are detailed below	an..35	A value taken from an observation status code list (e.g. CL_OBS_STATUS with format an1)
Observation-confidentia-lity	Conditional [not used in "delete messages"] the rules are detailed below	an..35	A value from an observation confidentiality code list (e.g. CL_OBS_CONF with format an1)
Pre-break-observation	Conditional [not used in "delete messages"]	an..15	A numeric observation or a missing value indicator

The following rules and explanations refer only to reporting data (updating actions). We'll discuss the role and the use of the ARR segment in the deletions section later in this guide.

Content and general usage rules when reporting data *(for deletions see p.153)*

- **Time series key**

A time series key is composed of values given to the (ordered) dimensions included in the key structure of a key family. To report the dimensions, separate them by the component separator ":"(colon), e.g.:

`Q:BE:100:4`.

In each ARR segment, you can have only one time series key present (just after the ARR++ characters), regardless of the technique you use (single observation or time range). Within one message, all the time series keys that you use in the ARR segment *must* belong to the same key family.

- **Periods, period ranges, time format and time format code list**

Always give the period and time format as a "pair" separated by the component separator ":" (colon).

Example: *single monthly period:* `199505:610` (= May 1995)
period range (quarterly): `1994119962:708` (= 1994 Q1 to 1996 Q2)

The table below indicates the formats for period specifications and the time format codes.

Table 8. period and period range format codes

Types for dates and specific periods			
Code	**format**	**explanation**	**dates and specific periods: examples**
203	CCYYMMDDhhmm	(for series with freq. higher than daily) year/month/day/hours/minutes	199511210850 = 21 Nov. 1995 (08:50) *date & format code:* 199511210850:203
102	CCYYMMDD	(for daily and business series) year/month/day	19951121= 21 Nov. 1995 *date & format code:* 19951121:102
616	CCYYWW	(for weekly series) year/ week	199252= 52nd week of 1992 *date & format code:* 199252:616
610	CCYYMM	(for monthly series) year/month	199511= Nov. 1995 *date & format code:* 199511:610
608	CCYYQ	(for quarterly series) year/quarter	19953=1995q3 *date & format code:* 19953:608
604	CCYYS	(for half yearly series) year/semester	*date & format code:* 19951:604
602	CCYY	(for annual series) year	*date & format code:* 1995:602
Types for period ranges			
Code	**format**	**explanation**	**period ranges: examples**
711	CCYYMMDD CCYYMMDD	(for daily and business series) from year/month/day to year/month/day	1992110419930124=4Nov92-24Jan1993 *date & format code:* 1992110419930124:711

716	CCYYWW CCYYWW	(for weekly series) from year/week to year/week	199227199550= from week no.27 of 1992 to week no.50 of 1995 *date & format code:* 199227199550:716
710	CCYYMM CCYYMM	(for monthly series) from year/month to year/month	199208199511= from Aug92 to Nov95 *date & format code:* 199208199511:710
708	CCYYQCCYYQ	(for quarterly series) from year/quarter to year/quarter	1992319954= from 1992q3 to 1995q4 *date & format code:* 1992319954:708
704	CCYYSCCYYS	(for half yearly series) from half-year to half-year	1995219961= from 1995h2 to 1996h1 *date & format code:* 1995219961:704
702	CCYYCCYY	(for annual series) from year to year	19951996= from 1995 to 1996 *date & format code:* 19951996:702

- **Data elements: observation and observation level attributes**

After you indicate the time series key, time period, and format, you report the actual observations as data elements. Each data element consists of at least 2 components: the **observation** itself and the mandatory (observation-level) attribute "**Observation status.**" Depending on the key family, you may add a third and fourth component element: the (observation-level) attribute "**Observation confidentiality**" and "**Pre-break observation.**" Within the data element, separate the components by the component separator ":" (colon).

Example:

Data element with 2 components: `1234.5:A`
(only observation status)

Data element with 3 components: `1234.5:A:C`
(observation status + confidentiality)

Data element with 4 components: `1234.5:B::1230.5`
(obs. status, confidentiality is not given and pre-break observation)

Data element with 4 components: `1234.5:B:C:1230.5`
(observation status, confidentiality and pre-break observation)

You must separate the *time format indication* and the *first data element with an observation* with a component separator ":". Then separate the *individual data elements* that follow by the data element separator "+" (plus sign). The plus sign follows every data element except for the *last* one: Owing to the GESMES truncation rules, the data-element separator that follows the last data element in the segment should be omitted and replaced by the segment terminator sign (', single quote).

The maximum number of data elements in an ARR segment is 9,999 (i.e., the overall maximum number of data element separators ("+") is 10,000 including the 2 immediately following the ARR statement). If a time series has more than 9,999 observations, you must split it over more than 1 ARR segment.

We recommend that applications use the maximum number of data elements as a parameter.

Example: ...+`1234.5:A+234.6:B+3456.7:A'` (3 data elements followed by segment terminator)

- **Rules, special cases, and recommendations concerning the handling of observation elements**

  ⇒ In GESMES/TS this **maximum length** of the component element holding the observation is 15 positions. As a consequence, the maximum number of significant figures for an observation is:

  ⇒ 15 for a positive integer
  ⇒ 14 for a positive decimal or a negative integer
  ⇒ 13 for a negative decimal

  ⇒ Use the point (".") *exclusively* as **decimal separator**.

  ⇒ You may use **scientific notation** for both the observation and the observation pre-break, if the parties involved agree. To express a number in scientific notation[1]:

  - omit any plus signs, because "+" is a reserved character in GESMES and may be misinterpreted;
  - use "E" before the exponent;
  - the value of the number before the "E" should be between -10 and 10; and
  - to express numbers between minus one and one, preface the exponent with a negative sign.

  Examples: The number 1,230,000 is written 1.23E6. The number 0.000001 is written as 1.0E-6. Again, in GESMES/TS, the total length of the field should not exceed 15 characters.

  ⇒ Pre-break observations do not have an observation status.

  ⇒ Indicate a **missing value** using a dash ("-") (e.g. `...+-:H+...`). The observation status that follows will explain the absence of the value. (Parameterise the applications for this "missing observation" character, taking its value from the second occurrence of the GIS segment.)

  ⇒ If there are **unchanged or nonreported values** within a reported time range, you must still write the corresponding separators. Therefore, a sequence "++" means that the observation, which would be expected between the two plus signs, is considered either as not reported or as unchanged vis-à-vis the last reported value (see also examples below when discussing the time-range technique).

  ⇒ **You must always report observations together with an observation status value**. If an observation changes and the status is not changed, report both components. Likewise, if the observation status changes and the observation remains unchanged, you would have to re-report both components. The same would apply also if the observation confidentiality status changed (relevant for the key families in which this attribute might be used); you would have to re-report all three elements.

[1] For more information on the rules applying see http://www.ex.ac.uk/cimt/dictunit/enote.htm (section on "E-format").

Similarly, if the observation confidentiality status or the pre-break observation changed, you would have to re-report all component elements. If you as a sender do not report, for a certain observation, an observation confidentiality attribute or an observation pre-break value, then the receiver should assume that for this period, values (for the confidentiality status and pre-break) never existed or they cease to exist from now on.

Example 1:

On March 10, 1999, the sender reported the segment:

```
ARR++M:YY:ZZ:199902:610:-7.9:E:C'
```

On March 15, 1999, the sender reported the segment:

```
ARR++M:YY:ZZ:199902:610:-7.9:A:F'
```

The second time the sender resent the observation, because the status (from Estimate to normal) and the observation confidentiality (from "Confidential" to "Free") were modified.

Example 2:

On March 10, 1999, the sender reported the segment:

```
ARR++M:YY:ZZ:199902:610:-7.9:E:C'
```

On March 15, 1999, the sender reported the segment:

```
ARR++M:YY:ZZ:199902:610:-7.9:A'
```

The second time the sender sent the segment without the observation confidentiality value; in this case, the receiver would need to decide how to interpret the fact that the confidentiality attribute was not reported. He could either delete the previously existing value (C) or leave it unchanged (i.e., continue to mark the observation as confidential).

Example 3:

On April 10, 1999, the sender reported the segment:

```
ARR++M:FG:T1:199902:610:10:B:F:12'
```

On April 15, 1999, the sender reported the segment:

```
ARR++M:FG:T1:199902:610:10:A:F'
```

Here, we assume that in the first transmission (April 10, 1999) the sender reported a break in series together with a pre-break value. In the second transmission, we now assume that the sender revised all back data, built a series without breaks, and re-reported the same observation without the "break" flag and without the pre-break observation.

- **Time range and single observation technique**

⇒ With the ***time-range technique***, the sender sets the period and time format specification in the ARR segment to define a range of periods. This is then followed by a number of data elements, which have to exactly match the number of periods in the defined time range.

Also, as we discussed earlier (see paragraph describing the data elements of the ARR segment):

- no more than 9,999 observations can be reported in the same ARR segment; and
- **you should explicitly write the first and the last observation of the range in the segment**: In this way, the number of observations reported in an ARR segment should always equal the number of periods indicated by the time range.

Example 1. - time range technique

```
ARR++M:YY:ZZ:199301199304:710:39.9:A+21.5:A+23.4:A+43.0:E'
```

This segment contains four observations (with their corresponding status) for the series M:YY:ZZ. The observation 39.9 (status: A) is for Jan.93, 21.5 (A) is for Feb.93, 23 (A) for Mar. 93, and the observation 43.0 (E) is for Apr.93.

⇒ With the ***single-observation technique***, you set the period and time format specification to define a single time period, which is then followed by a single component data element for the observation and its attribute(s).

Example 2. - single period technique

```
ARR++M:YY:ZZ:199301:610:39.9:A'
ARR++M:YY:ZZ:199302:610:21.5:A'
ARR++M:YY:ZZ:199303:610:23.4:A'
ARR++M:YY:ZZ:199304:610:43.0:E'
```

These four segments contain one observation each, and they are equivalent to the single segment shown in the previous example.

⇒ ***You can mix the two techniques in the same message*** (but, of course, not within the same ARR segment):

Example 3.

The following set of segments is valid (and equivalent to the ones shown in examples 1 & 2):

```
ARR++M:YY:ZZ:199301:610:39.9:A'
ARR++M:YY:ZZ:199302199304:710:21.5:A+23.4:A+43.0:E'
```

and the following one (example 4) is also valid and an equivalent form of the previous set of segments:

Example 4.

```
ARR++M:YY:ZZ:199301199302:710:39.9:A+21.5:A'
ARR++M:YY:ZZ:199303:610:23.4:A'
ARR++M:YY:ZZ:199304:610:43.0:E'
```

The examples below (5, 6, and 7) show the use of the two techniques when "reporting" **unchanged or nonreported observations** and their flags (we discussed it theoretically on a previous page). Of course,

it makes sense to refer to reporting of *unchanged* observations only when using the ***time-range technique***.

Example 5

We assume that you as an institution have to report for the series M:YY:ZZ only data for Nov.92, Jan.93, and Apr.93, since you revised or updated only these data. (The observation for April 1993 is an estimate and should be treated as confidential.) For Dec92, Feb93, and Mar93, the data that you had reported in the past remain still valid:

Series: `M:YY:ZZ`

Nov92	Dec92	Jan93	Feb93	Mar93	Apr93
-7.9 (A)	unchanged	37.8 (A)	unchanged	unchanged	43 (E, C)

When you use the time-range technique, you should explicitly write all separators (including the ones referring to the unchanged observations). Then, you report the sequence above as:

```
ARR++M:YY:ZZ:199211199304:710:-7.9:A++37.8:A+++43:E:C'
```

Example 6

The following three segments, written using the single-observation technique, are equivalent to the one presented just above (example 4):

```
ARR++M:YY:ZZ:199211:610:-7.9:A'
ARR++M:YY:ZZ:199301:610:37.8:A'
ARR++M:YY:ZZ:199304:610:43:E:C'
```

Example 7

In the following segments, we show another way to report the same data by mixing the two techniques:

```
ARR++M:YY:ZZ:199211199301:710:-7.9:A++37.8:A'
ARR++M:YY:ZZ:199304:610:43:E:C'
```

⇒ ***Practical application of the two techniques***

In general, because it eliminates redundant information, the *time-range technique* is the preferred method for exchanging complete time series or databases. (Sample instances would be when an institution first reports a new series to a center institution or when a central institution disseminates a full database to other institutions).

The *single-observation technique* is simple and efficient for reporting a single observation per series, for example in the updates and revisions context. It can be useful also in the case of series with sparse irregular observations. For instance, for some series with only some observations per year, it is sensible to report only these observations (e.g., using daily frequency with the single-observation technique). Usually, this is a matter that the center will administer (e.g., if the observations are not too sparse, it

might be preferable to keep a continuous track of the exchanged series using missing values in conjunction with the "cannot-exist" observation status).

In general, both the *time range* and the *single observation* techniques are equally valid.

- **The use of the pre-break value attribute: examples**

  The following examples correspond to Example 1 and 2 previously shown. The difference is that here they also include a pre-break value.

  Example 1. - time range technique

  ```
  ARR++M:YY:ZZ:199301199304:710:39.9:A+21.5:B::20.1+23.4:A+43.0:E'
  ```

  The observation for Feb.93, which is a break, includes also ***a pre-break-observation 20.1***.

  Example 2. - single period technique

  ```
  ARR++M:YY:ZZ:199301:610:39.9:A'
  ARR++M:YY:ZZ:199302:610:21.5:B::20.1'
  ARR++M:YY:ZZ:199303:610:23.4:A'
  ARR++M:YY:ZZ:199304:610:43.0:E'
  ```

  The same as before, using the single-period technique.

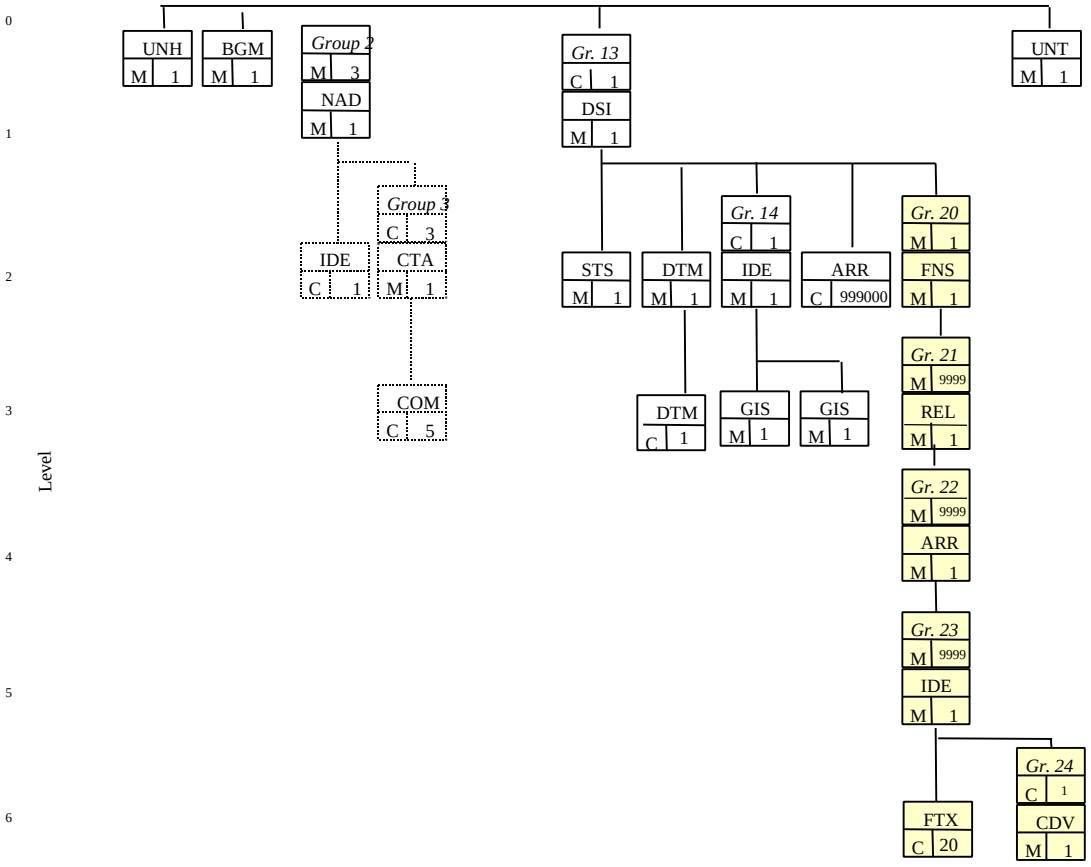
6.9. Attribute section

Scope

An attribute is an object relating to one of the following: (1) an observation, (2) a time series, (3) a group of sibling series, or (4) a data set. It provides information about the corresponding data (at that level), such as the measurement unit or the confidentiality status. An attribute can be coded (taking values from a code list) or uncoded (free text). The purpose of the segments included in the Attributes section is to allow you to exchange attributes at the levels foreseen by the data model.

GESMES segment usage

The Attributes section comprises all groups from Group 20 to 24 (they are shown shaded in the diagram below). You must use these groups in order to exchange attributes (that is why Group 20 is shown here as mandatory). Regardless whether the message carries observations as well (in which case you would also use the ARR segment that follows Group 14), you need to use Groups 13 and 14, because they provide information about the relevant data set and key family.

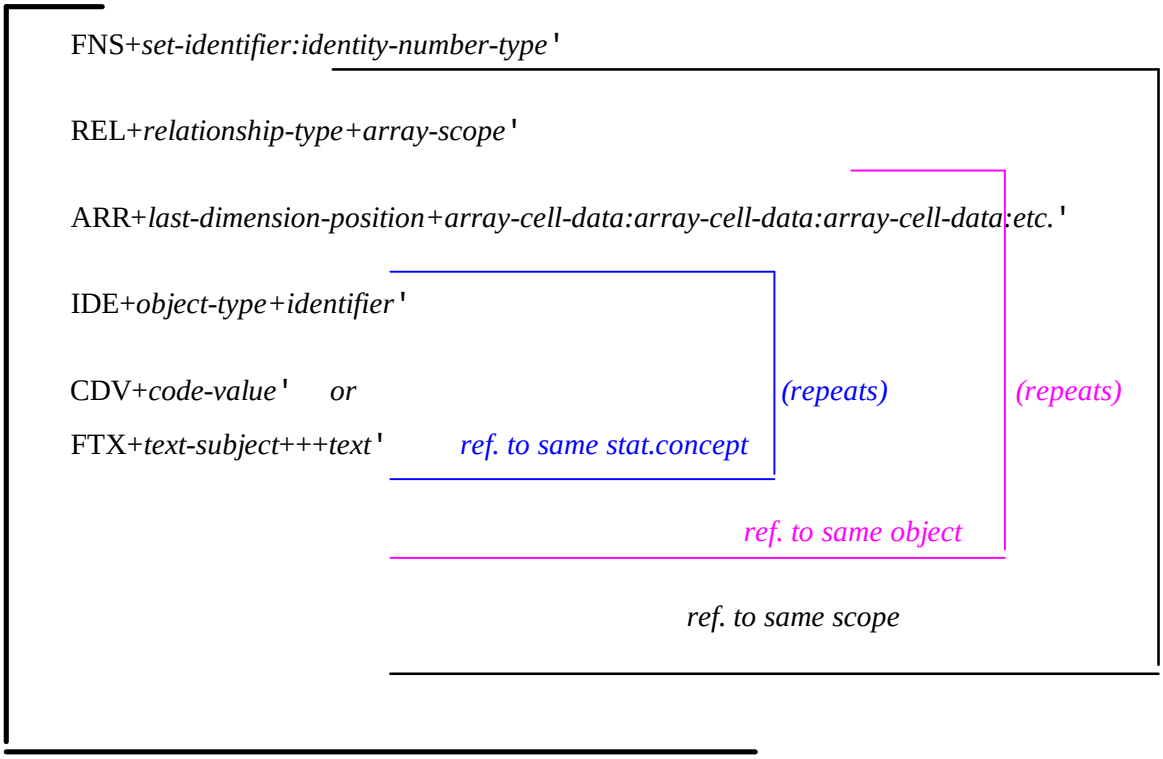


Guidelines for usage

Always use Group 20 in the Data and Attribute message (or in an "attribute only" update message). Use it also when you need to delete specific attributes.

- Include all the attributes in a set introduced by the FNS segment.
- Use the REL segment to define the scope of the attribute in terms of the object type to which the attributes relate.
- Use the ARR segment to define the statistical objects to which the attributes relate.
- Use the IDE segment to identify the statistical concept.
- If the statistical concept is coded, then use the CDV segment to provide its value; and
    - If it is uncoded, then use the FTX segment to contain the textual value.

Attribute section: segment structure



6.9.1. **Segment Group:** **Group 20 (FNS-Group 21)**

Max. Number of Occurrences: *1*

Status: *Conditional*

Content and General Usage Rules

This group contains a set of attributes relevant to the specified data set. It follows the specified key family definition in use throughout this message.

We have built up the **examples** used for the segments relating to the FNS group on attributes in such a way that with each added segment you retain the overall view of the whole group. We then present the completed examples after the description of the CDV segment (Group 24).

6.9.2. Segment Type: FNS

Segment Type Name: *Footnote Set*

Max. Number of Occurrences: *1*

Status: *Mandatory (trigger)*

Data Element Sequence

FNS+*set-identifier:identity-number-type*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
set-identifier	mandatory the identity of the set of attributes	an..35	(any text) e.g.: Attributes
identity-number-type	mandatory the object is “attribute” (=10)	an..3	10 - Attributes

Content and General Usage Rules

Use this segment to identify the attribute set. In principle, the reading applications will use the FNS segment to confirm that an attribute section starts, but they can ignore the *set identifier* given in the segment (it is enough to acknowledge the existence of the FNS segment).

Example

```
FNS+Attributes:10'
```

In the examples that follow in the next few subsections, we do not show the FNS segment again, because it has a constant format.

6.9.3. **Segment Group:** **Group 21 (REL-Group 22)**

Max. Number of Occurrences: *9999*

Status: *Conditional*

Content and General Usage Rules

This group specifies the scope of the attribute(s) (REL), followed by the attributes (group 22).

6.9.4. Segment Type: REL

Segment Type Name: *Relationship*

Max. Number of Occurrences: *1*

Status: *Mandatory (trigger)*

Data Element Sequence

REL+*relationship-type*+*array-scope*'

Data Element Usage Rules

Local name	usage rules	format	GESMES/TS code values
relationship-type	Mandatory	an..3	Z01 - relationship is with a statistical array
array-scope	Mandatory this provides information about the scope of the attributes which follow; the attributes can refer to: - the data set identified in the DSI segment (=1); - series or sibling series level (=4); - observation level (=5)	an..3	1 - data set. 4 - specific combination of dimension values 5 – observation

Content and General Usage Rules

Use this to provide the scope of the footnote(s) or attribute(s).

Dependencies

The *array-scope* that you give here should be consistent with the contents of the Group 22-ARR segment that follows (until a new REL is found).

If *array-scope=4*, then the ARR segments (which follow the REL segment) should refer to specific dimension values—they should identify either a time series or a sibling group (in the latter case, the frequency would be wildcarded).

If *array-scope=5*, then the references for the attributes will indicate attachments at the observation level (all dimensions plus time reference plus time-format).

If *array-scope=1*, then the ARR segments that follow should not contain dimension values, but all dimensions should be wildcarded (attributes attached at the data set level).

Example 1 (attribute for a data set)

```
DSI+ECB_TEST'          DSI identifies the data set ECB_TEST
. . .
```

`IDE+5+ECB_TEST_KF'`	Key family: ECB_TEST_KF
`. . .`	
`REL+Z01+1'`	**Relationship is with the data set**
`...`	ARR segment to follow

Example 2 (attributes for sibling series)

`REL+Z01+4'`	**Array scope is "combination of dimension values"**
`...`	ARR segment to follow

Example 3 (attributes for a time series)

`REL+Z01+4'`	**Array scope is "combination of dimension values"**
`...`	ARR segments to follow

Example 4 (attribute for an individual observation)

`REL+Z01+5'`	**Array scope is "observation"**
`...`	ARR segments to follow

***6.9.5.* Segment Group:**	**Group 22 (ARR-Group 23)**
Max. Number of Occurrences:	*9999*
Status:	*Conditional*

Content and General Usage Rules

This group contains an ARR segment. It defines the statistical objects that the attributes refer to. Then, the name (identifier) of the attribute follows (in an IDE segment), and its value follows. It can be either:

- text (FTX segment) or
- a value from a code list (group 24, CDV)

The examples that we present in this subsection include also relevant segments (e.g., REL). They should precede the appearance of the segments of the group.

6.9.6. Segment Type: ARR

Segment Type Name: *Array Information*

Max. Number of Occurrences: *1*

Status: *Mandatory (trigger)*

Data Element Sequence

ARR+*last-dimension-position+array-cell-data:array-cell-data:array-cell-data:etc.*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
last-dimension-position	Mandatory this element is used to indicate the number of dimensions that are relevant for the attachment of the attribute: (a) if the attribute qualifies a sibling group or a time series, then the *position* indicates the number of dimensions in the key family. (b) if the attribute qualifies an observation, then the *position* should indicate the number of dimensions in the key family increased by two (additional dimensions: period, time format). (c) if the attribute qualifies the whole data set (given by the data set-identifier in the DSI segment) , then the *position* should be set equal to zero.	an..12	This value can be: (a) the number of dimensions in the key family (for attributes at sibling group or time series level) (b) the number of dimensions in the key family increased by two (for the OBS_COM attribute) (c) 0 - for attributes attached at the data set level (as identified by the DSI segment).
Array-cell-data	Conditional this contains dimension values which specify uniquely the position in the array to which the attribute relates: - wildcarding of one dimension is valid for the sibling group level - if *array-cell-data* refers to the time dimension value (the case of an observation), then both the period and periodicity must be specified in the sequence `period:time format`	an..35	see examples; (not used for attributes attached at the data set level)

Content and General Usage Rules

Use this segment to specify the values in the data set for which the attribute(s) are relevant.

For attributes at the observation level, the *last-dimension-position* is the number of dimensions plus two. The reason for this is that the conceptual dimension time (needed in order to address the observation level) is represented in GESMES by two physical dimensions: period and time format (see also examples).

For attributes at the data set level, the *last-dimension-position* must be set equal to 0. This is the method to wildcard all dimension values.

This ARR segment plays an important role also in deleting attributes (see p. 153).

Example 1 (attribute for a data set)

`DSI+ECB_TEST'`	It identifies the data set ECB_TEST
`. . .`	
`IDE+5+ECB_TEST_KF'`	Key family: ECB_TEST_KF
`. . .`	
`REL+Z01+1'`	Relationship is with the data set
`ARR+0'`	**Attached at the data set level**
`...`	IDE segments to follow

Example 2 (attributes for sibling series)

`REL+Z01+4'`	Array scope is "combination of dimension values"
`ARR+4+:XX:ZZ:CC'`	**attached to sibling group :XX:ZZ:CC**
`...`	IDE segments to follow

Example 3 (attributes for a time series)

`REL+Z01+4'`	Array scope is "combination of dimension values"
`ARR+4+M:XX:ZZ:CC'`	**attached to time series M:XX:ZZ:CC**
`...`	IDE segments to follow

Example 4 (attribute for an individual observation)

`REL+Z01+5'`	Array scope is "observation"
`ARR+6+M:XX:ZZ:CC:199606:6`	**attached to obs. of June 1996 for series M:XX:ZZ:CC**
`10'`	
`...`	IDE segments to follow

These attribute segments refer to the observation for June 1996 of the series `M:XX:ZZ:CC`.
It is obvious that in this case we have to write ARR+6, because apart from the key family dimensions (=4) the reading applications should expect *two more dimension references.* We need these to refer to a specific period (the "`199606`" -date component- and the "`610`" -format component-). **This is an important point for you to take into account when you design both the *writing* and *reading* applications**.

6.9.7. **Segment Group:**	**Group 23 (IDE-FTX-Group 24)**
Max. Number of Occurrences:	*9999*
Status:	*Conditional*

Content and General Usage Rules

Use each occurrence of this group to identify an attribute relating to the object that you specified in the preceding ARR segment. The attribute present in the IDE segment can be assigned as value either:

- as a text (FTX segment) for an uncoded attribute or
- as a value from a code list (group 24, CDV) for a coded attribute

6.9.8. Segment Type: IDE

Segment Type Name: *Identity*

Max. Number of Occurrences: *1*

Status: *Mandatory (Trigger)*

Data Element Sequence

IDE+*object-type*+*identifier*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
Object-type	mandatory It indicates the type of the object	an..3	Z10 - coded attribute Z11 - uncoded attribute
Identifier	mandatory the identity of the object	an..35	attribute identifier

Content and General Rules

Use each occurrence of this segment to identify an attribute related to the object that you specified in the preceding ARR segment.

The *identifier* in this segment points either

- to the *text* in the FTX segment or
- to the *code-value* in Group 24 (CDV)

which follow.

Example 1 (attribute for a data set)

DSI+ECB_TEST'	It identifies the data set ECB_TEST
. . .	
IDE+5+ECB_TEST_KF'	Key family: ECB_TEST_KF
. . .	
REL+Z01+1'	Relationship is with the data set
ARR+0'	attached at the data set level
IDE+Z10+UNIT'	**Coded attribute UNIT**

`...`	CDV segment to follow

Example 2 (attributes for sibling series)

`REL+Z01+4'`	Array scope is "combination of dimension values"
`ARR+4+:XX:ZZ:CC'`	attached to sibling `:XX:ZZ:CC`
`IDE+Z10+AVAILABILITY'`	**Coded attribute AVAILABILITY**
`...`	CDV segment to follow
`IDE+Z11+TITLE'`	**Uncoded attribute TITLE**
`...`	FTX segment to follow

Example 3 (attributes for a time series)

`REL+Z01+4'`	Array scope is "combination of dimension values"
`ARR+4+M:XX:ZZ:CC'`	attached to time series `M:XX:ZZ:CC`
`IDE+Z10+COLLECTION'`	**Coded attribute COLLECTION**
`...`	CDV segment to follow
`IDE+Z11+COLL_DETAIL'`	**Uncoded attribute COLL_DETAIL**
`...`	FTX segment to follow

Example 4 (attribute for an individual observation)

`REL+Z01+5'`	Array scope is "observation"
`ARR+6+M:XX:ZZ:CC:199606:610'`	attached to obs. of June 1996 for series `M:XX:ZZ:CC`
`IDE+Z11+OBS_COM'`	**uncoded attribute OBS_COM**
...	FTX segment to follow

***6.9.9.* Segment Type:** **FTX**

Segment Type Name: *Free Text*

Max. Number of Occurrences: *20*

Status: *Conditional*

Data Element Sequence

FTX+*text-subject*+++*text*:*text*:*text*:*text*:*text*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
text-subject	mandatory	an..3	ACM - statistical description
Text	mandatory	an..70	text value (of an uncoded attribute); it could continue with up to four more component element (up to five in total).
text (2nd comp.element)	conditional	an..70	(text could continue)
text (3rd comp.element)	conditional	an..70	(text could continue)
text (4th comp.element)	conditional	an..70	(text could continue)
text (5th comp.element)	conditional	an..70	(text could continue only using a new FTX segment)

Content and General Usage Rules

Use each occurrence of this segment to provide the text value of an uncoded attribute. The overall allowed length of the text depends on how the key family definition specified the respective attribute. If you foresee the length to be longer than 70 characters (upper limit for original text inside a single-component element, not counting release characters), then you should split the *text* in (the maximum) 5 component-element texts. (Each one of them will be no longer than 70 characters and separated with a *component separator*).

The reading applications should read the components of the received text and consider that they form a string, which is the result of the added sub-strings. The writing applications should break the original text into pieces of 70 characters or shorter (it does not matter whether a word is broken between two component elements or not), keeping spaces as significant characters and adding release characters as required afterwards.

You can send multiple FTX segments (up to 20, one below the other), and the reading applications should consider them as an ordered sequence of sections of the same text. As before, your use of a new FTX does not imply that the receiver should interpret this as *line feed*.

The free text in FTX should respect the EDIFACT rules for treating the characters you use in the syntax (see also separate section on using text in uncoded attributes).

Example 1 (coded attribute for a data set)

`DSI+ECB_TEST'`	It identifies the data set ECB_TEST
`. . .`	
`IDE+5+ECB_TEST_KF'`	Key family: ECB_TEST_KF
`. . .`	
`REL+Z01+1'`	Relationship is with the data set
`ARR+0'`	attached at the data set level
`IDE+Z10+UNIT'`	Coded attribute UNIT
`...`	CDV segment to follow

Example 2 (attributes for sibling series)

`REL+Z01+4'`	Array scope is "combination of dimension values"
`ARR+4+:XX:ZZ:CC'`	attached to sibling `:XX:ZZ:CC`
`IDE+Z10+AVAILABILITY'`	Coded attribute AVAILABILITY
`...`	CDV segment to follow
`IDE+Z11+TITLE'`	Uncoded attribute TITLE
`FTX+ACM+++MONETARY AGGREGATE M1'`	**Attribute text value**

Example 3 (attributes for a time series)

`REL+Z01+4'`	Array scope is "combination of dimension values"
`ARR+4+M:XX:ZZ:CC'`	attached to time series `M:XX:ZZ:CC`
`IDE+Z10+COLLECTION'`	Coded attribute COLLECTION
`...`	CDV segment to follow
`IDE+Z11+COLL_DETAIL'`	Uncoded attribute COLL_DETAIL
`FTX+ACM+++COLLECTED ON 2 LAST WORKING DAYS OF THE MONTH'`	**Attribute text value**

Example 4 (attribute for an individual observation)

`REL+Z01+5'`	Array scope is "observation"
`ARR+6+M:XX:ZZ:CC:199606:610'`	attached to obs. of June 1996 for series `M:XX:ZZ:CC`
`IDE+Z11+OBS_COM'`	Uncoded attribute OBS_COM
`FTX+ACM+++NEW ACCOUNTING METHOD'`	**Attribute text value**

6.9.10. **Segment Group:** **Group 24 (CDV)**

Max. Number of Occurrences: *1*

Status: *Conditional*

Content and General Usage Rules

This group provides the value of a coded attribute.

6.9.11. **Segment Type:** **CDV**

Segment Type Name: *Code value*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

CDV+*code-value*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
Code-value	mandatory a code value	an..18	attribute code value

Content and General Rules

Use each occurrence of this segment to assign a value to the attribute specified in the preceding IDE segment.

Example 1 (attribute for a data set)

DSI+ECB_TEST'	It identifies the data set ECB_TEST
. . .	
IDE+5+ECB_TEST_KF'	Key family: ECB_TEST_KF
. . .	
REL+Z01+1'	Relationship is with the data set
ARR+0'	Attached at the data set level
IDE+Z10+UNIT'	Coded attribute UNIT
CDV+USD'	**code value for US dollar**

Example 2 (attributes for sibling series)

REL+Z01+4'		Array scope is "combination of dimension values"
ARR+4+:XX:ZZ:CC'		attached to sibling :XX: ZZ:CC
IDE+Z10+AVAILABILITY'		Coded attribute AVAILABILITY
CDV+A'		**code value: A** (=assumed to mean availability="free")
IDE+Z11+TITLE'		Uncoded attribute TITLE
FTX+ACM+++MONETARY M1'	AGGREGATE	attribute text value

Example 3 (attributes for a time series)

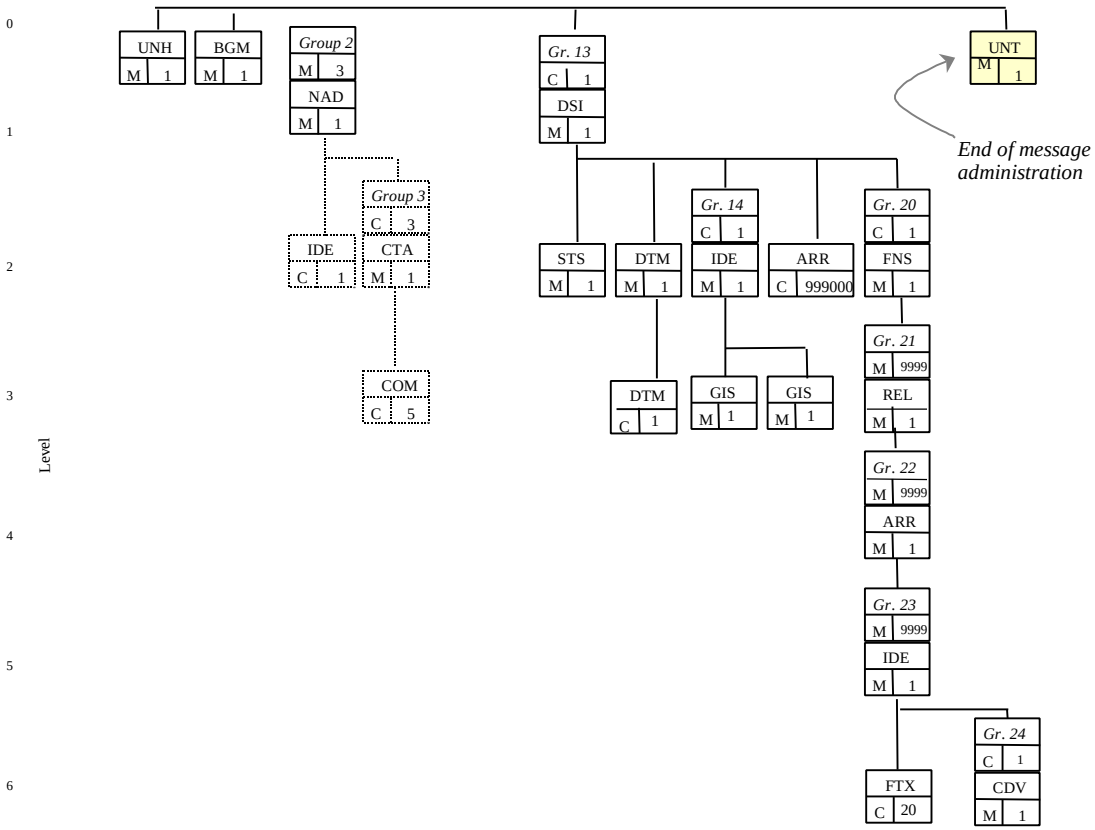
`REL+Z01+4'`	Array scope is "combination of dimension values"
`ARR+4+M:XX:ZZ:CC'`	attached to time series M:XX:ZZ:CC
`IDE+Z10+COLLECTION'`	Coded attribute COLLECTION
`CDV+A'`	**coded value for collection is A = "average of period"**
`IDE+Z11+COLL_DETAIL'`	Uncoded attribute COLL_DETAIL
`FTX+ACM+++simple arithmetic mean of daily values'`	attribute text value

Example 4 (attribute for an individual observation)

`REL+Z01+5'`	Array scope is "observation"
`ARR+6+M:XX:ZZ:CC:199606:610'`	attached to obs. of June 1996 for series M:XX:ZZ:CC
`IDE+Z11+OBS_COM'`	uncoded attribute OBS_COM
`FTX+ACM+++NEW ACCOUNTING METHOD'`	attribute text value

6.10. End-of-message administration section

The UNT segment ends the message and contains control-count data that the receiver can use to check the integrity of the data received.



6.10.1. Segment Type: UNT

Segment Type Name: *Message Trailer*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

UNT+*number-of-segments*+*message-reference-number*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
Number-of-segments	mandatory this is a total count of all the segments contained in the message, including the UNH and the UNT segments	n..6	
message-reference-number	mandatory this is the same value as the *message-reference-number* in the UNH at the beginning of the message	an..14	Format in GESMES/TS: MREFnnnnnn

Content and General Usage Rules

This segment ends the message and contains control data that a receiver can use to validate message integrity.

Example

```
UNT+59+MREF000001'
```

This message contains 59 segments, and the UNH that started the message has a message-reference-number of MREF000001

6.11. End-of-interchange administration section

The UNZ segment ends the interchange (UNZ) and contains control-count data that a receiver can use to check the integrity of the messages received.

6.11.1. **Segment Type:** **UNZ**

Segment Type Name: *Interchange Trailer*

Max. Number of Occurrences: *1*

Status: *Mandatory*

Data Element Sequence

UNZ+*number-of-messages*+*interchange-reference*'

Data Element Usage Rules

local name	usage rules	format	GESMES/TS code values
number-of-messages	mandatory this is a count of the number of messages (i.e. UNH/UNT pairs) that are in the interchange	n..6	
interchange-reference	mandatory this must be the same value that is placed in the *interchange-reference* in the UNB at the beginning of this interchange	an..14	Format: IREFnnnnnn

Content and General Usage Rules

This segment ends the interchange. You as a sender should put it immediately following the last UNT of the last message of the interchange. It contains a count of the number of messages in the interchange and a reference number that your receiver can use to validate the integrity of the interchange.

Example

```
UNZ+3+IREF000006'
```

There are three messages in the interchange, and the value of the interchange-reference in the UNB that relates to this UNZ is IREF000006

7. DELETE MESSAGES

A GESMES/TS message turns into a message containing *delete directives* when you write its STS segment as:

```
STS+3+6'
```

Once the receiver's system has found this segment, the message is considered as a **delete message.** All its ARR segments should then exclusively refer to values of objects that should be deleted. All references for deletion should refer exclusively to statistical objects belonging or related to the data set indicated by the DSI segment of this message.

7.1. Rules and techniques for deletions

References for deletions can appear in two different parts of a "delete" message, and we discuss them separately below:

- ***DELETIONS ACTIVATED VIA REFERENCES APPEARING IN THE MAIN SET OF ARR SEGMENTS (FOLLOWING GROUP 14) OF THE MESSAGE***

*(**Rule**: numeric observations cannot be present in the ARR segments of the message)*

1. **For specific observations and the corresponding observation level attributes (observation status, observation confidentiality, observation pre-break, observation comment):**

   For instance, the segments
   ```
   ARR++M:XXX:YYY:199201:610'
   ```
   and
   ```
   ARR++M:XXX:ZZZ:199203:610'
   ```
   would imply respectively:
   - the deletion of the `Jan-1992` observation (and the corresponding observation flags) for the series `M:XXX:YYY` and
   - the deletion of the `Mar-1992` observation (and the corresponding observation flags) for the series `M:XXX:ZZZ`

   Recommendation—You should avoid deletions that could create conceptually unjustified "holes" within a time series composed of continuous observations.

2. **For ranges of observations and the corresponding observation level attributes (obs. status, observation confidentiality, observation pre-break, observation comment):**

   For instance, the segment
   ```
   ARR++M:XXX:YYY:199201199205:710'
   ```
   implies the deletion of the observations of the series (and its attributes) from Jan.1992 to May 1992.

Recommendation—Again, you should avoid deletions that could create conceptually unjustified "holes" within a time series composed of continuous observations.

3. **For a specific time series:** *Rule—Dates/periods/time ranges cannot be present in the segment.*

   For instance, if M:BE:XXX:YYY is the key of a series, then the segment
   ```
   ARR++M:BE:XXX:YYY'
   ```
   implies the deletion of this series and all its attribute values at all levels not higher than the time series level (incl. the coded and uncoded attribute values)

4. **For a group of sibling series:** *Rule—Dates/periods/time ranges cannot be present in the segment.*

   For instance, if :BE:XXX:YYY is the key of the sibling group (second position: frequency wildcarded: for any frequency), then the segment
   ```
   ARR++:BE:XXX:YYY'
   ```
   implies the deletion of all series of the group (e.g. Q:BE:XXX:YYY and M:BE:XXX:YYY) and all their attribute values at all levels not higher than the sibling group level (incl. the coded and uncoded attribute values).

5. **For a whole data set:** *Rule—Dates/periods/time ranges cannot be present in the segment.*

   In this case, you should write the ARR segment in the following way: `ARR+0'`
   This segment should imply the deletion of all series and all attribute values of the data set indicated by the DSI segment of the message.

- ***DELETIONS ACTIVATED VIA REFERENCES APPEARING IN THE SET OF GROUP 21-ARR SEGMENTS:***

  ***Rule**: Both the ARR and IDE segments should be present in a delete message, but CDV or FTX segments (which follow in a normal update message) cannot be present in this case.*

6. **For specific observation comments (OBS_COM) at the observation level:**

   For instance, the segments
   ```
   ARR+7+M:ABB:A:DE:S1:199606:610'
   IDE+Z11+OBS_COM'
   ```
   imply the deletion of the observation comment value for Jun.96 for the series
   `M:ABB:A:DE:S1`.

7. **For specific attributes at a series level:**

   For instance, the segments
   ```
   ARR+5+M:ABB:A:DE:S1'
   IDE+Z11+COVERAGE'
   ```
   imply the deletion of the value of the attribute `COVERAGE` for the series `M:ABB:A:DE:S1`.

8. **For specific attributes at a sibling group level:**

   For instance, if in this key family, the attribute `COVERAGE` has been defined at the sibling group level and frequency is the first dimension of the key family, then the segments
   ```
   ARR+5+:ABB:A:DE:S1'
   IDE+Z11+COVERAGE'
   ```
   imply the deletion of the value of the attribute `COVERAGE` for the sibling group
   `:ABB:A:DE:S1`.

9. **For specific attributes at a data set level:**

For instance, if in this key family, the attribute OTHER_METH_EXPL has been defined at the data set level, then the segments

```
ARR+0'
IDE+Z11+OTHER_METH_EXPL'
```

imply the deletion of the text for the attribute OTHER_METH_EXPL of the data set.

Note: We assume that, before you write the ARR/IDE pairs, you have used appropriate settings for the REL segment:

- REL+Z01+4' for deleting time series' and sibling groups' attributes;
- REL+Z01+5' for deleting observation level attributes e.g. observation comments (but other than the observation status, observation confidentiality, observation pre-break); and
- REL+Z01+1' for data set attributes' deletions.

7.2. Comments on deletions

1. **Deletions** do affect:

    - objects identified by the ARR segment that comes immediately after Group 14 (for deletion of observations, series, sibling groups, and data sets); and

    - specific attribute values identified via references in the ARR segment of Group 22 (in conjunction with a corresponding attribute identifier).

2. You should not allow the **deletion of observations** or ranges of observations to create "holes" in a series containing continuous observations, if you cannot conceptually justify these holes. Usually, a receiving institution expects an observation deletion at the start or the end of a series. For example, if the need should arise for you to temporarily "delete" an observation in the middle of a series with continuous observations (for example if the value is wrong and the correct one is not known yet), then you should use an "update" instruction instead: You should send a "missing value" with an appropriate missing value explanation flag (a value of the corresponding OBS_STATUS attribute indicating the reasoning for the reported missing value).

3. Only the center can initiate the **deletion of a whole data set.** The center should inform the receiving ends in advance using other administrative means, provide a timetable, and describe the concrete actions that should take place. The deletion of a data set implies the deletion of all series included in the data set and of their corresponding attribute values at all levels (observation, time series, sibling, data set). The key family, from which the data set took its structure, is not affected by the deletion of the data set.

4. We do not foresee the **deletion of structural definitions (e.g. key families) using automated means.** If such a need arises, the center that administers the corresponding data exchange will provide guidance. (If the institution administering the data exchange is different from the institution devising the structural definitions used, the two central institutions may need to consult prior to this).

8. MESSAGE ACKNOWLEDGEMENTS

GESMES/TS does not currently support message acknowledgements.

Message acknowledgement services provided by a center may differ between centers.

9. TEXT IN UNCODED ATTRIBUTES: RULES AND CONSIDERATIONS

9.1. Character set

A GESMES/TS message uses the upper-case Latin character set, the service characters (: ' ? +), the dash (used in negative numbers and to denote missing values), and the dot (decimal point in numbers). **Text given as a value to an uncoded attribute can also use any character of the standard character set (e.g., lower-case characters) and characters of the** extended set (160-255), e.g. é, ñ, ê, ö, ç. However, before putting in production the characters of the extended set (or any other character set) in a data exchange context, all partners should confirm that this would be compatible with their applications.

9.2. Special characters

Use caution with special non-alphanumeric characters in uncoded attributes:

- Beyond the ANSI/ASCII code number 126, only the ANSI-accented-characters are allowed (depending on the outcome of the corresponding tests as we mentioned before). In particular, you should not use control characters (coded in positions 128 - 159) inside a GESMES/TS message. They can cause problems in applications.
- The release character ("?" in GESMES/TS) must precede any appearance of the GESMES/TS service characters (+ ' : ?), but it should not precede decimal points (see also Appendix I on the EDIFACT syntax). You should not include the release character in the character count for FTX component elements.

9.3. Uncoded attribute values: maximum length of text

The length of the text that you can use in an uncoded attribute value is specified in the key family definition. We discuss this issue extensively on page 81 and 144 (providing also a number of examples). For instance, an FTX segment can contain up to 350 characters (broken down to five component elements of maximum 70 characters of original text each[1]).

[1] Release characters are not counted in the 70 characters so that a component element in an FTX segment may actually be longer then 70 characters, depending on the number of release characters that had to be inserted.

10. EXAMPLES OF MESSAGES

10.1. Data update message

Let us assume that the National Bank of Belgium (NBB) has to send the following data to the ECB:

Key family: **PRICES_TEST_DATA** - Monthly data for five time series

	M:BE:PROD:GN:NS	M:BE:CONS:GN:NS	M:BE:WHOL:GN:NS	M:BE:WHOL:RM:NS	M:BE:WHOL:RM:SA
Sep.95	99.10 A	112.20 A	111.80 A	110.20 A	102.90 A
Oct.95	98.10 A	112.30 A	112.10 A	110.10 A	103.00 A
Nov.95	98.40 A	112.40 A	111.90 A	110.00 A	104.60 A
Dec.95	99.50 A	112.50 A	112.10 A	110.10 A	106.30 A
Jan.96	100.00 A	112.70 A	112.40 A	110.80 A	109.00 A
Feb.96	99.20 A	113.10 A	112.30 A	111.00 A	111.10 A
Mar.96	99.80 A	113.80 A	112.00 A	110.10 A	109.80 A

Key family: PRICES_TEST_DATA - Quarterly data for five time series

Series key:	Q:BE:PROD:GN:NS	Q:BE:CONS:GN:NS	Q:BE:WHOL:GN:NS	Q:BE:WHOL:RM:NS	Q:BE:WHOL:RM:SA
95q4	98.67 A	112.40 A	112.03 A	110.07 A	104.63 A
96q1	99.67 A	113.20 A	112.23 A	110.63 A	109.97 A

Key family: **INT_RATES_KF,** series: D:BE:IR:MM:THRM (211 daily observations: 1/9/95 to 29/3/96)

01. Sep.95	4.31 A	*11. Sep.95*	4.22 A	...	...
02. Sep.95	(na) H	*12. Sep.95*	4.23 A	*etc.*	
03. Sep.95	(na) H	*13. Sep.95*	4.17 A	...	...
04. Sep.95	4.30 A	*14. Sep.95*	4.16 A	*23. Mar.96*	(na) H
05. Sep.95	4.32 A	*15. Sep.95*	4.16 A	*24. Mar.96*	(na) H
06. Sep.95	4.25 A	*16. Sep.95*	(na) H	*25. Mar.96*	3.31 A
07. Sep.95	4.25 A	*17. Sep.95*	(na) H	*26. Mar.96*	3.31 A
08. Sep.95	4.23 A	*18. Sep.95*	4.16 A	*27. Mar.96*	3.31 E
09. Sep.95	(na) H	...	...	*28. Mar.96*	3.32 F C
10. Sep.95	(na) H	...	...	*29. Mar.96*	3.33 F C

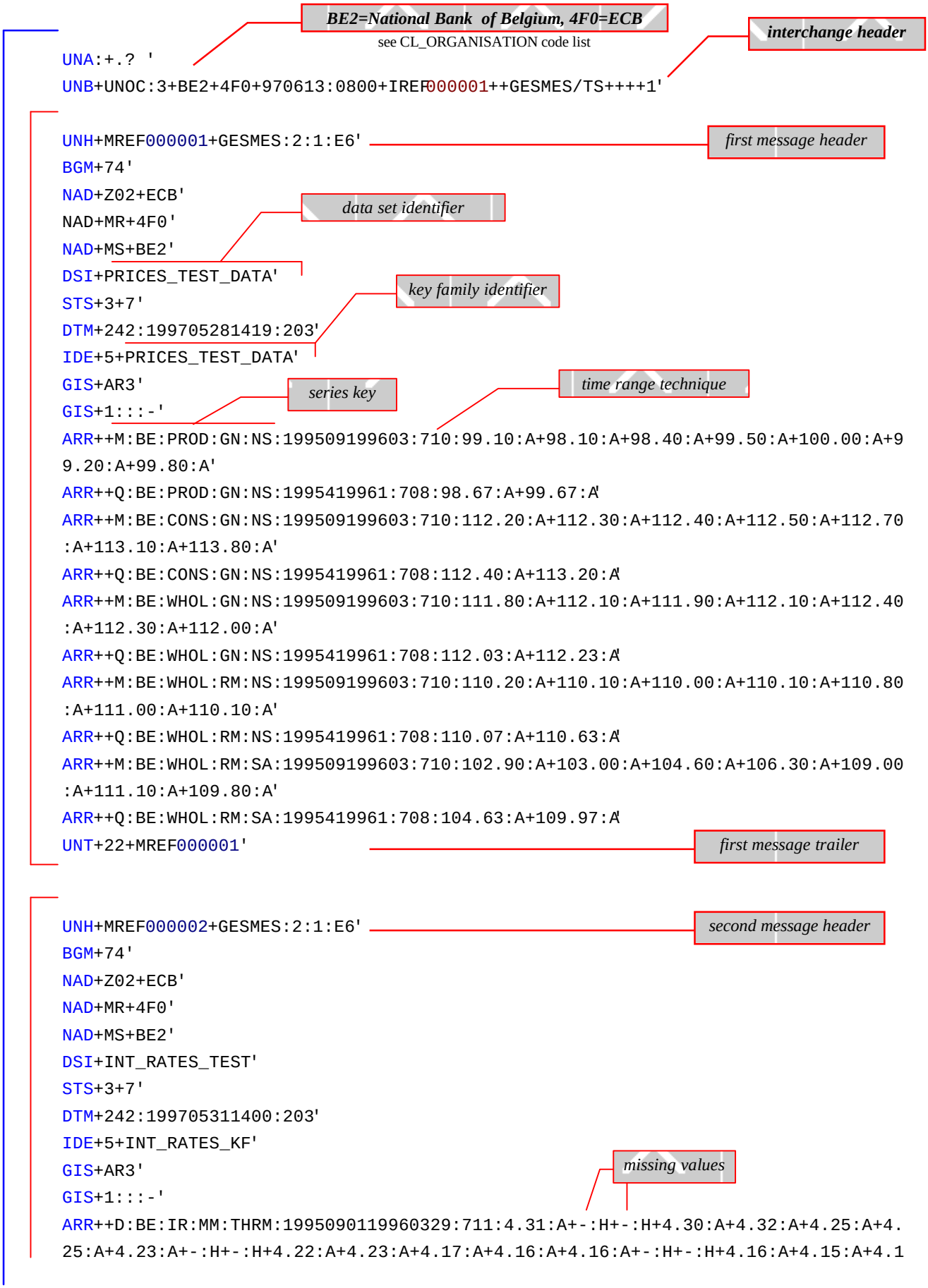
As shown on the table above, the last two daily observations have been flagged as confidential.

For the series M:BE:IR:MM:THRM, the NBB has revised the observation 4.10:A for February 1994 and needs to report it as well.

For the monthly series M:BE:IR:MM:THRM, the NBB also has to send the following observations:

Sep.95	Oct.95	Nov.95	Dec.95	Jan.96	Feb.96	Mar.96
4.24 A	4.22 A	3.98 A	3.84 A	3.53 A	3.32 A	3.31 A

The following sequence of segments is a full interchange containing two messages (one for each key family) and performs the required reporting:




```
UNA:+.? '
UNB+UNOC:3+BE2+4F0+970613:0800+IREF000001++GESMES/TS++++1'

UNH+MREF000001+GESMES:2:1:E6'
BGM+74'
NAD+Z02+ECB'
NAD+MR+4F0'
NAD+MS+BE2'
DSI+PRICES_TEST_DATA'
STS+3+7'
DTM+242:199705281419:203'
IDE+5+PRICES_TEST_DATA'
GIS+AR3'
GIS+1:::-'
ARR++M:BE:PROD:GN:NS:199509199603:710:99.10:A+98.10:A+98.40:A+99.50:A+100.00:A+9
9.20:A+99.80:A'
ARR++Q:BE:PROD:GN:NS:1995419961:708:98.67:A+99.67:A'
ARR++M:BE:CONS:GN:NS:199509199603:710:112.20:A+112.30:A+112.40:A+112.50:A+112.70
:A+113.10:A+113.80:A'
ARR++Q:BE:CONS:GN:NS:1995419961:708:112.40:A+113.20:A'
ARR++M:BE:WHOL:GN:NS:199509199603:710:111.80:A+112.10:A+111.90:A+112.10:A+112.40
:A+112.30:A+112.00:A'
ARR++Q:BE:WHOL:GN:NS:1995419961:708:112.03:A+112.23:A'
ARR++M:BE:WHOL:RM:NS:199509199603:710:110.20:A+110.10:A+110.00:A+110.10:A+110.80
:A+111.00:A+110.10:A'
ARR++Q:BE:WHOL:RM:NS:1995419961:708:110.07:A+110.63:A'
ARR++M:BE:WHOL:RM:SA:199509199603:710:102.90:A+103.00:A+104.60:A+106.30:A+109.00
:A+111.10:A+109.80:A'
ARR++Q:BE:WHOL:RM:SA:1995419961:708:104.63:A+109.97:A'
UNT+22+MREF000001'

UNH+MREF000002+GESMES:2:1:E6'
BGM+74'
NAD+Z02+ECB'
NAD+MR+4F0'
NAD+MS+BE2'
DSI+INT_RATES_TEST'
STS+3+7'
DTM+242:199705311400:203'
IDE+5+INT_RATES_KF'
GIS+AR3'
GIS+1:::-'
ARR++D:BE:IR:MM:THRM:1995090119960329:711:4.31:A+-:H+-:H+4.30:A+4.32:A+4.25:A+4.
25:A+4.23:A+-:H+-:H+4.22:A+4.23:A+4.17:A+4.16:A+4.16:A+-:H+-:H+4.16:A+4.15:A+4.1
```

```
4:A+4.15:A+4.44:A+-:H+-:H+4.32:A+4.24:A+4.25:A+4.27:A+4.30:A+-:H+-:H+4.24:A+4.23
:A+4.17:A+4.16:A+4.30:A+-:H+-:H+4.25:A+4.25:A+4.24:A+4.22:A+4.20:A+-:H+-:H+4.20:
A+4.21:A+4.26:A+4.23:A+4.25:A+-:H+-:H+4.26:A+4.20:A+4.21:A+4.22:A+4.20:A+-:H+-:H
+4.15:A+4.14:A+-:H+4.12:A+4.11:A+-:H+-:H+4.10:A+4.11:A+4.06:A+4.04:A+4.04:A+-:H+
-:H+-:H+4.01:A+4.00:A+3.98:A+4.00:A+-:H+-:H+3.95:A+3.91:A+3.94:A+3.91:A+3.90:A+-
:H+-:H+3.87:A+3.84:A+3.86:A+3.92:A+3.91:A+-:H+-:H+3.96:A+3.92:A+3.91:A+3.92:A+3.
93:A+-:H+-:H+3.90:A+3.89:A+3.88:A+3.75:A+3.79:A+-:H+-:H+3.79:A+3.81:A+3.79:A+3.8
0:A+3.79:A+-:H+-:H+-:H+-:H+3.76:A+3.74:A+3.73:A+-:H+-:H+-:H+3.71:A+3.69:A+3.69:A
+3.69:A+-:H+-:H+3.69:A+3.67:A+3.63:A+3.61:A+3.58:A+-:H+-:H+3.58:A+3.50:A+3.47:A+
3.45:A+3.44:A+-:H+-:H+3.43:A+3.44:A+3.41:A+3.39:A+3.44:A+-:H+-:H+3.43:A+3.41:A+3
.33:A+3.28:A+3.25:A+-:H+-:H+3.29:A+3.31:A+3.27:A+3.27:A+3.28:A+-:H+-:H+3.27:A+3.
27:A+3.32:A+3.32:A+3.33:A+-:H+-:H+3.42:A+3.42:A+3.42:A+3.36:A+3.31:A+-:H+-:H+3.3
4:A+3.35:A+3.31:A+3.32:A+3.31:A+-:H+-:H+3.30:A+3.30:A+3.31:A+3.30:A+3.29:A+-:H+-
:H+3.34:A+3.31:A+3.31:A+3.31:A+3.31:A+-:H+-:H+3.32:A+3.31:A+3.32:A+3.31:A+3.31:A
+-:H+-:H+3.31:A+3.31:A+3.31:E+3.32:F:C+3.33:F:C'
ARR++M:BE:IR:MM:THRM:199402:610:4.10:A'
ARR++M:BE:IR:MM:THRM:199509199603:710:4.24:A+4.22:A+3.98:A+3.84:A+3.53:A+3.32:A+
3.31:A'
UNT+15+MREF000002'

UNZ+2+IREF000001'
```

obs-confid (2nd -optional- attr.)

second message trailer

interchange trailer

In a completely realistic scenario, you do not need the *line feed* characters after the end of each segment, and you can transmit the whole interchange in only *one line*. This is shown below, in the following interchange:

```
UNA:+.? 'UNB+UNOC:3+BE2+4F0+970613:0800+IREF000002++GESMES/TS++++1'UNH+MREF000001+GES
MES:2:1:E6'BGM+74'NAD+Z02+ECB'NAD+MR+4F0'NAD+MS+BE2'DSI+PRICES_TEST_DATA'STS+3+7'DTM+
242:199705281419:203'IDE+5+PRICES_TEST_DATA'GIS+AR3'GIS+1:::-'ARR++M:BE:PROD:GN:NS:19
9509199603:710:99.10:A+98.10:A+98.40:A+99.50:A+100.00:A+99.20:A+99.80:A'ARR++Q:BE:PRO
D:GN:NS:1995419961:708:98.67:A+99.67:A'ARR++M:BE:CONS:GN:NS:199509199603:710:112.20:A
+112.30:A+112.40:A+112.50:A+112.70:A+113.10:A+113.80:A'ARR++Q:BE:CONS:GN:NS:199541996
1:708:112.40:A+113.20:A'ARR++M:BE:WHOL:GN:NS:199509199603:710:111.80:A+112.10:A+111.9
0:A+112.10:A+112.40:A+112.30:A+112.00:A'ARR++Q:BE:WHOL:GN:NS:1995419961:708:112.03:A+
112.23:A'ARR++M:BE:WHOL:RM:NS:199509199603:710:110.20:A+110.10:A+110.00:A+110.10:A+11
0.80:A+111.00:A+110.10:A'ARR++Q:BE:WHOL:RM:NS:1995419961:708:110.07:A+110.63:A'ARR++M
:BE:WHOL:RM:SA:199509199603:710:102.90:A+103.00:A+104.60:A+106.30:A+109.00:A+111.10:A
+109.80:A'ARR++Q:BE:WHOL:RM:SA:1995419961:708:104.63:A+109.97:A'UNT+22+MREF000001'UNH
+MREF000002+GESMES:2:1:E6'BGM+74'NAD+Z02+ECB'NAD+MR+4F0'NAD+MS+BE2'DSI+DAILY_INT_RATE
S_TEST'STS+3+7'DTM+242:199705311400:203'IDE+5+DAILY_INT_RATES_TEST'GIS+AR3'GIS+1:::-'
ARR++D:BE:IR:MM:THRM:1995090119960329:711:4.31:A+:H+-:H+4.30:A+4.32:A+4.25:A+4.25:A+
4.23:A+-:H+-:H+4.22:A+4.23:A+4.17:A+4.16:A+4.16:A+-:H+-:H+4.16:A+4.15:A+4.14:A+4.15:A
+4.44:A+-:H+-:H+4.32:A+4.24:A+4.25:A+4.27:A+4.30:A+-:H+-:H+4.24:A+4.23:A+4.17:A+4.16:
```

```
A+4.30:A+-:H+-:H+4.25:A+4.25:A+4.24:A+4.22:A+4.20:A+-:H+-:H+4.20:A+4.21:A+4.26:A+4.23
:A+4.25:A+-:H+-:H+4.26:A+4.20:A+4.21:A+4.22:A+4.20:A+-:H+-:H+4.15:A+4.14:A+-:H+4.12:A
+4.11:A+-:H+-:H+4.10:A+4.11:A+4.06:A+4.04:A+4.04:A+-:H+-:H+-:H+4.01:A+4.00:A+3.98:A+4
.00:A+-:H+-:H+3.95:A+3.91:A+3.94:A+3.91:A+3.90:A+-:H+-:H+3.87:A+3.84:A+3.86:A+3.92:A+
3.91:A+-:H+-:H+3.96:A+3.92:A+3.91:A+3.92:A+3.93:A+-:H+-:H+3.90:A+3.89:A+3.88:A+3.75:A
+3.79:A+-:H+-:H+3.79:A+3.81:A+3.79:A+3.80:A+3.79:A+-:H+-:H+-:H+-:H+3.76:A+3.74:A+3.73
:A+-:H+-:H+-:H+3.71:A+3.69:A+3.69:A+3.69:A+-:H+-:H+3.69:A+3.67:A+3.63:A+3.61:A+3.58:A
+-:H+-:H+3.58:A+3.50:A+3.47:A+3.45:A+3.44:A+-:H+-:H+3.43:A+3.44:A+3.41:A+3.39:A+3.44:
A+-:H+-:H+3.43:A+3.41:A+3.33:A+3.28:A+3.25:A+-:H+-:H+3.29:A+3.31:A+3.27:A+3.27:A+3.28
:A+-:H+-:H+3.27:A+3.27:A+3.32:A+3.32:A+3.33:A+-:H+-:H+3.42:A+3.42:A+3.42:A+3.36:A+3.3
1:A+-:H+-:H+3.34:A+3.35:A+3.31:A+3.32:A+3.31:A+-:H+-:H+3.30:A+3.30:A+3.31:A+3.30:A+3.
29:A+-:H+-:H+3.34:A+3.31:A+3.31:A+3.31:A+3.31:A+-:H+-:H+3.32:A+3.31:A+3.32:A+3.31:A+3
.31:A+-:H+-:H+3.31:A+3.31:A+3.31:E+3.32:F:C+3.33:F:C'ARR++M:BE:IR:MM:THRM:199402:610:
4.10:A'ARR++M:BE:IR:MM:THRM:199509199603:710:4.24:A+4.22:A+3.98:A+3.84:A+3.53:A+3.32:
A+3.31:A'UNT+15+MREF000002'UNZ+2+IREF000002'
```

10.2. Data and attribute update message

```
UNA:+.? '
UNB+UNOC:3+BE2+4F0+970926:1948+IREF000001++GESMES/TS++++1'
```
interchange administration section

```
UNH+MREF000001+GESMES:2:1:E6'
BGM+74'
NAD+Z02+EMI'
NAD+MR+4F0'
NAD+MS+BE2'
IDE+10+Test message (26-Sep-97)'
```
message administration section

```
DSI+EMI_TEST_BOP'
STS+3+7'
DTM+242:199709261948:203'
```
data set administration section

```
IDE+5+EMI_TEST_BOP'
GIS+AR3'
GIS+1:::-'
ARR++M:BE:N:2:269:1:199610199703:710:2.51:A+2.55:A+2.62:A+2.73:A+2.84:A+2.92:E'
ARR++M:BE:S:3:982:2:199703199708:710:0.74:A+0.76:A+0.79:A+0.85:A+0.97:A+1.07:A'
ARR++M:BE:N:4:379:3:199706199710:710:4.36:A+4.47:A+4.71:A+4.59:A+3.73:F:C'
```
array structure and data section

```
FNS+Attributes:10'
REL+Z01+4'
ARR+6+:BE:N:1:379:3'
IDE+Z11+TITLE'
FTX+ACM+++Test virtual title'
IDE+Z10+UNIT'
CDV+BEF'
IDE+Z10+UNIT_MULT'
CDV+6'
ARR+6+M:BE:S:2:379:3'
IDE+Z10+COLLECTION'
CDV+2'
ARR+6+M:BE:N:2:379:2'
IDE+Z10+COLLECTION'
CDV+C'
REL+Z01+5'
ARR+8+M:BE:N:4:379:3:199705:610'
IDE+Z11+OBS_COM'
FTX+ACM+++Test virtual observation comment'
```
attribute section

```
UNT+35+MREF000001'
```
end of message administration

```
UNZ+1+IREF000001'
```
end of interchange administration

The message above contains data and attributes. We have done the following:

- Given some numeric values (array structure and data section) for the series `M:BE:N:2:269:1`, `M:BE:S:3:982:2` and `M:BE:N:4:379:3.`
- Assigned the text "`Test virtual title`" as TITLE to the sibling group "`:BE:N:1:379:3`" (here, we assume "frequency" to be the first dimension, and it is wild-carded). For the same sibling group, we assign the value "`BEF`" to the attribute UNIT and the value "`6`" to the attribute UNIT_MULT (unit multiplier).
- Assigned the value "`C`" to the attribute COLLECTION for the series `M:BE:S:2:379:3` and the same attribute value also for the series `M:BE:N:2:379:2.`
- Assigned the text "`Test virtual observation comment`" as OBS_COM (observation comment) for the May 1995 observation for the series `M:BE:N:4:379:3`.

10.3. Key family definition message

In the previous example, the objects referred to series which, of course, belong to a key family. So, let us assume that this key family's definition is reflected in the following table[1]:

[1] This is just an example, it does not refer to any key family used in actual testing or reporting. Also, for reasons of simplicity, the TIME_PERIOD and TIME_FORMAT concepts are not shown in the table.

KEY FAMILY: Test key family ref. to balance of payments data
KEY FAMILY MNEMONIC: EMI_TEST_BOP
Release: 1.0, 26 Sep. 1997

		Concept (mnemonic)	Concept name	Code list (mnemonic)	Code list name	code values: N / AN
	ORDER	KEY FAMILY DIMENSIONS				
	1	FREQ	Frequency	CL_FREQ	Frequency codelist	AN1
	2	REF_AREA	Reference area	CL_AREA_EE	Area codelist (EUROSTAT-BoP/EMI)	AN2
	3	ADJUSTMENT	Adjustment indicator	CL_ADJUSTMENT	Adjustment indicator codelist	AN1
	4	DATA_TYPE	BoP type/position	CL_DATA_TYPE	BoP type/position codelist	AN1
	5	BOP_ITEM	BoP item	CL_BOP_ITEM	BoP item codelist	AN3
	6	BASIS	Data basis	CL_BASIS	Data basis codelist	AN1
Attachment level	**Status**					
I. Attributes (attached to the observation in the message)						
Observation	*M*	OBS_STATUS	Observation status	CL_OBS_STATUS	Observation status codelist	AN1
Observation	*C*	OBS_CONF	Observation confidentiality	CL_OBS_CONF	Observation confidentiality codelist	AN1
II. Attributes (handled in the FNS group)						
Sibling	*M*	TITLE	Title	*<uncoded>*	*<uncoded>*	AN..70
Sibling	*M*	UNIT	Unit	CL_UNIT	Unit codelist	AN..12
Sibling	*M*	UNIT_MULTIPLIER	Unit multiplier	CL_UNIT_MULT	Unit multiplier codelist	AN..2
Sibling	*M*	DECIMALS	Decimals	CL_DECIMALS	Decimals codelist	AN1
Time series	*M*	COLLECTION	Collection indicator	CL_COLLECTION	Collection indicator codelist	AN1
Time series	*M*	AVAILABILITY	Availability	CL_AVAILABILITY	Organisation availability code list	AN1
Sibling	*M*	COMPILATION	Compilation explanations	*<uncoded>*	*<uncoded>*	AN..70
Observation	*C*	OBS_COM	Observation comment	*<uncoded>*	*<uncoded>*	AN..70
Time series	*C*	BREAKS	Explanation for breaks	*<uncoded>*	*<uncoded>*	AN..70
Sibling	*C*	AGG_EQUN	Aggregation equation used	*<uncoded>*	*<uncoded>*	AN..70

Other concepts used in the reporting/dissemination messages (message administration section)					
	ORGANISATION	Organisation	*CL_ORGANISATION*	*Organisation codelist*	AN3
Other concepts used in the Gesmes/CB key family dissemination message (advanced use)					
Advanced use (ASI group)	USAGE_STATUS	Usage status	*USS*	*Usage status code list*	AN1
Advanced use (ASI group)	ATTACHMENT_LEVEL	Attachment level	*ALV*	*Attachment level code list*	AN1

Each message that refers to a key family, like the one presented in the previous example, has to comply with the corresponding key family definition.

A center (e.g., ECB=4F0) could disseminate the key family definition table, presented above, to its partners (ZZZ=unspecified), using GESMES/TS. For this, it would use the ASI group of segments. We assume that the code EMI represents the structural definitions maintenance agency. The message below is the coded version of the previous table:

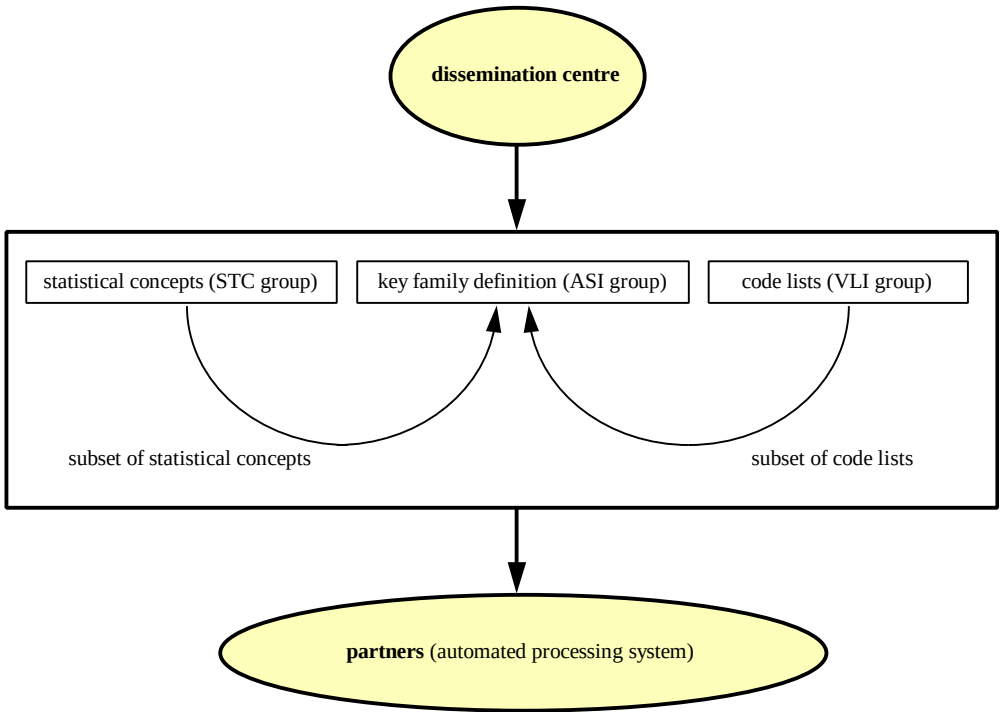
```
UNA:+.? '
UNB+UNOC:3+4F0+ZZZ+970525:1539+IREF000003++GESMES/TS'
UNH+MREF000001+GESMES:2:1:E6'
BGM+73'
NAD+Z02+EMI'
NAD+MR+ZZZ'
NAD+MS+4F0'
IDE+10+Ad hoc message - example key family'
ASI+EMI_TEST_BOP'
FTX+ACM+++Test kf for BoP Statistics'
SCD+13+FREQ++++:1'ATT+3+5+:::AN1'IDE+1+CL_FREQ'
SCD+4+REF_AREA++++:2'ATT+3+5+:::AN2'IDE+1+CL_AREA_EE'
SCD+4+ADJUSTMENT++++:3'ATT+3+5+:::AN1'IDE+1+CL_ADJUSTMENT'
SCD+4+DATA_TYPE++++:4'ATT+3+5+:::AN1'IDE+1+CL_DATA_TYPE'
SCD+4+BOP_ITEM++++:5'ATT+3+5+:::AN3'IDE+1+CL_BOP_ITEM'
```

```
SCD+4+BASIS++++:6'ATT+3+5+:::AN1'IDE+1+CL_BASIS'
SCD+1+TIME_PERIOD++++:7'ATT+3+5+:::AN..35'
SCD+1+TIME_FORMAT++++:8'ATT+3+5+:::AN3'
SCD+3+OBS_VALUE++++:9'ATT+3+5+:::AN..15'
SCD+3+OBS_STATUS++++:10'ATT+3+5+:::AN1'ATT+3+35+2:USS'ATT+3+32+5:ALV'IDE+1+CL_OBS_STA
TUS'
SCD+3+OBS_CONF++++:11'ATT+3+5+:::AN1'ATT+3+35+1:USS'ATT+3+32+5:ALV'IDE+1+CL_OBS_CONF'
SCD+Z09+TITLE'ATT+3+5+:::AN..70'ATT+3+35+2:USS'ATT+3+32+9:ALV'
SCD+Z09+UNIT'ATT+3+5+:::AN..12'ATT+3+35+2:USS'ATT+3+32+9:ALV'IDE+1+CL_UNIT'
SCD+Z09+UNIT_MULTIPLIER'ATT+3+5+:::AN..2'ATT+3+35+2:USS'ATT+3+32+9:ALV'IDE+1+CL_UNIT_
MULT'
SCD+Z09+DECIMALS'ATT+3+5+:::AN1'ATT+3+35+2:USS'ATT+3+32+9:ALV'IDE+1+CL_DECIMALS'
SCD+Z09+COLLECTION'ATT+3+5+:::AN1'ATT+3+35+2:USS'ATT+3+32+4:ALV'IDE+1+CL_COLLECTION'
SCD+Z09+AVAILABILITY'ATT+3+5+:::AN1'ATT+3+35+2:USS'ATT+3+32+4:ALV'IDE+1+CL_AVAILABILI
TY'
SCD+Z09+COMPILATION'ATT+3+5+:::AN..70'ATT+3+35+2:USS'ATT+3+32+9:ALV'
SCD+Z09+BREAKS'ATT+3+5+:::AN..70'ATT+3+35+1:USS'ATT+3+32+4:ALV'
SCD+Z09+AGG_EQUN'ATT+3+5+:::AN..70'ATT+3+35+1:USS'ATT+3+32+9:ALV'
SCD+Z09+OBS_COM'ATT+3+5+:::AN..70'ATT+3+35+1:USS'ATT+3+32+5:ALV'
UNT+88+MREF000001'
UNZ+1+IREF000003'
```

These three elements should be also defined

The partners could build fully automated systems by reading the *key family definitions* directly from GESMES/TS messages, like the one above.

In fact, the message needs all three GESMES/TS structural sections (each one could be in a different message), shown in the graph below, for a center to be able to disseminate the complete key family definition in GESMES/TS and for the receiving institutions not to require paper to interpret the received time-series data.



As a receiver, you could automate your system to "read" *the list of statistical concepts* and the relevant *code lists* and link them in the context of a *key family definition message*. The next two examples complete the whole example by providing the list of statistical concepts and the code lists.

10.4. List of concepts dissemination message

The example message below contains the concepts that we used in building the previous key family.

```
UNA:+.? '
UNB+UNOC:3+4F0+ZZZ+970525:1539+IREF000001++GESMES/TS'
UNH+MREF000001+GESMES:2:1:E6'
BGM+73'
NAD+Z02+EMI'
NAD+MR+ZZZ'
NAD+MS+4F0'
IDE+10+Statistical concepts-example list'
STC+REF_AREA'FTX+ACM+++Reference area'
STC+FREQ'FTX+ACM+++Frequency'
STC+ADJUSTMENT'FTX+ACM+++Adjustment indicator'
STC+DATA_TYPE'FTX+ACM+++BoP type/position'
STC+BOP_ITEM'FTX+ACM+++BoP item'
STC+BASIS'FTX+ACM+++Data basis'
STC+TIME_PERIOD'FTX+ACM+++Time period or range'
STC+TIME_FORMAT'FTX+ACM+++Time format code'
STC+OBS_VALUE'FTX+ACM+++Observation value'
STC+TITLE'FTX+ACM+++Title'
STC+UNIT'FTX+ACM+++Unit'
STC+UNIT_MULTIPLIER'FTX+ACM+++Unit multiplier'
STC+DECIMALS'FTX+ACM+++Decimals'
STC+AVAILABILITY'FTX+ACM+++Availability'
STC+COLLECTION'FTX+ACM+++Collection indicator'
STC+AVAILABILITY'FTX+ACM+++Availability'
STC+COMPILATION'FTX+ACM+++Compilation explanations'
STC+BREAKS'FTX+ACM+++Explanation for breaks'
STC+AGG_EQUN'FTX+ACM+++Aggregation equation used'
STC+OBS_COM'FTX+ACM+++Observation comment'
STC+ORGANISATION'FTX+ACM+++Organisation'
UNT+49+MREF000001'
UNZ+1+IREF000001'
```

10.5. Lastly—the Code-list dissemination message

The example message below contains the code lists. You should use these lists to give values to the coded statistical concepts of a key family (according to its definition). The message below is for demonstration purposes only—the code lists shown are much shorter than their actual length (e.g., countries and other codes are missing).

```
UNA:+.? '
UNB+UNOC:3+4F0+ZZZ+970525:1539+IREF000002++GESMES/TS'
UNH+MREF000001+GESMES:2:1:E6'
BGM+73'
NAD+Z02+EMI'
NAD+MR+ZZZ'
```

```
NAD+MS+4F0'
IDE+10+Message with example code lists'
VLI+CL_FREQ+++Frequency code list'
CDV+M'FTX+ACM+++monthly'
CDV+Q'FTX+ACM+++quarterly'
VLI+CL_AREA_EE+++Area code list (EUROSTAT/ECB)'
CDV+4D'FTX+ACM+++European Commission'
CDV+4F'FTX+ACM+++European Monetary Institute'
CDV+5B'FTX+ACM+++Bank for International Settlements'
CDV+BE'FTX+ACM+++Belgium'
CDV+DK'FTX+ACM+++Denmark'
CDV+FR'FTX+ACM+++France'
CDV+GB'FTX+ACM+++United Kingdom'
CDV+GR'FTX+ACM+++Greece'
CDV+LU'FTX+ACM+++Luxembourg'
CDV+NO'FTX+ACM+++Norway'
CDV+US'FTX+ACM+++United States of America'
VLI+CL_ADJUSTMENT+++Adjustment indicator code list'
CDV+N'FTX+ACM+++neither seasonally or working day adjusted'
CDV+S'FTX+ACM+++seasonally adjusted, not working day adjusted'
CDV+W'FTX+ACM+++working day adjusted, not seasonally adjusted'
CDV+Y'FTX+ACM+++working day and seasonally adjusted'
VLI+CL_DATA_TYPE+++BoP type/position code list'
CDV+1'FTX+ACM+++Stocks at the beginning of the period'
CDV+2'FTX+ACM+++Credit flows'
CDV+3'FTX+ACM+++Debit flows'
CDV+4'FTX+ACM+++Net flows'
CDV+5'FTX+ACM+++Price valuation adjustment'
CDV+6'FTX+ACM+++Exchange rate adjustments'
CDV+7'FTX+ACM+++Reclassification, other adjustments and other revaluations'
CDV+8'FTX+ACM+++Stocks at the end of the period / Gross stocks'
VLI+CL_BOP_ITEM+++BoP item code list'
CDV+269'FTX+ACM+++Merchanting and other trade-related services '
CDV+379'FTX+ACM+++Current transfers'
CDV+971'FTX+ACM+++Freight transport by air and sea'
CDV+981'FTX+ACM+++Other services'
CDV+982'FTX+ACM+++Services not allocated'
CDV+991'FTX+ACM+++Goods and services'
VLI+CL_BASIS+++Data basis code list'
CDV+1'FTX+ACM+++Transaction'
CDV+2'FTX+ACM+++Cash'
CDV+3'FTX+ACM+++Accruals'
CDV+Z'FTX+ACM+++Other'
VLI+CL_OBS_STATUS+++Observation status code list'
CDV+A'FTX+ACM+++normal value'
CDV+E'FTX+ACM+++estimated value'
CDV+F'FTX+ACM+++forecast value'
CDV+H'FTX+ACM+++missing, holiday or weekend'
VLI+CL_OBS_CONF+++Observation confidentiality code list'
CDV+F'FTX+ACM+++Free'
CDV+C'FTX+ACM+++Confidential (and non-publishable)'
VLI+CL_UNIT+++Unit code list'
CDV+BEF'FTX+ACM+++Belgian Franc'
CDV+DEM'FTX+ACM+++German Deutsche Mark'
CDV+ITL'FTX+ACM+++Italian Lira'
VLI+CL_UNIT_MULT+++Unit multiplier code list'
CDV+6'FTX+ACM+++millions'
CDV+9'FTX+ACM+++billions'
VLI+CL_COLLECTION+++Collection indicator code list'
CDV+C'FTX+ACM+++end of period'
CDV+M'FTX+ACM+++average of observations through period'
VLI+CL_ORGANISATION+++Organisation code list'
CDV+BE2'FTX+ACM+++Banque Nationale de Belgique (Belgium)'
CDV+DK2'FTX+ACM+++Danmarks Nationalbank (Denmark)'
CDV+DE2'FTX+ACM+++Deutsche Bundesbank (Germany)'
CDV+4F0'FTX+ACM+++European Central Bank (ECB)'
CDV+5B0'FTX+ACM+++Bank for International Settlements (BIS)'
CDV+4D0'FTX+ACM+++European Commission (Eurostat)'
CDV+ZZZ'FTX+ACM+++Unspecified'
VLI+USS+++Usage status code list'
CDV+1'FTX+ACM+++Conditional'
```

```
CDV+2'FTX+ACM+++Mandatory'
VLI+ALV+++Attachment level code list'
CDV+1'FTX+ACM+++Key family'
CDV+4'FTX+ACM+++Time series'
CDV+5'FTX+ACM+++Observation'
CDV+9'FTX+ACM+++Sibling group'
UNT+143+MREF000001'
UNZ+1+IREF000002'
```

11. APPENDIX I: THE EDIFACT SYNTAX

11.1. Introduction

You will find EDIFACT syntax fully outlined in ISO document 9735, Reference number ISO 9735: 1988 (E). The latest version of this document is dated 1990-11-01.

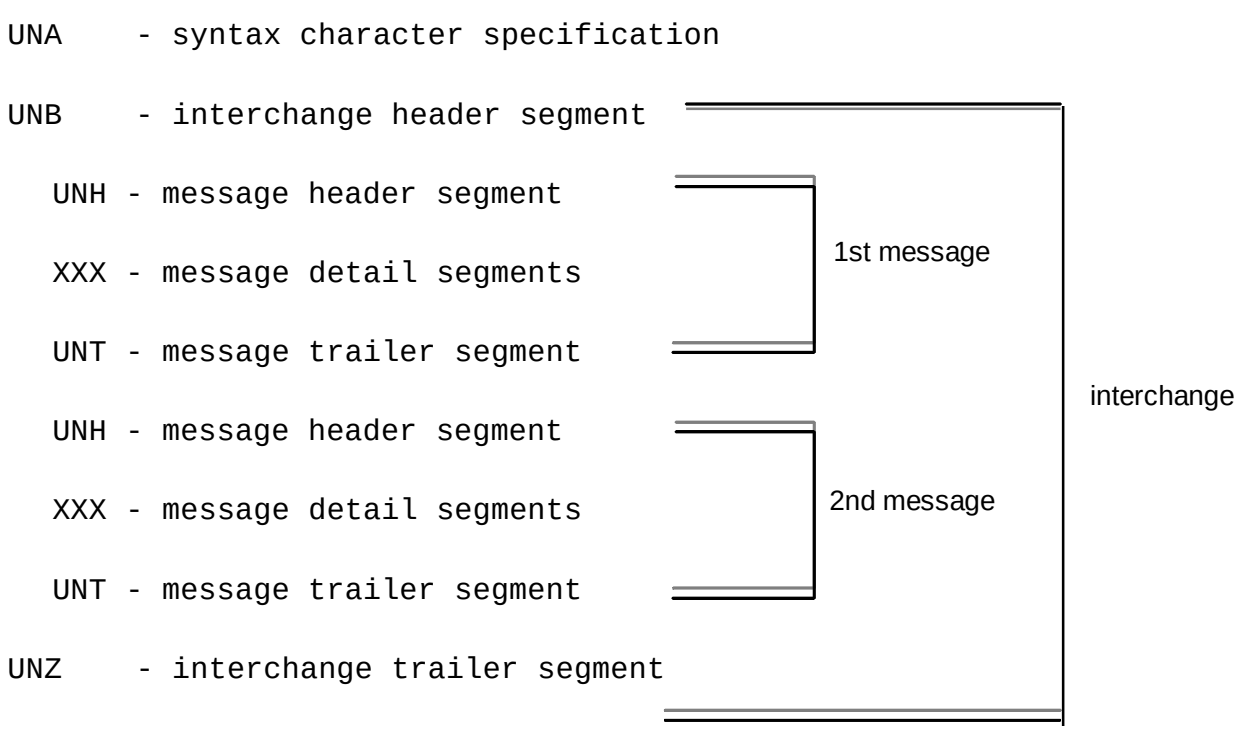
The explanation given below is a simplified version of some of the sections of ISO 9735. The purpose of this explanation is to highlight the parts of ISO 9735 that are relevant to the use of GESMES explained in this guide.

In general, if you are an organisation that wishes to process a GESMES message directly from an application (rather than use a commercial EDIFACT translator), then you should give particular attention to the EDIFACT truncation rules (although, in fact, the "fixed" rules of GESMES/TS incorporate and imply these truncation rules).

Organisations are advised to read ISO 9735 if they desire a fuller understanding of the EDIFACT syntax.

11.2. EDIFACT interchange and message structure

The structure of an EDIFACT interchange is shown below.



An EDIFACT interchange comprises a sequence of segments. Each segment has a unique 3-character identifier. Segment identifiers that begin with UN are called "service segments" and are defined as a part of the EDIFACT Syntax (ISO 9735). Other segments are called “user data segments” and are defined in the Segments Directory of the UN Trade Data Interchange Directory (UNTDID).

11.3. Purpose of the interchange and message envelope segments

UNA - Service String Advice

In general in GESMES, this segment is conditional and is used only if the sender wishes to define the service characters used in the interchange. Its use is advised unless a restricted character set is required (e.g., upper case only). **In GESMES/TS this segment is mandatory** and the specified set is `:+.? '`

UNB - Interchange Header

This contains information that identifies the interchange, the sender, receiver, date and time of preparation, and other optional data. An interchange can contain many messages.

UNH - Message Header

This contains information that identifies the type and version number of the message that follows.

User-Defined Segments

This contains one or more segments that have been designed to support the business function required of the message (e.g., DTM, NAD, DSI, etc.).

UNT - Message Trailer

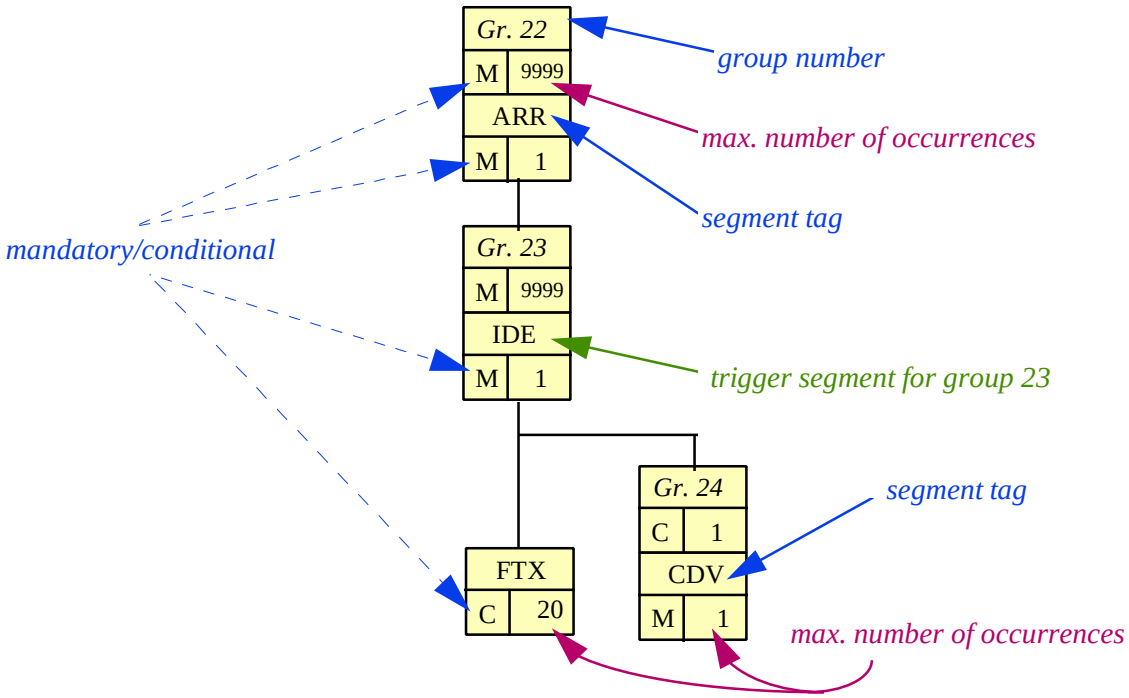
This indicates the end of the message identified in the UNH and contains data that support the checking of the integrity of the data in the message.

UNZ - Interchange Trailer

This indicates the end of the interchange identified in the UNB and contains data that support the checking of the integrity of the interchange.

11.4. Interpreting a Branching Diagram

An EDIFACT message can be represented in a diagrammatic form called a branching diagram. An extract from the branching diagram of GESMES/TS is shown below.



Group 22 is the collection of segments used to assign values to an attribute(s) that is (are) attached at a level or object defined in the ARR segment (group 22). The attribute(s) is (are) identified in the IDE segment, and its (their) value(s) is (are) given in the FTX (if the attribute is uncoded) or in the CDV segment (if it is coded). Group 22 can appear up to 9,999 times.

How do I interpret the diagram?

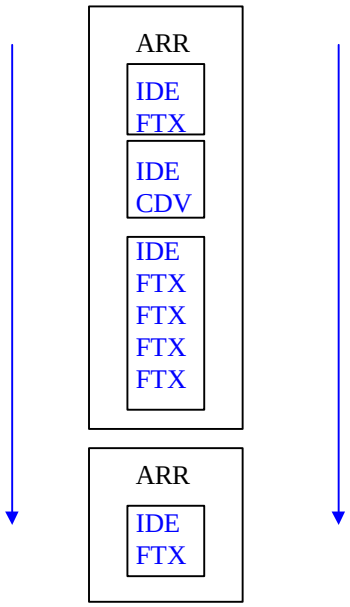
The diagram is interpreted hierarchically. The group number is not sent in the message. The existence of the group in a particular transmission is indicated by the existence of the first segment in the group—the "trigger" segment. The trigger segment is always mandatory if the group is used and can occur only once for each occurrence of the group. The trigger segment for group 22 is the ARR segment. It is followed by between 1 and 9,999 occurrences of group 23. Group 23 comprises between 1 to 20 occurrences of the FTX segment or 1 occurrence of the Group 24. (Here it is not a "genuine" group because it comprises only one segment. In GESMES/TS, it is shown as a *group* for reasons of presentation consistency with generic GESMES, in which Group 24 includes more segments). An example sequence of segments is shown below.

Example

You could present the following ordered sequence of segments in a message:

ARR, **IDE**, FTX, **IDE**, CDV, **IDE**, FTX, FTX, FTX, FTX, **ARR**, **IDE**, CDV.

This message contains 2 occurrences of Group 21. The first one contains three occurrences of Group 22, and the second one contains just one occurrence of Group 22. The sequence of these groups and segments (when reading/writing this message) is more clearly illustrated below:



11.5. EDIFACT service characters

(For a discussion concerning the use of control characters see Box 3, last section of this appendix)

11.5.1. The service-character set

EDIFACT uses the following service characters:

- Segment Terminator
- Data Element Separator
- Component Separator
- Release Indicator

To specify the characters that you use as separators, use the UNA segment—first segment of the transmission. If you do not specified them, then use the default characters specified in the EDIFACT syntax. These defaults depend upon the character repertoire used. The defaults for the Level A character repertoire are:

+ the data element separator
: the component separator

' the segment terminator
? the release indicator

These service characters are specified in the UNA segment of GESMES/TS.

11.5.2. Segment and data-element separators

Terminate a segment by the segment terminator.

Terminate each data element by a data element separator, except the last data element in the segment, which you terminate by a segment terminator. Separate the segment tag from the first data element in the segment by a data element separator.

Terminate a component data element by a component separator, except for the last component in a composite, which you terminate by a data element separator. If the component data element is the last data element in the segment, then terminate it by a segment terminator.

11.5.3. Release indicator

Use the release indicator when you find it necessary to transmit one of the separator, release, or terminator characters as data. Send it immediately before the character to be released. This is valid for the following character only:

For instance, consider the following text:

```
Is today's temperature more than +10 degrees?
```

You would send this in an EDIFACT message as follows:

```
Is today?'s temperature more than ?+10 degrees??
```

11.6. EDIFACT truncation rules

The EDIFACT syntax uses truncation rules to eliminate redundant data. Software creating EDIFACT messages must observe these rules. In order to explain them, we use the following segment:

```
TAG+DE1+CE1:CE2:CE3:CE4+DE3+DE4'
```

TAG *the segment tag (e.g. NAD)*
DE *a data element*
CE *a component element*
\+ *the data element separator*

: *the component separator*

' *the segment terminator*

The truncation rules are as follows:

i) The syntax eliminates leading zeros from numeric data/component elements and eliminates trailing spaces from alphabetic or alphanumeric data/component elements if the data/component element is defined as variable length.

ii) If no data exist for a data/component element, then the relevant separator follows immediately after the separator for the previous data/component element.

iii) The syntax terminates a composite element after the last component element for which there are data for the composite:

```
TAG+DE1+CE1:CE2:CE3:CE4++DE4'
```

data element (DE3) is omitted
(i.e., there are no data for DE3)

```
TAG+DE1+CE1::CE3:CE4+DE3+DE4'
```

a component element (CE2) is omitted
(i.e., there are no data for CE2)

iv) If all the component elements of a composite are omitted, then this is indicated by the data element separator:

```
TAG+DE1+CE1:CE2+DE3+DE4'
```

the last two component elements (CE3, CE4) are omitted
(i.e., there are no data for CE3 and CE4)

```
TAG+DE1++DE3+DE4'
```

the composite element is omitted
(i.e., there are no data for CE1, CE2, CE3, CE4)

v) The segment is terminated immediately after the last data element for which there are data:

```
TAG+DE1+CE1:CE2+DE3'
```

the segment is terminated before the last data element
(i.e., there are no data for DE4)

```
TAG+DE1+CE1:CE2'
```

the segment is terminated after the second component element of the composite
(i.e., there are no data for CE3, CE4, DE3 and DE4)

BOX 3. CONTROL CHARACTERS IN EDIFACT MESSAGES

- EDIFACT messages are simple, flat, text files.
- A whole message or a whole interchange (containing possibly several messages) can be written in one single line (as one very long string of characters) without any line feed character or other control characters.
- However, as without any *line feed* the messages are not well readable to the human eye, several institutions prefer *to insert a line feed at the end of each segment*.
- It has to be noted that in EDIFACT messages the control characters are not significant and they should not affect the "reading" or the interpretation of a message.
  (List of control characters: {from 0 to 0x1f and 0x7f to 0x9f})
- **Applications which "read" EDIFACT messages must ignore all control characters**, regardless of where such characters are located inside the incoming file.

12. APPENDIX II: CODE LISTS FOR OBSERVATION STATUS AND CONFIDENTIALITY

As a reference, the following two tables present the code lists for the observation status and the observation confidentiality attributes. We strongly recommend that in all implementations of GESMES/TS you use these two code lists *unchanged*, to guarantee interoperability when interpreting the body of the main ARR segments.

Code list for the Observation Status attribute (CL_OBS_STATUS)

Code_Value	Code_Description
A	Normal value
B	Break
E	Estimated value
F	Forecast value
H	Missing value; holiday or weekend
L	Missing value; data exist but were not collected
M	Missing value; data cannot exist
P	Provisional value
S	Strike

When more than one "condition" occurs for the same observation, then you should use the next table: it indicates the level of importance of each specific "event." (For instance, the information that an observation is a "break" is more important than that it is an "estimate," and you should use the flag B rather than E.)

Observation status hierarchy	Relevant in conjunction with...	
	numeric values	**missing values**
B / break	✓	✓
M / undefined, data cannot exist		✓
L / data not collected		✓
H / holiday or weekend		✓
S / strike	✓	✓
F / forecast value	✓	
E / estimated value	✓	
P / provisional value	✓	
A / normal value	✓	

Code list for the Observation Confidentiality attribute (CL_OBS_CONF)

Code_Value	Code_Description
C	Non-publishable and confidential
F	Free
N	Non-publishable, but non-confidential
R	Confidential statist. information due to identifiable

13. APPENDIX III: FREQUENTLY ASKED QUESTIONS ABOUT GESMES/TS

Which are the most important "quantitative" restrictions applying to a message?

1. More than one message (UNH/UNT) can be present in the same file/interchange (UNB/UNZ).

2. If a message contains statistical data and/or attributes, you can write only one DSI Group (group no. 13) inside this message.

3. If a message contains statistical definitions, then you cannot have a DSI present. And one or more repetitions of Groups VLI (no.4), STC (no.9), and ASI (no.10) can appear in the same message. See also the paragraph below on message types.

4. A message can be either a "deletion" (parameter "6" in STS, see page 112) message or an "update" (parameter "7" in STS) message (not both).

5. A message can be (only one of the following three types):
   - a "structural" message (parameter in BGM segment equal to "73") containing code lists (VLI), concepts (STC), and/or key family definitions (ASI);
   - a normal message containing data/attributes or instructions for deletions (parameter in BGM segment equal to "74", see page 65); or
   - a data set list message (parameter in BGM segment equal to "DSL").

6. Maximum length of fields (GESMES/TS limits):
   - **Code lists.** Identifiers: an..18, code list names: an..70, code values: an..18, code value descriptions: an..350
   - **Statistical concepts.** Identifiers: an..18, concept names: an..70.
   - **Key families** and data sets**.** Identifiers: an..18, key family and data set names: an..70.

7. In practice, center institutions make an effort to keep the length of series keys no longer than 35 characters.

8. You can write up to 999,000 ARR segments (following Group 14) in the same message.

9. In each ARR segment, you can have only one time series key present.

10. You can write up to 9,999 data elements in the same ARR (i.e., if a time series has more than 9,999 observations, you need to be split it over 2 or more ARR segments).

11. You can use up to 15 positions (n..15) inside a data element for writing a numeric value (including the place needed for the minus sign for negative values and/or the decimal point; you must never use the plus sign for positive numbers).

12. The only attributes you can have present in the main ARR segment are the observation status (mandatory coded attribute), the observation confidentiality (conditional coded attribute), and the pre-break value (conditional numeric field).

Can an "update" and a "delete" message be present in the same interchange?

Yes.

Is there a way to distinguish between "reporting new data" and "reporting corrections in previously reported data"?

GESMES/TS is not equipped with a specific mechanism for this, and it is left to the receiving application how to process the information contained in a GESMES/TS interchange. If you are a receiving application, you could check—before "writing" on the reception databases—whether each received observation/attribute/time series/sibling group is a new one or a replacement of a previously existing observation/attribute/time series/sibling group. Nevertheless, if in a specific data exchange circuit this was required, the distinction could be served by the second (conditional) DTM segment, which carries the "reporting period" information. For example, you could use this segment only in a message carrying new data (a reporting period could be anyway more relevant in this case).

Can I use spaces at the end of strings in the FTX segment?

No, as this could cause interpretation problems. According to the EDIFACT rules, insignificant trailing spaces are eliminated from alphabetic or alphanumeric component elements.

For example, the segment

```
FTX+ACM+++Test kf for BoP Statistics  '
```

is wrong, because it has redundant spaces at the end of the string.

The terms "mandatory" and "conditional" are confusing.

Indeed, without referring to a specific context, it is not possible to explain the meaning of the adjectives mandatory and conditional. Their meaning depends on the context in which they are used. The table below provides an overview of their specific meaning in each case.

Context	"mandatory" attributes	"conditional" attributes	Remarks
Data model	The values of mandatory attributes have to become known to partners. From a statistical point of view they are considered *essential* pieces of information to interpret the data.	The values of conditional attributes have to become known to partners, *if* they are or *when* they become available. From a statistical point of view they are considered *important* pieces of information to interpret the data.	The definition about which attributes are mandatory and which ones are conditional is provided by the key family definition.
Exchange: Attributes exchanged using the FNS group	Attribute values are exchanged the first time and then again every time they change ("updates and revision" principle).		
Exchange: attributes exchanged as elements of the ARR segment (and defined at the observation level): - Observation Status - Observation Confidentiality - Pre-break value	A value for the observation status attribute (OBS_STATUS) should be given with the exchange of every single observation, even if its value does not change. A need to update either the observation value or the observation status (or both) must imply (at least) the reporting of both the observation and the observation status.	Values for the observation confidentiality and/or the pre-break attributes are given when: - it is relevant to provide such a value(s) - if the obs. value or the observation status change and a relevant value for the observation confidentiality and/or the pre-break observation should be kept - the values of the observation confidentiality and/or pre-break observation need to be updated	A value for the observation status attribute (OBS_STATUS) should be given with the exchange of every single observation, even if its value does not change.
GESMES/TS segments EDIFACT segments	A segment is mandatory if it must be used in a message. Sometimes this depends on whether a previous segment or group is present. However, there are segments that must be present in any message (e.g. UNB, BGM).	A segment is conditional if there is the option not to use it. For example, for a message including numeric data (DSI and ARR present), there is the option to include or not to include the segments carrying attributes (FNS segment and other segments of the same group).	

14. APPENDIX IV: EXCHANGING LISTS OF DATA SETS

This appendix describes how you can use a special GESMES/TS message to communicate the list of data sets used in a data-exchange context. The message allows you to send the following information about the data sets in a structured form:

- data set identifier
- data set name
- identifier of the key family the data sets take their structure from

The message is identified by a special code in the BGM section so that it can be easily identified.

Example:

We assume that the ECB would like to communicate to its data-exchange partners information about the following data sets that they all (ECB and its partners) should use in exchanges. The first three sets use the key family ECB_BOP, which the ECB administers. The other data sets use the key family BIS_MACRO, which the BIS administers.

We assume further that, using non-GESMES/TS means, the center and its partners have agreed upon and communicated among themselves any additional information (e.g., details about the exact content of the various data sets).

Data set identifier	Data set name	Key family identifier	Key family maintained by
ECB_BOP_M	ECB Balance of payments, monthly data	ECB_BOP	ECB
ECB_BOP_Q	ECB Balance of payments, quarterly data	ECB_BOP	ECB
ECB_BOP_T	ECB Balance of payments, test data	ECB_BOP	ECB
ECB_MACRO_M	ECB Macro series, monthly data	BIS_MACRO	BIS
ECB_MACRO_Q	ECB Macro series, quarterly data	BIS_MACRO	BIS

You can transmit the information included in the table above as a GESMES/TS message. We implicitly assume that the sending organisation of a message of a list of data sets is the agency defining the included data sets. (In practice, usually, these definitions need to reflect mutual agreements between the sender and the receiver.)

We interpret the data set list in a similar way as a code list—you send it using the VLI, CDV, and FTX group. (See corresponding sections for further details). However, in the case of the data set list, you need to include an additional piece of information, namely, the key family identifier, from which the data set takes its structure. This is done via an *additional* IDE segment, that holds this information. This additional segment is defined on the next page. We reproduce a complete example of a message containing a data set list thereafter.

Segment Type: IDE

Segment Type Name: *Identity*

Max. Number of Occurrences: *1*

Status: *Conditional*

Data Element Sequence

IDE+*object-type+identifier+organisation-id'*

Data-Element Usage Rules

local name	usage rules	format	GESMES/TS code values
object-type	Mandatory	an..3	5 - data set structure
Identifier	mandatory the identifier of the key family the data set takes its structure from	an..18	Provided by center institution.
organisation-id list-qualifier maintenance-agency	mandatory the key family maintenance agency	an..18	

Content and General Rules

Use this segment only (and you must use it) in a "list of data sets" message.

This provides the key family identifier, from which the data set (identified in the preceding CDV and FTX segments) takes its structure. The organisation-id refers to the maintenance agency of the key family.

Example:

```
IDE+5+ECB_BOP+ECB'
```

Table 9: GESMES/TS message with data set lists

	Comment
UNA:+.? '	
UNB+UNOC:3+4F0+ZZZ+020626:1803+IREF004071++GESMES/TS'	
UNH+MREF000001+GESMES:2:1:E6'	
BGM+DSL'	Special BGM code: DSL for data set list.
NAD+Z02+ECB'	
NAD+MR+ZZZ'	
NAD+MS+4F0'	
IDE+10+Data set list for ECB-CBs exchanges'	
VLI+CL_DATA_SET_LIST+++List of ECB data sets'	The list of data sets is sent as a code list (using the VLI, CDV, FTX segments).
CDV+ECB_BOP_M' FTX+ACM+++ECB Balance of payments, monthly data' **IDE+5+ECB_BOP+ECB'**	Data set identifier (CDV) Data set name (FTX) Key family and maintenance agency (IDE)
CDV+ECB_BOP_Q' FTX+ACM+++ECB Balance of payments, quarterly data' **IDE+5+ECB_BOP+ECB'**	Data set identifier (CDV) Data set name (FTX) Key family and maintenance agency (IDE)
CDV+ECB_BOP_T' FTX+ACM+++ ECB Balance of payments, test data' **IDE+5+ECB_BOP+ECB'**	Data set identifier (CDV) Data set name (FTX) Key family and maintenance agency (IDE)
CDV+ECB_MACRO_M' FTX+ACM+++ECB_MACRO data set for monthly series' **IDE+5+BIS_MACRO+BIS'**	Data set identifier (CDV) Data set name (FTX) Key family and maintenance agency, in this case the agency is the BIS
CDV+ECB_MACRO_Q' FTX+ACM+++ECB Macro series, quarterly data' **IDE+5+BIS_MACRO+BIS'**	Data set identifier (CDV) Data set name (FTX) Key family and maintenance agency, in this case the agency is the BIS
UNT+000023+MREF000001'	
UNZ+000031+IREF004071'	

15. APPENDIX V: EXCHANGING DATA SETS AND DUPLICATES

The data exchange model states:

"Within the ETS (Exchanged Time series) a time series is uniquely identified by a data set identifier combined with the time series key."

All series in the ETS in practice come from a database where they are stored and (after having been exchanged) will end up stored in the recipient's database. One cannot exclude that, starting from one series in the source database this may be duplicated and reach as two different series (with the same key) through two different "data sets" the receiver's database. Such a possibility could not be excluded given that sometimes data sets simulate "paper reports" which, sometimes, may foresee the inclusion of a few (same) series in more than one data sets.

In general, there are four possibilities and two or more DSIs (using the same key family) can either:

1. have overlapping content with respect to data and series keys
2. have overlapping content with respect to series keys but not with respect to the data
3. be mutually exclusive on time series and sibling group level
4. be mutually exclusive on time series but not on sibling group level

Here are some examples:

Case 1: overlapping content with respect to data and series

Let us assume two data sets called BOP_SHORT and BOP_LONG. Both contain series on Balance of Payments statistics, BOP_SHORT only the main aggregates, BOP_LONG the aggregates as well as the components. Series TS1 (Current Account Net) shows up in both DSIs with the same data. There may be a timing difference: BOP_SHORT is sent out (received) first and some days later BOP_LONG arrives, containing again all series from BOP_SHORT but also the component details.

Possible processing issue: The recipient of these two data sets may want to continue to keep the series contained in them separate and create 2 separate instances of series TS1. The different DSIs allow for that. However, another recipient may wish to create only one unique instance of TS1, which is updated by any information coming in either via BOP-SHORT or BOP_LONG.

Case 2: overlapping content with respect to series keys but not with respect to the data

Let us assume two data sets called BUDGET_DATA and BUDGET_FORECAST. BUDGET_DATA contains the actual budget monthly return data whereas BUDGET_FORECAST contains the same series keys, but with the forecast of the monthly budget expenditure.

Possible processing issue: It is likely that the recipient of these two data sets will want to keep their contents separate, ie create 2 instances of the time series contained in the two data sets, one with the return data and one with the forecast data. It is up to the recipient to decide how this is actually done in the local database environment.

Case 3: no overlap at all; mutually exclusive on time series and sibling level

Let us assume two data sets called BOP_FRANCE and BOP_ITALY. Each of them contains the Balance of payments series for the country indicated in the DSI. Assuming that the individual series key contain a country identifier we will have a series key such as M:CURRACC:FR in BOP_FRANCE and series like M:CURRACC:IT in BOP_ITALY. There is no overlap between the two data sets. They are mutually exclusive on the time series and also on the sibling group level.

Possible processing issue: The recipient is free to do whatever s/he prefers: store the data in different (in this case "country") databases or ignore the DSI and store the series into one single database.

Case 4: mutually exclusive on time series but not on sibling level

Another example: BOP_MONTHLY and BOP_QUARTERLY. The first DSI contains only monthly series, the second one only quarterly series. These two data sets are mutually exclusive on the time series level, but not on the sibling group level.

Possible processing issue: Here, from a storage point of view, a sibling group may extend across two data sets. However, the recipient may want to bring the monthly and the quarterly series together as a single sibling in his storage environment. The data model states: "Within an ETS a sibling group is uniquely identified by a data set identifier combined with the sibling group key." One has to assume that series sibling SG1 (with a monthly series) in DSI BOP_MONTHLY and sibling SG1 (with a quarterly series) in BOP_QUARTERLY are each accompanied by the (same) sibling level attributes that are defined by the single KF the two DSIs take their structure from.

In cases 1, 3 and 4 we indicated that the recipient could use the DSI information to store the received data in different areas of the local database, i.e. create different "instances" of a time series or sibling that may have shown up in more than one of the data sets. Alternatively he could disregard the DSI and only create one single instance of the time series, which is updated by any incoming DSI, in which it shows up. Case 2 is different: here the recipient needs to keep the DSIs separate as otherwise he might be updating actual return data with forecast data.

Conclusion: some recipients (for their storage considerations) may need to know into which case the data sets they receive belongs to. Therefore, when defining data sets, the center institution needs to be clear about the type of data sets is defining, i.e. which of the cases described above applies. If that is the case, then this information should be communicated to the partners, with which the center plans to conduct these data exchanges. This communication could take the form of a document or a phone call.

16. APPENDIX VI: MAP OF ISO 8859-1 (UNOC) CHARACTER SET (LATIN 1 OR "WESTERN")

- Standard character set (decimal character codes 32-126):

	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

- Extended character set (decimal character codes 160-255):

	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

- Code positions 128 - 159 are reserved for control purposes only (you should not use them in messages).

17. APPENDIX VII: EDIFACT CODE LISTS USED IN GESMES/TS

The table below contains the relevant EDIFACT code values used in GESMES/TS segments. Please note that the provided code lists are not complete, because they list only those values used in GESMES/TS. The entries are sorted alphabetically by segment.

ATT segment 9017 - Attribute Qualifier (domain)

3 array structure component

ATT segment Attribute Qualifier (type)

5 presentation
35 usage status
32 object link type (attachment level)

ATT segment Attachment level code

1 data set
4 time series
5 observation
9 sibling group

ATT segment Code list qualifier

ALV attachment level
USS usage status

ATT segment Attribute status

1 Conditional
2 Mandatory

BGM segment Message function

73 statistical definitions
74 statistical data
DSL list of data sets

COM segment 3155 - Communication channel qualifier

EM electronic mail
TE telephone
FX telefax
XF X.400

CTA segment 3139 - Contact Function, Coded

CP Responsible person for computer data processing
CF Head of unit for information production
CC Responsible person for information production
CG Head of unit for information dissemination

DTM segment 2005 - Date/time/period qualifier

242 preparation date
Z02 reporting period

DTM segment 2379 - date/time/period format qualifier

101 YYMMDD
102 CCYYMMDD
201 YYMMDDHHMM
203 CCYYMMDDHHMM
602 CCYY
604 CCYYS
608 CCYYQ
610 CCYYMM
616 CCYYWW
702 CCYY-CCYY
704 CCYYS-CCYYS
708 CCYYQ-CCYYQ
710 CCYYMM-CCYYMM
711 CCYYMMDD-CCYYMMDD
716 CCYYWW-CCYYWW

FTX segment 4451 - Text Subject Qualifier

ACM statistical description

IDE segment Identification Qualifier

1 value list
4 code value
5 data set structure
10 message context
Z10 coded attribute
Z11 uncoded attribute

NAD segment Party type

MS Message sender
MR Message receiver
Z02 Maintenance agency for codes

REL segment 9141 - Relationship Qualifier

Z01 the relationship is with a statistical array

SCD segment 7497 - Component function qualifier

1 array time dimension
3 array cell
4 array dimension (other than time dimension)
13 dimension is “frequency”
Z09 attribute

STS segment 9011 - Status event, coded

6 Delete
7 add and replace

VLI segment C780 - Value list identification

3 Coded list

18. APPENDIX VIII: WHAT WAS NEW IN GESMES/CB RELEASE 2.0 VIS-À-VIS RELEASE 1.4

- We inserted the description of the data model (definitions, principles, rules) in the guide for 2.0 (before it was a separate document). In the section dealing with implementation issues of the data model, we added a new paragraph on the role of central institutions.
- We no longer based the identification of partners on a reference to a single-code list. Now, statistical centers administering data exchanges were expected to provide "an appropriate code list" for identifying themselves and their partner institutions. The release introduced the corresponding flexibility to allow more data-exchange circuits (not necessarily using the organisation code list of the initial user groups) to join the GESMES/TS community. Of course, in order to achieve a higher degree of interoperability, the guide still strongly recommended to all statistical centers to gradually converge to a single identification method for identifying organisations, across all data exchange circuits and partners.
- The release increased the format of the "communication number" from AN..25 to AN..512, in order to accommodate long e-mail addresses.
- The user could now describe code values with up to 350 character strings (AN..350); previously it was "AN..70".
- The release added a new (mandatory) FTX segment under the ASI group. It allowed center institutions to provide a textual description for each key family that they administered.
- The release introduced an implementation remark in the section describing the DSI segment.
- It added a new (conditional) segment—it gave the possibility, upon the center's request, of providing the reporting period in a code (DTM segment).
- We added four appendices:
  - a presentation of the code lists currently used for the observation status and observation confidentiality attribute (this was necessary, as several examples in the guide used their code values);
  - a list of frequently asked questions;
  - a map of the ISO 8859-1 character set; and
  - a table presenting the EDIFACT code lists used in GESMES/TS.

INDEX