

# Naming Tool

## developer documentation

### Purpose

Developer documentation, not user documentation. Help understand data, representation of data, flow of data. Visualisation.

### Table of content

- About
  - Document & Tool
- Overview
- Database & application
- Hierarchy of objects
  - ESS name and System/Device structure & examples
  - ESS name and System/Device structure & storage/objects
- Revisions
- Various aspects
  - ESS Naming Convention
    - Equivalence class representative
  - Notes
  - List of caches (and how used, UI/REST API)
  - Model and objects
  - Example(s) of database, content, objects
- UI
- REST API
- Integrations
- Reference

---

## About

The ESS Naming convention applies to devices and signals controlled and monitored by the Integrated Controls System (ICS).

The ESS Naming Convention names shall be used on operator screens, drawings, design schematics, computer software, project databases, equipment name tags, test procedures, and other sources of technical information at ESS.

## Document & Tool

Naming Tool implements Naming Convention.

Naming Convention considers System structure, Device structure, Property.  
Naming Tool considers System structure, Device structure but not Property.

This document concerns Naming Tool but may mention Property.

# Overview

## System structure

Which part of the facility does the device provide service to?

Parts

System Group	(Level 1)	(Sup)
System	(Level 2)	(Sys)
Subsystem	(Level 3)	(Sub)

## Device structure

What kind of device is it?

Parts

Discipline	(Level 1)	(Dis)
Device Group		
Device Type	(Level 3)	(Dev)

## ESS name

Combination of System structure and Device structure.

Instance Index (Level 4) (Idx)

## Property

Observable property of device, e.g. current, time, temperature.

## Examples of ESS names

Before

Sup-Sys-Sub:Dis-Dev-Idx (OFFSITE)  
Sys-Sub:Dis-Dev-Idx

Now

~~Sup-Sys-Sub:Dis-Dev-Idx~~ (OFFSITE not allowed)  
Sup (Accelerator as System)  
Sup:Dis-Dev-Idx (Accelerator as System)  
Sys  
Sys:Dis-Dev-Idx  
Sys-Sub  
Sys-Sub:Dis-Dev-Idx

*Note!*

*System structure need to be part of ESS name  
Device structure need not be part of ESS name  
Property not part of ESS name*

*System Group may be part of ESS name.  
Device Group not part of ESS name.*

*Changes (before, now) in releases 5.0.0, 5.1.0*

# Database & application

Application provides views of data, from different perspectives.

## Perspectives

- tabs in application
  - ESS Name Registry
  - System structure
  - Device structure
- REST API

At startup, database content is read into application (caches). During use, caches may be reinitiated.

Caches are used to build internal representations of data that are used in the different views of data, tabs in application and REST API.

Data for System structure and Device structure are stored in tables

- namepart
- namepartrevision

Data for ESS names are stored in tables

- device
- devicerevision

## Tables

- namepart
- device

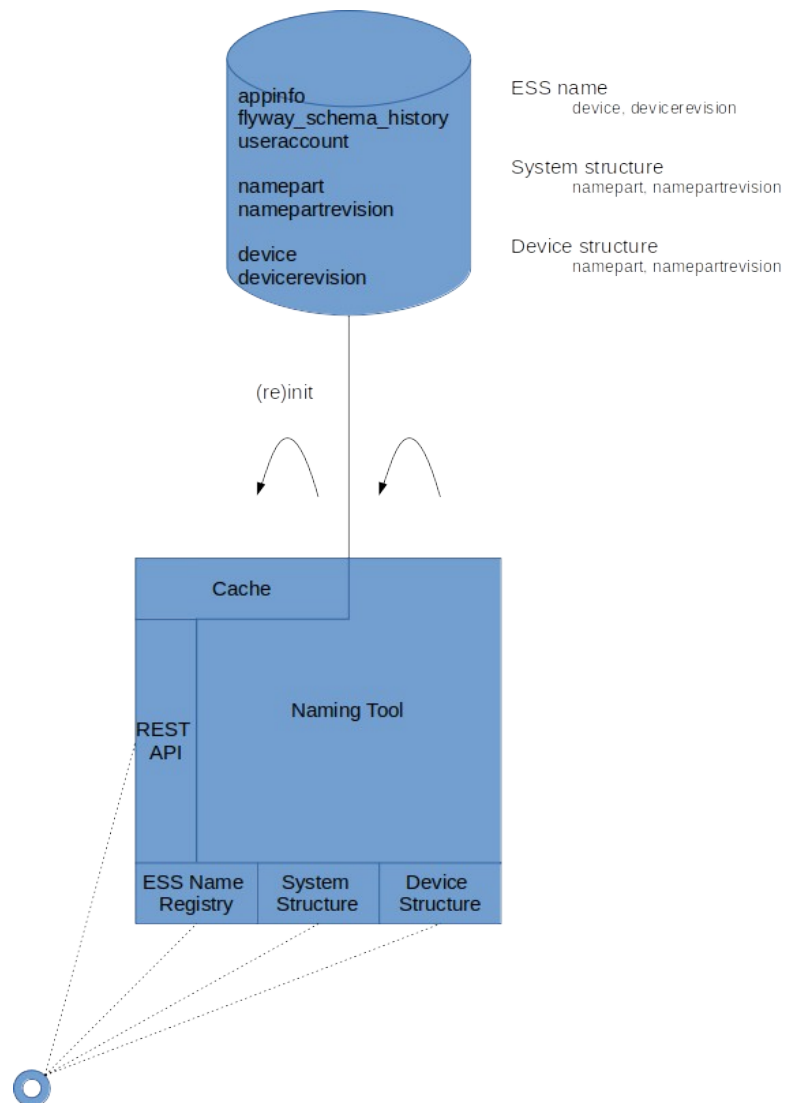
represent data that is not changed, identity (uuid) and type of structure (system, device).

## Tables

- namepartrevision
- devicerevision

represent other data, that may change, e.g. names, mnemonics, dates, etc.

Data that may change are changed through new rows/entries in database, i.e. new revisions. Thus, history is kept and may be traced.



### Note!

*namepart, device – identity - fixed*

*namepartrevision, devicerevision – other information - variable*

*hierarchy and history through recursive references in tables, foreign keys*

*data that may change are changed through new rows/entries in database, i.e. new revisions*

# Hierarchy of objects

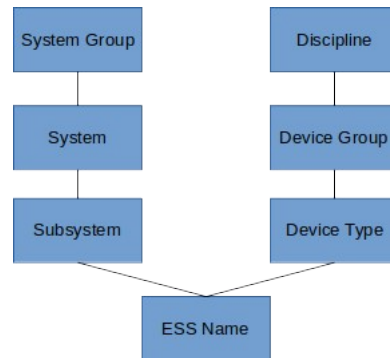
## ESS name and System/Device structure & examples

Before

Sup-Sys-Sub:Dis-Dev-Idx  
Sys-Sub:Dis-Dev-Idx

e.g.

A2T-010PRL:RFS-PrlTap-054

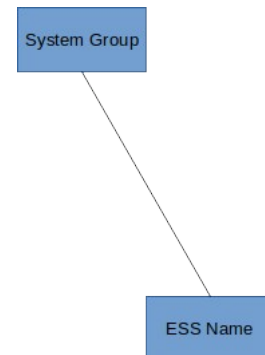


Now

Sup

e.g.

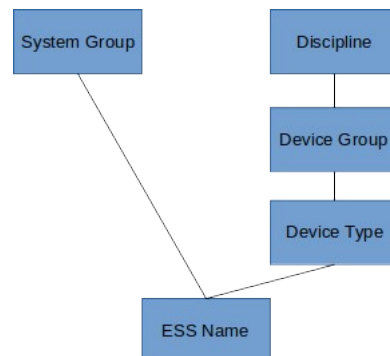
Acc



Sup:Dis-Dev-Idx

e.g.

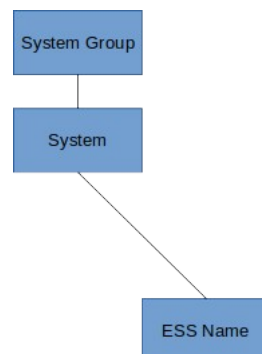
Acc:RFS-PrlTap-054



Sys

e.g.

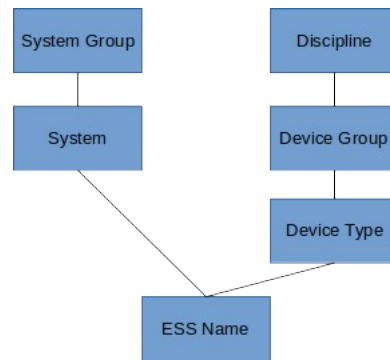
A2T



Sys:Dis-Dev-Idx

e.g.

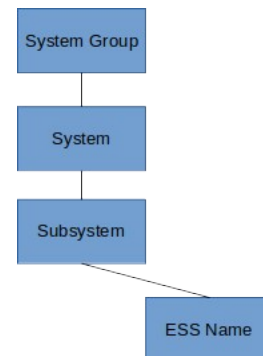
A2T:RFS-PrlTap-054



Sys-Sub

e.g.

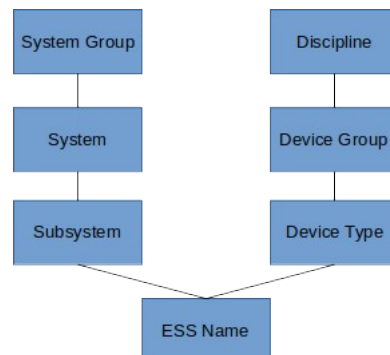
A2T-010PRL



Sys-Sub:Dis-Dev-Idx

e.g.

A2T-010PRL:RFS-PrlTap-054



*Note!*

*System Group mnemonic optional*

*one root node for System structure – not visible – level 0*

*one root node for Device structure – not visible – level 0*

*ESS name*

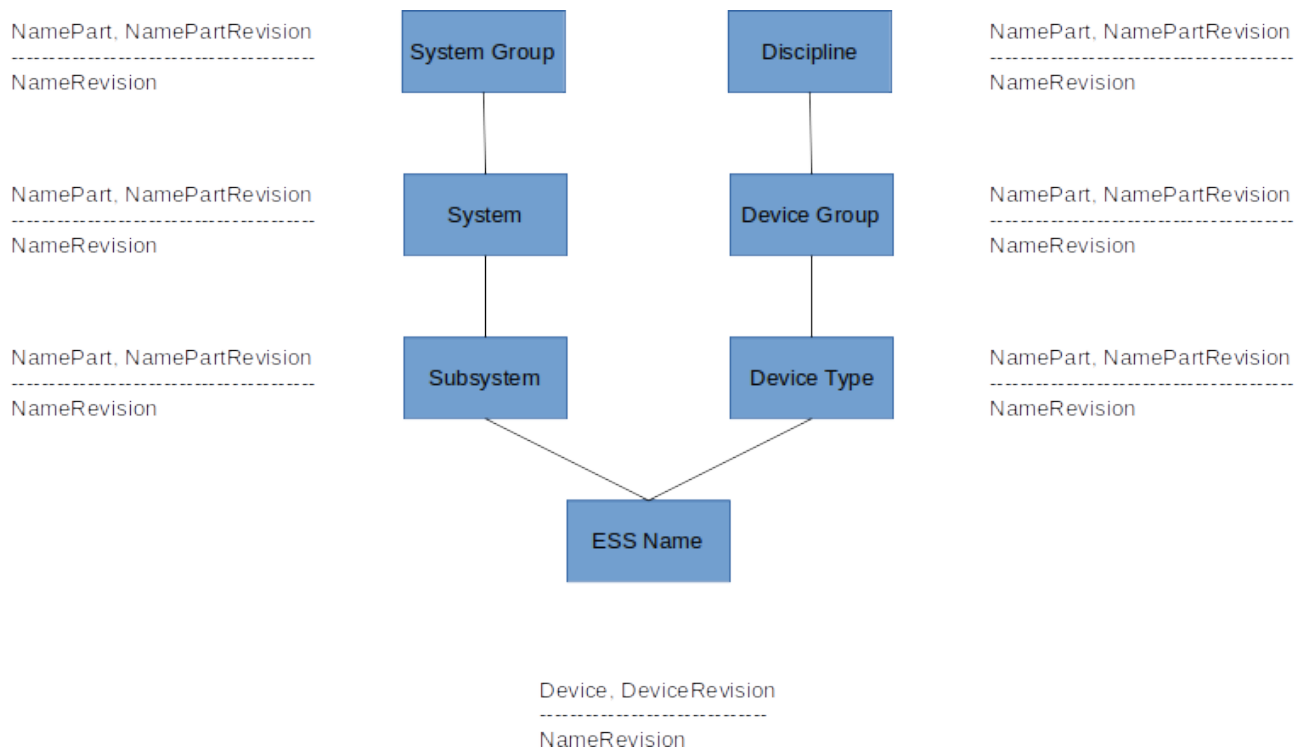
*level 4*

*must have System structure parent*

*may have Device structure parent*

*parents – level 1, 2, 3*

## ESS name and System/Device structure & storage/objects



Same object, NameRevision, is used to represent

- parents/children
  - system structure
  - device structure
- name registry

i.e. different levels in hierarchy.

Hierarchy/structure is important. Data is traversed and structures (caches) built at startup and if requested.

NameRevision

*database entry*

- parent data – *id, uuid, type* – system structure
- parent data – *id, uuid, type* – device structure
- data
  - ...
  - full name
  - mnemonic
  - description
  - status
    - APPROVED
    - CANCELLED
    - PENDING
    - REJECTED
  - ...

Identity of data is kept, when set . Other data may change but entries/rows in database are not changed. Instead, changes are handled through new entries/rows in database. The different entries, along with fixed data (identity), are called revisions. NameRevision object represents one revision. Important is “where we are” and “where we are going”, i.e. current and next revision, which is called a revision pair and is represented by a revision pair object.

## NameRevisionPair

*database entries, stage – where we are and where we are going*

- approved revision
- unapproved revision
- stage

Object contains the latest approved revision as well as the latest unapproved revision (which can be either pending approval or cancelled). Stage (phase) depend on approved and unapproved revisions. It can be thought of as “where we are and where we’re going”. History (“where we come from”) important for trace purposes but less so for current stage.

Each of the 7 boxes can be seen as NameView which is object with namerevision pair that may have type, level, parents, children.

## NameView

*data object with relations*

- parent object – *nameview* – System structure
- parent object – *nameview* – Device structure
- current nameview
  - namerevision pair
  - type
  - level
  - children

### *Note!*

*data may be non-existent in which case references have null value*

*hierarchy/structure important*

*data is traversed and structures (caches) built at startup*

# Revisions

An important aspect is how to consider

- if revision same as other revision
  - id = other id
  - status = other status
- if revision supersedes other revision
  - either
    - other null
    - id > other id
    - id = other id and (status not pending and other status pending)
- update namerevision pair
  - check namerevision
    - if supersedes approved revision
    - if supersedes unapproved revision
  - handle namestage

*Note!*

*since caches of system structure, device structure and ESS names used,  
revision handling important to consider current information  
“where we are” and “where we are going”*

## Various aspects

### ESS Naming Convention

Key concepts

- mnemonic required or not
- mnemonic valid or not
- if mnemonic can coexist with other mnemonics
- equivalence, uniqueness of names when treating similar looking names

Additional concepts

- mnemonic path is a list of mnemonics starting from the root of the hierarchy to the intended mnemonic
- root of hierarchy is node for System structure, Device structure, ESS Name Registry
- type is type of structure, i.e. System structure, Device structure, ESS Name Registry



## Equivalence class representative

A mnemonic or name may be considered with equivalence class representative. It is used to ensure uniqueness of similar looking names.

Among considered

- upper case / lower case
- 0 / O
- 1 / l / L / I / I
- v / V / w / W

## Notes

- read database at startup, traverse content (cache)
- build up hierarchies
  - system structure (parent-child)(cache)
  - device structure (parent-child)(cache)
- find ESS names
  - find out equivalence class representative
    - build structure of key/value (cache)
      - key - equivalence class representative
      - value – namerevision
  - add to system structure (cache)
- (cache) may be same as other (cache)
  - used in multiple contexts

## List of caches (and how used, UI/REST API)

See – Various aspects # Notes.

### Code

- NameViewProvider
  - NameRevisions
    - nameRevisionMap
- NameViews
  - nameViewMap

Both caches are initiated through NameViewProvider.

Name revision map used for REST API and to some extent also when adding new entries (batch add).

Name view map used to build system structure and device structure which then are used in UI. In addition, name revisions for ESS names added to system structure, which then are used in UI.

Since key is different in maps, content may differ and care to be taken in usage.

## Cache

Map<String, NameRevision> nameRevisionMap

Key – equivalence class representative of convention name

Value – name revision – database entry,  
for system structure, device structure or ESS name

## Cache

Map<UUID, NameView> nameViewMap

Key – uuid for system structure, device structure or ESS name

Value – name view – data object with relations,  
for system structure, device structure or ESS name

### Note!

*caches (re)initiated when either of system/device structure or ESS names changes*

***nameViewMap not only a map but has parent-child relations for  
System structure, Device structure and ESS names***

## Model and objects

In order of close to database

- org.openepics.names.model
- org.openepics.names.business
- org.openepics.names.nameviews

## Example(s) of database, content, objects

ESS name consists of System structure, Device structure and Device.

### Example

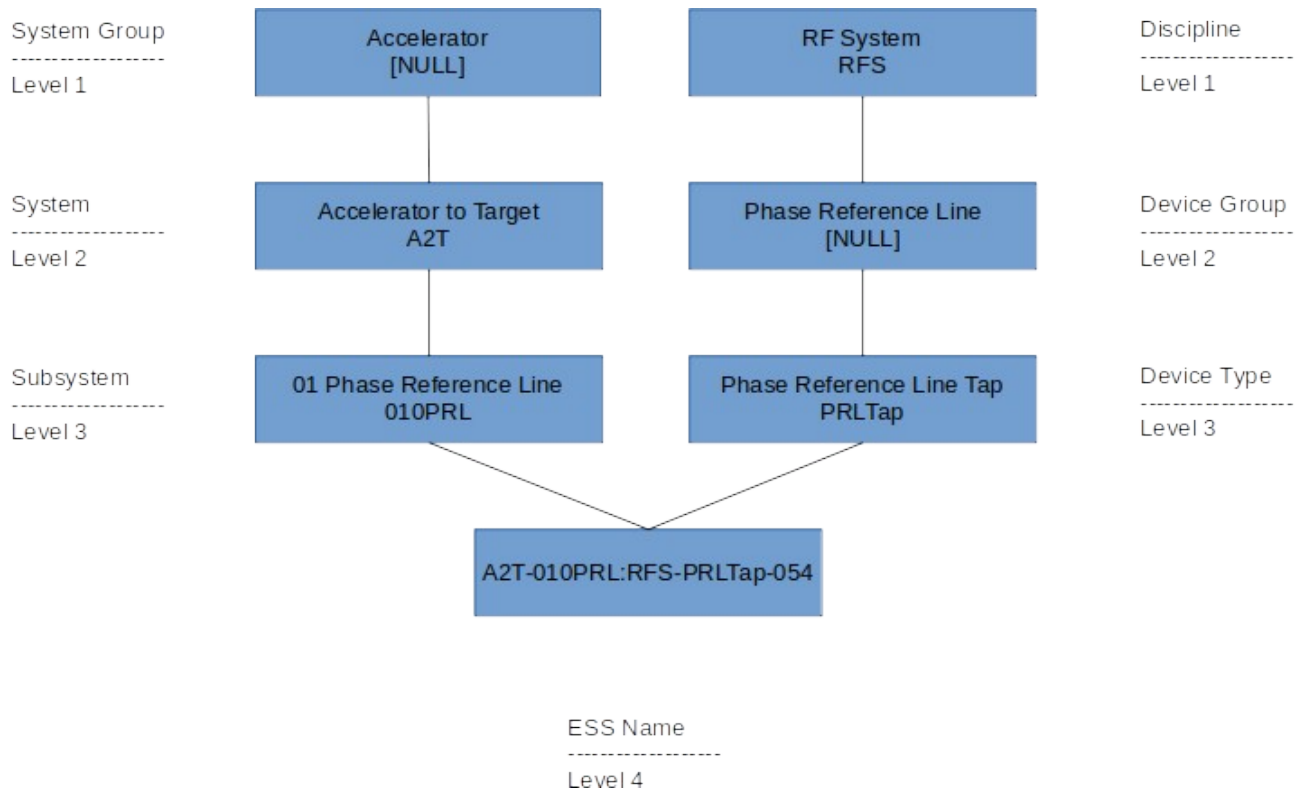
- A2T-010PRL:RFS-PrlTap-054

### In example

- System structure           A2T-010PRL
- Device structure           RFS-PrlTap
- ESS name                    054

### Looking in UI

- System Structure           0
  - System Group           1   Accelerator
  - System                2   Accelerator to Target           A2T
  - Subsystem            3   01 Phase Reference Line        010PRL
- Device Structure           0
  - Discipline            1   RF Systems                    RFS
  - Device Group        2   Phase Reference Line
  - Device Type         3   Phase Reference Line Tap       PRLTap
- Device                    4   A2T-010PRL:RFS-PrlTap-054   054



Looking in database – tables, subset of columns

### System structure

namepart, namepartrevision

id	id	version	deleted	mnemonic	name	status	namepart_id	parent_id
1	1	4	false	Acc	Accelerator	APPROVED	1	[NULL]
1	1050	1	false	[NULL]	Accelerator	APPROVED	1	[NULL]
16	16	2	false	A2T	Accelerator to Target	APPROVED	16	1
16	1708	1	false	A2T	Accelerator to Target	APPROVED	16	1
1359	2292	3	false	010PRL	01 Phase Reference Line	APPROVED	1359	16

### Device structure

namepart, namepartrevision

id	id	version	deleted	mnemonic	name	status	namepart_id	parent_id
242	242	2	false	RFS	RF Systems	APPROVED	242	[NULL]
1349	2282	1	false	[NULL]	Phase Reference Line	APPROVED	1349	242
1350	2692	3	false	PRLTap	Phase Reference Line Tap	APPROVED	1350	1349

### ESS name

device, devicerevision

id	id	conventionname	conventionname eqclass	deleted	instanceindex	device_id	devicetype_id	section_id
3957	32540	A2T-010PRL:RFS-PRLTap-054	A2T-10PR1:RFS-PR1TAP-54	false	054	3957	1350	1359

*Note!*

*namepart, device – identity - fixed*

*namepartrevision, devicerevision – other information - variable*

*hierarchy and history through recursive references in tables, foreign keys*

## UI

UI is divided into tabs – Home, ESS Name Registry, System Structure, Device Structure, Help.

Home, Help contain text, read-only information.

ESS Name Registry, System Structure, Device Structure contain information for read and write operations.

Menu options correspond to operations. ESS Name Registry handled by DeviceTableController. System Structure and Device structure handled by NamePartsController.

Menu option rendered – controller method rendered – operation method validateUser.

Menu option disabled – controller method can – operation method validateOnSelect.

### ESS Name Registry

#### Operations

Add

Add ESS name.

Selection of System structure required. Selection of Device structure optional. Instance index, Description to be entered if Device structure selected.

Modify

Modify ESS name.

Delete

Delete ESS names.

View History

View history for ESS name.

Batch Add

Import new ESS names from an Excel spreadsheet.

Batch Modify

Modify ESS names from an Excel spreadsheet.

Export

Export ESS names to an Excel spreadsheet.

Export of content in table in UI, according to current filter.

---

Text in parentheses correspond to operation in code.

---

Add

(AddDevice)

Single name operation. User.

Modify

(ModifyDevice)

Single name operation. User.

Delete

(DeleteDevices)

Multiple name operation. User.

View History

Single name.

Batch Add

(AddDevice)

Row – Single name operation. User.

Batch Modify

(ModifyDevice)

Row – Single name operation. User.

Export

Multiple names.

## System Structure / Device Structure

### Operations

Add	Propose to add a name.
Modify	Propose to modify a name.
Delete	Propose to delete names.
Cancel	Cancel requests.
Approve	Approve requests.
Reject	Reject requests.
Check Devices	Check and fix ESS names.
View History	View history for ESS name. User.

---

Text in parentheses correspond to operation in code.

---

Add	(Add)	Single name operation. User.
Modify	(Modify)	Single name operation. User.
Delete	(Delete)	Multiple name operation. User.
Cancel	(Cancel)	Multiple name operation. User.
Approve	(Approve)	Multiple name operation. Super user.
Reject	(Reject)	Multiple name operation. Super user.
Check Devices	(CheckDevices)	Multiple name operation. Super user.
View History		Single name

# REST API

REST API provide read-only information without authentication. All endpoints are requested with GET method.

## Purpose of endpoints

healthcheck	provide healthcheck of application
deviceNames	provide information about ESS names
parts	provide information about name parts – System Structure, Device Structure

## Endpoints

healthcheck		
	/healthcheck	Healthcheck of Naming application
deviceNames		
	/deviceNames	Lists all devices
	/deviceNames/devicetype/ search/{devicetype}	Finds all devices with given device type (search)
	/deviceNames/devicetype/ {devicetype}	Finds all devices with given device type (exact match)
	/deviceNames/discipline/ search/{discipline}	Finds all devices with given discipline (search)
	/deviceNames/discipline/ {discipline}	Finds all devices with given discipline (exact match)
	/deviceNames/search/ {searchText}	Finds all devices containing given text (search)
	/deviceNames/subsystem/ search/{subsystem}	Finds all devices by given subsystem (search)
	/deviceNames/subsystem/ {subsystem}	Finds all devices by given subsystem (exact match)
	/deviceNames/system/search/ {system}	Finds all devices by given system (search)
	/deviceNames/system/{system}	Finds all devices by given system (exact match)
	/deviceNames/{uuid}	Finds device by uuid or name
parts		
	/parts/mnemonic/search/ {searchText}	Finds all parts containing mnemonic with the specific search text.
	/parts/mnemonic/{mnemonic}	Finds all parts by specific mnemonic.

Attributes in return elements are set when there is value for attribute. Otherwise, attribute is not present in return element.

## Models

PartElement	
	<pre> PartElement{     type*           string     uuid*           string(\$uuid)                    UUID     level*          string     name*           string     mnemonic*       string     description*    string     status*         string     namePath*       string     mnemonicPath*   string } </pre>
DeviceNameElement	
	<pre> DeviceNameElement{     uuid*           string(\$uuid)                    UUID     systemGroup*    string     system*         string     subsystem*      string     discipline*     string     deviceType*     string     instanceIndex*  string     name*           string     description*    string     status*         string      The status can have three different values:     DELETED, ACTIVE, or OBSOLETE      Enum:     Array [ 3 ] } </pre>

# Integrations

## Known integrations

- CHESS integration to Naming REST API
  - endpoints
    - deviceNames
      - GET /deviceNames  
Finds all devices
      - GET /deviceNames/{uuid}  
Finds device by uuid or name
    - models
      - DeviceNameElement
        - uuid
        - name
        - status
- Awesome Naming Tool (Python script)
  - <https://gitlab.esss.lu.se/thomasfay/awesome-naming-tool>
  - endpoints
    - deviceNames
      - GET /deviceNames/search/{searchText}  
Finds all devices containing the specific search text
      - GET /deviceNames/{uuid}  
Finds device by uuid or name
    - parts
      - GET /parts/mnemonic/search/{searchText}  
Finds all parts containing mnemonic with the specific search text.
      - GET /parts/mnemonic/{mnemonic}  
Finds all parts by specific mnemonic.
  - Models
    - DeviceNameElement
    - PartElement
- Cable
  - Naming REST API
- CCDB
  - Naming REST API
- Other
  - REST API usage visible through Graylog

### *Note!*

*see Graylog, remoteAddress attribute, to view remote ip address using REST API  
REST API read-only without authentication*



# Reference

## Application

UI <https://naming.esss.lu.se/>  
REST API <https://naming.esss.lu.se/rest/>

## CCDB ecosystem

Naming  
Cable <https://cable.esss.lu.se/>  
<https://cable.esss.lu.se/rest/>  
CCDB <https://ccdb.esss.lu.se/>  
<https://ccdb.esss.lu.se/rest/>  
IOC Factory <https://iocfactory.esss.lu.se/>  
RBAC <https://rbac.esss.lu.se/>

## CHESS

<https://chess.esss.lu.se/>  
Document  
ESS-0000757  
<https://chess.esss.lu.se/enovia/link/ESS-0000757/21308.51166.18944.63533/valid>

## Confluence

<https://confluence.esss.lu.se/display/NC/ESS+Naming+Convention>  
ICS Software Catalog  
<https://confluence.esss.lu.se/display/ID/ICS+Software+Catalog>

## Gitlab

<https://gitlab.esss.lu.se/ics-software/naming-convention-tool>

## Grafana

<https://telegraph.tn.esss.lu.se>  
Docker and system monitoring  
Containers  
naming-01.cslab.esss.lu.se  
icsvs-app01.esss.lu.se  
icsvp-app01.esss.lu.se

## Graylog

<https://graylog-tn.tn.esss.lu.se/>  
Streams  
CCECO  
source: icsvs-app01.esss.lu.se AND application: naming-convention-tool  
source: icsvp-app01.esss.lu.se AND application: naming-convention-tool  
Dashboard  
CCECO

## Jira

### Projects

Naming System NT

<https://jira.esss.lu.se/projects/NT/issues>

ICS HW and Integration ICSHWI

<https://jira.esss.lu.se/projects/ICSHWI/issues>

INFRA

<https://jira.esss.lu.se/projects/INFRA/issues>

### Ticket(s)

<https://jira.esss.lu.se/browse/NT-278>

ESS Naming Convention – Update

<https://jira.esss.lu.se/browse/NT-279>

ESS Accelerator as System