

Universal Terminology

CHAPTER 3: IDENTIFIERS

The \mathbf{T}_{logy} is made of elementary building blocks which are named anatomical entities. Entities must be uniquely identified in order to organize the database according to the needs of its users.

The goal of this document is to present the unique identifier of entities in TAH. There are a number of constraints and properties which are necessary for a unique identifier in a database. Though this aspect is not specific to anatomy, it is important that all actors share a common understanding. Strict rules are defined and should be constantly applied for the life of the database. This document is the chapter 3 of the book Universal Terminology which presents a global documentation on the \mathbf{T}_{logy} .

Contents

3.1	Summary of the rules	2
3.2	The global situation	2
3.3	Meaning of the identifier	3
3.3.1	Short positive integer	3
3.3.2	Increasing sequential delivery	3
3.3.3	Existence of an identifier	4
3.3.4	Position of the last delivered identifier	4
3.4	Permanence of the identifier	5
3.4.1	Permanence of the link entity-identifier	5
3.4.2	Nobody can create an identifier	5
3.4.3	No reuse of a deleted identifier	5
3.5	Management of unused identifiers	6
3.6	Deletion of entities	6
3.6.1	Physical deletion is forbidden	6
3.6.2	Resuscitation in case of error only	7
3.7	Database minimal requirements	7
3.7.1	Control of the next delivered identifier	7
3.7.2	List the attributed identifiers by type	8
3.8	Identifiers and granularity	8
3.8.1	Unit identification	8
3.8.2	List identification	9
3.8.3	Word identification	9
3.9	Log of updates	10
3.10	Credentials	10

3.1 Summary of the rules

The rules applicable for the creation and the management of an entity **identifier** are the following. They are reviewed in details in this chapter.

- | | |
|----|--|
| 1 | An identifier is an integer value between 1 and 999'999. |
| 2 | An identifier is delivered in an increasing sequence within the allowable range. |
| 3 | The existence of an identifier cannot be inferred from the existence of another identifier. |
| 4 | No speculation on the current value of the next delivered identifier is allowed. |
| 5 | The link between the unique core of an entity and its identifier is guaranteed to remain constant. |
| 6 | Nobody can create an identifier on his own. |
| 7 | Never reuse an already publicly attributed identifier. |
| 8 | Manage a pool of unused identifiers and make it close to empty at any time. |
| 9 | Move any deleted entities to their cemetery. |
| 10 | Never resuscitate an entity unless an error of deletion is evident. |
| 11 | List and document the deleted entities. |
| 12 | Allow the reset of the last delivered value. |
| 13 | Prepare a listing of identifiers by type. |

Table 3.1: 13 rules applicable to the entity identifiers.

3.2 The global situation

The present number of entities in TAH is around 50'000 units and it should be ready to accommodate some substantial growth, depending on possible future developments, without any necessary redesign of the database structure. For this reason, the design has to be ready for a million of entities, without making problem or decreasing performance. This figure can easily be taken into account, without special effort, according to the systems available today.

An entity **identifier** is not only an internal pointer, it can be made visible to users, which may need occasionally to copy or transfer this identifier. This gives a first precondition for the unique identifier: it should be easy to memorize for a short time without difficulty. For example, reading the identifier and typing it in another window is a common situation. The solution is to use positive integers with five digits extensible to six digits. It would be counterproductive to use an identifier with more than six digits, because of the limits of human short time memory.

For external usage of the identifier, outside of the TAH database, someone wants to point to an entity in a scientific communication, it would be recommended to prefix the unique identifier by the three letters TAH. The TAH3993 identifier means: go to the TAH database and ask for the internal identifier 3993, which corresponds to *LA: pars sphenoidalis arteriae cerebri mediae*

It is important to understand that the unique TAH identifiers will exist by their own, outside of a strict database implementation. This is due to the fact

that IFAA is an international organization, with the specific goal of sharing knowledge with anyone else in the scientific community, without restriction either economical or political. Once we start to share an information, we lose immediately some ownership and in particular the freedom to arbitrarily change a critical content. This is exactly what happens regarding the entity identifiers. In the above example, attributing an identifier to an entity and sharing its value, implies a commitment of the authors of TAH not to alter this identifier at any future time. However, if this must occur anyway, it must be regulated by restrictive rules and accompanying information.

This document is a presentation of the rules to be applied by the managers and the users of the TAH database in any circumstances. Being necessarily approved by FIPAT and consequently endorsed by IFAA, this document is not depending on the willingness of any individual. In addition, this document should be permanently accessible on a public basis, which is the best way to enforce it.

The following organization has been retained:

- Meaning of the identifier,
- Permanence of identifiers,
- Management of unused identifiers,
- Deletion of entities,
- Database minimal requirements.

3.3 Meaning of the identifier

Though it would be best that the identifier has no particular meaning, it is necessary to discuss about what can be guaranteed or not by the instances of those identifiers. The arguments here may seem more or less trivial, but making important things explicit is never a lost of time. We will discuss in turn different rules applicable to the identifier.

3.3.1 Short positive integer

This aspect was already introduced above.

Entity identifier

An identifier (definition) is an integer value between 1 and 999'999.

3.3.2 Increasing sequential delivery

There are at least two possible solutions: to deliver the identifier as a sequence or to deliver the identifier randomly within the specified range. Both solutions are usually available and each one has advantages. The random distribution is more blind and secure than the sequential distribution, but pragmatically the sequential distribution allows better human controls. This is not the place to argue about two different techniques. The early authors of the TAH database have selected the sequential delivery. In reality, this is the most commonly used approach by database developers.

It will be seen later that during the development of a database, some identifiers are created by error and we do not want them to be lost. Because they have not been published, they can be recycled and reused without problem. A pool of unused identifiers will be created. Removing an identifier from the pool is an operation distinct but equivalent to delivering a new identifier.

The fact that identifiers are delivered in sequence means that some time elapses between their successive deliveries. Therefore, it is possible to infer some time interval from the difference between two identifiers. However, this is quite an approximative exercise for two reasons: the time scale could change without warning; the existence of a pool of unused identifier makes any break of sequence possible. Consequently, such inferences are considered as inadequate.

Sequential delivery

An identifier (definition) is delivered following an increasing sequence within the allowable range.

3.3.3 Existence of an identifier

The exercise of inferring the existence of an identifier from the fact that a greater identifier exists is to be strictly prohibited. As we will see below, care will be taken to maintain the sequence of delivered identifiers as compact as possible. But this never means that every previously delivered identifier still exists. The managers of the database have the right to organize themselves as they want and without warning.

When using database procedures, an error message will be delivered each time a user uses a non-existing identifier. This is a way to learn about the existence of an identifier. Other procedures outside of the database, but based on the actual database content, are subject to the same conditions.

Identifier existence

The existence of an identifier (definition) cannot be inferred from the existence of another identifier.

3.3.4 Position of the last delivered identifier

The database remembers of the last delivered identifier, in order to be able to deliver the next one. But in general, this value is not directly available to common users. For this reason, this value is to be considered as not accessible to users and their applications. However, the managers of the database may have access to this value and even can modify it. This is not a recommended exercise. In addition, this kind of operation is implementation-dependent. In no way could any user speculate on its availability.

No speculation on identifier

No speculation (definition) on the value of the next delivered identifier is allowed.

3.4 Permanence of the identifier

3.4.1 Permanence of the link entity-identifier

An identifier is automatically created every time a new entity is created in the table of entities and its value is attributed to this entity. But at the same time the entity still is empty and it could be filled with any content. This is to say that the database does not know about the fields that are defining an entity and the identifier. Moreover, an unexpected procedure may exchange the content of two entities, the result being that the two entities are always present but their identifiers have been exchanged. In other words, the database provides no guarantee about the content of entities to which an identifier has been attributed.

It is the responsibility of the database managers to maintain a coherent database. In particular, they guarantee the permanent link between an identifier and the core of an entity which are the attributes of its uniqueness.

However, the content of the database may be shared between several users and the control of a stability of the links entity-identifier is a difficult task.

Identifier permanence

The permanence of an identifier (definition) is the link between the unique core of an entity and its identifier, which is guaranteed to remain constant and permanent in time.

As a result of this definition, an identifier must never be reused. Therefore, when an entity is no more in use or has been replaced by other entities, it must be given the status *deleted* which is interpreted as a no show in the terminology. However, this entity still exists.

Numerous example can be found in TA98, where a few tens of entities have been deleted when this version of the terminology has been updated. For example *LA:arteriae nutriciae femoris* has been replaced by the *LA:arteria nutricia femoris proximalis* and the *LA:arteria nutricia femoris distalis*.

3.4.2 Nobody can create an identifier

Identifiers are created by a unique process internal to the database, in the reference version of this database and whatever are the distributed copies of the database. This is strictly the only way to create a new identifier.

Central generation of identifiers

A central database (definition) is the only way to create new identifiers and nobody can create an identifier on his own.

3.4.3 No reuse of a deleted identifier

Under the principle of permanence of the link entity-identifier, it is primordial to never reuse an identifier after a deletion, at least if the identifier as been published. If an identifier were re-used, an external user may inadvertently

consider that the user is always linked to the past entity although the link has disappeared.

As an example let us consider Wikipedia who is indexing articles to TA as well as other sources. The index that they use is based on the unique identifier. If the present rule is not applied, Wikipedia will point to the wrong entity, forcing them to revise of all their applications. The existence of TAH in the scientific community is highly depending on such criteria. There is no choice for FIPAT, and we must follow that common-sense rule.

Reuse is forbidden

Reuse of an identifier (definition) is forbidden as soon as this identifier is part of a publicly validated version of the terminology.

3.5 Management of unused identifiers

Within the daily operations of a living database currently accepting updates and upgrades, it is frequent to create a new entity by error. Typically one creates an already existing entity. When the error is discovered, the newly created entity is discarded and its identifier is available for another entity. It is available because the temporary new entity is unknown by the rest of the word. In fact hundred of such entities may be inadvertently created in a single week of work. A management strategy has to be set up.

Any identifier attributed and released by error should be recycled for further use when the mistake is discovered. A pool of such identifiers must be created. It is in fact a list of empty entities. When a false identifier is found, the entity is emptied and this empty entity is moved elsewhere in the database. When a new identifier is needed, instead of creating a new one, it is possible to take one from the pool. This strategy of management is easily supervised by a database manager who takes care of the existence of a too rich pool and empties it if necessary.

Management of unused identifiers

The management of unused identifiers (definition) is realized as a pool of unused identifiers available on request.

3.6 Deletion of entities

Whatever is the evolution of the database, sooner or later some entities are no more in use and must be discarded. This means that these entities become inaccessible to ordinary users and their applications. This does not mean that these entities should be physically deleted.

3.6.1 Physical deletion is forbidden

In the situation of TAH, the removal of entities should be a relative exception: what was once present in a human body is not expected to disappear tomorrow.

Therefore, for safety reasons and in order to keep an history of deletions, it must be decided to move the deleted entities to a pool or cemetery. This is a one-way transfer, because these entities have been published and their identifiers are banned forever.

Cemetery of deleted entities

A cemetery of deleted entities (definition) is a recipient where to retrieve all deleted entities.

A list of deleted entities must be available. The reason is that these entities have officially existed for some period of time and therefore they can never be lost. At any moment a casual user may come and ask about an entity: an answer is expected even for the deleted entities.

It should be remained that some 7500 identifiers have been made public in January 2013 as the TA98 online version. A number of these identifiers could be deleted during the ongoing revision.

3.6.2 Resuscitation in case of error only

Unless there is a need to resuscitate a deleted entity, such an operation is strictly forbidden. However, it may happen that an entity has been deleted by error and that the correction of this error is a resuscitation. It must be proved that the new entity and the past entity are really the same entity.

Resuscitation

A resuscitation (definition) is the recovering of a deleted entity by error.

In practice, resuscitation is more current than expected. In a database, new entities are created with new identifiers in sequence. But the deleted identifiers are lost, unless a process of resuscitation is programmed. Without such a process, one may be constrained to face unexplained holes in the list of identifiers. Even if resuscitation is not strictly speaking mandatory, the authors of the database of \mathbf{T}_{logy} have preferred to resuscitate unexpectedly deleted entities. In this way, the list of identifiers is continuous from zero to the last generated entity (close to 50000 in May 2023).

3.7 Database minimal requirements

The database system should fulfil a set of minimal requirements for the proper administration of the TAH database. Such requirements are explained now, without inferring any thing on the implementation system.

3.7.1 Control of the next delivered identifier

This is clearly a mandatory feature, which remains relatively undocumented in certain database systems. In theory such a need should never occur, but in reality it does. For example it may happen that some entities are deleted by accident, whatever its origin: user, system, etc. In this situation the database manager has to reset the last value of delivered identifier to any value, new

entities must be created from there, until the last value is finally restored to its initial value.

Reset of identifiers

A reset of identifiers (definition) is a special process which allows the restart of the generation of identifiers to any value.

This process allows the necessary operation of resuscitation.

3.7.2 List the attributed identifiers by type

It may be useful to be in position to deliver some statistics on the usage of identifiers. In particular some evidence about the compactness of the identifiers in their range of value may be necessary for optimization purpose.

Listing of identifiers

Listing of identifiers (definition) are listings by type of entities for the attributed identifiers and listings of available free identifiers for generic entities (below 25000) or other entities (above 25000).

Such listings are mandatory for the proper management of the database. They are only internally available in the database.

3.8 Identifiers and granularity

The unique identifier, as defined above, is applicable to the anatomical entities. How do we identify the other atoms of the granularity, starting from the entity identifiers? How do we identify the units, the words and the lists?

3.8.1 Unit identification

Units are made one to 5 entities depending on its type. Therefore, the Unit identifier is always selected from one of these entities. Of course when the unit is made of a single entity, the entity identifier becomes the Unit identifier. This is true for the single entity, the taxonomic entity, the vocabulary entity and the likes.

For the set entity, the mixed set entity, the pair entity, and the pair set entity, the rule is to select the identifier of the generic entity as identifier of the unit.

Unit identifier

A unit identifier (definition) is the identifier of its generic entity.

For example, consider the unit *LA:931 humerus (par)*: this unit is a pair and consequently is made of four entities: the generic *LA:931 humerus*, the specific pair *LA:32729 humerus (pair)*, the specific left member *LA:32730 humerus sinister* and the specific right member *LA:32731 humerus dexter*. The identifier of the generic entity is 931 and becomes the identifier of the unit. Because the generic

entities always have an identifier below 25000, this feature is also true for unit identifiers.

3.8.2 List identification

List are generally hierarchical lists according to one of the active hierarchies of the \mathbf{T}_{logy} : the partonomy, the taxonomy or the TA98 hierarchy. List are identified by the identifier of their top entity defining the list. Formally, the identifier must also contain the type of hierarchy, in order to prevent duplicates of identifier. But this aspect has not yet been implemented!

List identifier

A list identifier (definition) is the identifier of its hierarchically top entity.

3.8.3 Word identification

Words are represented by single entities and are therefore identified by the corresponding entity identifier.

3.9 Log of updates

30 Mar 2022 Standardisation of the file as a chapter.

21 Oct 2021 Creation of the file.

3.10 Credentials

This document is part of the book "Universal Terminology" accompanying the website on Terminologia Anatomica. It expresses the vision of the authors of the *T_{logy}* about the foundations of the science of ontology, supporting the here presented terminology. Despite it is as exact as possible, close to the reality of the database of the terminology and the surrounding software, approximations, errors and ambiguities are possible and should be considered as independent of their willingness and intents.

Identified comments about the content of the website and its presentation are welcome. An appropriate answer will be given when pertinent.

Authentic URL of this file: <https://ifaa.unifr.ch/Public/TNAEntryPage/help/Chap03.pdf>