

# **LBNF and DUNE Requirements Management Plan**

**19 September 2018**

**[DUNE-doc-7626]**

**Long-Baseline Neutrino Facility (LBNF) and  
Deep Underground Neutrino Experiment (DUNE)**

**Authors:**

**Nandhini Dhanaraj and Anne Heavey**

## Approvals and Version Control

This version of the document may not be the current or approved revision. The current revision is maintained in LBNF/DUNE's document Management system (DocDB) where all internal Project document approvals are managed. DocDB can be accessed through the web by authorized users (<http://docs.dunescience.org>) and this document can be identified by the document and version number as indicated in the Version Control Table below. Note that the version number in the table below and DocDB may not match. The current approved version is always available in DocDB.

Approvals for this document will be required from:

Responsible Person:

LBNF Project Manager Elaine McCluskey

DUNE Technical Coordinator Eric James

LBNF/DUNE Systems Engineering Manager Jack Fowler

<b>DocDB</b>	<b>Version</b>	<b>Author</b>	<b>Version Date</b>	<b>Description of Changes</b>
7626	v2	N. Dhanaraj and A. Heavey	19 SEP 2018	Updated responsible person list. Added sections 6 and 8. Updated section 7. Added figure with traceback structure for Near Site.

# Contents

Approvals and Version Control i

Contents ii

List of Figures iii

- 1 Overview ..... 5
- 2 Definitions ..... 5
- 3 Requirements traceback structure..... 6
- 4 Requirements standards ..... 8
  - 4.1 Content .....8
  - 4.2 Terminology .....9
  - 4.3 Examples .....9
- 5 Requirement metadata ..... 10
- 6 Features in DOORS NG for Requirements Management ..... 11
- 7 The requirement lifecycle..... 11
- 8 Requirements Management using DOORS NG ..... 12
  - 8.1 Requirements currently under configuration control:.....12
  - 8.2 Requirements NOT under configuration control:.....12
- 9 Roles..... 13
  - 9.1 LBNF and DUNE systems engineers.....13
  - 9.2 LBNF subproject managers and DUNE consortium leaders.....13
  - 9.3 LBNF and DUNE leadership.....13

## List of Figures

Figure 1: Requirements traceback structure for the Far Site. (Only a representative sample of Far Detector systems is shown. It is possible that further levels underneath the detector systems will exist. The Near Site structure will be done in an analogous manner.) .....	7
Figure 2: High-level Near Site CF requirements traceback structure .....	8



# 1 Overview

LBNF and DUNE are using the IBM DOORS Next Generation (DOORS NG) software package to manage project requirements and their linkages to one another. Fermilab owns two floating licenses to DOORS NG, which we believe to be adequate for the LBNF/DUNE needs.

This application allows import and export of Excel spreadsheets, which gives project members a choice as to how to conduct reviews of and make updates to their sets of requirements: within DOORS NG itself, or through spreadsheets.

In this document we describe what is encompassed by the term “requirement” in the LBNF/DUNE methodology, define related terms, outline the requirement standards and the structure into which they fit, and the process and roles for managing them throughout the project lifecycle.

# 2 Definitions

The following types of objects are defined in the Fermilab DOORS NG configuration. Note that DOORS NG considers virtually everything an “artifact,” including modules (A module is a container for a set of requirements.), and therefore DOORS calls these object types “Artifact Types.”

In alphabetical order, the defined artifact types are:

**Assumption:** A statement of a circumstance or boundary condition outside the control of LBNF/DUNE that the documented objects will nonetheless depend upon.

**Design choice:** A numerical or non-numerical choice for a piece of the design (e.g., a parameter or a material, a technology, a make and model); it must trace back to and be consistent with a specification(s) that satisfies the associated requirement(s).

**Heading:** Heading text to go into a requirements output document for grouping requirements. Headings do not require traceback. (Grouping can also be done via metadata.)

**Objective:** This is used solely for DUNE scientific objectives.

**Parameter:** A value for a requirement, specification or design choice, with units as needed.

**Requirement:** A statement of what the experiment/detector/system/subsystem/component needs to do to enable DUNE to meet its science objectives. Anything tagged as a requirement for a given subsystem or component is to be fulfilled by that same subsystem or component. (E.g., in the *Subsystem A* requirement set, each requirement begins with: “Subsystem A shall ...”) In general, requirements will not contain numerical values; they prescribe or constrain specifications and design choices from which design parameters are chosen.

**Specification:** A value or range of values for a design choice or parameter, with units as needed, that meet the associated requirement(s).

**Use with “Module” format:** This is the artifact type to associate with each module that you create. A module is an artifact that acts as a container for a set of requirements.

Note that in this document, the term “requirement” is often used to generically denote all these artifact types.

### 3 Requirements traceback structure

With the right structure, it is possible to trace requirements (and other artifact types) back to their drivers, and ultimately to the science objectives of DUNE. With the right metadata, it is possible to sort and filter the requirements for a variety of purposes. In DOORS NG, the LBNF/DUNE requirements are organized into a WBS-like hierarchical structure of modules – where a module corresponds to a single spreadsheet, and each module contains all the requirements (and other artifact types listed in Section 2) for a given portion of the project.

The structure defined for the Far Site is shown in Figure 1. Each label on the chart corresponds to one spreadsheet, i.e., to one DOORS NG module.

## Requirements traceback structure, Far Site

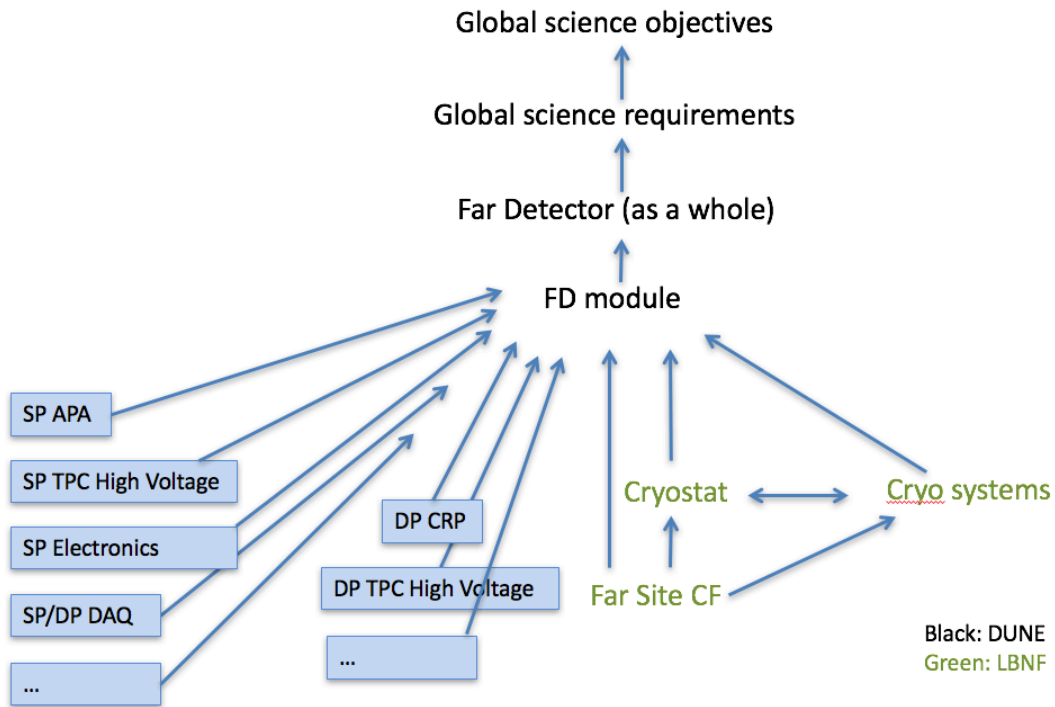


Figure 1: Requirements traceback structure for the Far Site. (Only a representative sample of Far Detector systems is shown. It is possible that further levels underneath the detector systems will exist. The Near Site structure will be done in an analogous manner.)

The structure defined for the Near Site is shown in Figure 2: High-level Near Site CF requirements traceback structure.





- NOT about how a thing is designed, but rather about WHAT function is needed. (test: ask yourself why the requirement is needed; if this leads to another requirement, the current one is probably a specification or design choice);
- specified to the appropriate level, considering function, cost, schedule, etc. Avoid making requirements more stringent than necessary;
- (specifications) appropriately stringent (e.g., don't say “exactly 100 ft” if it can be 100 +/- 3 ft).


## 4.2 Terminology

- Requirements use SHALL; they do NOT use: “is”, “was” and “must”
- Requirements do NOT contain unverifiable words like “maximize,” “user-friendly”, “adequate”, “enough” and so on, without further qualification (e.g., “high enough to allow measurement of xyz” is ok).
- Requirements NEVER contain: “but not limited to”, “etc.”, or “and/or.” This would make a requirement impossible to verify.
- Specifications (and parameters) typically use SHALL since they follow from requirements. However, they can also just be names of quantities with value ranges (or specific values), without verbs at all, if appropriate.
- Objectives use verbs or nouns appropriate to reach goal: “measure xyz,” or “measurement of xyz” and so on.

## 4.3 Examples

- (in the section Cryostat) The cryostat shall provide mechanical support for the TPC.
- (in the section Beam Primary Beam, Primary Water System) The primary water system shall have capacity and flows that are adequate for maintaining magnet temperatures consistent with established Lab standards.

**Artifact Attributes**  [New Attribute...](#)

Name	Data Type		
Alternative Spelling	String		
Artifact Format	ArtifactFormats		
Associated changes	String		
Associated risks	String		
Category	Category		
Class	Class		
Contributor	User		
Created On	DateTime		
Creator	User		
Custodian	String		
Description	String		
Identifier	Integer		
Legacy ID	String		
Legacy Parent ID	String		
Modified On	DateTime		
Note	String		
Primary Text	String		
Priority	Priority		
Rationale	String		
Reference	String		
Risk	Risk		
Status	Status		
Title	String		
Verification method	Verification method		
Verification Owner	String		

## 5 Requirement metadata

In DOORS NG, the requirement metadata is contained in fields called “Artifact Attributes,” listed at left.

Some attributes are system-maintained (Alternative Spelling, Artifact Format, Created On, Creator, Identifier, Description, Modified On, Title).

Attributes with restricted values have a choice menu (Category, Class, Priority, Risk, Status, Verification method).

Notes about other fields:

Attribute type (not an attribute) “requirement” requires a traceback to a parent (which may initially be identified using “Legacy Parent ID”). For other types, either Rationale or Parent is required.

Custodian is the person responsible for ensuring that the requirement is met.

Reference is for a document reference; e.g., docdb number. (Hyperlinks are not supported in attribute fields besides *Primary Text*)

## 6 Features in DOORS NG for Requirements Management

The requirements management tool DOORS NG has several features and functionalities that can be used to effectively manage requirements. The following are some of them.

**Baseline:** The 'Baseline' feature allows the user to capture a snapshot in time of the requirements. Baselines can be created periodically and compared one against another to check the changes between two configurations.

**Locks:** This feature allows users to manually lock each requirement thus preventing any uncontrolled changes.

**Review:** Requirements can be reviewed and approved by reviewers/approvers through the review process. This feature allows to select individual requirements and assign it to reviewers and approvers. DOORS NG can be configured to send automated emails to the reviewers and approvers to solicit their approvals. The progress of the review process can be tracked in DOORS.

**Status:** This feature allows the assignment of different status to the requirements that educates the user regarding the disposition of the requirement. The 'status' can be set to any of the List of Values (LOVs), such as 'Draft', 'Approved', 'Deprecated', 'Config controlled', 'Verified-pass', 'Verified-fail'.

## 7 The requirement lifecycle

This section describes the lifecycle of requirements from their genesis to their realization in the project. The lifecycle begins with the definition/creation of requirements, establishing the parent-child links, reviewing and approving the requirements and linkages, implementing the requirements, and finally verifying them. Once the requirement is verified according to an approved method, the requirement lifecycle is considered to be complete.

The requirements and their metadata are drafted/defined by Subject Matter Experts (SMEs) for a given subproject (typically the subproject manager), and assigned a status of "Draft." The parent requirements are then identified. If the requirements are drafted in a spreadsheet rather than in the DOORS NG application, at this stage they are uploaded to DOORS NG and the traceback links to the parent requirements are established.

Requirement modules can be baselined at a given point in time, typically after they are approved, providing a snapshot against which future changes can be compared. Any design change occurring after the establishment of an official baseline will be routed through a change control procedure (documented in DocDB 82, The LBNF/DUNE Configuration Management Plan).

If a requirement does not pass the verification step, then it will be reviewed by the LBNF and DUNE Leadership and the necessary actions will take place, which is beyond the scope of this document.

## **8 Requirements Management using DOORS NG**

This section describes the process by which requirements will be managed using DOORS NG. The process is slightly different for requirements depending upon whether they are currently under configuration control or not.

### **8.1 Requirements currently under configuration control:**

For requirements that are already under configuration control, the ‘status’ of the requirement will be set to ‘Config controlled’ in DOORS NG and they will also be locked for any edits, using the ‘lock’ feature in DOORS.

Any subsequent changes/addition to the requirements will be managed through the LBNF/DUNE change control process using the BCR (Baseline Change Request) tool. This is accomplished by initiating a change request which then undergoes an appropriate level of review, described in the “The LBNF/DUNE Configuration Management Plan”, in DUNE Docdb-82. Once the change is approved through this process, a comment will be entered in DOORS NG by the Systems Engineering team, stating and citing the change.

### **8.2 Requirements NOT under configuration control:**

For requirements that are NOT under configuration control, the ‘status’ of the requirement will be set to ‘draft’ or ‘approved’ depending on the sub-project manager’s direction. Any changes to the requirements will be managed through the ‘review’ feature available in DOORS NG. This feature allows to pick individual requirements or artifacts and include them in the ‘review’ process. Once the review process is complete the ‘status’ of the requirement is assigned appropriately and a comment entered stating the change.

## **9 Roles**

### **9.1 LBNF and DUNE systems engineers**

- Elicit requirements from LBNF subproject managers and DUNE consortium leaders;
- Define requirements lifecycle;
- Ensure that requirements follow lifecycle;
- (With supporting team) Manage requirements in DOORS NG;
- (With supporting team) Produce and provide requirements reports to project members as requested.

### **9.2 LBNF subproject managers and DUNE consortium leaders**

- Draft requirements with appropriate metadata;
- Identify parent requirements;
- Review requirements according to lifecycle stages and update content and status as needed;
- Appoint custodians and verification owners for each requirement;
- Ensure that verification takes place.

### **9.3 LBNF and DUNE leadership**

- Review requirements
- Approve verification