



# **SDMX USER GUIDE**

## **Getting Started**

**VERSION 0.3**

February 2006



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35

© SDMX 2005

<http://www.sdmx.org/>



## Contents

36		
37	1	GETTING STARTED..... 4
38	1.1	Overview of this section..... 4
39	1.2	What is SDMX? ..... 4
40	1.3	What are the goals of SDMX? ..... 4
41	1.4	When can SDMX be used? ..... 5
42	1.5	What kinds of metadata can be exchanged with SDMX ..... 6
43	1.6	First steps in using SDMX ..... 6
44	1.7	How to structure data ..... 9
45	1.8	Understanding data structure definitions..... 12
46	1.9	The different kinds of standard messages ..... 12
47	1.10	Deriving one SDMX-ML message from another ..... 14
48	1.11	Including reference metadata in messages ..... 15
49	2	ANNEX 1: KEY FAMILIES – A TUTORIAL ..... 16
50	2.1	Introduction..... 16
51	2.2	What is a Key Family?..... 16
52	2.3	Grouping Data ..... 17
53	2.4	Attachment Levels ..... 18
54	2.5	Keys..... 18
55	2.6	Code Lists and Other Representations ..... 19
56	2.7	Cross-Sectional Data Structures ..... 20
57		



## 58 **1 GETTING STARTED**

### 59 **1.1 Overview of this section**

60 This section of the User Guide aims to take potential users of SDMX through the first  
61 steps in using SDMX to structure data and metadata. It forms a simple introduction to  
62 the basics of SDMX, which is intended to be understandable for statisticians and  
63 other users who have not been involved in the development of the SDMX standards  
64 and tools, but are interested in applying them.

### 65 **1.2 What is SDMX?**

66 The name “Statistical Data and Metadata eXchange”<sup>1</sup> refers to an international  
67 initiative aimed at developing and employing more efficient processes for exchange  
68 and sharing of statistical data and metadata among international organisations and  
69 their member countries. The initiative, started in 2001, is sponsored by 7  
70 international organisations: Bank for International Settlements (BIS), European  
71 Central Bank (ECB), Eurostat, International Monetary Fund (IMF), Organisation for  
72 economic Co-operation and Development (OECD), United Nations (UN) and the  
73 World Bank (WB), who are committed to establish, implement and comply with  
74 common standards.

75 The SDMX version 1.0 set of technical standards has been approved by ISO as a  
76 technical specification (TS17369 2005). Version 2.0 has been publicly released in  
77 November 2005 and its approval process by ISO is ongoing.

### 78 **1.3 What are the goals of SDMX?**

79 The rationale of SDMX is standardisation for statistical data and metadata access  
80 and exchange.

81 With the ever increasing ease of use of the Internet, the electronic exchange and  
82 sharing of data is becoming more and more easy, frequent and important. This  
83 stresses the need for a set of common standards for exchange and sharing of  
84 statistical data and metadata, and making processes more efficient. As statistical  
85 data exchange takes place continuously, the gains to be realised from adopting  
86 common standards are considerable both for data providers and users.

87 The objective is to establish a set of commonly recognised standards, adhered to by  
88 all players, making it possible not only to have easy access to statistical data,  
89 wherever these data may be, but also access to metadata that makes the data more  
90 meaningful and usable. The standards will allow national organisations to fulfil their  
91 responsibilities towards users and partners, including international organisations, in a  
92 very efficient way, among other things by using their general online databases to give  
93 access as soon as the data are released.

94 The SDMX standards also aim to ensure that appropriate metadata always come  
95 along with the data, making the information immediately understandable and useful.  
96 For this reason, standards for metadata exchange are extremely important in SDMX.

---

<sup>1</sup> See [www.sdmx.org](http://www.sdmx.org)

97 **1.4 When can SDMX be used?**

98 As mentioned above, the SDMX standards are designed for exchange or sharing of  
99 statistical information between two or more partners<sup>2</sup>. Evidently, the SDMX  
100 standards have been developed by the sponsors in order to accommodate the  
101 constituencies of the sponsoring organisations (national statistical offices, central  
102 banks, ministries, etc.). Within and across these constituencies, the standards are  
103 intended for reporting (or sharing) statistical data and metadata in the most efficient  
104 way.

105 SDMX standards can also be used within a national system for transmitting or  
106 sharing statistical data and metadata and by private data providers (such as re-  
107 sellers of statistical databases). This is particularly interesting in countries with a  
108 federal structure or a fairly decentralised statistical system. In such cases, a close  
109 link can be established between the national system for data sharing and the  
110 international ones, allowing for additional efficiency gains for the involved  
111 organisations.

112 If data are made available for exchange using the pull mode (see Box 1) according to  
113 SDMX standards, this could easily evolve to open SDMX-based dissemination; such  
114 dissemination may respond well to user demands for well-structured data and  
115 metadata in reuseable formats, and should be considered as an option for national  
116 authorities as well as international organisations.

117 **BOX 1**

118 **Push and pull**

119 Messages can be exchanged in two different modes, the push mode and the pull  
120 mode:

121 **Push** mode means that the data provider takes action to send the data to the party  
122 collecting the data. This can take place using different means, such as e-mail or file  
123 transfer, and in some cases the transfer can be supported by systems such as  
124 Eurostat's Stadium and Statel. These are the "traditional" modes of data collection,  
125 carried out by international organisations for many years.

126 **Pull** mode implies that the data provider makes the data available via the Internet.  
127 This may be as simple as placing a structured (SDMX-ML) file on a website or it may  
128 involve accessing a database service, available via the web and capable of  
129 processing a standard SDMX query. The data collector then fetches the data on his  
130 own initiative. In this case, more than one data collector may be allowed to take the  
131 pieces of data needed by each collector. This mode also resembles dissemination in  
132 the sense that access might be given to final users of information, who will then,  
133 according to their needs, access multiple web sites all using the same formats.

134 While all combinations of the modes above are supported by SDMX standards, it is  
135 the aim of the SDMX initiative to further promote data sharing exchange using the  
136 pull mode.

---

<sup>2</sup> This is further described in the document Framework for SDMX standards (version 1.0). See  
<http://www.sdmx.org/news/document.aspx?id=125&nid=49>

## 137 **1.5 What kinds of metadata can be exchanged with SDMX**

138 Statistical metadata (defined as “data about statistical data”) provide information on  
139 data and about processes involved in the production and usage of data. SDMX can  
140 in principle be used for all kinds of metadata.

141 SDMX supports the use of a common terminology when exchanging and sharing  
142 data. For this purpose, SDMX comprises a Metadata Common Vocabulary (MCV)<sup>3</sup>.  
143 The MCV contains a comprehensive set of definitions of metadata terms used for  
144 describing the statistics and processes used in their compilation by national agencies  
145 and international organisations.

146 SDMX metadata standards build on the distinction between “structural” and  
147 “reference” metadata (see Box 2).

### 148 **BOX 2**

#### 149 **Structural metadata and reference metadata**

150 **Structural metadata** are those metadata acting as identifiers and descriptors of the  
151 data, such as names of variables or dimensions of statistical cubes. Structural  
152 metadata must be associated with the data, otherwise it becomes impossible to  
153 identify, retrieve and browse the data.

154 **Reference metadata** are metadata that describe the contents and the quality of the  
155 statistical data (conceptual metadata, describing the concepts used and their  
156 practical implementation, methodological metadata, describing methods used for the  
157 generation of the data, and quality metadata, describing the different quality  
158 dimensions of the resulting statistics, e.g. timeliness, accuracy). While these  
159 reference metadata exist and may be exchanged independent of the data and its  
160 structural metadata, they are often linked (“referenced”) to the data.

161 The idea is that it should be possible, using the SDMX standards, to exchange or  
162 share the data and the metadata that will allow a thorough understanding and  
163 interpretation of the corresponding statistical data.

## 164 **1.6 First steps in using SDMX**

165 The advantages of SDMX can best be understood by using it. The following sections  
166 explain

- 167 • How to structure data using a data structure definition (sometimes known as a  
168 “key family”)
- 169 • The different kinds of standard messages, what they are used for and how  
170 they inter-relate.

171 The examples and explanations given here all use SDMX-ML, which is the XML-  
172 based version of the SDMX standards. SDMX makes extensive use of XML and  
173 some related technologies (see Box 3). It is not necessary to be an XML expert, but  
174 a basic understanding of these technologies is essential for working with SDMX.

175 It should be possible to open any of the XML example files in a normal web browser.

---

<sup>3</sup> MCV: <http://www.sdmx.org/knowledge/document.aspx?id=66>

176

177 **BOX 3**178 **XML technologies for SDMX**179 **XML**

180 The eXtensible Markup Language

181 Designed to describe data and what data is

182 **Schema**183 Describes the structure of an XML document in terms of a Schema Definition  
184 language (XSD)185 **XSLT**186 The eXtensible Stylesheet Language describes how an XML file can be transformed  
187 into another format

188

189 **XML carries data**

```
190 <note date="12/05/2005">  
191 <to>Bernhardt</to>  
192 <from>Giuseppe</from>  
193 <heading>Reminder</heading>  
194 <body>Don't forget the slides!</body>  
195 </note>
```

196

- 197 • It is eXtensible
- 198 • It does not do anything
- 199 • It can be used to exchange and share data
- 200 • It can be used to create new languages

201 **Validation**

```
202 <?xml version="1.0"?>  
203 <note  
204 xmlns=http://eurostat.cec.eu.int  
205 xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance  
206 xsi:schemaLocation="http://eurostat.cec.eu.int/note.xsd">  
207 ...  
208 </note>
```

209

- 210 • A schema defines all valid elements
- 211 • The schema is contained in a separate file (.xsd)
- 212 • Errors will stop an XML application
- 213 • Example: browsers parse and visualise XML files
- 214 • Use of stylesheets (XSL or CSS)

215

## 216 XML Schema Definition (note.xsd)

```

217 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
218   targetNamespace="http://eurostat.cec.eu.int"
219   xmlns="http://eurostat.cec.eu.int" elementFormDefault="qualified">
220
221   <xs:element name="note">
222     <xs:complexType>
223       <xs:sequence>
224         <xs:element name="to" type="xs:string"/>
225         <xs:element name="from" type="xs:string"/>
226         <xs:element name="heading" type="xs:string"/>
227         <xs:element name="body" type="xs:string"/>
228       </xs:sequence>
229     </xs:complexType>
230   </xs:element>
231 </xs:schema>
  
```

## 233 Namespaces

```

234 <?xml version="1.0"?>
235 <note
236   xmlns=http://eurostat.cec.eu.int
237   xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
238   xsi:schemaLocation="http://eurostat.cec.eu.int note.xsd">
239
  
```

- 240 • Name conflicts:
- 241 • Same name for different elements in different files
- 242 • This is a prefix for any name in the file
- 243 • Is not a place where a validation file can be found

## 245 eXtensible Stylesheet Language Transformations

- 246 • XSL = XML Style Sheet Language
- 247 • XSL describes how the XML document should be displayed
- 248 • XSLT - a language for transforming XML documents
- 249 • Transforms an XML document into another XML document
- 250 • Parts of the source document that match one or more predefined templates are
- 251 transformed into the result document

## 253 An XML file

```

254 <?xml version="1.0" encoding="ISO-8859-1"?>
255 <?xml-stylesheet type="text/xsl" href="cdcatalog.xsl"?>
256 <catalog>
257   <cd>
258     <title>Amandla</title>
259     <artist>Miles Davis</artist>
260     <country>USA</country>
261     <company>Noadvertshere</company>
262     <price>10.90</price>
263     <year>1989</year>
  
```

```

264 </cd>
265 . . .
266 </catalog>
267
268 XML to HTML via XSLT
269 <?xml version="1.0" encoding="ISO-8859-1"?>
270 <xsl:stylesheet version="1.0"
271     xmlns:xsl="http://www.w3.org/1999/XSL/Transform"> <xsl:template
272 match="/">
273     <html>
274         <body>
275             <h2>My CD Collection</h2>
276             <table border="1">
277                 <tr bgcolor="#9acd32">
278                     <th align="left">Title</th>
279                     <th align="left">Artist</th>
280                 </tr>
281                 <xsl:for-each select="catalog/cd">
282                     <tr>
283                         <td><xsl:value-of select="title"/></td>
284                         <td><xsl:value-of select="artist"/></td>
285                     </tr>
286                 </xsl:for-each>
287             </table>
288         </body>
289     </html>
290 </xsl:template>
291 </xsl:stylesheet>
  
```

## 292 **1.7 How to structure data**

293 A typical time series might identify a data value using the following dimensions:

Dimensions (concepts)	Example (dimension value)
1 Frequency	M
2 Reference area/country	BE
3 Adjustment	W
5 Indicator	STS_PROD
6 Activity	NS0020

294

295 A time series of five monthly values could be represented in SDMX-ML in the  
 296 following message.

297 The first part of the message contains information about the XML namespaces and  
 298 schemas used (see Box 3). It should be noted that these are identified using names  
 299 which look like URLs for web sites, but this is just a convention: these are unique  
 300 identifiers which in most cases do not point to real web sites.

```

301 <?xml version="1.0" encoding="UTF-8"?>
302 <CompactData
303     xmlns="http://www.SDMX.org/resources/SDMXXML/schemas/v1_0/message"
304     xmlns:eurostat_sts="http://epp.eurostat.cec.eu.int/portal/structure/key_families/EUROSTAT_STS/compact"
305     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  
```



```
307 xsi:schemaLocation="http://www.SDMX.org/resources/SDMXML/schemas/v1_0
308 /message SDMXMessage.xsd
309 http://epp.eurostat.cec.eu.int/portal/structure/key_families/EUROSTAT
310 _STS/compact EUROSTAT_STS_Compact.xsd">
311
```

312 The next part of the message is the header, which identifies the particular data being  
313 sent in this case, including the sender and receiver.

```
314 <Header>
315 <ID>SODI_IPI_M_EU_2005_0001_V0001_N</ID>
316 <Test>>true</Test>
317 <Name xml:lang="en">EUROSTAT_STS_STS_IND_PROD_M_BE_SODI_Test</Name>
318 <Prepared>2005-03-15T13:44:17+01:00</Prepared>
319 <Sender id="BE">
320 <Name>Institut National de Statistiques de Belgique</Name>
321 </Sender>
322 <Receiver id="EUROSTAT">
323 <Contact>
324 <Name>Leonhard Maqua</Name>
325 <Department>Statistical Information Technologies</Department>
326 <Email>Leonhard.Maqua@ xxx.xxx.xxx</Email>
327 </Contact>
328 <Contact>
329 <Name>Bengt-Ake Lindblad</Name>
330 <Department>Statistical Information Technologies</Department>
331 <Email>Bengt-Ake.Lindblad@xxx.xxx.xxx</Email>
332 </Contact>
333 <Contact>
334 <Name>Giuseppe Sindoni</Name>
335 <Department>Statistical Information Technologies</Department>
336 <Email>Giuseppe.Sindoni@ xxx.xxx.xxx</Email>
337 </Contact>
338 </Receiver>
339 <DataSetAgency>EUROSTAT</DataSetAgency>
340 <DataSetID>SODI_IPI_M</DataSetID>
341 <DataSetAction>Update</DataSetAction>
342 <Extracted>2005-03-15T13:44:17+01:00</Extracted>
343 </Header>
344
```

345 The third part of the message contains the data values, or “observations” in SDMX  
346 terminology. In this example, there are two related time series, each with five values.

```
347 <eurostat_sts:DataSet>
348 <eurostat_sts:SiblingGroup REF_AREA="BE" ADJUSTMENT="W"
349 STS_INDICATOR="PROD" STS_ACTIVITY="NS0020" STS_INSTITUTION="1"
350 STS_BASE_YEAR="2000" UNIT="BEL" UNIT_MULT="0" DECIMALS="2"
351 TITLE_COMPL="Elements of the full national title, or sub-title, which
352 are not included in the title or which appear there in a
353 significantly modified or abbreviated form.usage: In English"/>
354 <eurostat_sts:SiblingGroup REF_AREA="BE" ADJUSTMENT="Y"
355 STS_INDICATOR="PROD" STS_ACTIVITY="NS0020" STS_INSTITUTION="1"
356 STS_BASE_YEAR="2000" UNIT="BEL" UNIT_MULT="0" DECIMALS="2"
357 TITLE_COMPL="Elements of the full national title, or sub-title, which
358 are not included in the title or which appear there in a
359 significantly modified or abbreviated form.usage: In English"/>
360 <eurostat_sts:Series FREQ="M" REF_AREA="BE" ADJUSTMENT="W"
361 STS_INDICATOR="PROD" STS_ACTIVITY="NS0020" STS_INSTITUTION="1"
```

```

362 STS_BASE_YEAR="2000" TIME_FORMAT="P1M" COLLECTION="A"
363 AVAILABILITY="A">
364 <eurostat_sts:Obs TIME_PERIOD="2004-10" OBS_VALUE="113.30"
365 OBS_STATUS="A" OBS_CONF="F"/>
366 <eurostat_sts:Obs TIME_PERIOD="2004-09" OBS_VALUE="111.75"
367 OBS_STATUS="A" OBS_CONF="F"/>
368 <eurostat_sts:Obs TIME_PERIOD="2004-08" OBS_VALUE="97.06"
369 OBS_STATUS="A" OBS_CONF="F"/>
370 <eurostat_sts:Obs TIME_PERIOD="2004-07" OBS_VALUE="97.49"
371 OBS_STATUS="A" OBS_CONF="F"/>
372 <eurostat_sts:Obs TIME_PERIOD="2004-06" OBS_VALUE="110.40"
373 OBS_STATUS="A" OBS_CONF="F"/>
374 </eurostat_sts:Series>
375 <eurostat_sts:Series FREQ="M" REF_AREA="BE" ADJUSTMENT="Y"
376 STS_INDICATOR="PROD" STS_ACTIVITY="NS0020" STS_INSTITUTION="1"
377 STS_BASE_YEAR="2000" TIME_FORMAT="P1M" COLLECTION="A"
378 AVAILABILITY="A">
379 <eurostat_sts:Obs TIME_PERIOD="2004-10" OBS_VALUE="105.71"
380 OBS_STATUS="A" OBS_CONF="F"/>
381 <eurostat_sts:Obs TIME_PERIOD="2004-09" OBS_VALUE="105.73"
382 OBS_STATUS="A" OBS_CONF="F"/>
383 <eurostat_sts:Obs TIME_PERIOD="2004-08" OBS_VALUE="105.14"
384 OBS_STATUS="A" OBS_CONF="F"/>
385 <eurostat_sts:Obs TIME_PERIOD="2004-07" OBS_VALUE="105.75"
386 OBS_STATUS="A" OBS_CONF="F"/>
387 <eurostat_sts:Obs TIME_PERIOD="2004-06" OBS_VALUE="104.40"
388 OBS_STATUS="A" OBS_CONF="F"/>
389 </eurostat_sts:Series>
390 </eurostat_sts:DataSet>
391 </CompactData>
392

```

393 In order to understand this example, look at the definitions in Box 4.

### 394 BOX 3

395 **time series:** A set of ordered observations on a quantitative characteristic of an  
 396 individual or collective phenomenon taken at different points of time.

397 **observation:** The value, at a particular period, of a particular variable.

398 **statistical concept:** a characteristic of an observation. It can be coded (= taking  
 399 values from a code list) or uncoded

400 **group of sibling time series:** the group of time series referring to exactly the same  
 401 economic phenomenon as expressed in different frequencies

402 **dimension:** A statistical concept used, in combination with other statistical concepts,  
 403 to identify a statistical series or single observations (e.g. frequency, economic  
 404 phenomenon, reference area, other concepts)

405 **attribute:** statistical concept qualifying observations, time series or datasets (e.g.  
 406 title, availability, observation status, units, title in national language, source). In  
 407 general terms, the attribute is a characteristic of an object or entity.

408 **key:** set of values taken by the dimensions describing a specific time series

409 **data structure definition (sometimes called a key family):** Set of structural  
 410 metadata associated to a data set, which include information about how concepts are  
 411 associated with the measures, dimensions, and attributes of a data cube, along with  
 412 information about the representation of data and related descriptive metadata.

413 a **series key** uniquely identifies a series within the data set which it belongs to  
 414 **key family identifier**: a mnemonic to refer to a key family

415 The data structure definition for this message is contained in another message called  
 416 a **structure definition message**. As this message is very much longer than the  
 417 message with the data, it is provided as a linked file [<EUROSTAT\\_ST5.xml>](#).

## 418 **1.8 Understanding data structure definitions**

419 The idea of a data structure definition – also known as a key family – is fundamental  
 420 to SDMX. For a more extensive explanation, see Annex 1 which contains a tutorial  
 421 on key families which is taken from the SDMX Version 2 standards documents. The  
 422 Version 2 package contains more explanatory material and example files, so you  
 423 should download it from the SDMX web site (<http://www.sdmx.org>) and use it to  
 424 explore further.

## 425 **1.9 The different kinds of standard messages**

426 In order to appreciate the flexibility and elegance of SDMX, it is advisable to look at  
 427 the whole collection of SDMX-ML messages and to understand how they relate to  
 428 each other. There are six standard messages:

	Name of message	Short description	Schema file
1	Structure Definition Message	Contains a data structure definition	Fixed
2	Generic Data Message	Conveys data in a form independent of a data structure definition	Fixed
3	Compact Data Message	Exchange of large data sets in a data structure definition-dependent form	Derived from data structure definition
4	Utility Data Message	For schema-based functions in a data structure definition-dependent form	Derived from data structure definition
5	Cross-sectional Data Message	Exchange of many observation types in a data structure definition-dependent form	Derived from data structure definition
6	Query message	To query a database to obtain an SDMX-ML message as the result	Fixed

429

430 Any SDMX-ML message is constructed according to an XML schema (contained in a  
 431 file with the extension .xsd). In some cases the schema file is already fixed in the  
 432 SDMX standards. In other cases, the schema is derived from the data structure  
 433 definition.

434 The following sections give more explanations for each message type, with links to  
 435 examples.

### 436 **1.9.1 Structure definition message**

- 437 • Contains a data structure definition: all SDMX-ML message types share this  
 438 common XML expression of the metadata needed to understand and process  
 439 a data set;

- 440 • Supports annotations;
- 441 • Concepts and code lists could also be referenced by the message;
- 442 • This leverages Internet referencing mechanisms;
- 443 • Example: [Short Term Statistics \(STS\)](#).

#### 444 **1.9.2 Generic Data Message**

- 445 • Conveys data in a form independent of a data structure definition;
- 446 • Used when applications receiving the data do not have detailed  
447 understanding of the data set structure before they obtain the data set itself;
- 448 • Not a particularly compact format;
- 449 • All aspects of the data set easily available;
- 450 • Does not provide strict validation between the data set and its structural  
451 definition using a generic XML parser (the parser cannot validate the codes  
452 since they are not contained in the schema). In terms of XML syntax, all  
453 codes and observation values are elements.
- 454 • All key values specified at the series level;
- 455 • Attribute values attached at the same level as in key family;
- 456 • Example: [STS Generic Message](#).

#### 457 **1.9.3 Compact Data Message**

- 458 • Exchange of large data sets in a data structure definition-dependent form;
- 459 • Specific to the data structure definition of the data set it encodes;
- 460 • Follows mappings between constructs in Structure Definition message and  
461 compact format;
- 462 • For exchange of large data sets in XML format;
- 463 • For transmission of partial data sets (incremental updates) and whole data  
464 sets;
- 465 • In terms of XML syntax, all codes and observation values are attributes. The  
466 permissible values of the codes are defined in the schema (which is specific  
467 to the data structure definition) so that a a generic XML parser can be used to  
468 validate a data file against its structural definition.
- 469 • Key values can be at Group level;
- 470 • Example: [STS compact message](#) (this is the example used in section 1.7).

#### 471 **1.9.4 Utility Data Message**

- 472 • This message may be considered a special-purpose message whose usage  
473 is outside the scope of this chapter. It is intended for schema-based functions  
474 in a data structure definition-dependent form;
- 475 • Specific to the key family of the data set;
- 476 • For validation and other XML schema functions;
- 477 • Cannot be used for incremental updates (requires complete data set);
- 478 • Key values at the series level;
- 479 • Example: [STS utility message](#).

#### 480 **1.9.5 Cross-Sectional Data Message**

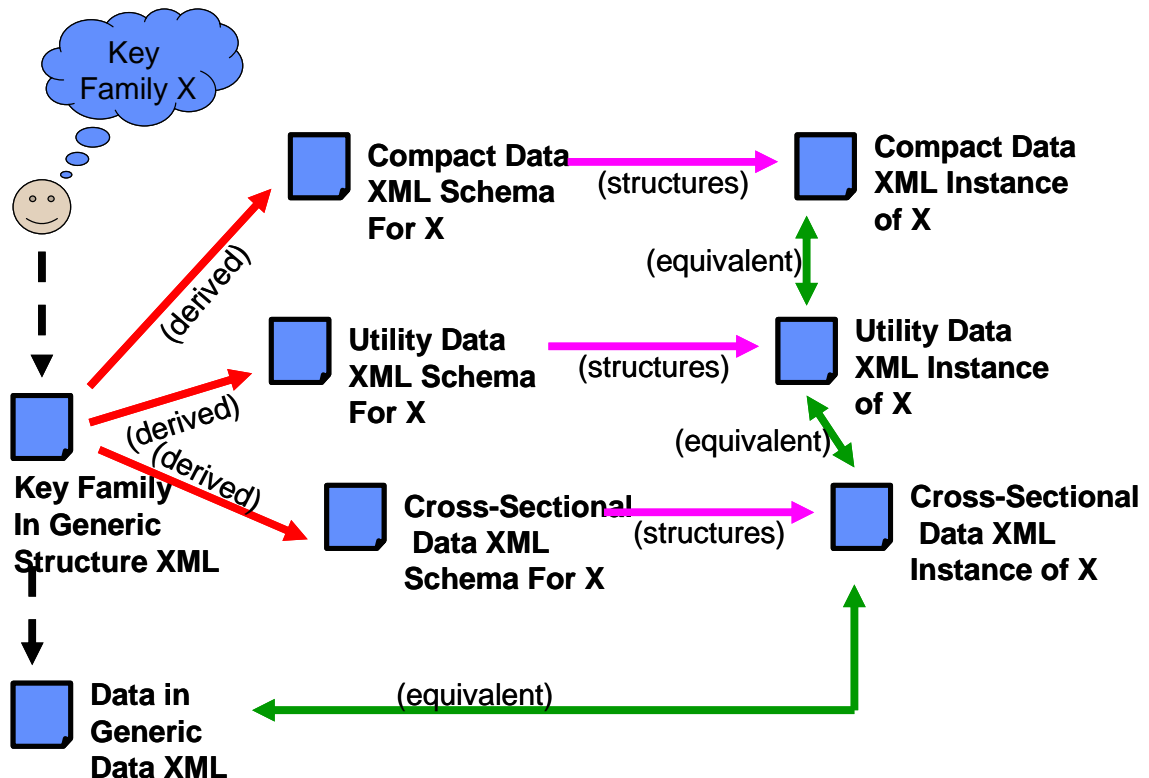
- 481 • Exchange of many observation types in a data structure definition-dependent  
482 form; it is intended for situations where the statistical data consist of multiple  
483 observations at a single point in time;
- 484 • Specific to key family of the data set;
- 485 • Key values from the Group down to the Observation level;
- 486 • Multiple observation values with different “measures”;
- 487 • Time at the Group level;
- 488 • Example: [Joint Debt cross sectional message](#)

#### 489 **1.9.6 Query Message**

- 490 • This is another special-purpose message whose usage is outside the scope  
491 of this chapter. It is used to convey a query to a database which then returns  
492 an SDMX-ML message;
- 493 • For web services and database-driven applications;
- 494 • Queries regard both data and structural metadata;
- 495 • Example: [Joint Debt query message](#).

### 496 **1.10 Deriving one SDMX-ML message from another**

497 Much of the flexibility and elegance of SDMX standards arises from two facts. First,  
498 since all the messages are built on the underlying SDMX Information Model, some  
499 SDMX-ML messages can be derived from others, as shown in Figure 1.

500 **FIGURE 1: Model-based equivalence of SDMX-ML messages**


501

502

503 Second, because the SDMX-ML messages use standard XML, these  
 504 transformations, and any other operations which the user may wish to perform on  
 505 data conveyed with SDMX-ML, can be carried out using normal XML tools, such as  
 506 XML editors and XSLT processors. High-quality tools are available – often for free –  
 507 and well supported with documentation and training.

### 508 **1.11 Including reference metadata in messages**

509 Version 2 of the SDMX standards provides extensive facilities for building SDMX-ML  
 510 messages which contain reference metadata. Have a look at the [Reference](#)  
 511 [Metadata Example](#) to see how such a message is constructed.

## 512 **2 ANNEX 1: KEY FAMILIES – A TUTORIAL**

### 513 **2.1 Introduction**

514 This document is intended to explain "key families" to those who are completely  
515 unfamiliar with the concept. Key families are an important part of the SDMX family of  
516 standards for exchanging statistical data, and they are modelled and explained in  
517 much greater detail in other documents. However, those documents are not written to  
518 explain the basics, and will make difficult reading for those new to the idea. This  
519 document provides a basic tutorial, to help provide the basic level of understanding  
520 needed to make sense of the SDMX standards.

### 521 **2.2 What is a Key Family?**

522 In order to answer this question, we need to look at statistical data. Statistical data is  
523 represented with numbers, such as:

524 17369

525 If you are presented with a number - as above - you will have no idea of what it  
526 actually represents. You know that it is a piece of statistical data, and therefore is a  
527 measurement of some phenomenon - also known as an "observation" - but you can't  
528 tell from the number alone what it is a measurement of. A number of questions come  
529 immediately to mind:

- 530 - What is the subject of the measurement?
- 531 - What units does it measure in?
- 532 - What country or geographical region, if any, does it apply to?
- 533 - When was the measurement made?

534 The list of questions is potentially endless. Behind each of these questions is a  
535 particular idea, or "concept", which is used to describe the data. In our questions  
536 above, these descriptor concepts are Subject, Unit of measure, Country, and Time. If  
537 I tell you the answers to these questions, the data will begin to make sense:

- 538 - the Subject is "total population"
- 539 - the Unit of measure is "thousands of people"
- 540 - the Country is "Country ABC"
- 541 - the Time is "1 January 2001"

542 This is a simplified and fictional example, but it does demonstrate how we can begin  
543 to make sense of statistical data with a set of descriptor concepts. We now know that  
544 our number represents the fact that the total population of Country ABC on 1  
545 January, 2001, was 17,369,000.

546 The simplest explanation of a key family is that it is a set of descriptor concepts,  
547 associated with a set of data, which allow us to understand what that data means.  
548 There is more to it, however.



## 549 **2.3 Grouping Data**

550 Numbers are often grouped together in various ways, to serve as useful packages of  
551 information. One very common approach is to have a set of observations - known as  
552 a "series", or a "time series" - made over time. This allows us to see trends in the  
553 phenomenon being measured. Thus, if I measure the total population in Country ABC  
554 on 1 January of every year, I can see whether the population is growing or declining.

555 A time series always has a "frequency". This is a descriptor concept which describes  
556 the intervals of time between observations. Usually, this is a regular interval, so that  
557 the frequency can be expressed as "annual" or "monthly" or "weekly". Sometimes,  
558 the intervals are irregular. Notice that a single observation does not have a frequency  
559 - only series of observations have frequencies. Frequency is an example of a  
560 descriptor concept which only applies to series of data.

561 There are other, higher-level groupings of data as well. A number of series are often  
562 grouped together into a "Group". Traditionally, the Group was known as a "Sibling  
563 Group", and it contained a set of Series which were identical except that they were  
564 measured with different frequencies. Thus, a given phenomenon would be measured  
565 as daily, monthly, and annually, and these Series, taken together, would be a "Sibling  
566 Group".

567 It is possible to have Groups which have variable values for descriptor concepts  
568 other than frequency, however: if I want to express the US daily exchange rate for all  
569 of the world's currencies over the past year, I have a different kind of group. All of the  
570 "frequency" descriptors would be the same - "daily" - but the descriptor concept  
571 which gives the "foreign currency" would be different for each series.

572 There is also a higher level of package known as a "Data Set". This represents a set  
573 of data that may be made up of several Groups. Typically, it is maintained and  
574 published by an agency, so that it becomes a known source of statistical data.

575 A basic structure is emerging: We have Observations, grouped into Series, which are  
576 grouped into Groups, which are grouped into Data Sets.

577

578 Note: It should be mentioned that there is another way of packaging Observations,  
579 which we call "cross-sectional" data. In cross-sectional data, a large number of  
580 related Observations are presented for a single point or period in time. This  
581 organization of data is very similar to Time Series data in the way a set of descriptor  
582 concepts can be associated with it. A Key Family can be used to describe both cross-  
583 sectional and time series data. For the purposes of this part of the tutorial, however,  
584 we will focus on time series data. Once we have described the Key Family for time  
585 series data, we will go back and see how cross-sectional data are structured.

586

---

### 587 ***What is a key family? (Answer #1)***

588 A key family is a way of associating a set of descriptor concepts with a specific set of  
589 statistical data, as well as a technique for packaging or structuring that set of data  
590 into groups and sub-groups. This is only one way of understanding the structure and  
591 meaning of statistical data, but it provides us with a solid, generic model.

592

---

## 593 **2.4 Attachment Levels**

594 Some descriptor concepts are not meaningful at the Observation level, but only at a  
595 higher level. The example we saw earlier was frequency, which means nothing for a  
596 single Observation, but has meaning when applied to a Series of Observations. This  
597 is because it represents the interval of time between Observations. Time, on the  
598 other hand, is meaningful at the Observation level - every Observation is associated  
599 with a specific point or period in Time. Key families provide information about the  
600 level at which a particular descriptor concept is attached: at the Observation level,  
601 the Series level, the Group level, or the Data Set level. This is known as the  
602 "attachment level" of the descriptor concept.

603 If we think about Groups, particularly, we can see how this works. Within a group,  
604 some descriptor concepts have values that are the same for all Series within the  
605 Group, while other descriptor concepts are changeable. For the Group described  
606 above, of all US exchange rates measured daily for all of the world's currencies, the  
607 descriptor concepts of Subject ("US exchange rate") and Frequency ("daily") will be  
608 the same for all members of the Group. The descriptor concept "Foreign Currency",  
609 however, will change for each Series within the group: there will be a Series for  
610 "Swiss Francs," a Series for the "Euro," a Series for "New Zealand dollars," etc.

611 The rule is that descriptor concepts are "attached" to the grouping level where they  
612 become variable. Thus, if, within a single set of data, all the contents of a Series  
613 share a single value for a descriptor concept, then that descriptor concept should be  
614 attached at the Series level. This rule also assumes that the chosen level is the  
615 highest structural level where all sub-groups will share the same value. (While it is  
616 true that all Series in a Group where the country is "Switzerland" share a single  
617 value, if every Group in the Data Set would always also have the value "Switzerland"  
618 for country, then the attachment level should be the Data Set, not the Group.)

619 Attachment levels of descriptor concepts are always at least at the level where the  
620 concept is meaningful: thus, you cannot attach the descriptor concept frequency at  
621 the Observation level, because as a concept it only operates at the level of Series  
622 (that is, with multiple Observations made over time).

## 623 **2.5 Keys**

624 A "key family" is so called because of the term "key". "Key" refers to the values for  
625 the descriptor concepts which describe and *identify* a particular set of data. Let's take  
626 a simple example:

627 I have a set of statistical data which uses the following descriptor concepts:

- 628 - Time
- 629 - Frequency
- 630 - Topic
- 631 - Country

632 Time is always attached at the Observation level - the value for Time is the time at  
633 which the Observation was made. Time - because it is a concept connected to all  
634 statistical data - does not form part of the key. The other descriptor concepts -  
635 frequency, topic, and country - are all attached at the series level. For any given  
636 Series of Observations, they will all have a single value.



637 If we have a Series of data which is the monthly measurement of the total population  
638 of Country ABC, we will have a key made up of the following values for each  
639 descriptor concept:

640 Frequency = "monthly"

641 Topic = "total population"

642 Country = "Country ABC"

643 This set of values - "Monthly - total population - Country ABC" is the "key" for this  
644 data Series: it identifies what the data is.

645 Keys are most often associated with data at the Series level, but they also exist at  
646 other levels. For example, we could enlarge our example to be a Group including the  
647 monthly total population data for all of the countries in the world. At the Group level,  
648 Frequency would have a value of "monthly", and Topic would have a value of "total  
649 population", but we would not specify the Country descriptor concept, because it  
650 would change from Series to Series. The key for the Group is known as a "Group  
651 Key" - it identifies what the Group is, rather than identifying the Series. (In order to  
652 completely understand the Group, of course, we also need to know which descriptor  
653 concepts are changeable - in this case, Country.)

654 The key values are attached at the Series level, and are given in a fixed sequence.  
655 Frequency is the first descriptor concept, and the other concepts are assigned an  
656 order for that particular data set. This makes it much easier to share and understand  
657 statistical data.

658 If you look back to our initial use of this example, you will notice that we have not  
659 been discussing the "Unit of measure" descriptor concept. This is because the "key"  
660 only contains values for those descriptor concepts which identify the data. If we have  
661 the measurements made in thousands or in millions, the data are the same - they  
662 can be derived from one another by simply multiplying the numbers in the data by the  
663 appropriate conversion factor.

664 This points out a major distinction between the two types of descriptor concepts: the  
665 ones which both identify and describe the data are called "dimensions", and those  
666 which are purely descriptive are called "attributes". Only "dimensions" - that is, the  
667 descriptor concepts which also identify the data - are used in the "key", because the  
668 "key" is fundamentally a way of identifying a set of data.

## 669 **2.6 Code Lists and Other Representations**

670 In order to be able to exchange and understand data, a key family tells us what the  
671 possible values for each dimension are. This list of possible values is known as a  
672 "code list." Each value on that list is given a language-independent abbreviation - a  
673 "code" - and a language-specific description. This helps us avoid problems of  
674 translation in describing our data: the code can be translated into descriptions in any  
675 language without having to change the code associated with the data itself.  
676 Wherever possible, the values for code lists are taken from international standards,  
677 such as those provided by ISO for countries and currencies.

678 As stated, dimensions are always represented with codes. Attributes are sometimes  
679 represented with codes, but sometimes represented by numeric or free-text values.  
680 This is allowed because the attributes do not serve an identification function, but  
681 merely describe the data.

682

---



683 ***What is a key family? (Answer #2)***

684 We now have a more sophisticated understanding of a what a key family does: it  
685 specifies a set of concepts which describe and identify a set of data. It tells us which  
686 concepts are dimensions (identification and description), and which are attributes  
687 (just description), and it gives us an attachment level for each of these concepts,  
688 based on the packaging structure (Data Set, Group, Series, Observation). It also tells  
689 us which code lists provide possible values for the dimensions, and gives us the  
690 possible values for the attributes, either as code lists or as numeric or free text fields.

691

---

692 **2.7 Cross-Sectional Data Structures**

693 Given the explanation of Key Families thus far, we understand that a Key Family  
694 associates descriptor concepts with data, some of which also serve to identify the  
695 data – the “dimension” concepts which make up the Key.

696 Cross-sectional data structures do not apply a different set of concepts to the data:  
697 the same concepts still apply in describing and identifying the data. It attaches the  
698 concepts to the data differently, to create a different presentation of the data.

699 If we go back to our earlier example, we had the following concepts:

- 700 - Time
- 701 - Frequency
- 702 - Topic
- 703 - Country

704 If we want to take a set of data which is described and identified by this set of  
705 concepts, and present it in a cross-sectional fashion, we would not change these  
706 concepts – we would merely change the way in which they are represented – that is,  
707 attached – to the data structure.

708 Take, as an example, the total population of each country in the world on January 1,  
709 2001 as a set of data. In our earlier example, we measured the population of Country  
710 ABC over a period of years – that is, over time. Time was the concept we used to  
711 organize our data in a sequence of observations.

712 If we organize our data to reflect only a single point in time – in this case, January 1,  
713 2001 – then organizing our data over time makes less sense. It is still a possible way  
714 to structure the data, but we may wish to view it as a cross-section.

715 Think about the term “cross-section” – it can be understood to mean a group of  
716 parallel series over time, from which a section is taken, across time. Thus, a cross-  
717 section is created.

718 In our example, it is easy to see how this applies: instead of organizing our data over  
719 time – that is, using the time concept - we are choosing to organize it over the  
720 Country concept. Thus, instead of having a single value for Frequency, Topic, and  
721 Country for all Observations in our series, with a Time value associated with each  
722 Observation, we will have a Country value associated with each Observation, and a  
723 single value for Frequency, Topic, and Time. Instead of calling the group of  
724 Observations a “Series”, we now use the term “Section”.

725 In our earlier example, we had a key which existed mostly at the Series level:



726 Frequency = "monthly"

727 Topic = "total population"

728 Country = "Country ABC"

729 Time – our remaining concept, was associated with the Observations, with a different  
730 value for each one. Thus, we could have a Series which looks like this:

731 January 1, 2001 – 17369

732 February 1, 2001 – 17370

733 March 1, 2001 – 17405

734 For our cross-sectional presentation, we would have most of our key at the Section  
735 level (or, potentially, at a higher level of grouping):

736 Frequency = "monthly"

737 Topic = "total population"

738 Time = "January 1, 2001"

739 With each Observation, we now have a Country value, instead of a Time value:

740 Country ABC = "17369"

741 Country XYZ = "24982"

742 Country HIJ = "37260"

743 In this cross-sectional presentation of our data set, we have chosen to present each  
744 Observation paired with a Country value, taken from our Codelist of values for the  
745 concept Country. Other dimensions could as easily produce a cross-sectional view,  
746 by attaching their values at The Observation level, instead of the values for Country,  
747 as in our example.

748 Because the concepts themselves do not change, but only the way in which they are  
749 attached to the data structure, a single key family can be used to describe both time-  
750 series and cross-sectional presentations.

751 In the version 1.0 SDMX standards, formats are capable of presenting cross-  
752 sectional data for any single dimension concept, as well as presenting the data as a  
753 time series. It is up to the key family creator to select which non-Time concept, used  
754 as a dimension, will serve to organize a cross-sectional presentation. In future  
755 versions, it is possible that more complete support for the possible cross-sectional  
756 views for a key family will be provided.