XML format for statistical import/export (FiStatXML)

Management Summary

Author	Yury Shatz
Version	1.2
Project	FIGIS
Distribution	
Created	22 June 2001 10:05
Saved	01/03/2002 1:52 PM
Printed	18 March 2002 11:35

Revision History		
Date	**Author**	**Summary**
22/06/01	M. Taconet	Comments made saved in document FI-Stat-XML_v1.doc
07/02/02	Y. Shatz	Version for wide distribution
17/03/2002	Y. Jaques	Added DTD Appendix

Contents

1 Purpose of this document

To provide background information to FiStatXML and explain its structure.

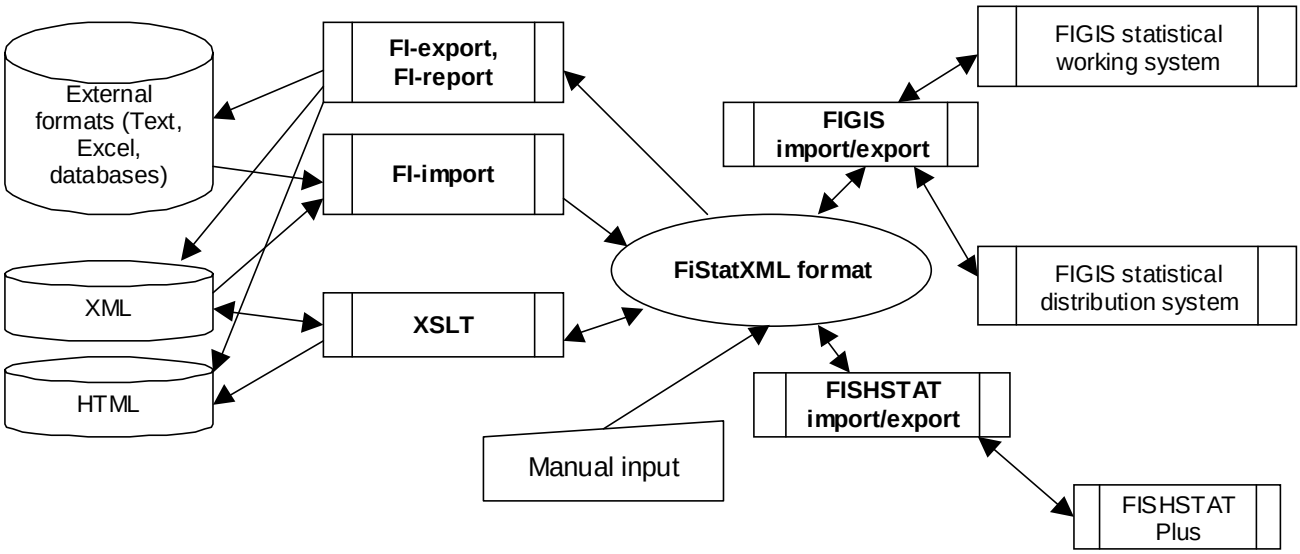
2 What is FiStatXML for?

To provide an intermediary format between FIGIS statistical module and reference tables, Fishstat Plus, external databases and files, reports, and other XML/HTML formats.

The goal is to reduce complexity and to reuse certain components related to data import, export and transformation. The advantages of using a common import/export format are:

- most of the import/export components can be reused between FIGIS, Fishstat and any future system,
- internal import/export functionality of these systems can be made relatively simple, as it only has to process one format.
- any new component to be developed will have access to multitude of import & export functions by other components.

This is the scheme of interaction between the systems



In the context of this document,

- FiStatXML dataset is a file or set of files that define a statistical dataset and follow proposed standard.
- Source system – system that produced the document (possibly converted from some other format). For instance, FI-import
- processing system – system that is processing the document (possibly converting it to some other format). For instance, Fishstat import or FI-export.
- name – usually a combination of names in several languages.

3 What is a FiStatXML dataset?

Everything necessary for creation of a statistical dataset from ground zero. The main four modules are:

1. Time series (TS_DATA) – actual data matrix: keys, values, symbols.
2. Time series metadata (TS_METADATA) – describes structure of part 1, as well as other information about the dataset
3. Reference data (REF_DATA) – what are the reference objects behind the key IDs? What are names, codes, attributes of each and how they relate to each other.
4. Reference metadata (REF_METADATA) – what are types of these key values? What attributes they may have, how to sort them, how to build trees of them, etc.

Additional component that is not actually a part of dataset but may in certain cases affect processing, is:

5. Processing instructions.

The four modules are interrelated in the following way:

- TS_DATA depends on TS_METADATA and REF_DATA;
- TS_METADATA depends on REF_METADATA
- REF_DATA depends on REF_METADATA.

This does not mean the parts cannot be used separately. Depending on the purpose of the transfer and data available on destination system, some parts may be omitted. Indeed, destination system is not always clean slate, it may have data and especially metadata already available. Some examples include:

- Import only REF_METADATA - create object types etc
- Import only REF_DATA - create objects based on object types existing in the database
- Import only TS_METADATA - create an empty dataset
- Import only TS_DATA - fill an empty dataset
- Import REF_METADATA and REF_DATA - create reference metadata and data
- Import TS_METADATA and TS_DATA - create dataset based on existing reference data.

It depends on the destination system which of these operations will be available.

Actually, that's the reason to separate XML into the modules. These modules can be produced separately; and they can be processed separately.

When some of the data is available both on destination system and in FiStatXML files, it is up to the processing system to select whether data being imported has higher priority than imported. Processing instructions may provide hints, but their interpretation is up to processing system.

3.1 How is a dataset stored?

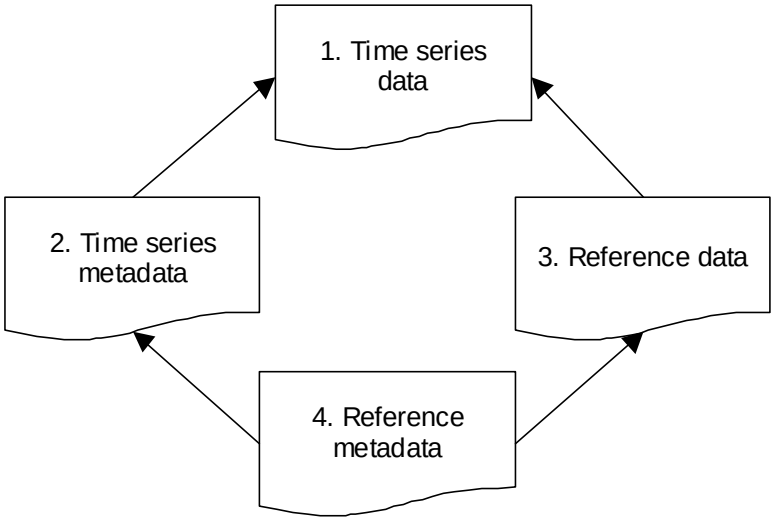
A FiStatXML dataset is stored as an XML file or a set of files. Some of these files can be independent (a country reference table), some may require others. Some files (especially reference metadata and data parts) can be shared between datasets. Usually a complete dataset will be an XML file including a number of other files.

Modules 1-4 may, and often will, be stored in separate files. For example, metadata for Capture dataset changes very rarely, reference data – more often, time series – even more often. So it is sensible to store these components separately and only edit/process the modified ones.

Each dataset must be valid (in XML sense) according to FiStatXML DTD. This DTD must be used by anyone for validating datasets in FiStatXML.

3.2 What’s in each module?

Modules are interrelated. "Higher" modules refer to "lower" ones. The DTD is designed in such a way that these references (both in DTD and in XML!) are uni-directional. In other words, time series data needs reference metadata, but not vice versa. This is how the modules interact (arrow from 2 to 1 means: module 1 needs module 2 and needs data from it)



This allows us to reuse the same reference metadata, reference data and even dataset metadata in several datasets. We will use **bold font in square brackets** to indicate references to objects declared in another module.

XML may refer to the objects existing (or possibly existing) in the destination system through the use of foreign IDs. These are IDs specific for a destination system. They have an ID scheme, e.g. Figis or Fishstat. This is mainly used for reference data and metadata (parts 3 and 4). If processing system determines that an object already exists on destination system, it should not create the second instance. In the absence of foreign ID, processing system may still attempt to map object to existing ones by using names or unique attributes.

3.2.1 Module 1: time series data.

Time series data is an array of records. For each record we need to store

1. Key values. These are links to reference objects. They should point to the keys known either to the processing system [**module 3**] or to the destination system (as pre-existing reference objects)
2. Several Time series (variables). Each is identified by a name or acronym [**module 2**], has a measurement unit [**module 2**] and contains an array of cells. Each cell is for one time unit (usually, a year) [**module 2**] and contains a numeric value, a symbol [**module 2**] and optionally a note

3.2.2 Module 2. Time series metadata

This is needed to interpret module 1. It describes a time series record. The following information must be included:

1. Keys. For each key, system needs to know: object type or types [**module 4**]; name; optional list of acceptable values; list of relevant attributes and relationships [**module 4**]; default display format and other formats, either as references to [**module 4**] or as ad-hoc definitions; Default hierarchy and other relevant hierarchies [**module 4**]
2. Known measurement units.
3. List of allowed symbols.
4. Are notes allowed?
5. Time axis definition.
6. Variables. For each, system needs to know an acronym, a full name, measurement units, time axis, number format, symbols, default value and symbol

3.2.3 Module 3. Reference data

This is data about the actual reference objects. Keys from part 1 refer to these objects. Therefore, what objects need to be described depends on the dataset. If it's a dataset with 10 species, it is enough to describe these 10. If it has 1000 distinct species, all 1000 **may have** to be described. Actually, it is not a requirement to describe ALL reference objects found in the dataset. Some or all objects may be pre-existing in the database.

For each object the following information is stored:

1. ID (or several IDs with schemes)
2. Type [**module 4**]
3. Attribute values for some or all attributes defined in [**module 4**] for this type.
4. References to relates objects (described elsewhere in [**this module**]) for relationships defined in [**module 4**]

In addition, custom groups can be stored, with name, ID and a list of members [**this module**]

3.2.4 Module 4. Reference metadata

This is the description of keys. It describes Object types, Attributes, Relationships and Hierarchies. For examples, objects that may be described here are “Species”, “ISSCAAP group”, “Taxonomic hierarchy”, “Scientific name”, etc.

This description mirrors most of the information found in Figis metadata (see //fiweb01/figis/~~DOCUMENTS/Figis/Ref_Tables_management/~design/data/metadata.doc), and partly in Fishstat metadata. This way it allows mapping to either system. It is expected to be sufficient for most other applications. When a destination system does not have its own structure for metadata, import procedure should map metadata to RDBMS structure, remember this mapping and import REF_DATA.

What actually needs to be described is determined by the time series metadata: e.g. if one of dataset keys is “Species” we need to describe what a species is. This is what may be contained in the file:

1. There is object type called “Species”.
2. This type has ID "SPECIES" (possibly, several IDs with different schemes)
3. Type has attribute "Taxonomic code" with ID "TaxoCode". This attribute is unique and mandatory and contains a string.
4. There is display format "scientific name" which displays scientific name if it is available otherwise common name
5. Species has relationship "Species in ISSCAAP group" with ID "SpeciesInISSCAAP". It is not mandatory, is many to one, and type on the other end of relationship must be "ISSCAAPGroup" [**defined in this section**]

Any types mentioned in (5) must be defined in the same way.

Another type of entity described is hierarchy. It is a chain of pair (ObjectType + Relation) that allows destination system to build a tree. The idea is that for any object system tries to find a Relation matching object type, and through this relation locates the parent. Then repeats the process for the parent.

3.2.5 Part 5: Processing instructions

These are elements understood by some processing systems but not others. DTD does not describe them. Indeed there is no way to describe PI's in the DTD. They are regular XML processing instructions in format `<?system-name instruction?>.`

One example is conditional processing instructions. Key metadata (part 4) for Capture, Aquaculture, Production, regional capture etc is very similar. It would be nice to have them all in the same file and mark dataset-specific parts with some flag (might look something like <?conditional dataset=”capture, production”?>. Another example is some elements that are used in one system but not others. For instance, <?fishstat ignore?>. Processing instruction refers to the element in which it is found.

Appendix A

Time Series Import-Export DTD

Contents

Master module

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ============================================================= -->
<!-- Project: Time Series Universal Import Export -->
<!-- Component:
-->
<!-- Original author: Yves Jaques -->
<!-- Organisation: FAO of the UN, FIDI, FIGIS project -->
<!-- Date: 07/09/2001 -->
<!-- Modifications: -->
<!-- Date          Author              Note -->
<!-- =======    ============     ================================== -->
<!-- dd/mm/yy    who?                why? -->
<!-- dd/10/2001   Yury Shatz            ver. 2-->

<!--                                    -->
<!-- ============================================================= -->
<!--This DTD is a top-level DTD for the Time Series Universal Import Export that calls the other DTD's and contains a root element IMPEXP that contains the other DTD's root level elements. -->
<!-- ============================================================= -->
<!--FigisTimeSeriesTagLib entity calls a group of common elements shared by the various DTDs.
-->
<!ENTITY % FigisTimeSeriesTagLibContent PUBLIC
    "Figis Time Series Import Export Tag Library 1.0"
    "TSTagLibContent1.0.dtd">
%FigisTimeSeriesTagLibContent;
<!ENTITY % TagLibInclude 'IGNORE'>
<!-- ============================================================= -->
<!-- FigisTimeSeriesMetaData calls a DTD that holds elements used to contain Metadata about the keys and structures of a dataset. -->
<!ENTITY % FigisTimeSeriesMetaData PUBLIC
    "Figis Time Series Import Export Metadata 1.0"
    "tsmetadata1.0.dtd">
%FigisTimeSeriesMetaData;
<!-- ============================================================= -->
<!-- FigisTimeSeriesRefMetaData calls a DTD that contains elements used to describe the data held in the elements used to hold the labels and the relationships of the Key ID's - names, codes, attributes and relations.-->
<!ENTITY % FigisTimeSeriesRefMetaData PUBLIC
    "Figis Time Series Reference MetaData 1.0"
    "TSRefMetaData1.0.dtd">
%FigisTimeSeriesRefMetaData;
<!-- ============================================================= -->
<!-- FigisTimeSeriesRefData calls a DTD that contains elements used to hold the labels and the relationships of the Key ID's - names, codes, attributes and how they relate.-->
<!ENTITY % FigisTimeSeriesRefData PUBLIC
    "Figis Time Series Reference Data 1.0"
    "TSRefData1.0.dtd">
%FigisTimeSeriesRefData;
<!-- ============================================================= -->
<!-- FigisTimeSeriesData calls a DTD that contains elements used to hold the actual data values.
-->
<!ENTITY % FigisTimeSeriesData PUBLIC
    "Figis Time Series Import Export Data 1.0"
    "TSData1.0.dtd">
```

```
%FigisTimeSeriesData;
<!-- ============================================================= -->
<!-- TS_IMPORT_EXPORT is the root element for the Times Series Import/Export DTD..
Element  . -->
<!ELEMENT TS_IMPORT_EXPORT (REF_METADATA*, REF_DATA*, TS_METADATA?,
TS_DATA*)>
<!ATTLIST TS_IMPORT_EXPORT
    %REM;
>
<!-- TS_DATA contains base data values: keys and data values for each record-->
<!-- TS_METADATA describes dataset structure, e.g. the names of keys, units, symbols. -->
<!-- REF_DATA contains reference tables, i.e. the elements to which TS_DATA refers. E.g. if a
column is described in TS_DATA as being 'Italy' then in REF_DATA object Italy would be
described, such as its codes, names, relations, etc.-->
<!-- REF_METADATA describes structure of these reference objects. Most importantly,
it defines object types to which objects in REF_DATA belong. For example, if REF_DATA
contains 'Italy' and says that 'Italy' is a 'Country', REF_METADATA describes
what a country is, what attributes it must have, etc-->

<!-- Example: too large to show example. It is the root element.-->

<!-- ============================================================= -->
<!-- This is the end of the Master DTD. -->
```

Time Series Metadata module

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ============================================================= -->
<!-- Project: Time Series Universal Import Export -->
<!-- Component:
-->
<!-- Original author: Yves Jaques -->
<!-- Organisation: FAO of the UN, FIDI, FIGIS project -->
<!-- Date: 07/09/2001 -->
<!-- Modifications: -->
<!-- Date          Author              Note -->
<!-- =======    ============     ================================== -->
<!-- dd/mm/yy    who?                why? -->
<!--                                    -->
<!-- dd/10/2001   Yury Shatz             ver. 2-->
<!-- dd/01/2002   Yves Jaques              ver. 2.5, updates after various inputs from Taconet and
meetings.-->
<!--============================================================= -->
<!--This DTD contains the structures that hold the descriptions of the structures that hold the
actual data-->
<!-- ============================================================= -->
<!--FigisTimeSeriesTagLib entity calls a group of common elements shared by the various DTDs.
-->
<!ENTITY % FigisTimeSeriesTagLib PUBLIC
    "Figis Time Series Import Export Tag Library 1.0"
    "TSTagLib1.0.dtd">
<!ENTITY % TagLibInclude 'INCLUDE'>
%FigisTimeSeriesTagLib;
<!-- ============================================================= -->
<!--TS_METADATA is the root container for time series dataset metadata.
It describes structure of a dataset, its keys and time series.
 -->
<!ELEMENT TS_METADATA (NAME_LIST, KEY_LIST, UNIT_LIST?, SYMBOL_LIST?,
TIME_AXIS, VARIABLE_LIST?)>
<!ATTLIST TS_METADATA
    FAMILY CDATA #IMPLIED
    SHORT_NAME CDATA #IMPLIED
    ACTIVE_DATASET (yes | no) "yes"
    %REM;
>
<!--NAME_LIST contains the NAMEs for the dataset. -->
<!--KEY_LIST contains all the table KEYS in the DATASET -->
<!--UNIT_LIST contains all the measrement UNITs for the DATASET. -->
<!--SYMBOL_LIST contains all the SYMBOLs (flags that qualify data) for the DATASET. -->
<!-- TIME_AXIS describes time axis of the dataset -->
<!--VARIABLE_LIST contains the VARIABLES (aka series, parameters) for the DATASET. -->
<!--FAMILY is the name of the group (if any) to which the DATASET is associated. -->
<!--SHORT_NAME is an abbreviated convenience name for the DATASET. -->
<!--ACTIVE_DATASET indicates whether or not the dataset is active or inactive (discontinued.) --
>
<!-- Example: too large to show physical example. A theoretical example would be  a set of
names for the dataset, followed by a list of keys each containing countries or species together
with their types and unit values for the data. -->
<!-- =============================================================-->
```

```
<!--KEY_LIST contains all the table KEY(s) for a DATASET. It has a numeric attribute
NUM_KEYS that stores the total number of keys found in the dataset.  -->
<!ELEMENT KEY_LIST (KEY)+>
<!ATTLIST KEY_LIST
    NUM_KEYS CDATA #REQUIRED
    %REM;
>
<!--KEY is the root container for metadata about a dataset key. -->
<!-- NUM_KEYS contains the number of keys found in the KEY_LIST container.
Example: -->
<!-- A group of Keys. -->
<!-- ============================================================-->
<!--KEY describes a dataset key.
It does not contain actual key values, but just contains metadata.
-->
<!ELEMENT KEY (NAME_LIST?, OBJ_TYPE_REF+, REL_REF*, FORMAT_REF_LIST?,
VALUE_FILTER?)>
<!ATTLIST KEY
    %REM;
>
<!--NAME_LIST contains the NAME(s)  (if any) for the key. -->
<!--OBJ_TYPE_REF references the object classes that are found within the key. -->
<!--REL_REF lists the relationship(s) (if any) to other keys. -->
<!--optional FORMAT_REF_LIST lists the output format(s) for the key. -->
<!--optional VALUE_FILTER contains a list of values allowed for the key. These are
not values actually encountered but definition of all allowed values.
Import procedure may use this to filter the incoming data.
-->
<!-- Example: <KEY>
            <NAME_LIST>
                <NAME LANG="en">Country</NAME>
                <NAME LANG="fr">Pays</NAME>
            </NAME_LIST>

            <OBJ_TYPE_REF>
                <ID>Country</ID>
                <FOREIGN_ID ID_SCHEME="Fishstat">32</FOREIGN_ID>
                <FOREIGN_ID ID_SCHEME="Figis">13001</FOREIGN_ID>
            </OBJ_TYPE_REF>

            <REL_REF>
                <ID>CountryOnContinent</ID>
            </REL_REF>
        </KEY>
        -->
<!-- ============================================================-->
<!--VALUE_FILTER - list of values allowed for the key -->
<!ELEMENT VALUE_FILTER (CUST_GRP_REF | RELATION_FILTER | OBJ_REF)+>
<!ATTLIST VALUE_FILTER
    %REM;
>
<!-- Allowed values are defined in one of three ways: -->
<!--CUST_GRP_REF: references custom groups. Keys in these
groups are allowed -->
<!--RELATION_FILTER contains references to related objects. Any objects
related to those listed (e.g. their children) are allowed-->
```

```
<!--OBJ_REF: as references to individual objects  -->
<!-- Example: -->
<!-- List of allowed values for species, including a custom group,
two ISSCAAP groups and two individual species -->
<!--
<VALUE_FILTER>
    <CUST_GRP_REF><ID>MarineFishes</ID></CUST_GRP_REF>
    <RELATION_FILTER>
        <OBJ_TYPE_REF><ID>Species</ID></OBJ_TYPE_REF>
        <REL_REF><ID>SpeciesInIsscaapGroup</ID></REL_REF>
        <RELATED_OBJ_LIST>
            <OBJ_REF><ID>IsscaapGroup11</ID></OBJ_REF>
            <OBJ_REF><ID>IsscaapGroup12</ID></OBJ_REF>
        </RELATED_OBJ_LIST>
    </RELATION_FILTER>
    <OBJ_REF><ID>SpeciesABC</ID></OBJ_REF>
    <OBJ_REF><FOREIGN_ID ID_SCHEME="Figis">2124</FOREIGN_ID></OBJ_REF>
</VALUE_FILTER>
-->
<!-- ============================================================-->
<!--CUSTGRP_REF contains the NAME and ID(s) of a custom group in the dataset. -->
<!ELEMENT CUST_GRP_REF (%REF;)+>
<!ATTLIST CUST_GRP_REF
    %REM;
>
<!-- Example: -->
<!-- see above -->
<!-- ============================================================-->
<!--RELATION_FILTER defines a list of objects through their related objects
(usually groups)-->
<!ELEMENT RELATION_FILTER (OBJ_TYPE_REF, REL_REF, RELATED_OBJ_LIST)>
<!ATTLIST RELATION_FILTER
    %REM;
>
<!-- It consists of
- OBJ_TYPE_REF: object type we are selecting;
- REL_REF: relation we are using for selection;
- RELATED_OBJECT_LIST: related objects.
-->
<!-- Example: -->
<!-- see above -->
<!-- ============================================================-->
<!--RELATION_OBJ_LIST is a list of references to objects
(usually groups)-->
<!ELEMENT RELATED_OBJ_LIST (OBJ_REF+)>
<!ATTLIST RELATED_OBJ_LIST
    %REM;
>
<!-- Example: see above in example for VALUE_FILTER -->
<!-- ============================================================-->
<!--FORMAT_REF_LIST contains a set of FORMAT_REFs that hold the ID's for the format
strings for a key. -->
<!ELEMENT FORMAT_REF_LIST (FORMAT_REF)+>
<!ATTLIST FORMAT_REF_LIST
    %REM;
>
```

```
<!-- FORMAT_REF hold the ID(s) for a format string for output display. -->
<!-- Example: -->
<!-- none -->
<!-- ============================================================-->
<!--FORMAT_REF contains a formatting reference for a KEY (including its NAME and any IDs)
for a format that is defined elsewhere. -->
<!ELEMENT FORMAT_REF (%REF;)+>
<!ATTLIST FORMAT_REF
    %REM;
>
<!-- Example: -->
<!-- none -->
<!-- ============================================================-->
<!--UNIT_LIST contains a list of UNITs for a DATASET or a VARIABLE. -->
<!ELEMENT UNIT_LIST (UNIT)+>
<!ATTLIST UNIT_LIST
    %REM;
>
<!--UNIT contains the ID(s) and names for a unit of measurement. -->
<!-- Example:
    <UNIT_LIST>
        <UNIT>
            <ID>MT</ID>
            <NAME_LIST>
                <NAME LANG="en">Metric ton</NAME>
            </NAME_LIST>
        </UNIT>
        <UNIT>
            <ID>$1000</ID>
            <NAME_LIST>
                <NAME LANG="en">Thousand US dollars</NAME>
            </NAME_LIST>
        </UNIT>
    </UNIT_LIST>
-->
<!-- ============================================================-->
<!--UNIT contains an internal ID, zero or more FOREIGN_IDs and a NAME_LIST containing one
or more unit NAMEs. -->
<!ELEMENT UNIT (ID, FOREIGN_ID*, NAME_LIST?)>
<!ATTLIST UNIT
    ABBREV CDATA #IMPLIED
    DEFAULT (yes | no) "no"
    %REM;
>
<!--ID contains an ID value for a measurement unit. -->
<!--FOREIGN_ID contains an ID value for a measurement unit external to the IMP/EXP program.
-->
<!--NAME_LIST contains one or more unit NAMEs. -->
<!-- Example:
            <UNIT>
            <ID>MT</ID>
            <NAME_LIST>
                <NAME LANG="en">Metric ton</NAME>
            </NAME_LIST>
        </UNIT>
    -->
```

```
<!-- ============================================================-->
<!--SYMBOL_LIST contains a set of SYMBOLs for a DATASET or a VARIABLE. These symbols
are flags that qualify the data. -->
<!ELEMENT SYMBOL_LIST (SYMBOL)+>
<!ATTLIST SYMBOL_LIST
    %REM;
>
<!--SYMBOL (data qualifying flag) contains a NAME_LIST of symbol names for a DATASET or a
VARIABLE. -->
<!-- Example:
    <SYMBOL_LIST>
        <SYMBOL CHAR=".">
            <NAME_LIST>
                <NAME LANG="en">Unknown</NAME>
            </NAME_LIST>
        </SYMBOL>
        <SYMBOL CHAR="0">
            <NAME_LIST>
                <NAME LANG="en">Negligible</NAME>
            </NAME_LIST>
        </SYMBOL>
        <SYMBOL CHAR="-">
            <NAME_LIST>
                <NAME LANG="en">Zero</NAME>
            </NAME_LIST>
        </SYMBOL>
        <SYMBOL CHAR="F">
            <NAME_LIST>
                <NAME LANG="en">FAO Estimate</NAME>
            </NAME_LIST>
        </SYMBOL>
        <SYMBOL CHAR="R">
            <NAME_LIST>
                <NAME LANG="en">Repetition</NAME>
            </NAME_LIST>
        </SYMBOL>
    </SYMBOL_LIST>
-->
<!-- ============================================================-->
<!--SYMBOL (a flag that qualifies data) contains a NAME_LIST of symbol names for a DATASET
or a VARIABLE. -->
<!ELEMENT SYMBOL (NAME_LIST)>
<!ATTLIST SYMBOL
    CHAR CDATA #REQUIRED
    DEFAULT (yes | no) "no"
    ZERO (yes | no | both) "no"
    %REM;
>
<!--NAME_LIST contains the NAME(s) for the SYMBOL. -->
<!--CHAR holds the actual string value of the SYMBOL (usually 1 character).-->
<!--DEFAULT indicates whether or not it is the default flag for the value.-->
<!--ZERO indicates whether this symbol is used with zeroes, with non-zeroes
or both -->
<!-- Example:
        <SYMBOL CHAR="F">
            <NAME_LIST>
```

```
            <NAME LANG="en">FAO Estimate</NAME>
        </NAME_LIST>
    </SYMBOL>
    -->
<!-- ================================================================-->
<!--
TIME_AXIS describes dataset's time axis.
Time axis can be regular or irregular. Regular axis is a list of time periods
at regular intervals, e.g. years, quarters or months. It can be described using attributes only - you
do not have to list the intervals.
Irregular axis is a list of time periods not necessarily at regular intervals.
Such intervals must be explicitly listed.
-->
<!ELEMENT TIME_AXIS ((REGULAR, TIME_PERIOD_LIST?) | TIME_PERIOD_LIST)>
<!ATTLIST TIME_AXIS
    TIME_UNIT (year | quarter | month | other) "year"
    COLUMNS_PER_TIME_UNIT CDATA "1"
    START_DATE CDATA #REQUIRED
    NUM_COLUMNS CDATA #REQUIRED
    %REM;
>
<!-- Regular time axis is identified by empty element REGULAR. If it is present,
TIME_PERIOD_LIST is optional, otherwise it is mandatory.

A regular TIME_AXIS has exactly COLUMNS_PER_TIME_UNIT columns for each
TIME_UNIT starting at START_DATE, for total of NUM_COLUMNS. -->
<!-- TIME_PERIOD_LIST holds the TIME_PERIODs for irregular axes.
It may be used for regular axes as well, if you need to change
names or IDs for time periods
-->
<!--TIME_UNIT and COLUMNS_PER_TIME_UNIT together define dataset resolution.
For example,
TIME_UNIT="year" COLUMNS_PER_TIME_UNIT="1" is an annual dataset.
TIME_UNIT="year" COLUMNS_PER_TIME_UNIT="5" is a dataset with 5 columns per year,
e.g. 4 quarters and 'unknown quarter'. Etc.

Each TIME_PERIOD has a sequence number (1...), an ID and a screen name (see
TIME_PERIOD_LIST definition).
These ID, name or sequence number can be used to refer to the columns from TS_DATA
(although it may be more convenient to use column numbers instead)

If TIME_PERIOD_LIST is missing for a regular axis, these IDs and names are generated
automatically.
Format is
year[.time_unit][.column_within_unit]

year is a 4-digit year number
time_unit is a number of TIME_UNIT within the year (e.g. 1..4 for quarters etc)
column_within_unit is a number of column within time unit.

So, if we define our dataset as
TIME_UNIT="year" COLUMNS_PER_TIME_UNIT="1" START_DATE="1990",
columns are 1990,1991,1992...
For
TIME_UNIT="year" COLUMNS_PER_TIME_UNIT="5" START_DATE="1990",
they are 1990.1, 1990.2,...1990.5, 1991.1 etc.
```

```
and for TIME_UNIT="month" COLUMNS_PER_TIME_UNIT="2" START_DATE="1995.1"
(twice a month data)
they are 1995.1.1, 1995.1.2,1995.2.1, 1995.2.2...

User can ovveride column IDs and/or names for regular time axis, and MUST do so
for irregular-->
<!-- ============================================================-->
<!-- REGULAR, Empty 'marker' element -->
<!ELEMENT REGULAR EMPTY>
<!ATTLIST REGULAR
    %REM;
>
<!-- Its presence within TIME_AXIS indicates that it is a regular time axis.-->
<!-- ============================================================-->
<!--TIME_PERIOD_LIST contains a set of TIME_PERIODs (slices of time) for which a VARIABLE
contains data. -->
<!ELEMENT TIME_PERIOD_LIST (TIME_PERIOD)+>
<!ATTLIST TIME_PERIOD_LIST
    %REM;
>
<!-- ============================================================-->
<!--TIME_PERIOD contains a numeric attribute VALUE containing the name or number of a time
period (slice) for which a VARIABLE contains data. -->
<!ELEMENT TIME_PERIOD (NAME_LIST?)>
<!ATTLIST TIME_PERIOD
    SEQ_NUMBER CDATA #IMPLIED
    ID CDATA #IMPLIED
    NUMERIC_VALUE CDATA #IMPLIED
    %REM;
>
<!--SEQ_NUMBER is sequence number of TIME_PERIOD (starting from 1)
if omitted its position within TIME_PERIOD_LIST is implied
ID is unique identifier. TS_DATA may refer to ID or to sequence number.
If omitted, it is automatically assigned.
NAME_LIST are screen names in various languages. If omitted, same as ID
NUMERIC_VALUE is a number (not necessarily integer!)
showing place on X axis. This may be needed for highly irregular datasets
(e.g. to display graph). If omitted it is
automatically generated from START_DATE and SEQ_NUMBER (or ignored) -->
<!-- Example:
1) annual dataset from 1970 to 1999 (Capture)
<TIME_AXIS START_DATE="1970" NUM_COLUMNS="30">
    <REGULAR/>
</TIME_AXIS>

2) quarterly dataset (with fake 5th quarter) from 1950 to 2000 (Tuna Atlas)
<TIME_AXIS START_DATE="1950" NUM_COLUMNS="255"
COLUMNS_PER_TIME_UNIT="5">
    <REGULAR/>
</TIME_AXIS>

3) quarterly dataset (with normal 4 quarters) from 1990 to 2000
<TIME_AXIS START_DATE="1990" NUM_COLUMNS="44" TIME_UNIT="quarter">
    <REGULAR/>
</TIME_AXIS>
```

```
4) Irregular dataset: Fleets from 1970..1995 with some years missing:
<TIME_AXIS START_DATE="1970" NUM_COLUMNS="21" TIME_UNIT="year">
    <TIME_PERIOD_LIST>
        <TIME_PERIOD ID="1970"/>
        <TIME_PERIOD ID="1975"/>
        <TIME_PERIOD ID="1977"/>
        <TIME_PERIOD ID="1978"/>
        <TIME_PERIOD ID="1979"/>
        <TIME_PERIOD ID="1980"/>
        <TIME_PERIOD ID="1981"/>
        <TIME_PERIOD ID="1982"/>
        <TIME_PERIOD ID="1983"/>
        <TIME_PERIOD ID="1984"/>
        etc
    </TIME_PERIOD_LIST>
</TIME_AXIS>

-->
<!-- ============================================================-->
<!--VARIABLE_LIST contains one or more VARIABLES found in a DATASET. Attribute
NUM_VARIABLES holds the number of instances of VARIABLE found in the VARIABLE_LIST. --
>
<!ELEMENT VARIABLE_LIST (VARIABLE+)>
<!ATTLIST VARIABLE_LIST
    NUM_VARIABLES CDATA #REQUIRED
    %REM;
>
<!--TIME_AXIS holds all the temporal data for the VARIABLE(s). -->
<!--VARIABLE contains timeseries metadata-->
<!--NUM_VARIABLES holds the number of instances of VARIABLE contained in the
VARIABLE_LIST. -->
<!-- Example: -->
<!-- none -->
<!-- ============================================================-->
<!--VARIABLE is a single variable within a dataset. For example,
"Quantity", "Value", "Tonnage" etc.
Synonyms for VARIABLE could be 'parameter', 'dimension', or even 'data concept'.
In addition to identifying information,
(ID, NAME_LIST, ABBREV), it may override dataset 'global' metadata
such as measurement UNITs, SYMBOLs and even TIME_AXIS
if any of those differs between variables.
 -->
<!ELEMENT VARIABLE (ID, NAME_LIST?, UNIT_LIST?, SYMBOL_LIST?, TIME_AXIS?)>
<!ATTLIST VARIABLE
    ABBREV CDATA #REQUIRED
    PRECISION (0 | 1 | 2 | 3 | 4 | 5) "0"
    MULTIPLIER CDATA "1"
    %REM;
>
<!--ID is the VARIABLE id.-->
<!--NAME_LIST contains one or more variable NAMEs. -->
<!--UNIT_LIST contains a list of UNITs for the VARIABLE. -->
<!--SYMBOL_LIST contains a set of SYMBOLs for the VARIABLE. These symbols are flags that
qualify the data. -->
<!--TIME_AXIS holds the TIME_PERIODs covered by a VARIABLE. -->
<!--ABBREV is the variable short name. -->
```

```
<!--PRECISION is '1' for highest, '5' for lowest and '0' for unspecified.-->
<!--MULTPLIER is the multiplier if any applied to the timeseries data.-->
<!-- Example:
<VARIABLE ABBREV="V">
    <ID>VALUE</ID>
    <NAME_LIST>
        <NAME xml:lang="en">Value</NAME>
        <NAME xml:lang="fr">Valeur</NAME>
    </NAME_LIST>
    <UNIT_LIST>
        <UNIT ABBREV="$1000">
            <ID>$1000</ID>
            <NAME_LIST>
                <NAME xml:lang="en">Thousand US Dollars</NAME>
            </NAME_LIST>
        </UNIT>
    </UNIT_LIST>
    symbol list and time axis are taken from TS_DATASET
</VARIABLE>
-->
<!-- ============================================================-->
<!--This is the end of the TimeSeriesMetadata DTD. -->
```

Time Series Reference Data module

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ============================================================= -->
<!-- Project: Time Series Universal Import Export -->
<!-- Component:
-->
<!-- Original author: Yves Jaques -->
<!-- Organisation: FAO of the UN, FIDI, FIGIS project -->
<!-- Date:12/09/2001 -->
<!-- Modifications: -->
<!-- Date          Author              Note -->
<!-- =======    ============     ================================== -->
<!-- dd/mm/yy    who?                why? -->
<!--                                    -->
<!-- dd/10/2001   Yury Shatz             ver. 2-->
<!-- ============================================================= -->
<!--This DTD contains the structures that hold the labels and the relationships of the Key ID's - names, codes, attributes and how they relate.-->
<!-- ============================================================= -->
<!--FigisTimeSeriesTagLib entity calls a group of common elements shared by the various DTDs.
-->
<!ENTITY % FigisTimeSeriesTagLib PUBLIC
    "Figis Time Series Import Export Tag Library 1.0"
    "TSTagLib1.0.dtd">
<!ENTITY % TagLibInclude 'INCLUDE'>
%FigisTimeSeriesTagLib;
<!-- ============================================================= -->
<!--REF_DATA is the root element for all reference data and holds two main containers, REF_OBJ for key labels and relationships, and CUST_GRP for custom group lists of reference objects. -->
<!ELEMENT REF_DATA (REF_OBJ | CUST_GRP)+>
<!ATTLIST REF_DATA
  %REM;
>
<!--REF_OBJ stores information about the labels and relationships of the keys -->
<!--CUST_GRP stores custom group lists of reference objects. -->
<!-- Example: too large to show physical example. A theoretical example would be a list of countries as REF_OBJ instances each containing names and associated ID's -->
<!-- ============================================================= -->
<!--REF_OBJ is the root element for reference information on objects. It allows for labels to be assigned to objects and for those labels to also be associated to various foreign ID scheme types (classes) using OBJ_TYPE_REF. The ATT_VAL field contains foreign ID instances while NAME_LIST contains a set of label name instances, often for various languages. ID_LIST contains the actual ID's used in the import/export of the data. REL_VAL contains data on group class and instances. -->
<!ELEMENT REF_OBJ (ID_LIST, OBJ_TYPE_REF, NAME_LIST, ATT_VAL*, REL_VAL*)>
<!ATTLIST REF_OBJ
  %REM;
>
<!--ID_LIST contains the various ID's and FOREIGN_IDs for the REF_OBJ instance.-->
<!--OBJ_TYPE_REF references the REF_OBJ class ID(s), internal and/or foreign. -->
<!--NAME_LIST contains a set of label name instances, often for various languages. -->
<!--The ATT_VAL field contains foreign ID instances.-->
<!--REL_VAL contains data on group class and instances. -->
<!-- Example:
```

```
<REF_OBJ>
        <ID_LIST>
            <ID>Continent1</ID>
            <FOREIGN_ID ID_SCHEME="Fishstat">1</FOREIGN_ID>
        </ID_LIST>
        <OBJ_TYPE_REF>
            <ID>Continent</ID>
            <FOREIGN_ID ID_SCHEME="Fishstat">36</FOREIGN_ID>
            <FOREIGN_ID ID_SCHEME="Figis">11002</FOREIGN_ID>
        </OBJ_TYPE_REF>
        <NAME_LIST>
            <NAME xml:lang="en">Africa</NAME>
            <NAME xml:lang="fr">Afrique</NAME>
            <NAME xml:lang="es">Africa</NAME>
        </NAME_LIST>
        <ATT_VAL>
            <ID>ContinentCode</ID>
            <VALUE>1</VALUE>
        </ATTRIB>
</REF_OBJ>
-->
<!-- ============================================================= -->
<!--REL_VAL stores reference data for group classes and instances.-->
<!ELEMENT REL_VAL ((ID | FOREIGN_ID), OBJ_REF+)>
<!ATTLIST REL_VAL
  %REM;
>
<!-- ID and FOREIGN_ID identify REL element defined elsewhere -->
<!--Each instance of ID holds one ID value internal to the import tool.-->
<!--Each instance of FOREIGN_ID holds one foreign ID value together with attribute
ID_SCHEME that identifies the foreign ID system.-->
<!-- OBJ_REF references the custom group members.-->
<!-- Example:
  <RELATION>
     <ID>CountryOnContinent</ID>
    <OBJ_REF>
      <ID>Continent4</ID>
    </OBJ_REF>
  </RELATION>
-->
<!-- ============================================================= -->
<!--CUST_GRP stores lists of custom groups. -->
<!ELEMENT CUST_GRP (ID_LIST, NAME_LIST, (OBJ_REF)+)>
<!ATTLIST CUST_GRP
  %REM;
>
<!--ID_LIST contains ID's and FOREIGN_IDs for the CUST_GRP.-->
<!--NAME_LIST contains a set of names for CUST_GRP, often for various languages. -->
<!-- OBJ_REF references the custom group members.-->
<!-- Example:
  none
-->
<!-- ============================================================= -->
<!--This is the end of the TimeSeriesReferencedata DTD. -->
```

Time Series Reference Metadata module

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Maria Barbuzzi (FAO
Headquarters) -->
<!-- ============================================================= -->
<!-- Project: Time Series Universal Import Export -->
<!-- Component:
-->
<!-- Original author: Yves Jaques -->
<!-- Organisation: FAO of the UN, FIDI, FIGIS project -->
<!-- Date: 07/09/2001 -->
<!-- Modifications: -->
<!-- Date          Author              Note -->
<!-- =======    ============    ================================== -->
<!-- dd/mm/yy    who?                why? -->
<!--                                    -->
<!-- dd/10/2001   Yury Shatz            ver. 2-->
<!-- ============================================================= -->
<!--This DTD contains the structures that describe the labels, names, codes, attributes and
relations of the KEY ID values described in the Time Series Reference Data.-->
<!-- ============================================================= -->
<!--FigisTimeSeriesTagLib entity calls a group of common elements shared by the various DTDs.
-->
<!ENTITY % FigisTimeSeriesTagLib PUBLIC
    "Figis Time Series Import Export Tag Library 1.0"
    "TSTagLib1.0.dtd">
<!ENTITY % TagLibInclude 'INCLUDE'>
%FigisTimeSeriesTagLib;
<!-- ============================================================= -->
<!-- REF_METADATA is a container for object types (metaobjects)
hierarchies for an import data file.-->
<!ELEMENT REF_METADATA (OBJ_TYPE | HIERARCHY)*>
<!ATTLIST REF_METADATA
  %REM;
>
<!-- OBJ_TYPE is an object type or class.-->
<!-- HIERARCHY is a top-level object (independent of OBJ_TYPE) that is a set of one or more
pairs of type and relationship. -->
<!-- Example: too large to show physical example. A theoretical example would be two
metaobjects, COUNTRY and CONTINENT described using two instances of OBJ_TYPE and
then their hierarchy described in HIERARCHY. -->
<!-- ============================================================= -->
<!-- OBJ_TYPE is an object type or class.-->
<!ELEMENT OBJ_TYPE (ID_LIST, NAME_LIST, ATT*, REL*, HIER_REF*)>
<!ATTLIST OBJ_TYPE
  %REM;
>
<!-- ID_LIST contains the IDs that identify OBJ_TYPE.-->
<!-- NAME_LIST contains the NAME(s) for OBJ_TYPE.-->
<!-- ATT contains the attribute class if any.-->
<!-- REL contains relationships to other OBJ_TYPEs if any.-->
<!-- HIER_REF contains references to the HIERARCHY if any.-->
<!-- Example: too large to show physical example. Theoretical example would be a 'continent'
metaobject containing it's various ID's (internal and foreign), names and attributes followed by its
```

```
related types such as a list of countries, followed at last by references to hierarchies of which the object is a part. -->
<!-- ========================================================== -->
<!-- REL is information about relationship class. It is contained in OBJ_TYPE to which relationship applies. The relationship contains attributes TYPE and MULTiplicity as well as whether the relationship is MANDATORY and/or REVERSible -->
<!ELEMENT REL (ID_LIST, NAME_LIST, RELATED_TYPES, REVERSE_REL?)>
<!ATTLIST REL
  TYPE (child | parent | exact_equivalent | partial_equivalent | inexact_equivalent | broader | narrower | associative | unknown) "child"
  MULT (one-to-one | one-to-many | many-to-one | many-to-many) #REQUIRED
  MANDATORY (yes | no) "no"
  REVERSE (yes | no) "no"
  %REM;
>
<!-- ID_LIST contains the IDs that identify REL. -->
<!-- NAME_LIST contains the NAME(s) for REL.-->
<!-- RELATED_TYPES contains references to the objects on the other end of the relationship-->
<!-- REVERSE_REL references the reverse of the relationship if it exists.-->
<!-- TYPE describes the relationship type. Default is CHILD.-->
<!-- MULT describes whether the relationship has multiplicity.-->
<!--MANDATORY toggles whether the relationship is mandatory or not. -->
<!--REVERSE toggles whether the relationship is reversible or not-->
<!-- Example:
<REL MULT="many-to-one">
     <ID_LIST>
       <ID>SpeciesInIsscaapGroup</ID>
       <FOREIGN_ID ID_SCHEME="Figis">32001</FOREIGN_ID>
       <FOREIGN_ID ID_SCHEME="Fishstat">SpeciesInISSCAAPGroup</FOREIGN_ID>
     </ID_LIST>
     <NAME_LIST>
       <NAME xml:lang="en">ISSCAAP Group</NAME>
     </NAME_LIST>
     <RELATED_TYPES>
       <OBJ_TYPE_REF><ID>ISSCAAPGroup</ID></OBJ_TYPE_REF>
     </RELATED_TYPES>
   </REL>
   -->
<!-- ========================================================== -->
<!-- REVERSE_REL is a reference to the reverse of a relationship. It contains the NAME(s), internal ID(s) and FOREIGN_ID(s) of the reverse relation type.-->
<!ELEMENT REVERSE_REL (REL_REF)+>
<!ATTLIST REVERSE_REL
  %REM;
>
<!-- REL_REF contains the NAME(s), internal ID(s) and FOREIGN_ID(s) of the reverse relationship.-->
<!-- Example: (where the metaobject is IsscaapGroup and the related types are single species)
            <REVERSE_REL>
            <REL_REF>
                <ID>SpeciesInIsscaapGroup</ID>
            </REL_REF>
            </REVERSE_REL>
  -->
<!-- ========================================================== -->
<!-- ATT contains metadata information about an attribute.  -->
```

```
<!ELEMENT ATT (ID_LIST, NAME_LIST, FORMAT*)>
<!ATTLIST ATT
  UNIQUE (no | yes) "no"
  PRIORITY (0 | 1 | 2 | 3 | 4 | 5) "0"
  MANDATORY (yes | no) "no"
  MULTILINGUAL (yes | no) "no"
  TYPE (string | number | date) "string"
  %REM;
>
<!-- ID_LIST contains the IDs that identify ATT. -->
<!-- NAME_LIST contains the NAME(s) for ATT.-->
<!-- PRIORITY is '1' for highest, '5' for lowest and '0' for unspecified.-->
<!-- MANDATORY toggles whether the relationship is mandatory or not. -->
<!-- MULTILINGUAL toggles whether the attribute value is multilingual
(e.g. short name). Such attribute will have multiple ATT_VALs referring to
it -->
<!-- TYPE defines attribute value type. -->
<!-- Example:
      Country has a 3-letter ISO code - its unique attribute. It has the highest priority when
      identifying a country in the absence of ID.
    <ATT UNIQUE="yes" PRIORITY="1">
      <ID_LIST>
        <ID>ISO3Code</ID>
        <FOREIGN_ID ID_SCHEME="Figis">21</FOREIGN_ID>
        <FOREIGN_ID ID_SCHEME="Fishstat">Country3Alpha</FOREIGN_ID>
      </ID_LIST>
      <NAME_LIST>
        <NAME xml:lang="en">ISO 3-letter Code</NAME>
      </NAME_LIST>
    </ATT>
  -->
<!-- ============================================================= -->
<!-- HIERARCHY is a top-level object (independent of OBJ_TYPE) that is a set of one or more
pairs of type and relationship. -->
<!-- It determines how trees are built. For each type there are 1 or more relationships within
HIERARCHY with different priorities.
For each object a parent within a hierarchy is found in the following way:
1) object type is taken
2) all HIER_RELs for this object type are found
3) a relation with priority 1 is taken
4) if object has related objects, they are its parents. Otherwise,
5) next-priority relation is taken, return to step 4
-->
<!ELEMENT HIERARCHY (ID_LIST, NAME_LIST, HIER_REL+)>
<!ATTLIST HIERARCHY
  %REM;
>
<!-- ID_LIST contains the IDs that identify the HIERARCHY. -->
<!-- NAME_LIST contains the NAME(s) for the HIERARCHY.-->
<!-- HIER_REL defines relationships within a hierarchy. -->
<!-- Example:
  <HIERARCHY>
    <ID_LIST>
        <ID>CountriesByContinent</ID>
        <FOREIGN_ID ID_SCHEME="Figis">10002</FOREIGN_ID>
        <FOREIGN_ID ID_SCHEME="Fishstat">CountriesByContinent</FOREIGN_ID>
```

```
    </ID_LIST>
    <NAME_LIST>
        <NAME LANG="en">Countries By Continent</NAME>
    </NAME_LIST>
    <HIER_REL>
        <OBJ_TYPE_REF>
            <ID>Country</ID>
        </OBJ_TYPE_REF>
        <REL_REF>
            <ID>CountryOnContinent</ID>
        </REL_REF>
    </HIER_REL>
  </HIERARCHY>
   -->
<!-- ================================================================= -->
<!-- HIER_REL is a relationship within a hierarchy. It says, "use this relation for this type to build
the tree."-->
<!-- If hierarchy contains several relations for the same type, they are used in order of priority.-->
<!-- Relationships with lower priority are used only in the case that an object does not have
relations of higher priority. -->
<!-- NOTE: Everywhere 1 is highest ('first') priority and 0 is unspecified. -->
<!ELEMENT HIER_REL (OBJ_TYPE_REF, REL_REF)>
<!ATTLIST HIER_REL
  PRIORITY (0 | 1 | 2 | 3 | 4 | 5) "0"
  DIRECTION (up | down) "up"
  %REM;
>
<!--OBJ_TYPE_REF contains the reference to the object type on the close end of the hierarchy.--
>
<!--REL_REF contains the reference to the object type on the far end of the hierarchy.-->
<!--PRIORITY is '1' for highest, '5' for lowest and '0' for unspecified.-->
<!--DIRECTION defines in which direction the related object lies in the hierarchy. -->
<!-- Example:
    <HIER_REL>
        <OBJ_TYPE_REF>
            <ID>Country</ID>
        </OBJ_TYPE_REF>
        <REL_REF>
            <ID>CountryOnContinent</ID>
        </REL_REF>
    </HIER_REL>
-->
<!-- ================================================================= -->
<!--This is the end of the TimeSeriesRefMetadata DTD. -->
```

Time Series Data module

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ============================================================= -->
<!-- Project: Time Series Universal Import Export -->
<!-- Component:
-->
<!-- Original author: Yves Jaques -->
<!-- Organisation: FAO of the UN, FIDI, FIGIS project -->
<!-- Date: 07/09/2001 -->
<!-- Modifications: -->
<!-- Date          Author                Note -->
<!-- =======    ============     ================================== -->
<!-- dd/mm/yy        who?                  why? -->
<!-- dd/10/2001   Yury Shatz              ver. 2-->
<!-- ============================================================= -->
<!--Keys, Values, Units. This DTD contains the structures that hold the actual data-->
<!-- ============================================================= -->
<!--FigisTimeSeriesTagLib entity calls a group of common elements shared by the various DTDs.
-->
<!ENTITY % FigisTimeSeriesTagLib PUBLIC
    "Figis Time Series Import Export Tag Library 1.0"
    "TSTagLib1.0.dtd">
<!ENTITY % TagLibInclude 'INCLUDE'>
%FigisTimeSeriesTagLib;
<!-- ============================================================= -->
<!--TS_DATA is the root data-holding element and consists of a group of one or more
RECORDs.-->
<!ELEMENT TS_DATA (TS_RECORD+)>
<!ATTLIST TS_DATA
  %REM;
>
<!-- ============================================================= -->
<!--TS_RECORD is the container for a single data record. It contains a KEYDATA_LIST
(consisting of OBJ_REFerences) that hold the IDs of the keys for a record. TS_RECORD also
contains either one or more CELL_SERIES or one or more CELLs holding data values.-->
<!-- Creator of XML can choose what is more convenient -
perhaps generate CELL_SERIES if all cells of the same series are grouped
in source file, or generate individual CELLs if they are mixed..
-->

<!ELEMENT TS_RECORD (KEYDATA_LIST, (CELL_SERIES+ | CELL+), NOTE?)>
<!ATTLIST TS_RECORD
  %REM;
>
<!-- KEYDATA_LIST holds references to the keys for a record,
it must contain as many OBJ_REFs as many keys there are in the dataset
(see NUM_KEYS in KEY_LIST in TS_METADATA), in the same order
NOTE can be attached to the whole record, to a series or to an
individual cell
-->
<!-- CELL holds a data value.-->
<!-- Example:
<TS_RECORD>
 <KEYDATA_LIST>
  <OBJ_REF><ID>Country008</ID></OBJ_REF>
```

```
  <OBJ_REF><ID>Species1230100909</ID></OBJ_REF>
 </KEYDATA_LIST>
  <CELL VARIABLE="Q" TIME_PERIOD="1979" SYMBOL="F" VALUE="200"/>
  <CELL VARIABLE="V" TIME_PERIOD="1979" SYMBOL="F" VALUE="0.0"/>
  <CELL VARIABLE="Q" TIME_PERIOD="1980" SYMBOL="F" VALUE="800"/>
  <CELL VARIABLE="V" TIME_PERIOD="1980" SYMBOL="F" VALUE="0.0"/>
  <CELL VARIABLE="Q" TIME_PERIOD="1981" SYMBOL="F" VALUE="500"/>
</TS_RECORD>
-->
<!-- ============================================================= -->
<!-- KEYDATA_LIST holds references to the keys for a record,
it must contain as many OBJ_REFs as many keys there are in the dataset
(see NUM_KEYS in KEY_LIST in TS_METADATA), in the same order
-->
<!ELEMENT KEYDATA_LIST (OBJ_REF)+>
<!ATTLIST KEYDATA_LIST
  %REM;
>
<!--OBJ_REF identifies a key object (instance of OBJ defined in REF_DATA),
using ID, ATT_VAL, or NAME
-->
<!-- Example:
 <KEYDATA_LIST>
  <OBJ_REF><ID>Country008</ID></OBJ_REF>
  <OBJ_REF><ID>Species1230100909</ID></OBJ_REF>
  <OBJ_REF><ID>OceanStatArea05</ID></OBJ_REF>
  <OBJ_REF><ID>EnvironmentIN</ID></OBJ_REF>
 </KEYDATA_LIST>
-->

<!-- ============================================================= -->
<!--CELL_SERIES is a container for a single series (cells for a single variable),
and specifies VARIABLE and UNIT for all of them. This way individual CELL
elements don't have to have VARIABLE and UNIT attributes
 -->
<!ELEMENT CELL_SERIES (CELL+, NOTE?)>
<!ATTLIST CELL_SERIES
    ORDERED (yes | no ) "yes"
    VARIABLE CDATA #IMPLIED
    UNIT CDATA #IMPLIED
    %REM;
>
<!-- ORDERED="yes" means that there is a CELL per each column inside the CELL_SERIES,
i.e. first CELL is for first column etc.
This way individual CELLs do not need COL and TIME_PERIOD attribute
-->
<!--VARIABLE is the ID or ABBREV for VARIABLE element as in 'Q' for quantity of 'V' for value.
See VARIABLE element in TS_METADATA. You must specify VARIABLE if dataset contains
more than one. -->
<!--UNIT is the ID or ABBREV for measurement unit for the value.
See UNIT element in TS_METADATA. If omitted, default unit for this variable is taken
-->


<!-- ============================================================= -->
<!--CELL is the core data-holding element. It holds one numeric value for
```

```
a cell along with attributes that identify the CELL -->
<!ELEMENT CELL (NOTE?)>
<!ATTLIST CELL
    COL CDATA #IMPLIED
    TIME_PERIOD CDATA #IMPLIED
    VARIABLE CDATA #IMPLIED
    UNIT CDATA #IMPLIED
    SYMBOL CDATA #IMPLIED
    VALUE CDATA #IMPLIED
    %REM;
>
<!-- NOTE contains optional remarks about the cell. -->
<!-- COL is sequence number for a column -->
<!--TIME_PERIOD (a time slice) is an ID for a TIME_PERIOD (column). You can specify either
COL or TIME_PERIOD  -->
<!--VARIABLE is the ID or ABBREV for VARIABLE element as in 'Q' for quantity of 'V' for value.
See VARIABLE element in TS_METADATA. You must specify VARIABLE if dataset contains
more than one. -->
<!--UNIT is the ID or ABBREV for measurement unit for the value.
See UNIT element in TS_METADATA. If omitted, default unit for this variable is taken
-->
<!--SYMBOL is a data symbol (flag) or a string consisting of several symbols.
Each of them is a CHAR of SYMBOL element (see TS_METADATA)
-->
<!-- VALUE contains the actual data. -->
<!-- Example:
  <CELL TIME_PERIOD="1979" SYMBOL="F" VALUE="200"/>
-->
<!-- ============================================================= -->
<!-- NOTE  -->
<!ELEMENT NOTE (#PCDATA)>
<!ATTLIST NOTE
    %REM;
>
<!-- NOTE contains optional remarks about a cell or cell series. -->
<!-- ============================================================= -->
<!--This is the end of the Time Series Data DTD.-->
```

Time Series Tag Library

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Maria Barbuzzi (FAO
Headquarters) -->
<!-- ============================================================= -->
<!-- Project: Time Series Universal Import Export -->
<!-- Component:
-->
<!-- Original author: Yury Shatz -->
<!-- Organisation: FAO of the UN, FIDI, FIGIS project -->
<!-- Date: 07/09/2001 -->
<!-- Modifications: -->
<!-- Date          Author              Note -->
<!-- =======    ============     ================================== -->
<!-- dd/mm/yy    who?                why? -->
<!--                                    -->
<!-- ============================================================= -->
<!--This DTD calls the Tag library.-->
<!-- ============================================================= -->
<!-- The trick here is to include taglib only once. -->
<!-- master file ignores this file TagLibInclude and includes TagLibCOntent directly. -->
<!-- Sub-files include this file.
-->
<![%TagLibInclude;[

<!ENTITY % FigisTimeSeriesTagLibContent PUBLIC
    "Figis Time Series Import Export Tag Library 1.0"
    "TSTagLibContent1.0.dtd">

%FigisTimeSeriesTagLibContent;

]]>
<!-- ============================================================= -->
<!--This is the end of the DTD Tag library.-->
```

Time Series Tag Library Content

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Maria Barbuzzi (FAO
Headquarters) -->
<!-- ============================================================= -->
<!-- Project: Time Series Universal Import Export -->
<!-- Component:
-->
<!-- Original author: Yury Shatz -->
<!-- Organisation: FAO of the UN, FIDI, FIGIS project -->
<!-- Date: 07/09/2001 -->
<!-- Modifications: -->
<!-- Date          Author              Note -->
<!-- =======    ============     ================================== -->
<!-- dd/mm/yy    who?                why? -->
<!-- dd/10/2001   Yury Shatz            ver. 2-->
<!-- ============================================================= -->
<!--This DTD contains basic generic elements and structures used in various parts of the Time
Series Universal Import Export DTD.-->
<!-- ============================================================= -->
<!--REM is a convenience entity called as an attribute when the developer needs to insert
comments within the data structure.-->
<!ENTITY % REM "REM CDATA #IMPLIED">
<!-- ============================================================= -->
<!--REF is a convenience entity containing the common elements used in all REF elements.-->
<!ENTITY % REF "ID | FOREIGN_ID | NAME">
<!-- %REF is an entity that contains:
        NAME contains a label together with attribute xml:lang that identifies the language of the
name data.
        ID contains an ID internal to the IMP/EXP system.
        FOREIGN_ID contains an ID external to the IMP/EXP system together with a coding
scheme name.
        References are used to establish relationships between elements. For instance,
        SOMEELEMENT_REF refers to SOMEELEMENT, having the same ID, FOREIGN_ID
        or NAME. These three ways of identification are tried in this order:
            ID
            FOREIGN_ID
            NAME

        SOMEELEMENT matching these IDs must exist either in import XML
        or in destination system.
 -->
<!-- ============================================================= -->
<!--ID_LIST contains a list of internal ID's and/or FOREIGN_ID's.-->
<!ELEMENT ID_LIST (ID | FOREIGN_ID)+>
<!ATTLIST ID_LIST
  %REM;
>
<!--ID contains an ID internal to the IMP/EXP system. -->
<!--FOREIGN_ID contains an ID external to the IMP/EXP system together with a coding scheme
name. -->
<!-- Example:

        <ID_LIST>
            <ID>Species799XXXXXXX005</ID>
```

```
            <FOREIGN_ID ID_SCHEME="Fishstat">GQXD</FOREIGN_ID>
        </ID_LIST> -->
<!-- ============================================================= -->
<!--Each instance of ID holds one ID value internal to the import tool.-->
<!ELEMENT ID (#PCDATA)>
<!ATTLIST ID
  %REM;
>
<!-- Example:

            <ID>Species799XXXXXXX005</ID> -->
<!-- ============================================================= -->
<!--Each instance of FOREIGN_ID holds one foreign ID value together with attribute
ID_SCHEME that identifies the foreign ID system.-->
<!ELEMENT FOREIGN_ID (#PCDATA)>
<!ATTLIST FOREIGN_ID
  ID_SCHEME CDATA #REQUIRED
  %REM;
>
<!-- Example:

            <FOREIGN_ID ID_SCHEME="Fishstat">GQXD</FOREIGN_ID> -->
<!-- ============================================================= -->
<!--NAME_LIST contains a set of label NAMEs.-->
<!ELEMENT NAME_LIST (NAME)+>
<!ATTLIST NAME_LIST
  %REM;
>
<!--NAME contains a label together with attribute xml:lang that identifies the language of the
name data.-->
<!-- Example:

        <NAME_LIST>
            <NAME xml:lang="en">Seaweeds nei</NAME>
            <NAME xml:lang="fr">Algues nca</NAME>
            <NAME xml:lang="es">Algas nep</NAME>
        </NAME_LIST> -->
<!-- ============================================================= -->
<!--NAME contains a label together with attribute xml:lang that identifies the language of the
name data.-->
<!ELEMENT NAME (#PCDATA)>
<!ATTLIST NAME
  xml:lang CDATA #IMPLIED
  %REM;
>
<!-- xml:lang identifies the language of the name data.-->
<!-- Example:

            <NAME xml:lang="en">Seaweeds nei</NAME> -->
<!-- ============================================================= -->
<!--OBJ_TYPE_REF contains the NAME(s), internal ID(s) and FOREIGN_ID(s)
for an object class to which an object belongs. Object type must either be defined
as OBJ_TYPE element or exist in destination system -->
<!ELEMENT OBJ_TYPE_REF (%REF;)+>
<!ATTLIST OBJ_TYPE_REF
  %REM;
```

```
>
<!--%REF is ID|NAME|FOREIGN_ID needed to identify Object Type -->
<!-- Example:

        <OBJ_TYPE_REF>
            <NAME>Species</NAME>
            <ID>SPE</ID>
            <FOREIGN_ID ID_SCHEME="Fishstat">34</FOREIGN_ID>
            <FOREIGN_ID ID_SCHEME="Figis">31005</FOREIGN_ID>
        </OBJ_TYPE_REF> -->
<!-- ============================================================= -->
<!--HIER_REF identifies a hierarchy to which another element refers,
using NAME(s), internal ID(s) and FOREIGN_ID(s).
Hierarchy must either be defined as HIERARCHY element or exist in destination system
-->
<!ELEMENT HIER_REF (%REF;)+>
<!ATTLIST HIER_REF
  %REM;
>
<!--%REF is ID|NAME|FOREIGN_ID needed to identify Hierarchy -->
<!-- Example:
    <HIER_REF>
        <ID>IsscaapHier</ID>
    </HIER_REF>
-->
<!-- ============================================================= -->
<!--REL_REF identifies a relation class to which another element refers
using NAME(s), internal ID(s) and FOREIGN_ID(s).
relation class must either be defined as REL element or exist in destination system
-->
<!ELEMENT REL_REF (%REF;)+>
<!ATTLIST REL_REF
  %REM;
>
<!--%REF is ID|NAME|FOREIGN_ID needed to identify relation -->
<!-- Example:
    <REL_REF>
        <ID>SpeciesInIsscaapGroup</ID>
        <FOREIGN_ID ID_SCHEME="figis">32001</FOREIGN_ID>
    </REL_REF>

        -->
<!-- ============================================================= -->
<!--UNIT_REF identifies a measurement unit to which another element refers
using NAME(s), internal ID(s) and FOREIGN_ID(s).
Unit must either be defined as UNIT element or exist in destination system-->
<!ELEMENT UNIT_REF (%REF;)+>
<!ATTLIST UNIT_REF
  %REM;
>
<!--%REF is ID|NAME|FOREIGN_ID needed to identify relation -->
<!-- Example:
    <UNIT_REF>
        <ID>mt</ID>
    </UNIT_REF>
        -->
```

```
<!-- ============================================================= -->
<!--ATT_REF identifies an attribute to which another element refers,
using NAME(s), internal ID(s) and FOREIGN_ID(s). Attribute must be either
defined with ATT element ort exist in the destination system.-->
<!ELEMENT ATT_REF (%REF;)+>
<!ATTLIST ATT_REF
  %REM;
>
<!--%REF is ID|NAME|FOREIGN_ID needed to identify relation -->
<!-- Example:
    <ATT_REF>
        <FOREIGN_ID ID_SCHEME="figis">13</FOREIGN_ID>
        <NAME xml:lang="en">3-alpha code</NAME>
    </ATT_REF>

         -->
<!-- ============================================================= -->
<!--FORMAT contains the internal ID or NAME of the format in which a piece of data is written.
Attribute STRING contains a string that describes the data format for output. Syntax of this string
is to be developed.
In fact, it may be system-dependent.
-->
<!ELEMENT FORMAT (ID | NAME)>
<!ATTLIST FORMAT
  STRING CDATA #IMPLIED
  DEFAULT (yes | no) #REQUIRED
  %REM;
>
<!--ID contains an ID internal to the IMP/EXP system. -->
<!--NAME contains a label together with attribute xml:lang that identifies the language of the
name data.-->
<!--STRING contains a string that describes the data format for output.-->
<!--DEFAULT declares the format instance to be the default for the object in which it is
contained.-->
<!-- Example:

         -->
<!-- ============================================================= -->
<!--VALUE contains a data value: attribute or time series value.-->
<!ELEMENT VALUE (#PCDATA)>
<!ATTLIST VALUE
  %REM;
>
<!-- Example:
            <VALUE>5</VALUE>
         -->
<!-- ============================================================= -->
<!--ATT_VAL contains an attribute value together with a reference to the attribute.
Reference created using an ID or FOREIGN_ID-->
<!ELEMENT ATT_VAL ((ID | FOREIGN_ID), VALUE)>
<!ATTLIST ATT_VAL
  xml:lang CDATA #IMPLIED
  %REM;
>
<!--ID contains an ID internal to the IMP/EXP system. -->
```

```
<!--FOREIGN_ID contains an ID external to the IMP/EXP system together with a coding scheme
name. -->
<!--VALUE contains a metadata value.-->
<!-- Example:
        <ATT_VAL>
            <ID>ScientificName</ID>
            <VALUE>Algae</VALUE>
        </ATT_VAL>
-->
<!-- ============================================================= -->
<!--OBJ_REF identifies an object to which another element refers,
using NAME(s), internal ID(s), FOREIGN_ID(s) and ATT_VALs. Object must be either
defined with OBJ element or exist in the destination system.
IDs, names, attribute values are used for object identification in
following order:
ID
FOREIGN_ID
ATT_VAL
NAME
-->
<!ELEMENT OBJ_REF (%REF; | OBJ_TYPE_REF | ATT_VAL)+>
<!ATTLIST OBJ_REF
  %REM;
>
<!--%REF is ID|NAME|FOREIGN_ID needed to identify relation -->
<!--OBJ_TYPE_REF refers to object type. It may be necessary to include
OBJ_TYPE_REF if several objects of different types may have the same ID or
FOREIGN_ID. In this situation system should match only
objects of correct type.
ATT_VAL is additional type of identification. If neither ID nor
FOREIGN_ID is known, object is matched using ATT_VAL.-->
<!-- Example:
    <OBJ_REF>
        <ID>SpeciesABC001</ID>
        <OBJ_TYPE_REF><ID>Species</ID></OBJ_TYPE_REF>
        <ATT_VAL><ID>Code3Alpha</ID><VALUE>ABC</VALUE></ATT_VAL>
    </OBJ_REF>
         -->

<!-- ============================================================= -->
<!-- RELATED_TYPES is a reference to the object type on the other end of the relationship.-->
<!ELEMENT RELATED_TYPES (OBJ_TYPE_REF)+>
<!ATTLIST RELATED_TYPES
    %REM;
>
 <!--OBJ_TYPE_REF contains the references to the related object types, NAME(s), internal ID(s)
and FOREIGN_ID(s).-->

<!-- Example:
    <RELATED_TYPES>
      <OBJ_TYPE_REF><ID>ISSCAAPGroup</ID></OBJ_TYPE_REF>
    </RELATED_TYPES>
  -->


<!-- ============================================================= -->
```

```
<!--This is the end of the Time Series Universal Import Export Common Library Content DTD.-->
```