

ewcommand

Trick High Level Architecture MODEL Product Requirements

Simulation and Graphics Branch (ER7)
Software, Robotics and Simulation Division
Engineering Directorate

Package Release TrickHLA v2.9.0

Document Revision 1.0

DATE



National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas

**Trick High Level Architecture
MODEL
Product Requirements**

**Document Revision 1.0
DATE**

YOUR NAME

**Simulation and Graphics Branch (ER7)
Software, Robotics and Simulation Division
Engineering Directorate**

**National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas**

Abstract

This is the abstract of the MODEL.

Contents

Chapter 1

Introduction

MODEL introduction.

1.1 Identification of Document

This document describes the as-written requirements on the MODEL developed for use in the Trick Simulation Environment. This document adheres to the documentation standards defined in NASA Software Engineering Requirements Standard [?].

1.2 Scope of Document

This document provides information on the requirements for the MODEL.

1.3 Purpose and Objectives of Document

The TrickHLA package, which includes the MODEL, is classified as a Class C (Mission Support Software) product per NASA NPR 7150.2[?] to enable the use of the package in a critical application such as a dynamic simulation used to certify the flight readiness of a space vehicle.

The purpose of this document is to define the set of requirements that the MODEL must achieve to enable its use in Class C products.

1.4 Documentation Status and Schedule

The information in this document is current with the TrickHLA v2.9.0 implementation of the MODEL. Updates will be kept current with module changes.

Author	Date	Description
YOUR NAME	DATE	Initial Version

Revised by	Date	Description

1.5 Document Contents

This document is organized into the following sections:

Chapter ??: Introduction - Identifies this document, defines the scope and purpose, present status, and provides a description of each major section.

Chapter ??: Related Documentation - Lists the related documentation that is applicable to this project.

Chapter ??: Requirements - Presents the requirements for the MODEL.

Bibliography - Informational references associated with this document.

Chapter 2

Related Documentation

2.1 Parent Documents

The following documents are parent to this document:

- [Trick High Level Architecture \(\)](#) [?]
- [MODEL](#) [?]

2.2 Applicable Documents

The following documents are referenced herein and are directly applicable to this document:

- [MODEL Product Specification](#) [?]
- [MODEL User Guide](#) [?]
- [MODEL Inspection, Verification, and Validation](#) [?]
- *The Trick User's Guide: Trick 2005.0 Release* [?]
- *Trick Simulation Environment: User Training Materials: Trick 2005.0 Release* [?]
- *Trick Simulation Environment: Version Description: Trick 2005.0 Release* [?]
- *The Trick Design Document: Trick 2005.0 Release* [?]
- *NASA Software Engineering Requirements* [?]

Chapter 3

Requirements

3.1 General Requirements

This section identifies general requirements for the MODEL.

Requirement MODEL_1: Documentation

Requirement:

The documentation for the model shall include

1.1 Software requirements specification.

1.2 Software, interface, and software version descriptions.

1.3 Software test procedures and results.

1.4 User Guide.

Rationale:

The listed items are needed to comply with NASA NPR 7150.2 as a Class C product.

Verification:

Inspection

Requirement MODEL_2: Header File Trick Header

Requirement:

All header files associated with the model shall have an appropriate Trick header. The Trick header for a header file shall include

2.1 Purpose A brief description of the file.

2.2 References A list of applicable references that describe the model.

2.3 Assumptions and limitations A list of the assumptions made in developing the model and any limitations on the use of the model.

2.4 Programmer A list of the developers who created or modified the file.

Rationale:

- The Trick header in a header file indicates that Trick should process the file.
- Properly documenting the TrickHLA package models is a key goal of the TrickHLA verification, validation, and documentation task.
- Maintaining a version history is good programming technique and is mandatory per NPR 7150.2.

Verification:

Inspection

Requirement MODEL_3: Trick Comments for Enumerated Types

Requirement:

Each tag defined in an enumeration type in a model header file *should* have a comment describing the tag that follows the tag declaration.

Rationale:

Short tag names may not suffice in establishing the meaning of the tag.

Verification:

Inspection

Enumerated types that fail to meet this optional requirement shall be noted as such in the model verification document.

Requirement MODEL_4: Trick Comments for Data Structures

Requirement:

Each element of a data structure defined in a model header file shall have a Trick-compliant comment describing the element that follows the element declaration.

Rationale:

The element comment is required by Trick.

Verification:

Inspection

Requirement MODEL_5: Source File Trick Headers

Requirement:

Each externally visible function defined in the source files associated with the model shall have an appropriate Trick header. The Trick header for a function shall include

5.1 Purpose A brief description of the function.

5.2 References A list of applicable references that describe the function.

5.3 Assumptions and limitations A list of the assumptions made in developing the function and any limitations on the use of the function.

5.4 Class The default Trick job classification of the function.

5.5 Library dependency A list of the object files upon which the function depends, starting with the current file.

5.6 Programmer A list of the developers who created or modified the file.

Rationale:

- The Trick header that precedes a function indicates that the function is available for use in a simulation *S_define* file.
- Properly documenting the TrickHLA package models is a key goal of the TrickHLA verification, validation, and documentation task.
- Maintaining a version history is good programming technique and is mandatory per NPR 7150.2.

Verification:

Inspection

Requirement MODEL_6: Trick Comments for Function Definitions

Requirement:

Each function shall be commented with a Trick-compliant set of comments that describe the return value from the function and that describe the nature of the arguments passed to the function.

Rationale:

The function definition comments are required by Trick.

Verification:

Inspection

ADD ANY OTHER GENERAL REQUIREMENTS THAT APPLY TO THE MODEL AS A WHOLE.

3.2 Data Requirements

This section identifies requirements on the data represented by the MODEL. These as-built requirements are based on the MODEL data definition header files.

EXAMPLE, universal time, not complete:

Requirement MODEL_7: Time Representation

Requirement:

The universal time model shall represent time in the each of the following time systems:

7.1 TAI

International Atomic Time, a very accurate and stable time scale calculated as a weighted average of the time kept by about 200 cesium atomic clocks in over 50 national laboratories worldwide.

7.2 UT1

Universal Time, a measure of the rotation angle of the Earth as observed astronomically.

7.3 UTC

Coordinated Universal Time, the basis for the worldwide system of civil time.

7.4 and others

THIS IS AN INCOMPLETE EXAMPLE.

Rationale:

The purpose of the universal time module is to represent time in the multiplicity of time scales that are expected to be need by various simulation developers.

Verification:

Inspection, Test

3.3 Functional Requirements

This section identifies requirements on the functional capabilities provided by the MODEL. These as-built requirements are based on the MODEL source files.

EXAMPLE, universal time, not complete:

Requirement MODEL_8: Time Initialization

Requirement:

The universal time model shall be capable of initializing time using any one of the following mechanisms:

8.1 incomplete THIS IS AN INCOMPLETE EXAMPLE.

Rationale:

The purpose of the universal time module is to represent time in the multiplicity of time scales that are expected to be need by various simulation developers.

Verification:

Test

Requirement MODEL_9: Time Update

Requirement:

9.1 The universal time model shall correctly update all represented time values as the simulation clock advances.

9.2 The passage of one second of simulation time shall correspond to the passage of one second of International Atomic Time.

Rationale:

The purpose of the universal time module is to represent time in the multiplicity of time scales that are expected to be need by various simulation developers.

Verification:

Test