

# Comments to template assigning\_identifiers

## Introduction

This document presents comments and questions from the Norwegian implementation project based on a review of the capability assigning\_identifiers. The focus of the review was section “Capability templates”.

## General comments from implementers (that has some PLCS experience)

- Documentation in “Reading templates” (ref. DEXLib Help file) is necessary for the understanding of section “Caqpability templates”
- The document “Reading templates” is easy to understand and the documentation level is satisfactory from an implementers point of view
- Similar documentation as “Reading templates” is needed to achieve an understanding of the “logic” in the reference data library structure
- Section “Capability templates” is understandable from the implementers point of view
- It might be a question if the sub-section “Reference-path” is needed as long as the input parameter list and the instantiation examples are part of the documentation
- Documentation of rules/constraints should be made more explicit
- Rules/constraints should be possible to identify (i.e. unique name or number)
- The textual expression of rules/constraints should be unambiguous (not only by text)
- The section “Information model overview” is of minor interest for the implementers. It might be of interest if the implementers have no PLCS experience at all
- Where to store instantiated parameter sets of the templates? Ref. Table below Figure 5 in “Reading templates” document in Help section of DEXLib

*For more detailed comments and questions, please see comments below.*

## Capability templates

The following sections define a set of templates for the capability, where a template is a specification of a set of entities that need to be instantiated to represent a given set of information.

### Template: assigning\_identification (Short name: asg\_id)

This section specifies the template assigning\_identification.

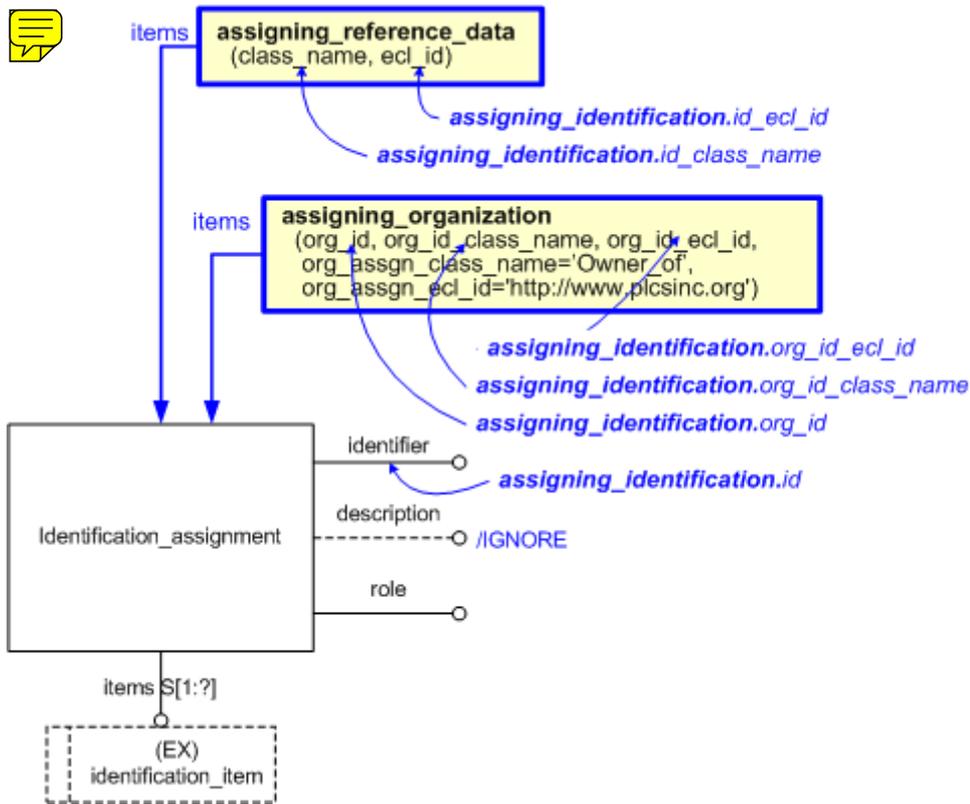
NOTE An explanation of a template and the associated instantiation path is provided [\[triha1\]](#) in the [Template overview](#) section.

#### Description:

This template describes the assignment of an identifier to something. [\[triha2\]](#) identifier must be defined in the context of an organization. [\[triha2\]](#) type or role of the identifier [\[triha3\]](#) provided by assigning reference data to the identification assignment [\[triha3\]](#) and thereby, [\[triha4\]](#) ifying the identification assignment as a sub class of “[Identifier\\_type](#)”. [\[triha4\]](#)

#### Model diagrams

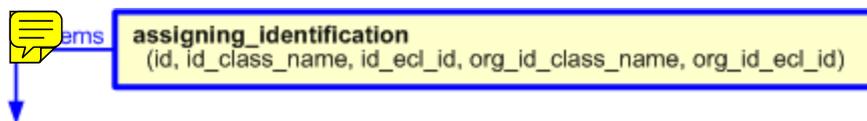
[\[triha5\]](#) EXPRESS-G diagram below shows EXPRESS entities that are required to represent the template “[assigning\\_identification](#)” [\[triha5\]](#). The text highlighted in blue, shows the template parameters.



[triha6]

Figure 5.1 – Template Configuration for Assigning Identification

The graphic for the template to be used in other EXPRESS-G diagrams is shown below.



[triha7]

Figure 5.2 – Template Summary

### Input parameters:

The following input parameters are defined for this template.

**id**(Type='STRING')

The identifier being assigned.

**id\_class\_name**(Type='CLASS')

The name of the class used to classify the identifier and so provide the role or reason for the identification.

The following classes and their sub-classes  e used [triha8].

**classifications:**  fier\_type [triha9]

**id\_ecl\_id**(Type=)

The id of the [External class library](#) storing the id\_class\_name class

**id**(Type='STRING') [triha10]

 identifier of the organization that "owns" the identifier.

**org\_id\_class\_name**(Type='CLASS')

The name of the class being used to classify the identification of the organization. For example CAGE code.

The following classes and their sub-classes can be used.

**classifications:** [Organization\\_identification\\_code](#), [Organization\\_name](#)

**org\_id\_ecl\_id**(Type='URN')

The id of the [External class library](#) storing the org\_id\_class\_name class

**items**(Type='SELECT (identification\_item)')

The items to which the identification is assigned

## Reference parameters:

The following reference parameters are defined for this template.

sgn([triha11]Type='Identification\_assignment')  
by the [Identification\\_assignment](#) entity instantiated in this path to be referenced when this template is used. [triha12]

Note: The [Identification\\_assignment](#) entity can be referenced in a template path by:

```
%^target = $assigning_identification.id_assgn%
```

where target is the parameter to which the [Identification\\_assignment](#) is bound.

For example, to identify the date of an identification by assigning a date to the [Identification\\_assignment](#) entity. E.g.

```
-- The template assigning_identification has an external
reference
-- id_assgn. Bind this to the local external reference
-- id_assignment. It can now be the target for assigned dates
%^id_assgn = $assigning_identification.id_assgn%
-- assign date to identification_assignment in the role of
"created"
/assigning_calendar_date(items=^id_assgn,
    date_class_name='Date_actual_start',
    date_ecl_id=p://www.plcsinc.org/plcs-proposed', [triha13]
    year=@creation_year,
    month=@creation_month,
    day=@creation_day)/
```

## ference path: [triha14]

The reference path shown below specifies the entities that are to be instantiated by the template.

A description of templates and the syntax for the reference path is provided in the [Reading Templates](#) help section.

### [Identification\\_assignment](#)

```
-- Mark the Identification_assignment entity as
-- referable when this template is used by binding it to the reference
-- parameter id_assgn
%^id_assgn = Identification\_assignment%
Identification\_assignment.identifier = @id
Identification\_assignment.role = '/IGNORE'
Identification\_assignment.description = '$'
Identification\_assignment.items -> @items

-- provide the role of the identification by classifying the
Identification_assignment
/assigning_reference_data(
    items=^id_assgn,
    class_name=@id_class_name,
    ecl_id=@id_ecl_id)/

-- ign an organization to the identifier and classify it as 'Owner_of' [triha15]
/assigning_organization(
    items=^id_assgn,
    org_id=@org_id,
    org_id_class_name=@org_id_class_name,
    org_id_ecl_id=@org_id_ecl_id,
    org_assgn_class_name=ner_of' [triha16],
    org_assgn_ecl_id='http://www.plcsinc.org/plcs-proposed')/
```

The following entities are instantiated with attributes as specified:

Entity in path	Value	Inherited from
----------------	-------	----------------

<a href="#">Identification_assignment.identifier</a>	@id	—
<a href="#">Identification_assignment.role</a>	'/IGNORE'	—
<a href="#">Identification_assignment.description</a>	'\$'	—

## Instance diagrams

The instance diagram below shows an example of the EXPRESS entities that are instantiated by the template:

```
/assigning_identification(items='#1', id='ZXS10345',
id_class_name='Part_identification_code',
id_ecl_id='http://www.plcsinc.org/plcs-proposed', org_id='Big Planes Inc',
org_id_class_name='Organization_identification_name',
org_id_ecl_id='http://www.plcsinc.org/plcs-proposed')/
```

Note that the [assigning\\_reference\\_data](#) and [assigning\\_organization](#) templates are used in the diagram. Namely:

```
/assigning_reference_data(items='#1', class_name='Part_identification_code',
ecl_id='http://www.plcsinc.org/plcs-proposed')/
/assigning_organization(items='#1', org_id='Big Planes Inc',
org_id_class_name='Organization_identification_name',
org_id_ecl_id='http://www.plcsinc.org/plcs-proposed',
org_assgn_class_name='Owner_of', org_assgn_ecl_id='http://www.plcsinc.org/plcs-
proposed')/
```

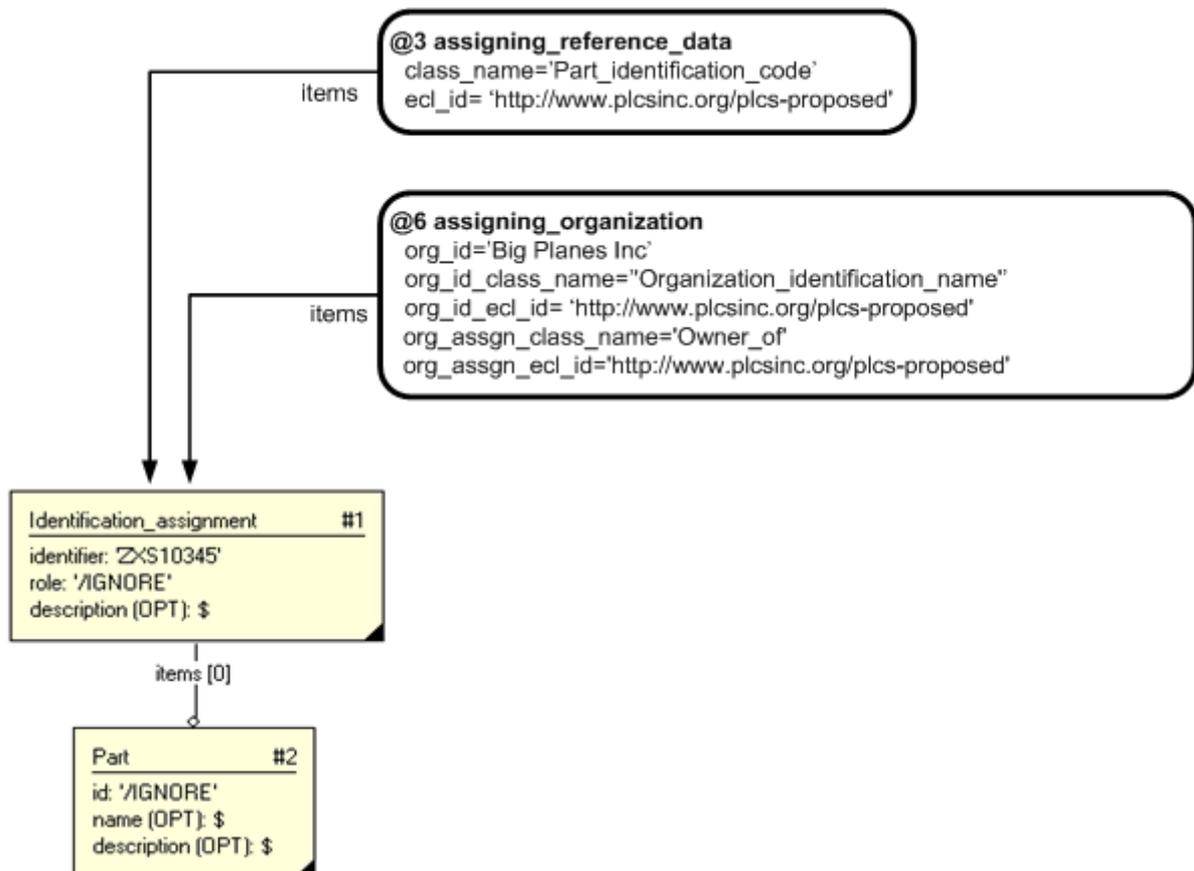
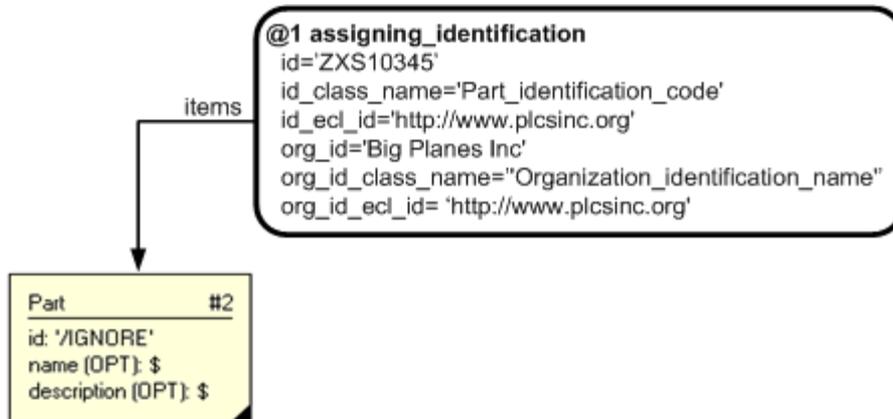


Figure 5.3 — Instantiation of Template Components

The instance diagram below shows the graphic symbol for the template that is to be used in other instance diagrams. The example template is:

```
/assigning_identification(items='#1', id='ZXS10345',
id_class_name='Part_identification_code',
```

```
id_ecl_id='http://www.plcsinc.org/plcs-proposed', org_id='Big Planes Inc',
org_id_class_name='Organization_identification_name',
org_id_ecl_id='http://www.plcsinc.org/plcs-proposed')/
```



[triha17][triha18]Figure 5.4 – Consolidated Template Instantiation

### Characterizations:

The assigning\_identification template can be optionally characterized by assigning other constructs to it. The following characterizations may apply.

#### Characterization: Assigning date

Dates can be associated with the assignment of an identifier by using the template [assigning\\_calendar\\_date](#). For example, the date on which an identifier was created is represented by assigning a date to the [Identification\\_assignment](#).

The model for assigning a date to an identifier is shown below.

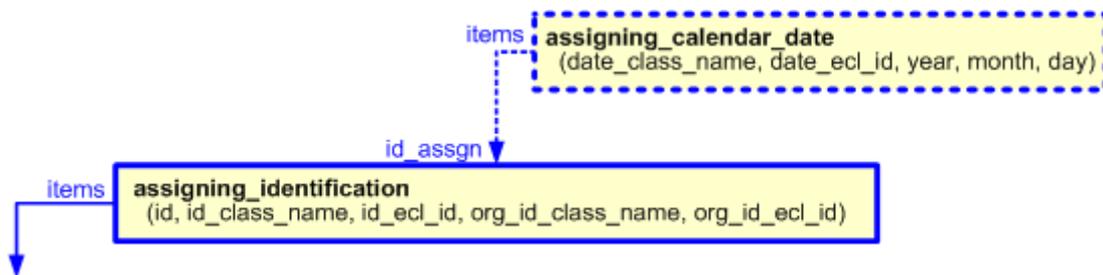


Figure 5.5 – Template Configuration with Optional Calendar Date Assignment

An example showing the assignment a start date to an identifier is shown below.

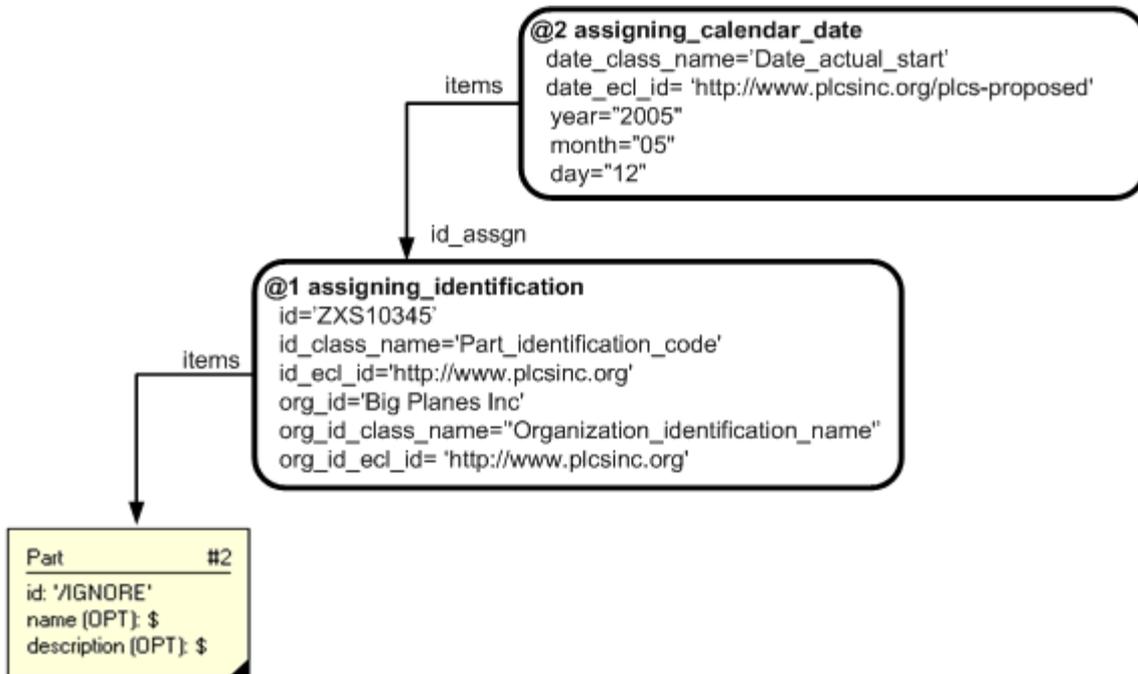


Figure 5.6 — Instantiation of Optional Template Configuration

The instantiated templates for assigning a identification to a part is:

```

/assigning_identification(items='#1', id='ZXS10345',
id_class_name='Part_identification_code',
id_ecl_id='http://www.plcsinc.org/plcs-proposed', org_id='Big Planes Inc',
org_id_class_name='Organization_identification_name',
org_id_ecl_id='http://www.plcsinc.org/plcs-proposed')/
  
```

The instantiated templates for assigning a date to an identification is shown below. Note the use of the reference parameter `$assigning_identification.id_assgn` to identify the [Identification\\_assignment](#) instantiated by the template `assigning_identification`.

```

/assigning_calendar_date(items='$assigning_identification.id_assgn',
date_class_name='Date_actual_start', date_ecl_id='http://www.plcsinc.org/plcs-
proposed', year='2005', month='05', day='12')/
  
```