THROUGH LIFE SUPPORT STANDARD

Guidance on Writing TLSS Business Objects

LSC REFERENCE: ECSMODTLSS5025.119
Eurostep REFERENCE: Eurostep.ESUKPC09.000156

Authors	Checked	Approved
Brad Harris - Eurostep	Rob Bodington - Eurostep Tim Turner – LSC Group	Simon Dick - Eurostep Paul Clark – LSC Group
Issue	**Date**	**Client**
1.0	31 October 2007	TLSD Policy Coordination – TLSS Project Manager

LSC Group Ltd.
Lincoln House
Fradley Park
Lichfield
Staffordshire
WS13 8RZ
United Kingdom

Eurostep Limited,
Cwttir Lane,
St. Asaph,
Denbighshire,
LL17 0LQ,
United Kingdom

Amendment Record

Issue	Date	Summary of changes
1.0	2007-10-31	First Issue, delivered by TLSS Work Package 1, Sub-task 1

Distribution

This document has been distributed to:

Copy No	Media	Location	Title
1	PDF File		TLSD Policy Coordination – TLSS Project Manager

TABLE OF CONTENTS

1 INTRODUCTION

1. Given the scope statement, information requirements and business data planning model of a TLSS DEX, the detailed structure and content of the TLSS Business Objects[1] that constitute the DEX, need to be specified.
2. The objective is to re-use as much previous TLSS work as possible and to closely align the business object model with the information requirements where this is appropriate, thereby iteratively and incrementally developing an integrated TLSS Business Object model that comprises:
   - The Information classes defined by the TLSS process model information flows.
   - The information classes derived from the TLSS information requirements repository.
   - New information classes that have not yet been defined by the previous two sets of classes, but which are required by a TLSS data exchange.
3. In order to facilitate this, and to ensure that the business object model has firm foundations, and isn't starting from a blank sheet of paper, the TLSS information requirements have been translated into a set of UML classes. These can now be used alongside the TLSS process model information flow classes as the starting point for the development of the TLSS Business Object Model.
4. It is assumed that the accounts and software listed below are activated and set up / installed.

2 KNOWLEDGE, SKILLS AND TOOLS REQUIRED

1. The process requires a thorough understanding of:
   - The TLSS process model and the information flow classes that it defines.
   - The TLSS information requirements.
   - UML information modelling methods and practices.
2. The following accounts and software are required:
   - MagicDraw (version 12.5) – for creating the UML representation of the business objects.[2]
   - TLSS Information Requirements repository read only account – for accessing the requirements repository and lodging issues against it. Contact LSC for access information

3 GUIDANCE NOTES ON CREATION OF TLSS BUSINESS OBJECTS

3.1 Representation

1. TLSS Business Objects shall be created as UML classes using MagicDraw version 12.5.
2. The file in which they shall be defined is "Common Business Information Structure.mdzip"
3. Each business object shall also be depicted graphically as a Composite Structure diagram.
4. Each diagram shall be stored as part of the Business Template that will be created to map the Business Object to its PLCS representation[3]. This shall be in jpeg or png format in the "\images" folder of the Business Template.

[1] Here after referred to as "the business object"

[2] Note: MagicDraw will be used to define the Business Objects for the first set of TLSS DEXs. The MooD system may be used subsequent to that.

3.2 Naming

1. The name of each business object shall be the name given in the process of identifying the constituent business objects of the DEX in accordance with [1] .
2. Names shall always be full names and never abbreviated. They shall not contain acronyms.
3. Names will be written in CamelCase, as for all other UML class names defined by TLSS previously.

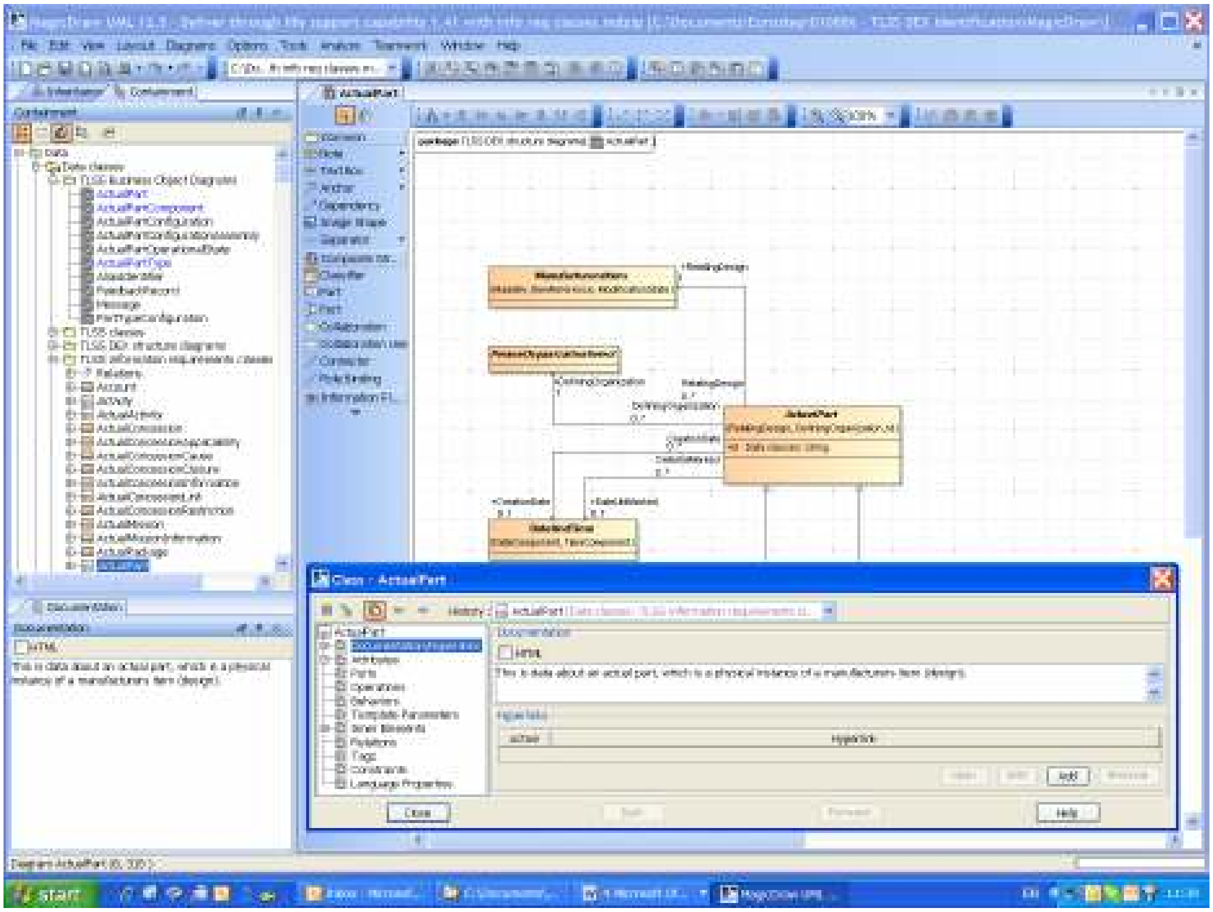
3.3 Definition

1. The business object is defined by a process similar to that employed when defining the data planning model in a TLSS Business Data Exchange Specification document [2] :
    - selecting from the existing pool of business objects, one that closely matches the requirements of the exchange(s) under consideration;
    - modifying it where required, by subtyping/supertyping, adding attributes or relationships etc;
    - adding to the existing pool of business objects where no appropriate business object already exists.
    - ensuring that the semantics of the resulting business objects do not conflict with the semantics already defined – i.e. all modifications need to viewed from the perspective of the overall model as well as from the perspective of all other individual exchange requirements.

    The difference here, is that closely related objects and relationships that may be required in other TLSS data exchange contexts, shall be considered for inclusion in model. This is because each business object may be used by several different TLSS DEXs, and in order to develop a coherent information model that serves all such purposes, it cannot be constrained by any one data exchange requirement.
2. For example, an “ActualPart” business object is defined in the “Documentation/Hyperlinks” property of the class as follows:

    This is data about an actual part, which is a physical instance of a manufacturers item (design).

[3] Note that there is a one:one correspondence between a business object and the template that defines its mapping to the PLCS data model.



This is data about an actual part, which is a physical instance of a manufacturers item (design).

Figure 1 Business Object Definition for ActualPart

3. The graphical UML representation of the business object is shown in the following diagram:

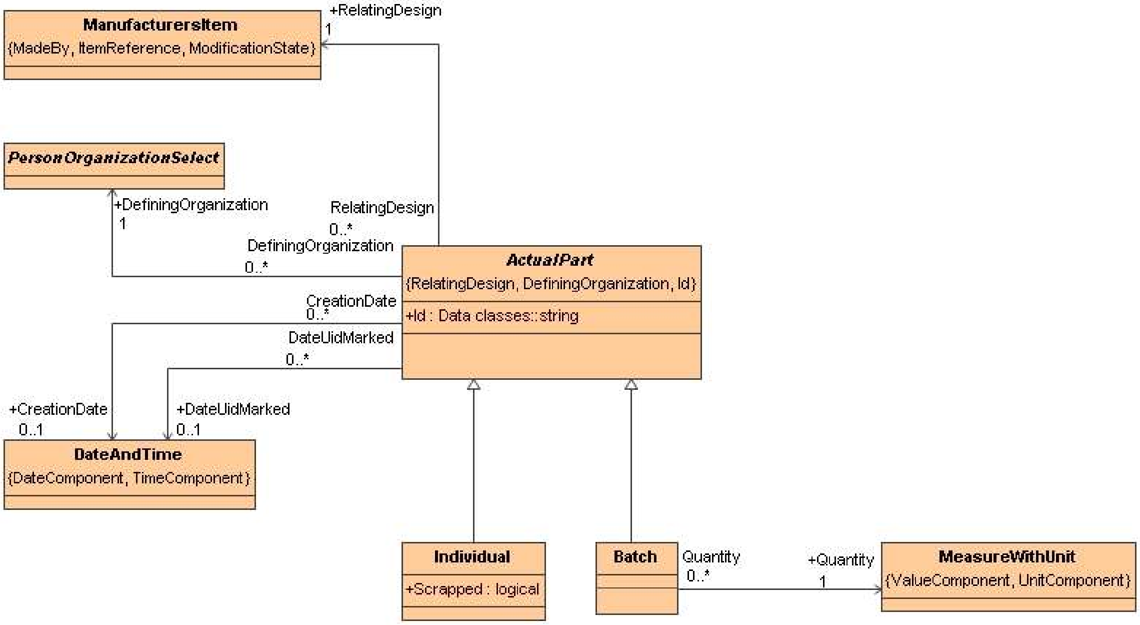
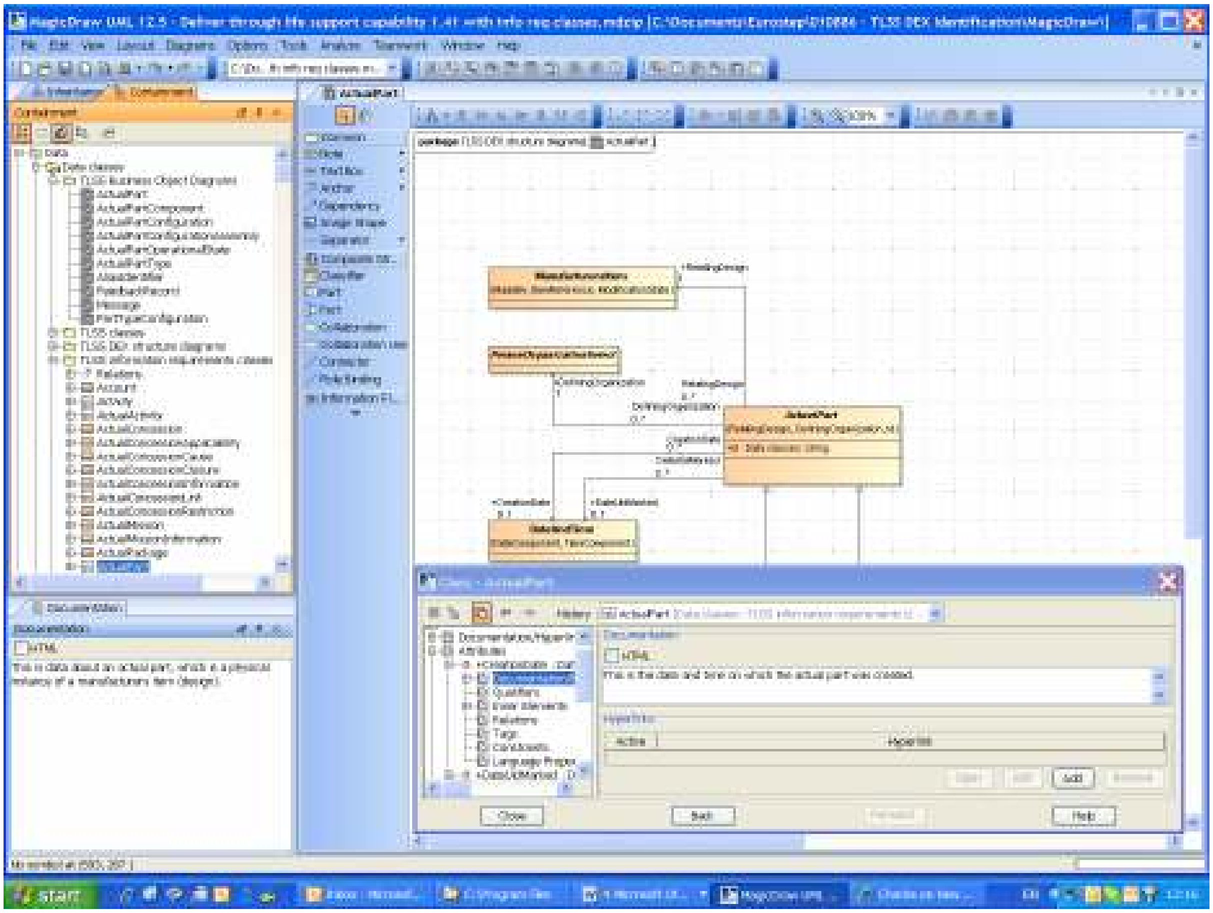


Figure 2 Graphical Representation for Business Object ActualPart

4. In this case:
    a. All of the attributes/relationships of the business object were included from the original objects created from the TLSS information requirements.
    b. No additional objects, attributes or relationships needed to be created.
5. In other cases, new business object may be required, or additions to existing business objects may be required in the form of further attributes and / or relationships. For example, when the “Nato_item_of_supply_codification” TLSS Business DEX was developed, there was a need for a “PartsList” business object and this had not previously been defined. As stated in [1] when this is the case, the new business object shall be created within the context of existing business objects to ensure that duplicate, redundant and inconsistent business objects are not proliferated.
6. In order that the business object is unambiguous, every attribute/relationship shall be fully defined using the “Documentation/Hyperlinks” property of the attribute/relationship. This shall include optionality, cardinality and defining the type of the attribute/relationship.
7. For example, the “CreationDate” attribute of an “ActualPart” Class can be defined as follows:

    This is the date and time on which the actual part was created.



MagicDraw UML 12.5 - Deliver through life support capability 1.41 with info req classes.mdzip [C:\Documents\Eurostep\D1088 - TLSS DEX Identification\MagicDraw\]

Documentation
□ HTML
This is data about an actual part, which is a physical instance of a manufacturers item (design).

Class - ActualPart

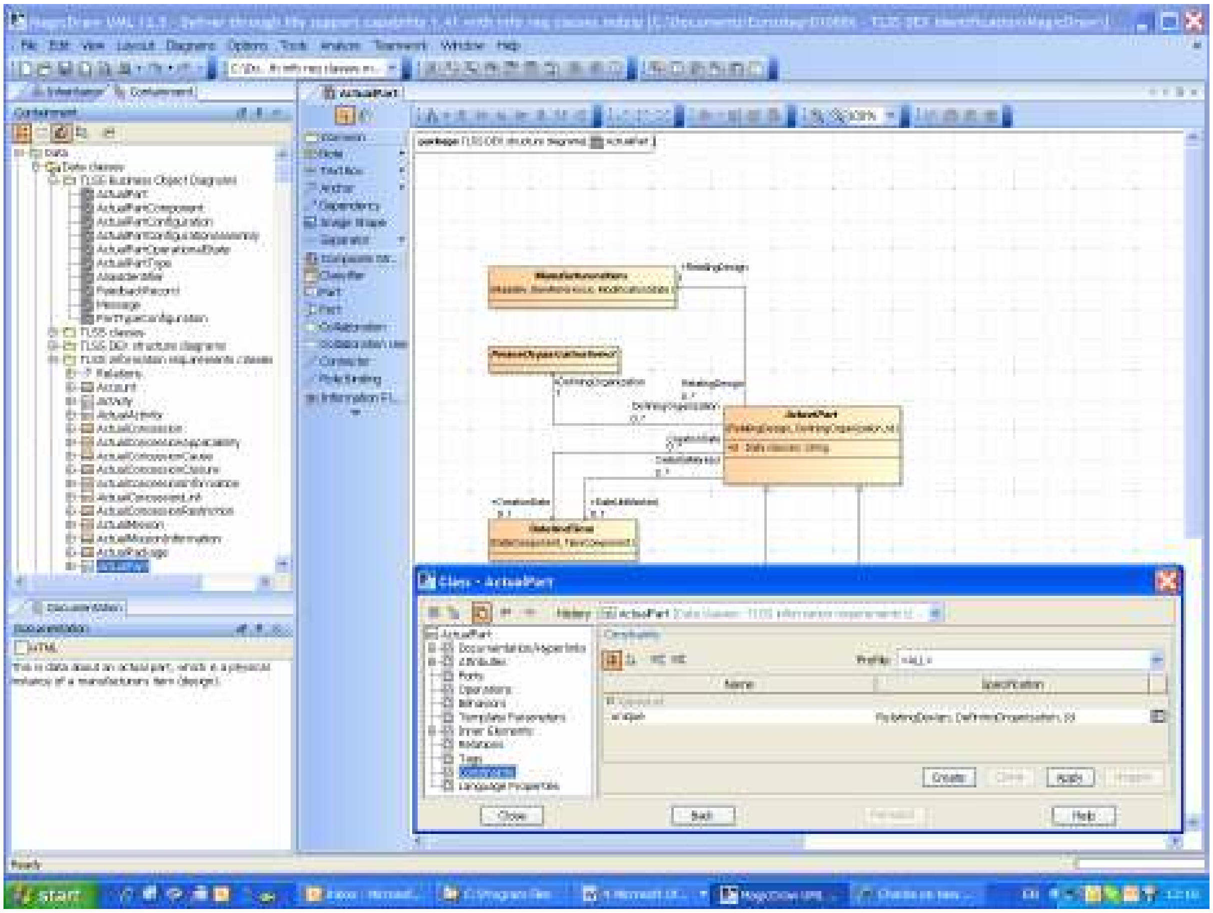
Documentation
□ HTML
This is the date and time on which the actual part was created.

Figure 3 Business Object Attribute Definition for ActualPart.CreationDate

3.4 Identification

1. Since each business object, by definition, is a human interpretable item of information, it is necessary to define the set of attributes/relationships that enable a human to distinguish one instance of a business object from another instance of the same type.
2. This shall be achieved by creating an instance of a constraints property of the class, called “unique”, that lists the set of attributes/relationships that together uniquely identify each instance of the business object. This analogous to defining the primary key of a relational database table.
3. For example, the “unique” constraint for the “ActualPart” class can be defined to be:

   unique RelatingDesign, DefiningOrganization, Id



Name	Specification
unique	RelatingDesign, DefiningOrganization, Id

Figure 4 Unique Constraint Definition for Business Object ActualPart

4. This constraint will be used in the corresponding business template to define its “UNIQUE” clause in terms of PLCS attributes and relationships that form part of the mapping from the business object to PLCS Templates and entities.

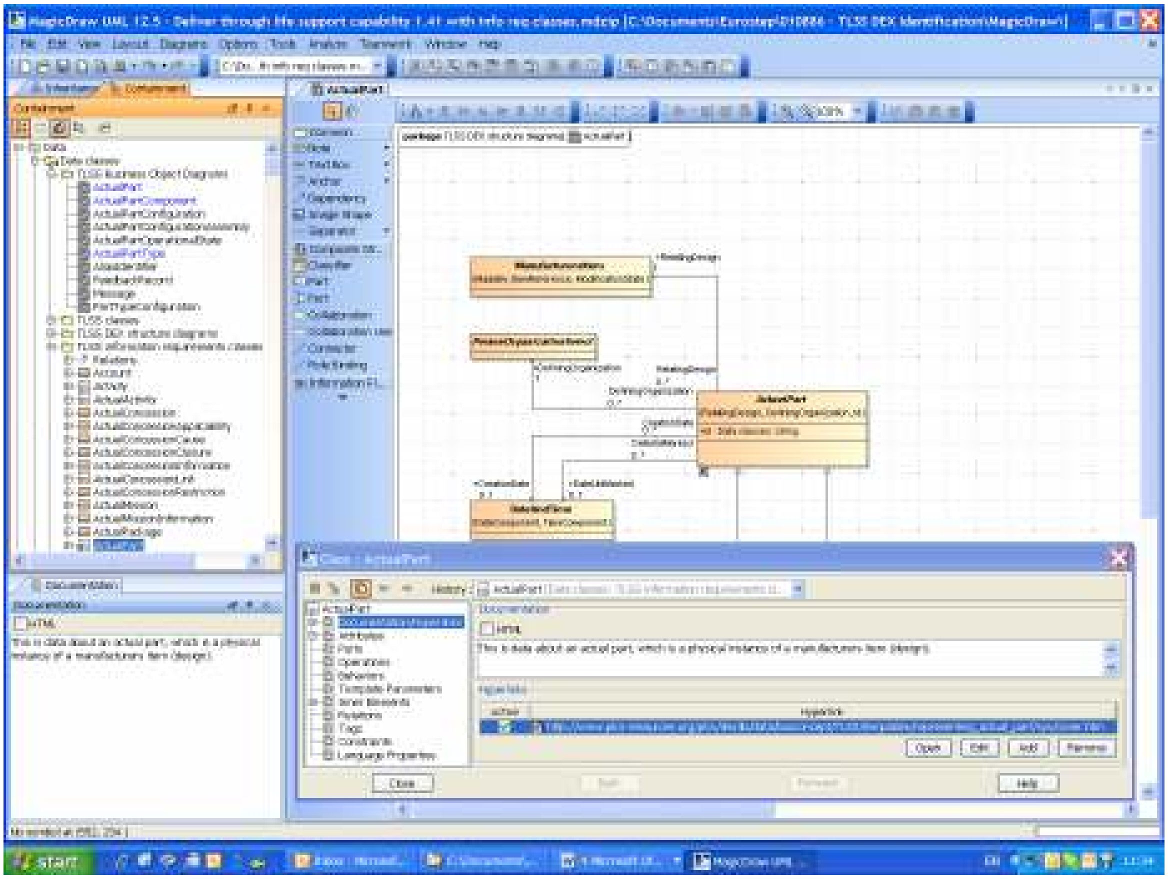
3.5 Modelling Issues

1. When modelling questions or issues arise, these shall be discussed with other TLSS Business Object developers whereby a common understanding and agreement can be reached. If agreement can not be reached, then the MOD TLSS Project Manager (or nominated deputy) shall have the final decision.

3.6 DEXLib Business Template Reference

1. When a business object has been defined a hyperlink reference to the DEXLib Business Template(s) that defines its mapping to PLCS model elements, shall be inserted in the “Documentation/Hyperlinks” property of the class.[4]
2. For example, for the “ActualPart” business object, the hyperlink to the plcs-resources web site that contains the corresponding DEXLib Business Template for the object shall be inserted:

   http://www.plcs-resources.org/plcs/dexlib/data/busconcept/TLSS/templates/representing_actual_part/sys/cover.htm



This is data about an actual part, which is a physical instance of a manufacturers item (design).

Figure 5 DEXLib Business Template Hyperlink Reference for Business Object ActualPart

3. Note that in order to insert this hyperlink the corresponding DEXLib Business Template for the business object must have been created and uploaded to Sourceforge. See [3] .

4 APPROVAL OF TLSS BUSINESS OBJECTS

1. Once completed from a development perspective, each TLSS Business Object shall be peer reviewed by ESL / LSC personnel who have not been directly responsible for the creation of the business object.
2. Issues arising from the review process shall be raised by the reviewer with the author(s).

[4] Note that there may be more than one Business Template referenced depending on the complexity of the Business Object – see [3] .

3. Once a business object has been reviewed, its corresponding DEXLib Business Template can be developed – see [3] .
4. The approval process however is with respect to the TLSS Business Object Model consisting of all business objects, rather than being against each individual business object i.e. the MagicDraw file referenced at 3.1 is the item that will be approved, and it will be the owner of the TLSS Business Object Model who is formally responsible for this.

5 MAINTENANCE OF TLSS BUSINESS OBJECTS

1. Once the TLSS Business Object Model has been approved, changes need to be proposed and approved before being implemented.

6 REFERENCES

[1] TLSS Data Exchange Specification Development Methodology.
[2] Guidance on writing DEXlib TLSS business data exchange specifications.
[3] Guidance on writing DEXlib TLSS business templates.