National Aeronautics and
Space Administration

SLS-VDD-177

BASELINE

EFFECTIVE DATE: APRIL 12, 2013

SPACE LAUNCH SYSTEM PROGRAM FLIGHT SOFTWARE BUILD CONTENT DEFINITION

Approved for Public Release; Distribution is Unlimited

The electronic version is the official approved document.
Verify this is the correct version before use.

Revised: 11/01/2011

REVISION AND HISTORY PAGE

Status	Revision No.	Change No.	Description	Effective Date
Baseline	-		Baseline per PCBD SV2-01-0086, dated 04/12/13; (CR-SLS-00114); PCN: SV00221	04/12/13

NOTE: Updates to this document, as released by numbered changes (Change XXX), are identified by a black bar on the right margin.

TABLE OF CONTENTS

PARAGRAPH **PAGE**

APPENDIX

TABLE

1.0 INTRODUCTION

The National Aeronautics and Space Administration (NASA) has called for the development of a Space Launch System Program (SLSP) launch vehicle. The Space Launch System (SLS) is being developed to provide heavy-lift capability to enable human exploration missions beyond Earth orbit (BEO), which supports NASA's strategic goal of extending and sustaining human activities across the solar system. The SLS will deliver exploration elements safely to desired orbits, providing sufficient lift mass and volume to execute the desired missions. To support these missions, this document defines the approach and content of the Flight Software builds.

1.1 Purpose

This build content definition focuses on the Marshall Space Flight Center (MSFC) in-house developed flight software (FSW) and its planned usage, including integration, verification, and flight. This document defines the identified usages of the SLS Flight Software and the planned build content for each release. The processes for the development of this software are discussed in SLS-PLAN-073, SLSP Flight Software Development Plan.

1.2 Scope

This build content definition applies to the MSFC-developed SLS flight software releases as identified in SLS-SCHE-164, SLS Program Systems Engineering & Integration (SE&I) Control Milestones. The FSW is defined as all the code that executes on the SLS Flight Computers (FC). The three flight computers composing the Flight Computer Operational Group (FCOG) run identical software. As discussed in SLS-PLAN-073, SLSP Flight Software Development Plan, the FSW is composed of the Flight Computer Software (FCSW) Computer Software Configuration Item (CSCI).

The FCSW CSCI provides for five modes of operation, including Standby Mode, Special Test Mode, Ascent Simulation Mode, Countdown Demonstration Mode, and Flight Operations Mode. SLS-RQMT-095, SLSP Flight Software Requirements Specification (SRS), Section 1.4, defines these modes.

1.3 Change Authority/Responsibility

The NASA Office of Primary Responsibility (OPR) for this document is ES51.

Proposed changes to this document will be submitted by an SLS Program Change Request (CR) to the Program Control Board (PCB) for disposition. All such requests will be in accordance with to the SLS-PLAN-008, SLS Program Configuration Management Plan.

2.0 DOCUMENTS

2.1 Applicable Documents

The following documents include specifications, models, standards, guidelines, handbooks, and other special publications. The documents listed in this paragraph are applicable to the extent specified herein. Versions are not specified as applicability of document versions to specific flight software releases are defined going into the implementation of that release.

SLS-PLAN-008C	Space Launch System Program Configuration Management Plan
SLS-PLAN-074	Space Launch System Program Flight Software Configuration Management Plan (SCMP)

2.2 Reference Documents

The following documents contain supplemental information to guide the user in the application of this document.

NPR 7150.2A	NASA Software Engineering Requirements
SLS-PLAN-020A	Space Launch System Program Concept Of Operations (Con Ops) Document
SLS-SPEC-032C	Space Launch System Program System Specification
SLS-ICD-029C	Space Launch System Program Stages to Integrated Spacecraft Payload Element (ISPE) Interface Control Document (ICD)
SLS-SPEC-043 (Baseline Pending)	Space Launch System Vehicle Operations and Maintenance Requirements
SLS-SCHE-069	Space Launch System Program Integrated Master Schedule (IMS)
SLS-PLAN-073	Space Launch System Program Flight Software Development Plan
SLS-PLAN-130 (Release Pending)	Space Launch System Program Avionics and Integration Test Plan (AITP)
SLS-PLAN-075 (Release Pending)	Space Launch System Program Flight Software Verification and Validation Plan (SVVP)
SLS-SPEC-079	Space Launch System Program Vehicle Management Specification
SLS-RQMT-095	Space Launch System Program Flight Software Requirements

(Baseline Pending)	Specification (SRS)
SLS-ICD-021C	Space Launch System Program Core Stage (CS) To Booster Interface Control Document (ICD)
SLS-ICD-039-01C	Space Launch System Program (SLSP) Stages to Engine(s) Interface Control Document (ICD) Volume I – Core Stage To Core Stage Engine(s)
SLS-ICD-052-05 (Baseline Pending)	SLSP-to-GSDOP ICD Volume 5: SLSP -to-GSDOP Command and Data Handling (C&DH) Detailed Design
SLS-RPT-087-01	SLSP Integrated Mission and Fault Management (M&FM) Design Analysis & Performance Assessment Document Volume 1: Monitored Conditions Report
SLS-RPT-087-02	SLSP Integrated Mission and Fault Management (M&FM) Design Analysis & Performance Assessment Document Volume 2: Goal Tree/Success Tree (GT/ST)
SLS-RPT-087-03	SLSP Integrated Mission and Fault Management (M&FM) Design Analysis & Performance Assessment Document Volume 3: Abort Triggers Package
<TBD-001>	SLS to Multi-Purpose Crew Vehicle (MPCV) ICD
SLS-SCHE-164 A	SLS Program Systems Engineering & Integration (SE&I) Control Milestones
SLS-STD-172 (Baseline Pending)	Space Launch System (SLS) Command and Telemetry Representation and Metadata Exchange Standards (CTRMES)
SLS-ICD-176	SLS to Mission Systems (MS) ICD
STG-AV-0002	SLS Stages to Elements Communication and Data Handling (C&DH) Document

3.0 SOFTWARE BUILD CONTENT APPROACH AND STRATEGY

An iterative, multi-step process is being used to develop and maintain the build content definition for the SLS flight software. The first step is the identification of the critical stakeholders. These stakeholders provide key inputs to the software development effort; they are also in many cases consumers of the flight software. The key inputs are identified as part of an

accepted risk at the SLS Program System Engineering and Integration (SE&I) Level: Risk # 11957 "SLS FSW Dependencies". The critical dependencies identified in this risk are linked and tracked in SLS-SCHE-069, SLSP IMS, as a part of the mitigation plan. These stakeholders are discussed in section 3.1. The second step is identifying the planned usages for the flight software, so that needed (potentially new) capabilities/functionalities can be captured. The known FSW usages are captured in section 3.2. The stakeholders and usage needs are integrated based on priority and planned input maturity to develop and update the build content as defined in section 4.0.

3.1 Stakeholders

Key stakeholders for the SLS flight software are identified in Section 3.1 of SLS-PLAN-073, SLSP Flight Software Development Plan. Each stakeholder provides critical information relevant to the operations of the FSW on SLS and is therefore necessary to the successful development effort. Table 3-1 below identifies expected inputs from key stakeholders.

Table 3-1. FSW Stakeholder Inputs

Stakeholder	Expected Inputs
SE&I	SLS-SPEC-032, SLS Program System Specification
Integrated Avionics and Software (IAS)	Simulation Environment (Artemis/Maestro); SLS-STD-172, SLSP CTRMES; IMACS Reports
Vehicle Management (VM)	SLS-SPEC-079, SLSP Vehicle Management Specification; Mission and Fault Management (M&FM) Algorithms; Guidance, Navigation, and Control (GN&C) Algorithms; SLS-RPT-087, M&FM Design Analysis & Performance Assessment Document Volumes 1, 2, and 3
Stages Element/Boeing	STG-AV-0002, SLS Stages to Elements Communication and Data Handling (C&DH) Document; Avionics functionality through M&FM
Booster Element	SLS-ICD-021, Space Launch System Program Core Stage (CS) To Booster Interface Control Document (ICD)
Liquid Engines	SLS-ICD-039-01, Space Launch System Program (SLSP) Stages to Engine(s) Interface Control Document (ICD) Volume I – Core Stage To Core Stage Engine(s); Functionality
Operations (OPS)	SLS-PLAN-020, SLS Concept of Operations; Integrated Mission Timeline through M&FM; SLS-SPEC-043, Vehicle Operations and Maintenance Requirements (VOMRS)
Ground System Development and Operations Program	SLS-ICD-052-05, SLSP-to-GSDOP ICD Volume 5: SLSP -to-GSDOP Command and Data Handling (C&DH) Detailed Design

(GSDOP)	
Multi-Purpose Crew Vehicle (MPCV)	<**TBD-001**>, SLSP-to-MPCV ICD
Mission Systems (MS)	SLS-ICD-176, SLSP-to-MS ICD
Spacecraft & Payloads Integration Office (SPIO)	SLS-ICD-029, SLSP Stages to Integrated Spacecraft Payload Element (ISPE) ICD

3.2 Planned Flight Software Usage

The following are currently planned uses for the SLS flight software releases.

3.2.1 Verification and Validation of the SLS Flight Software

Verification and validation (V&V) of the MSFC-developed SLS flight software releases occurs in Software Development Facility (SDF) Line 3 using the simulation environment (A Real-time Environment for Modeling, Integration, and Simulation (ARTEMIS) / Managed Automation Environment for Simulation Test and Real-time Operation (MAESTRO)) and is the responsibility of the MSFC FSW development organization. V&V of all modes of the flight software is performed prior to all releases of the flight software to organizations outside the development organization.

3.2.2 System Integration Test Facility (SITF) Avionics Hardware Integration

Core Stage is responsible for performing avionics hardware integration in the SITF. An identified use of Special Test Mode of the flight software is to support avionics hardware integration. During avionics hardware integration, the avionics components are brought into the SITF and integrated one by one to check out the interfaces, connectivity and functionality between them. The NASA provided flight software is to be used to support this activity in the Special Test Mode starting with Release 7.X. The software requirements for this mode are captured in Volume II of SLS-RQMT-095, SLSP Flight Software SRS, which has been coordinated with Stages/Boeing.

3.2.3 Integrated Vehicle Avionics Testing

V&V of the integrated vehicle avionics and software is to be performed by the Flight Systems Integration and Test Branch/ES61. Refer to SLS-PLAN-130, SLSP Avionics Integration and Test Plan for details on this test activity. Release 10 and 11 of the NASA provided flight software are used to support integration activities while Release 12 and beyond are used to support this V&V activity in all modes.

3.2.4 Michoud Assembly Facility (MAF) Checkout

MAF Checkout is the responsibility of Core Stage/Boeing. During MAF checkout, the Core Stage is integrated with all the avionics and other hardware. The MAF Checkout software is used to check out the interfaces, basic functionality, and connectivity of the avionics components

of the Core Stage. The MAF checkout software is to be developed by Core Stage/Boeing using Release 8 of the NASA provided flight software as the basis.

3.2.5 Core Stage Green Run

Green Run is the responsibility of Core Stage/Boeing. During Green Run, the Core Stage is integrated into a test stand at Stennis and several tests are performed, including a full duration burn of all four core stage engines. Software functionality required to support this activity includes managing the pressurization of the tanks, the power system, the thrust vector control, and the Core Stage engines. Additionally, a limited amount of fault management such as abort condition detection, redundancy management, and caution and warning monitoring is provided. Release 10 of the NASA provided flight software is to be used to support this activity in the Ascent Simulation Mode.

3.2.6 Vehicle Integration and Checkout

Vehicle Integration and Checkout is the responsibility of the Ground System Development and Operations Program (GSDOP). During Vehicle Integration, the Core Stage is integrated with the Boosters, Interim Cryogenic Propulsion Stage (iCPS), and Multi-Purpose Crew Vehicle (MPCV) at Kennedy Space Center (KSC) within the Vehicle Assembly Building (VAB). Software functionality required to support this activity includes: Countdown Simulation, Ascent Simulation, and **<TBD-002>.** Release 11 and beyond of the NASA provided flight software are to be used to support this activity in **<TBD-003>** modes.

3.2.7 Pad Integration and Checkout

Pad Integration and Checkout is the responsibility of GSDOP. During Pad Integration, the SLS Vehicle is rolled out to the launch pad and support services are integrated. Software functionality required to support this activity includes: Countdown Demonstration Test, communications checks, wet/dry tanking tests, and **<TBD-004>.** Release 11 and beyond of the NASA provided flight software are to be used to support this activity in **<TBD-005>** modes.

3.2.8 Pre-launch and Ascent

Management of SLS on-pad operational activities through liftoff is the responsibility of GSDOP. Management of the SLS from liftoff through end of mission is the responsibility of MS. Software functionality required to support this activity includes all nominal and off-nominal subsystem management and vehicle flight control through iCPS separation. Release 11 and beyond of the NASA provided flight software are to be used to support this activity in Flight Operations Mode.

4.0 SOFTWARE BUILD CONTENT DEFINITION

The Software Build Content Definition is documented in the Table 4-1 below. Release date information is captured within the schedule milestones in SLS-SCHE-164, SLS Program

Systems Engineering & Integration (SE&I) Control Milestones. SLS-SCHE-164 is the controlling authority if there are conflicts between the control milestones and this document. FSW requirements to release are captured in SLS-RQMT-095, SLSP FSW SRS.

Table 4-1. Flight Software Release Content

	Release						
	Prototype	**7.X**	**8**	**9**	**10**	**11**	**12**
Planned Release Date	Dec-12	May-13	Nov-13	Jun-14	Dec-14	Jun-15	Dec-15
Usage		MAF Integration			Green Run Supported	First FQT for Flight	Defect Repairs
FC Hardware	SDU/SWTB	SWTB	EDU				
Functionality							
Special Test Mode		Fully Implemented except MPCV interface	Fully Implemented (as Functional Capability)				
Ascent Simulation Mode			Basic Implementation		Fully Implemented		
Countdown Demonstration Mode (<TBD-006>)						Fully Implemented	
Functional Capability Switching (Standby Mode)			Fully Implemented				
Mission Management	Prototype		For Included Subsystems	Update	Update	Fully Implemented	
Time synch with ground			Fully Implemented				
Countdown/Autonomous Launch Sequence (ALS)	Prototype		Fully Implemented				

	Release						
	Prototype	**7.X**	**8**	**9**	**10**	**11**	**12**
Planned Release Date	Dec-12	May-13	Nov-13	Jun-14	Dec-14	Jun-15	Dec-15
Launch Commit Criteria (LCC)				Green Run Support		Fully Implemented	
Main Propulsion System (MPS) Management	Prototype		Nominal	Off-Nominal			
Core Stage Thrust Vector Control (TVC) Management	Prototype		1553 interface update only	Fully Implemented			
EPS Management	Prototype		Nominal	Off-Nominal			
Redundant Inertial Navigation Unit (RINU) Management	Prototype		Nominal	Off-Nominal			
Rate Gyro Assembly (RGA) Management				Nominal	Off-Nominal		
GN&C	Initial GN&C Algorithms		Jun 2013 DAC2 GN&C Algorithm updates	Dec 2013 DAC 3 GN&C Algorithm updates	Jun 2014 DAC3 Algorithm updates	Dec 2014 Algorithm updates	
Flight Safety System (FSS) Management						Fully Implemented	
Command and Telemetry Controller (CTC) Management	Prototype		Fully Implemented				
Imaging					Fully Implemented		
Telemetry Format Switching			Fully Implemented				
Flight Computer Redundancy Management (FCRM)	Prototype						

	Release						
	Prototype	**7.X**	**8**	**9**	**10**	**11**	**12**
Planned Release Date	Dec-12	May-13	Nov-13	Jun-14	Dec-14	Jun-15	Dec-15
FCOG Synchronization	Prototype	Fully Implemented					
Input Data Exchange (IDE)/Output Data Exchange (ODE)	Prototype		Fully Implemented				
Telemetry Management	Prototype			Initial Implementation to support Ground	Fully Implemented		
File Management							
iload/kload (parameterized data)			Initial Infrastructure	Green Run Support	Fully Implemented		
FCOG to Ground System (GS) File Transfer	Prototype		Fully Implemented				
GS to FCOG File Transfer	Prototype		Fully Implemented				
Integrated Measurement and Command System(IMACS) Integration				Fully Implemented			
Booster Management	Prototype		Update		Nominal	Off-nominal	
Core Stage Engine Control	Prototype			Nominal	Off-Nominal		
Fault Management							
Abort Condition Detection	Prototype (Simple)			Green Run Support		Fully Implemented	
Redundancy Management				Green Run Support		Fully Implemented	
Caution and Warning				Green Run Support		Fully Implemented	

	Release						
	Prototype	**7.X**	**8**	**9**	**10**	**11**	**12**
Planned Release Date	Dec-12	May-13	Nov-13	Jun-14	Dec-14	Jun-15	Dec-15
Abort Condition Response				Green Run Support		Fully Implemented	
Command and Telemetry Processing (External Interfaces)							
422 (MPCV)			Demo Can Work as Defined	Green Run Support		Fully Implemented	
Manual Control						Fully Implemented	
422 (GSDOP via CTC)			Defined Subsystem are available	Green Run Support	Fully Implemented		
1553 (ICPS)			Demo Can Work as Defined		Fully Implemented		
Internal Command and Data Handling (C&DH) Infrastructure							
422 Interface Manager	Initial Implementation	Update	Update	Fully Implemented			
1553 Interface Manager	Initial Implementation	Update	Update	Fully Implemented			
Data Distribution System (DDS)	Initial Implementation	Update	Update	Fully Implemented			

4.1 Software Releases

The software releases identified in Table 4-1 above are discussed briefly below.

4.1.1 Prototype

A series of prototype flight software releases have been developed to allow for the maturation of the flight software architecture. This activity also supported the development of the FSW development and test facilities along with helping mature the supporting simulation environment. The initial prototype was developed on an Ares Software Development Unit triplex, but as of the final prototype, the software was ported to the SLS Software Test Beds (SWTB). The transition to Release 7.X from the prototype activity is also the point where the development processes start to be formalized.

The prototype supports a nominal fly out of the SLS vehicle with a specific day and time of launch. It includes prototype implementations of the bus architecture, tank pressurization, engines, boosters, and the GN&C algorithms.

4.1.2 Release 7.X

Release 7.X is planned to be used for SITF integration and as part of the basis for the MAF Integration software developed by Boeing. This release provides the Special Test functionality as documented in SLS-RQMT-095, SLSP FSW SRS. In this mode the Flight Computer (FC) constructs a message or messages, based on the very limited set of external commands available in this Release, and posting the command to the appropriate 1553 bus. These commands define bus schedules and command or command chains, and control the execution of the schedules and commands. Release 7.X executes on SLS SWTBs. A Functional Test of planned functionality is to be performed.

A Functional Test in the context of the SLS flight software development effort is an informal test of the implemented requirements against test procedures. Software releases are performed in accordance with SLS-PLAN-074, SLSP Flight Software Configuration Management Plan.

4.1.3 Release 8

Release 8 is planned to port the existing software architecture onto the SLS Engineering Development Units (EDUs). The highest priority and most mature subsystem managers required to support the Green Run software are to be implemented to support nominal functionality. All external interfaces are to have at least sufficient functionality implemented to demonstrate the interface. The support for functionality capability switching is to be provided. Updates to the GN&C algorithms are incorporated. A Functional Test of planned functionality is to be performed.

4.1.4 Release 9

Release 9 is planned to implement additional required Green Run functionality by adding nominal functionality of subsystems not ready for Release 8 and completing the off-nominal functionality for already included subsystems. Additionally, fault management functionality required to support green run activities is implemented. Updates to the GN&C algorithms are incorporated. A Functional Test of planned functionality is to be performed.

4.1.5 Release 10

Release 10 is planned as an update to complete any final needed Green Run functionality as it is slated for use in the Green Run test. All nominal functionality is to be incorporated with this build. Updates to the GN&C algorithms are incorporated. A Formal Qualification Test (FQT) of implemented functionality is to be performed. The FQT process is documented in SLS-PLAN-075, SLSP Flight Software Verification and Validation Plan (SVVP).

4.1.6 Release 11

Release 11 is planned to be the first complete release that could support a flight. All nominal and off-nominal functionality is complete. All fault management is complete. The flight versions of the GN&C algorithms are incorporated. A FQT of all functionality is to be performed.

4.1.7 Release 12

Release 12 is planned to incorporate any defects found prior to the first flight. An analysis is to be performed to identify the extent of needed FQT and regression testing based on required updates.

APPENDIX A
ACRONYMS AND ABBREVIATIONS

A1.0 ACRONYMS AND ABBREVIATIONS

AITP	Avionics Integration and Test Plan
ALS	Autonomous Launch Sequence
BEO	Beyond Earth Orbit
C&DH	Command and Data Handling
CR	Change request
CSCI	Computer Software Configuration Item
CTC	Command and Telemetry Controller
CTRMES	Command and Telemetry Representation and Metadata Exchange Standards
DDS	Data Distribution System
DLE	Discipline Lead Engineer
EDU	Engineering Development Unit
FC	Flight Computer
FCOG	Flight Computer Operational Group
FCRM	Flight Computer Redundancy Management
FCSW	Flight Computer Software
FQT	Formal Qualification Test
FSS	Flight Safety System
FSW	Flight Software
GN&C	Guidance, Navigation, and Control
GS	Ground System
GSDOP	Ground System Development and Operations Program
IAS	Integrated Avionics and Software
ICD	Interface Control Document
iCPS	Interim Cryogenic Propulsion Stage
IDE	Input Data Exchange
IMACS	Integrated Measurement and Command System
IMS	Integrated Master Schedule
ISPE	Integrated Spacecraft Payload Element
KSC	Kennedy Space Center
LCC	Launch Commit Criteria
MAF	Michoud Assembly Facility
M&FM	Mission and Fault Management
MPCV	Multi-Purpose Crew Vehicle
MPS	Main Propulsion System
MSFC	Marshall Space Flight Center
NASA	National Aeronautics and Space Administration
ODE	Output Data Exchange
OPR	Office of Primary Responsibility

RGA	Rate Gyro Assembly
RINU	Redundant Inertial Navigation Unit
SDP	Software Development Plan
SDU	Software Development Unit
SE&I	System Engineering and Integration
SITF	System Integration and Test Facility
SLS	Space Launch System
SLSP	Space Launch System Program
SPIO	Spacecraft and Payloads Integration Office
SRS	Software Requirements Specification
STAS	Special Test Application Software
SWTB	Software Test Bed
TVC	Thrust Vector Control
VAB	Vehicle Assembly Building
VOMRS	Vehicle Operations and Maintenance Requirements

APPENDIX B
OPEN WORK

All resolved TBDs, TBRs, and forward work items should be listed on the Change Request (CR) the next time the document is updated and submitted for formal review and that will serve as the formal change record through the configuration management system.

B1.0 TO BE DETERMINED

Table B1-1 lists the specific To Be Determined (TBD) items in the document that are not yet known. The TBD is inserted as a placeholder wherever the required data is needed and is formatted in bold type within carets. The TBD item is sequentially numbered as applicable (i.e., **<TBD-001>** is the first undetermined item assigned in the document). As each TBD is resolved, the updated text is inserted in each place that the TBD appears in the document and the item is removed from this table. As new TBD items are assigned, they will be added to this list in accordance with the above described numbering scheme. Original TBDs will not be renumbered.

Table B1-1. To Be Determined Items

TBD	Section	Description
TBD-001	2.2	Document number for SLS to MPCV ICD is unknown
TBD-002	3.2.6	Software functionality required to support this activity includes: Countdown Simulation, Ascent Simulation, and **<TBD-002>.** FSW capability definitions to support VAB activities are still in development. The FSW Team anticipates the development of the Vehicle Operations and Maintenance Requirements Specification (VOMRS) to help in this activity (this document is a Draft Cat 1 for SLS PDR).
TBD-003	3.2.6	The NASA provided flight software is to be used to support this activity in **<TBD-003>** modes. The modes used to execute VAB integration and checkout activities will depend on the needed capabilities (see TBD-001).
TBD-004	3.2.7	Software functionality required to support this activity includes: Countdown Demonstration Test, communications checks, wet/dry tanking tests, and **<TBD-004>.** FSW capability definition to support Pad activities are still in development. The FSW Team anticipates the development of the Vehicle Operations and Maintenance Requirements Specification (VOMRS) to help in this activity (this document is a Draft Cat 1 for SLS PDR).
TBD-005	3.2.7	The NASA provided flight software is to be used to support this activity in **<TBD-005>** modes.

TBD	Section	Description
		The modes used to execute Pad integration and checkout activities will depend on the needed capabilities (see TBD-003).
TBD-006	4.0 Table 4-1	**Countdown Demonstration Mode <TBD-006>** FSW capability definitions to support the Countdown Demonstration Mode are still in development. The FSW Team anticipates the development of the Vehicle Operations and Maintenance Requirements Specification (VOMRS) to help in this activity (this document is a Draft Cat 1 for SLS PDR).

B2.0 TO BE RESOLVED

Table B2-1 lists the specific To Be Resolved (TBR) issues in the document that are not yet known. The TBR is inserted as a placeholder wherever the required data is needed and is formatted in bold type within carets. The TBR issue is sequentially numbered as applicable (i.e., **<TBR-001>** is the first unresolved issue assigned in the document). As each TBR is resolved, the updated text is inserted in each place that the TBR appears in the document and the issue is removed from this table. As new TBR issues are assigned, they will be added to this list in accordance with the above described numbering scheme. Original TBRs will not be renumbered.

Table B2-1. To Be Resolved Issues

TBR	Section	Description
TBR-001		