820-019
Deep Space Mission System (DSMS)
Interface Design Standards
D-9597

MON-2
DSN Monitor and Control Standard Practices

Initial Release: November 15, 1998
Change B: TBS

Document Owner

Ron Norman
NMC Subsystem Engineer

Date

Approved by:

Emily Law
Monitor & Control Systems
Service System Manager

Date

Kenneth Kimball
Implementation Engineering Manager

Date

Jeane Stipanuk
Interface Engineer and Release Authority

Date

Reviewed by:

John Milligan CSOC M&C System Engineer	Date	David Recce DSMS Chief Operations Engineer	Date
Erik Barkley Monitor & Control Systems Service System Development Engineer	Date	Jeff Berner Telecommunications Services Service System Development Engineer	Date
Mike Levesque Telecommunications Services Service System Development Engineer	Date	Tomas J Martin-Mur Tracking & Navigation Services Service System Development Engineer	Date
Arthur Freiley Antenna, Microwave & CoF Service System Development Engineer	Date	Michael Connally Antenna, Microwave & CoF Service System Development Engineer	Date
Jay Holladay Network Infrastructure Services, Data Service System Development Engineer	Date		

Change Log

Revision	Issue Date	Pages/Section Affected	Change Summary
New Issue	11/15/98	All	

Revision	Issue Date	Pages/Section Affected	Change Summary
Revision A	03/31/02	All	The following changes have been made in this version of the document: - Removed all references to the Network Planning and Preparation Subsystem (NPP) and the NPP produced Activity Support Package (ASP). This included updating the standard connection data published by the NMC (see Section 6). - Removed all references to the Distributed Computing Environment (DCE) and the Distributed File System (DFS) from the document. Replaced DFS references, where appropriate, with NMC File System (NMCFS) references. - Section 12 updated to describe new procedures for the delivery of subsystem Data Definition and UDS files. Several other sections throughout the document were also updated to reflect these new procedures. - Added a new appendix describing the NMC File System (NMCFS) directory structures for storing shared and subsystem files, and the access granted to MON-2 subsystems to these files. - Section 8 updated to describe the new delivery procedures and file naming conventions for MON-2 subsystem support data files. - Added footnotes in Section 3.3 and Appendix G noting that the NMC does not currently support the Reassign CCN. Also added a footnote in Section 5.2.1.2.5 noting that the NMC and MCIS do not currently support the Recovered Advisory. - Updated Figure 6-1 and associated text to incorporate new SPMC rules for subsystem program and version IDs. - Updated the guidelines in Sections 4.3.1 and 7.3.1 for developing the directives.dat and disp.dat files, respectively. - Section 12 further updated to clarify procedures for SPMC delivery of Data Definition and UDS files. - Removed the mandatory requirement that the MDDS be used to generate the NMC Monitor Data Definition files from several sections. - A general rewrite to improve the clarity of the document. - Section 12 further updated to change 'NMC OE' to 'SS OE' in Figures 12-3 and 12-5 and associated text. - Section 8, Table 8-1 further updated to correct the example given for the Set Revision entry.

Revision	Issue Date	Pages/Section Affected	Change Summary
Revision B	TBS	TBS	The following changes have been made in this version of the document: - Changed all NMC Global Control Engine (GCE) references to NMC Complex Supervisor (CS). - Changed all Translator (TS) references to Translator Plus (TSP). - Updated Table 6-1 (and accompanying text) to allow MON-2 susystems to specify a separate NMCFS path for their UDS display executables. - Changed the functional address in Table 6-4 (and the accompanying text in Section 6.4) under which the NMC publishes the standard monitor data for a connection. - 'XDS' deleted as a NMC supported display type. - Removed all references to the MDDS. Additionally, Monitor Data Definition (MDD) files renamed to Subsystem Data Definition (SDD) files throughout document. - Removed carrier suppression support references. - Removed the footnotes in Section 3.3 and Appendix G noting that the NMC does not currently support the Reassign CCN. - Updated Section 4.2.2 to clarify how NMC handles subsequent subsystem directives when awaiting a processing/wait response from the subsystem. - A general rewrite to improve the clarity of the document. -- Deleted Figures 1-1 and 1-2 -- Removed obsolete or outdated acronyms and references

Possible changes still to make:

- Correct the descrepancy with MON-3 'unassigned' code in STATUS word?

Contents

Figures

Tables

This page is intentionally left blank.

Section 1
Introduction

1.1 Purpose and Scope

The purpose of this module is to provide Deep Space Network (DSN) subsystem engineers (SSEs) and cognizant development engineers (CDEs) with the information needed to interface with the DSN Network Monitor and Control (NMC) subsystem, assemblies, and personnel. It establishes the standards and practices for subsystems monitored and controlled by the NMC.

This document provides a road map to the various standards, references, and procedures needed by subsystem developers. While the focus of this document is to describe the behavior of an 820-019, MON-2-compliant subsystem, issues pertinent to DFL-1-2/890-132 subsystems in the context of the 820-019 MON-1 environment are addressed as well. The primary reference, however, for these legacy subsystems remains the DFL-1-2 and 890-132 standards, although MON-2 provides an update to 890-132 monitor data formats (see Appendix B, Section B.2). In addition, this document provides a brief introduction to DSN procedures pertaining to documentation, testing, and delivery.

1.1.1 Applicability of this Release

This module is authorized under ECR 03.0068 for the Network Control Task (NCT). It is effective with NCT Delivery 1.6, NMC version NSM-5620-OP-F.

Revision A: All references to the Network Planning and Preparation Subsystem (NPP) and the NPP-produced Activity Support Package (ASP) have been removed from the document. (This included updating the standard connection data published by the NMC to remove ASP-related monitor data.). All references to the Distributed Computing Environment (DCE) and the Distributed File System (DFS) have also been removed from the document. (Replaced DFS references, where appropriate, with NMC file system references.). Section 12 has been updated to describe new procedures for the delivery of subsystem Data Definition and UDS files. Several other sections throughout the document have also been updated to reflect these new procedures. Additionally, Appendix H has been added to describe the NMC File System (NMCFS) directory structures for storing shared and subsystem files, and the access granted to MON-2 subsystems to these files. Section 8 of the document has been updated to describe the new delivery procedures and file naming conventions for MON-2 subsystem support data files. Footnotes have been added in Section 3.3 and Appendix G noting that the NMC does not currently support the Reassign CCN. Finally, many sections of the document have been edited to improve the clarity of the document.

Revision B: All references to the NMC Global Control Engine (GCE) in the document have been changed to NMC Complex Supervisor (CS). Table 6-1, and the accompanying text in Section 6.3.2, have been updated to allow MON-2 subsystems to specify a separate NMCFS path for their UDS display executables. The functional address in Table 6-4 under which the NMC publishes the standard monitor data for a connection, and the accompanying text in Section 6.4, have been updated. Section 7.2.1.1 and Table 7-1 have been updated to remove XDS as a display type supported by the NMC. All references to the MDDS have been removed from the document. Appendix C has been updated to remove carrier suppression as a supported data type. Section 3.3 and Appendix G have been updated to remove the footnotes noting that the NMC does not currently support the Reassign CCN. Section 4.2.2 has been updated to clarify how the NMC handles subsequent subsystem directives when awaiting a processing/wait response from the subsystem.

1.1.2 Intended Audience

The primary audience for this document are personnel responsible for the development of DSN subsystems that interface with the NMC. Thus, this document is intended for those personnel responsible for subsystems using either MON-2 (DFL-1-7/MON-1) or DFL-1-2/890-132 standards. Those responsible for legacy subsystems include those who may be partially upgrading their subsystem to the new protocols, or those who may be interested in understanding how the changes in the monitor and control paradigm and practices will or will not impact them.

1.2 Overview of Monitor and Control Documentation

MON-2 sits at the top of the monitor and control standards hierarchy, and can be viewed as a starting point for understanding DSN monitor and control. Other applicable monitor and control documents are listed below. Each of these provides specifics on their respective topics.

MON-1, Monitor and Control Services Standard defines the protocols and application programming interfaces (APIs) for the exchange of monitor and control data between the DSN Network Monitor and Control (NMC) subsystem and other DSN subsystems. [A companion document to MON-1 is the *Monitor and Control Infrastructure Services (MCIS) User's Guide*. The MCIS User's Guide provides instructions and procedures for installing, configuring and using the Monitor and Control Services.]

MON-3, Design Requirements for DSN User Interfaces defines requirements for the design and layout of DSN user interfaces. It focuses primarily on issues pertaining to the "look and feel" of the operator's interface, such as the design of displays, menus, windows, user guidance, reports, directive syntax, operations manuals, etc.

MON-7, Uniform Display Service (UDS) Standard defines the UDS libraries and APIs for creating X/Motif displays which adhere to the MON-3 standards and can be executed on the NMC Workstation. [A companion document to MON-7 is the *UDS User's Guide*. The UDS User's Guide describes how to compile, link and run UDS applications and how to use the UDS APIs.]

DFL-1-6, TMOD Fault Tolerant Data Delivery Services Standard addresses the standards for the fault tolerant data delivery (FTDD) services provided by the Ground Communications Facility (GCF) Data Delivery Subsystem (GDD) of the Ground Communications System (GCS), a service system to the DSN application systems.

DFL-1-7, *DSN Functional Addressing and Transport Standard* essentially replaces the DFL-1-2 standard, commonly referred to as "890-131".

The standards for legacy subsystems (commonly referred to as '890-132 standards') continue to be defined by DFL-1-2, 890-132, and 890-133. These three legacy standards have their counterparts in the DFL-1-7, MON-1/MON-2, and MON-3 standards, respectively.

1.3 Reference Documents

The following documents are referenced in this document.

810-047	*DSN Antenna and Facility Identifiers*
813-001	*DSN Engineering Documentation Standards and Guidelines*
813-024	*SPMC Configuration Management Plan, Draft*
813-101	*Guidelines For SCD Reviews : Service Capability Development (Scd) Standard Practice*
813-109	*Preparation Guidelines and Procedures for Deep Space Mission System (DSMS) Interface Specification*
813-110	*Preparation of Operator's Manuals for DSN Subsystems, SCD Standard Practice*
813-112	*DSMS Testing Standards and Guidelines*
820-016	*DSMS Subsystems - Software Interface*
820-019	*Interface Design Standards*
DFL-1-2	*DSCC General Data Flow Standard* [replaces 890-131]
DFL-1-6	*TMOD Fault Tolerant Data Delivery Service Standard*
DFL-1-7	*DSN Functional Addressing and Transport Standard*
MON-1	*Monitor and Control Services Standards*
MON-3	*DSN User Interface Design Guide*
MON-7	*Uniform Display Service (UDS) Standard*
820-061	*DSMS Subsystem, Configuration Item and Responsibility Definitions*
820-062	*DSMS Abbreviations and Acronyms*
826	*DSMS/System Configuration Files*
831-012	*Network Control Project (NCP) Functional Design Document*
837-009	*Network Monitor and Control Subsystem (NMC) Software Operator's Manual*
837-031	*Network Monitor and Control File System(NMCFS) Software Operator's Manual*
890-131	*MARK IVA Deep Space Network DSCC General Data Flow Standard* [replaced by 820-019, DFL-1-2]
890-132	*Mark IVA DSN DSCC Monitor and Control Data Interchange Standards*
890-133	*DSN Man-Machine Interface Standards for Subsystems Controlled by the DMC*
ANSI/IEEE STD	*IEEE Standard for Binary Floating-Point 754-1985Arithmetic*
ANSI X3.4-1977	*American National Standard Code for Information Interchange (ASCII)*
ISO/IEC 9945-1: 1990, IEEE Std 1003.1-1990	*Information Technology--Portable Operating System Interface (POSIX) Part 1: System Application Programming Interface (API) [C Language]*
JPL D-4000, V3.0	*Software Management Standards, Level I and II*
SOM-NBK-6081-OP	*Support Products Provider Assembly (SPPA) Software Operator's Mauual*
UG-DOI-5555-OP	*Multi-Use Software User's Guide*

UG-DSI-5605-OP	*Uniform Display Service (UDS) User's Guide*
UG-DSI-6000-OP	*Monitor and Control Infrastructure Services (MCIS) User's Guide*

1.4 Reader's Guide to this Document

This document is largely organized by DSN data types, e.g., monitor data, directives/ responses, events, configuration change notices, etc. Section 2 discusses general monitor and control practices that are not specific to any one data type. Sections 3 through 9 constitute the heart of MON-2, and discuss standard practices for each data type. Section 10 provides a brief introduction to Jet Propulsion Laboratory (JPL)/DSN software documentation, and then addresses subsystem documentation required by the NMC. Section 11 provides a brief overview of testing in the DSN, and then addresses NMC-specific testing issues, as well as providing a MON-2-based checklist. Section 12 provides an overview of DSN delivery practices for subsystem SDD/UDS files.

Appendix A contains an acronym list and glossary. Appendix B describes the standard monitor data formats for MON-2 subsystems. It also contains an update to the monitor data formats specified in 890-132. Appendix C addresses issues of interest to DFL-1-2/890-132 subsystems, as well as MON-2 subsystems interfacing with legacy subsystems. Appendix D provides a summary of required information from the subsystems. Appendix E lists the subsystems categories used by the NMC to organize the various display elements, including icons and menus. Appendix F is a summary of MON-2 "shall" statements (requirements) defined in this document. Appendix G shows typical subsystem failure recovery scenarios supported by the NMC. Appendix H shows the NMC File System (NMCFS) directory structure used to store the subsystem Data Definition files required by the NMC.

1.5 Definition of MON-2 Terms

advisory: An event notification reporting minor malfunctions, changes in status, routine progress, that does not require an operator to take corrective action. Advisory types are: deviational, completion, log-only, progress, and recovery. Refer to Section 5.2.1.2 of this document.

alarm: An event notification reporting a serious, anomalous incident that requires an operator to take corrective action. Alarm types are: critical, emergency, warning. Refer to Section 5.2.1.1 of this document.

assembly: Hardware or software logical and physical entities.

assignment: The process whereby a DSN equipment is assigned to support an activity.

checkpoint: A periodic snapshot taken of the subsystem data necessary to reestablish connection support should the subsystem controller require resetting or restarting.

configuration: The internal arrangement, processing mode, or state of a component, assembly, or resource.

configuration data: Configuration data is generally expressed as the current values of variables over which the NMC has control.

connection: A connection is a set of DSN resources/equipment needed to track a spacecraft or perform an activity. A connection is controlled and monitored by a Connection Controller using a NMC workstation.

directive: Directives are used to initiate a control action on a controlled application. Refer to Section 4 of this document. Directives are sometimes referred to as “ODs” (operator directives), and in even earlier terminology as “OCIs” (operator control inputs).

event notification: A short text message issued by a subsystem to provide feedback, typically to an operator, regarding an event or occurrence in the subsystem. Events are further classified as prompts, advisories, or alarms. Refer to Section 5 of this document.

functional address: A mechanism for identifying resources (usually, subsystem assemblies) within a connection by their functional assignment. Refer to Document 820-019, DFL-1-7.

group: (As defined in document 813-001:) A configuration of major assemblies that operate as a unit; usually to satisfy availability and redundant strings requirements.

legacy subsystem: Any subsystem that does not run the MON-2 protocols.

major assembly: (As defined in document 813-001:) A set of hardware and software assemblies which are designed to deliver a specific capability. These are major design elements of a DSN subsystem.

Monitor and Control Infrastructure Services (MCIS): MCIS consists of four services: Monitor Data Service (MDS), Functional Addressing Service (FAS), Monitor and Control Services (MCS), and the Event Notification Service (ENS). The MDS implements the publication and subscription of monitor data. The FAS is the service that maintains and transports functional addresses around the DSN. The MCS provides services used for processing events, directives/responses, and configuration change notices (CCNs). The ENS provides services to subsystems for subscribing to all or selected events that the DSN subsystems send to the NMC (or to other subsystems).

monitor by exception: The subsystems are able to interpret the health and performance of the entities they control and to relay the information in a timely manner, in a format that is understandable by an operator and by other subsystems. Refer to Section 2.3 of this document.

monitor data: Subsystem data describing the subsystem (assembly, subassembly) status, configuration, or performance.

parameter descriptor: Any data item, or group of data items, may have a descriptor that is composed of (1) parameter validity and (2) parameter analysis. Refer to Appendix B, Sections B.1.3 and B.2.3 of this document.

performance: The functioning of a component, assembly, or subsystem function with respect to the environment in which it operates.

performance data: Performance data contains measurements of both internal and external quantities as well as variables derived from those measurements.

permanent functional address: The permanent address by which a subsystem is known to the NMC (independent of the subsystem’s assignment mode).

positive closed-loop control: The concept of a subsystem providing verification that a requested action has been accomplished. Refer to Section 2.2 of this document.

resource: General term designating any equipment, item, or group of items to be scheduled or monitored and controlled.

standards and limits: Standards are the expected values for configuration data; deviations from the expected values require operator attention or action. Limits are quality measures for performance data, beyond which operator attention or action is required.

status: The operational readiness or the internal functioning of equipment and software; may apply to an individual component, assembly, function, or entire subsystem. Refer to Appendix B, Sections B.1.1 and B.2.1 of this document.

status data: Data reported by a subsystem on the overall operational health of the subsystem and the health of its individual assemblies.

Subsystem: This term is used to refer to a subsystem that interfaces with the NMC. The term is used to refer, in general, to a subsystem's assemblies and subassemblies, both hardware and software.

Temporal Dependency Network: Defines dependencies between Automation Blocks and the order in which these Automation Blocks are executed to complete a scenario**.**

TDN block: Logical sequence of activities to perform a task (e.g., configure Metric Data Assembly (MDA) or move antenna to point).

1.6 Abbreviations and Acronyms

See Appendix A.

Section 2
General Monitor and Control Practices

This section identifies the general capabilities required of a subsystem in order for the subsystem to be deemed MON-2 compliant.

2.1 Automation

Automation is the primary mechanism for achieving the DSN goal of providing DSN services to projects while operating under the constraint of a reduced operations workforce. The NMC Automation was developed to support this goal through the use of Temporal Dependency Networks (TDNs) and Automation blocks, which are run within the NMC.

The CDSCC Center For Automation is responsible for the development and maintenance of DSN TDNs and Automation blocks. Working with subsystem CDEs and Operations, the CDSCC Center For Automation develops, tests and distributes TDNs and Automation Blocks for use within the DSN, and works with subsystem CDEs to maintain these as subsystems upgrade.

Automation blocks may reference subsystem monitor data, directives, displays, and/or events. Thus, any subsystem modification (including additions, deletions, or modifications to monitor data, directives, displays, and events) that affects automation **shall** be coordinated with the CDSCC Center For Automation. Subsystems **shall** test with NMC Automation following any subsystem changes.

2.2 Positive Closed-Loop Control

Positive closed-loop control is the concept of providing verification that a requested action has been accomplished. Thus, if a directive is issued to effect a change in a subsystem, evidence that the change has in fact occurred should be available to an inquiring entity, whether an operator or a software process. For directives issued by the NMC Automation Assembly, a visual-only clue to an operator would not be sufficient. Examples of how a change in a subsystem’s state in response to a directive could be reflected include monitor data, events, and directive responses. The need for subsystem implementations to accommodate closed-loop control is discussed below in the sections on directives, monitor data, and events.

2.3 Monitor by Exception

Monitor by Exception allows operations to proceed normally with the operator being alerted only if an exception (non-nominal operation) occurs. The subsystems are able to interpret the health and performance of all controlled entities and to relay this information in a timely manner, in a format that is understandable by an operator and by other subsystem.

Monitor by Exception provides the environment for an operator to be solely responsible for several simultaneous tracking activities without being overloaded with low-level data and decision making. This capability is provided through interactive displays, event messages, audio, and possibly other indications that allow an operator to concurrently monitor and control several tracking activities. The operator should also have an easily accessible lower level view of the problem causing the exception, once an exception is identified by the

subsystem. (See 820-019, *MON-3 Appendix E* for a more complete description of *Monitor by Exception*.)

2.4 Self-Configuration

Another basic concept in the DSN monitor and control philosophy is for subsystems to be able to configure themselves with (at most) a single directive. Depending on the subsystem, self-configuration may be tied to specific spacecraft, antenna, or site-specific characteristics, which are defined in configuration tables. These DSN configuration files are downloaded to the subsystems prior to intended use. (See Section 8.)

The NMC will publish a standard set of connection monitor data for each MON-2 subsystem in a connection (See Section 6.4) to support the subsystem self-configuration process. This standard connection data will be published by the NMC before Assign CCNs are sent to the subsystems. The following guidelines apply if a subsystem is unable to access this standard connection data after it receives an Assign CCN:

(1) The subsystem **shall** wait indefinitely for the standard connection data to become available, i.e., the subsystem should not timeout.

(2) If the subsystem cannot complete its configuration based upon available data, it **shall** issue an appropriate alarm notification to the NMC (See Section 5.2.1.1) under its connection functional address.

(3) The subsystem **shall** continue to accept and process directives appropriate for its state. For example, subsystem directive may be used to supply data needed to complete the subsystem's configuration.

(4) A subsystem **shall** always accept an Unassign CCN from the NMC.

2.5 Self-Evaluation

Subsystems **shall** evaluate their performance and report deviations to the NMC. For example, a subsystem may determine a performance deviation by comparing actual performance data against subsystem 'Standards and Limits' or by comparing the delta of actual vs. predicted calculations against subsystem "Standards and Limits." The results of these comparisons **shall** be made immediately available to operations personnel through event notifications and/or monitor data.

Subsystems **shall** also monitor their output of all subsystem products, and alert operations personnel when the required outputs are either impaired or nonexistent.

2.6 Remote Restart

It is a DSN goal to provide operators with the capability to remotely restart a subsystem, hardware and software. For subsystems that provide this capability in whole or in part (e.g., remote login and restarting of a software process), the following requirements should apply. To reset a subsystem controller, the NMC would issue a remote initialization stimulus across the local area network (LAN). Hardware or software within the controller should detect the stimulus and cause the controller to be reset or reloaded. The controller should immediately begin its internal initialization; within 30 seconds, the controller should complete its internal initialization and enter the unassigned communication mode. (Refer to Section 3.2.1.1 on unassigned mode.)

The specific mechanism to be used by the NMC to restart a subsystem controller **shall** be defined in the NMC-subsystem interface agreement.

2.7 Standard Deployment and Software Versions

The occurrence of multiple versions of subsystem controller software deployed through the DSN complicates the ability to achieve automation, maintain automation scripts, and provide meaningful high-level display needed in an environment where there are fewer operators and higher workload. It is a DSN goal to achieve standard deployment of subsystem controllers, such that there is one version of a subsystem controller that is common to, and deployed at all DSN sites. Additionally, subsystem files, particularly those used by the NMC, **shall** be identified with specific software versions. The NMC software is coupled to subsystem program/version ID, and as such is very sensitive to changes in subsystem software and data files. Refer to Section 6.2.6, Program/Version ID Standard.

2.8 Subsystem Addresses

Each DSN subsystem is assigned one physical address and at least one functional address. A subsystem's physical address is permanently assigned to the subsystem and is used by the subsystem to register with the MCIS. Subsystem physical addresses are documented in Document 826, *DSMS/System Configuration Files*. The manner in which a subsystem sets its physical name with MCIS is described in DFL-1-7.

The different types of functional addresses a subsystem may have are introduced below and are described in more detail later in this document. The manner in which subsystems communicate using these functional addresses is described in DFL-1-7.

2.8.1 Subsystem Permanent Functional Addresses

Each subsystem controller (whether connection-assignable, antenna group, multi-connection, or non-assignable) must have a permanent functional address. This address is permanently assigned to a subsystem and allows communications with the subsystem independent of the subsystem's current assignment state.

Standard syntax for permanent functional addresses are defined in Table 6-3 of this document.

The permanent functional addresses assigned to DSN subsystem controllers are documented in Document 826, *DSMS/System Configuration Files*.

2.8.2 Subsystem Connection Addresses

Each connection-assignable or multi-connection subsystem controller is also assigned a connection address when it is assigned to a connection. This connection address should be used for all connection-related communications by (or with) the subsystem.

A connection address may be one of two types: an 'antenna group functional address' or a 'connection functional address'. In accordance with DFL-1-7, subsystems that are hard-wired to a specific antenna communicate using an 'antenna group functional address', and subsystems that are antenna independent communicate using a ‘connection functional address’.

The standard syntax for connection addresses are defined in Table 6-3 of this document.

The connection addresses assigned to DSN subsystem controllers are documented in Document 826, *DSMS/System Configuration Files* (However, a connection-assignable or multi-connection subsystem does not need to know its connection address beforehand, since this address is provided in the Configuration Change Notice (CCN) sent to the subsystem to assign it to a connection.)

2.9 NMC-to-Subsystem Communications

The NMC communicates with a subsystem only through the subsystem’s controller assembly. Other assemblies of a subsystem are monitored and controlled by the NMC indirectly through the subsystem’s controller assembly.

Section 3
Assignment

Assignment is the process by which a DSN subsystem is notified of its role in a connection. DSN subsystems are typically organized into connections to support a particular activity. For instance, a downlink channel can be assigned to perform the telemetry process function for a connection that is supporting a scheduled activity. Some subsystems can support only one connection at a time, other subsystems can concurrently support multiple connection, and still other subsystems perform a global function and are not directly assignable to a specific connection. This section describes the different types of subsystems and the assignment rules applicable to each type.

3.1 Subsystem Types and Functional Addresses

Subsystems controllers are classified into one of three categories depending on how they behave with respect to connection assignment: Connection-assignable, Multi-connection assignable, and Non-assignable. These subsystem categories and the types of functional addresses assigned to each type are summarized in Table 3-1 and described in more detail below.

(1) Connection-assignable

A connection-assignable subsystem controller is designed to participate in only one connection at a time. A connection-assignable controller may be antenna independent or it may be hard-wired to a specific antenna. Examples of antenna independent, connection assignable controllers are the Downlink Channel Controller (DCC) and the Array Control Processor (ACP). Examples of antenna specific, connection assignable controllers are the Uplink Controller (ULC), the Microwave Generic Controller (UGC), and the Antenna Pointing Controller (APC).

A connection-assignable subsystem controller **shall** have either a connection functional address or an antenna group functional address and a permanent functional address.

(2) Multi-connection assignable

A multi-connection subsystem controller is designed to participate in more than one connection at a time. An example of a multi-connection controller is the Telemetry Simulation Assembly II(TSA 2).

A multi-connection subsystem controller **shall** be capable of having multiple connection functional addresses and a permanent functional address.

(3) Non-assignable

A non-assignable subsystem controller is one which does not have a connection-specific function. Examples of non-assignable subsystem are the Frequency and Timing Subsystem (FTS), and the Media Calibration Subsystem (DMD).

Non-assignable subsystem controllers **shall** only have permanent functional addresses.

Table 3-1. Subsystem vs. Functional Address Types

Subsystem Type	Assigned Address Type	Permanent Address Required?
Connection Assignable	Connection Functional or Antenna Group	Yes
Multi-Connection Assignable	Connection Functional	Yes
Non-Assignable	N/A	Yes

3.2 Communication Modes and Subsystem Behavior

The communication modes and rules appropriate for connection-assignable, multi-connection and non-assignable subsystems are presented in this section.

3.2.1 Connection-Assignable Subsystems

A connection-assignable subsystem controller can be in one of two possible modes: unassigned or assigned. A subsystem selected for a connection, and informed of that role, is said to be "assigned." When a subsystem is not in a connection, it is in the "unassigned mode" and is available for assignment.

3.2.1.1 Unassigned

A subsystem in an unassigned mode is not assigned to a connection, but is available for assignment. A well-behaved subsystem **shall** place itself into the unassigned mode following bootup or reset.

A connection-assignable subsystem in an unassigned mode **shall** register under its permanent functional address to receive CCNs. Additionally, while in an unassigned mode, an assignable subsystem **shall** use its permanent functional address for all external communications.

Assignable subsystems in the unassigned mode are not restricted in their communications, although subsystems should not engage in communications which would generate spurious LAN traffic. It is reasonable for assignable subsystems in the unassigned mode to be able to:

- receive directives and CCNs
- subscribe to NMC monitor data
- send or receive events (When a subsystem is in unassigned mode, event notifications to the NMC should always be directed to the Complex Supervisor (CS) assembly of the NMC.)
- receive support data
- publish monitor data.

Unassigned subsystems are also not restricted as to report generation in response to directives.

3.2.1.2 Assigned

A subsystem in an assigned mode has been assigned to a specific connection.

An assignable subsystem in the assigned mode **shall** use the functional address supplied in the CCN as its source address in all subsequent connection-related communications. Such a subsystem is not restricted in receiving or initiating communications from or to other DSN subsystems (presumably, those which are in the same connection).

3.2.2 Multi-Connection Subsystems

In general, a multi-connection subsystem is expected to support the same communication modes and rules as a connection-assignable subsystem. However, since a multi-connection subsystem may be in more than one connection at a time, special interpretations of these modes/rules may be required. The interface agreement between the NMC and a multi-connection subsystem **shall** define any specific assignment protocol for these subsystems.

3.2.3 Non-Assignable Subsystems

A non-assignable subsystem **shall** ensure that its permanent functional address is published in accordance with DFL-1-7. The subsystem **shall** assume an assigned communications mode, following a subsystem reset/restart; such subsystem will not receive CCNs.

3.3 Configuration Change Notice (CCN)

The Configuration Change Notice (CCN) is the vehicle used by the NMC to inform a subsystem as to what connection(s) it is in, as well as to what communication mode the subsystem should transition. There are three types of CCNs: Unassign, Assign, and Reassign. (The NMC will continue to support other CCN types for legacy subsystems only.)

MON-1 provides specifics on the content of configuration change notices as well as the assignment API. The CCN communication mode transitions supported by the NMC are shown in Figure 3-1 and described below.

3.3.1 Unassigned to Assigned

The NMC changes the mode of a subsystem from unassigned to assigned by sending the subsystem controller an Assign CCN or a Reassign CCN. When a subsystem is selected to be in a connection, the NMC transmits an Assign CCN to the subsystem instructing it to transition to an Assigned mode.

A Reassign CCN is used to inform a subsystem controller that it is being assigned to an operational connection and that the controller should use the subsystem's checkpoint information to configure itself and continue connection operations. A Reassign CCN might be used, for example, to reconfigure a subsystem controller, or after a reset of the subsystem controller (or a restart of the controller's application) to instruct the controller to transition to an Assigned mode, but to continue to support an operational connection (without disrupting the connection). A Reassign CCN might also be used by the NMC to assign a backup controller to take the place of a failed controller. (See the scenarios in Appendix G.)

3.3.2 Assigned to Unassigned

The NMC changes the mode of a subsystem from assigned to unassigned by sending the subsystem an Unassign CCN.

3.3.3 *Assigned to Assigned*

In the case where a "Standby Backup" controller exists, the NMC will send a Reassign CCN to an assigned backup controller to instruct this controller to take over as the primary unit. (See the scenarios in Appendix G.)

The NMC will not attempt to assign a subsystem directly from one connection to another connection without transitioning the subsystem through the unassign mode. A subsystem **shall** not allow a change to the connection number while in the assigned mode.

3.4 *Subsystem-Provided Information*

Subsystems **shall** provide the following information in their interface agreement with the NMC:

(1) identification of subsystem type with respect to assignability (multi-connection, non-assignable, or connection-assignable)

(2) identification of subsystem functional and permanent functional addresses.

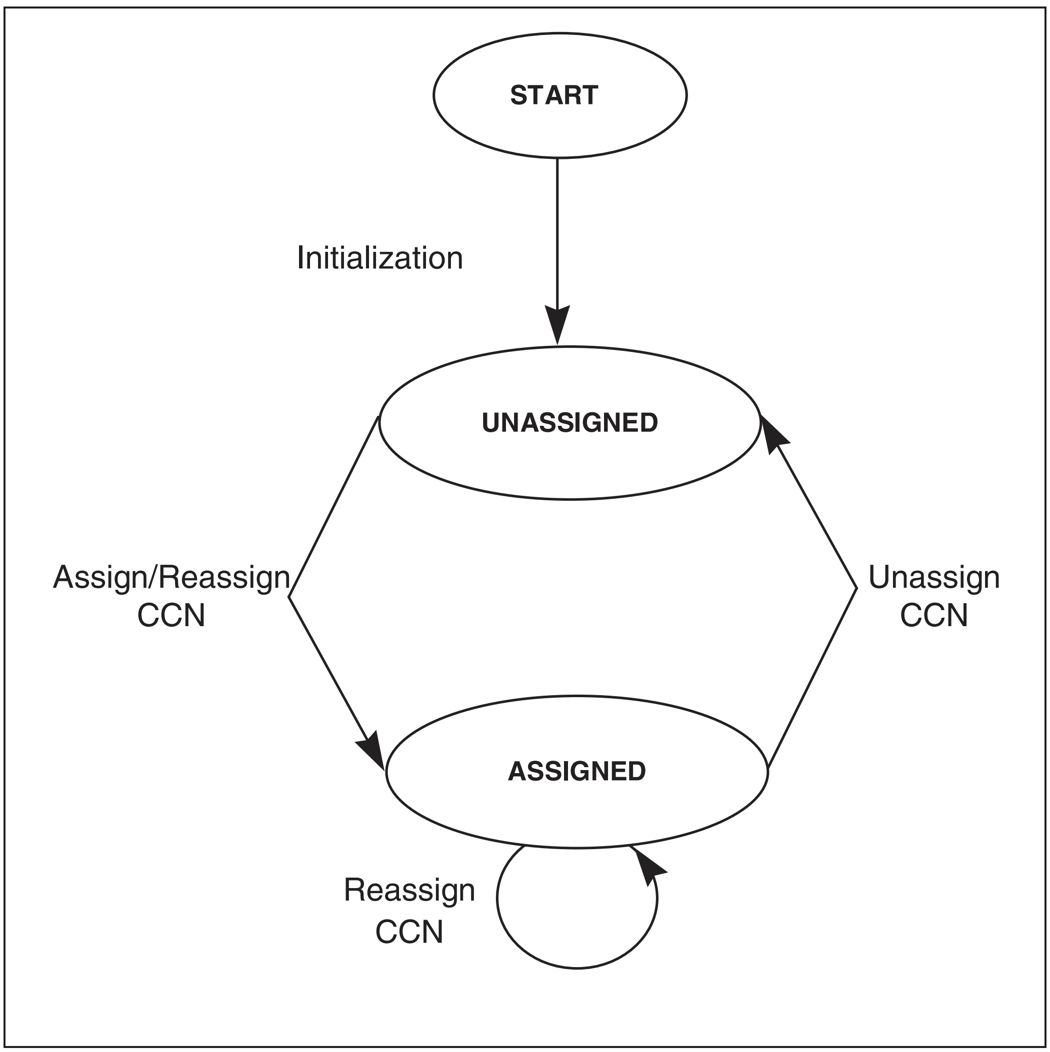

Figure 3-1. Communication Mode State Transitions

This page is intentionally left blank.

Section 4
Directives/Responses

4.1 Introduction

Directives are issued from the NMC to monitor and control DSN subsystem controllers. This section defines the directive/response sequence and timing, and NMC dependencies with regard to subsystem directives. Directives can be initiated by software (i.e., automation scripts and operator scripts) or by DSN personnel. Directive responses are issued by the subsystem controller receiving a directive in order to provide feedback to the NMC indicating the controller has received and is reacting to the directive.

4.2 Practices

4.2.1 Protocol and Content

MON-1 describes the protocol for exchanging directives and responses.

MON-3 provides specifics regarding the content and format of directives and directive responses.

All directives issued to subsystems **shall** be entered into the appropriate NMC log(s). For instance, a directive can be initiated to a subsystem from one of the subsystems's UDS displays. In this case, the subsystem will ensure that the directive is also entered into the appropriate NMC log(s).

Subsystems **shall** provide text in the directive response message to further describe and clarify the meaning of the response.

As described in Section 6.2.4(3), directives that control status or configuration parameters **shall** be associated with the monitor data reflecting the directed changes.

4.2.2 Directive Response Categories and Criteria for Use

In general, directive responses issued by subsystem controllers are intended to communicate one of the following conditions:

(1) The directive was received and executed, or

(2) The directive was received and is either being executed or will be executed momentarily, or

(3) The directive was rejected because the syntax is in error, parameters are out of limit, values are missing, the directive itself is inappropriate, etc.

Subsystems will communicate these conditions with the following types of responses: Completed, Processing, Processing/Wait, Started, and Rejected (as described below; also, see Figure 4-1).

Subsystems **shall** use an estimated-time-to-complete criterion in determining what response category to use when issuing the first response to a directive. The response categories are described below with respect to the condition under which each will be issued. Note that the estimated completion times are intended to be pre-execution estimates of the completion time of the process invoked by the directive.

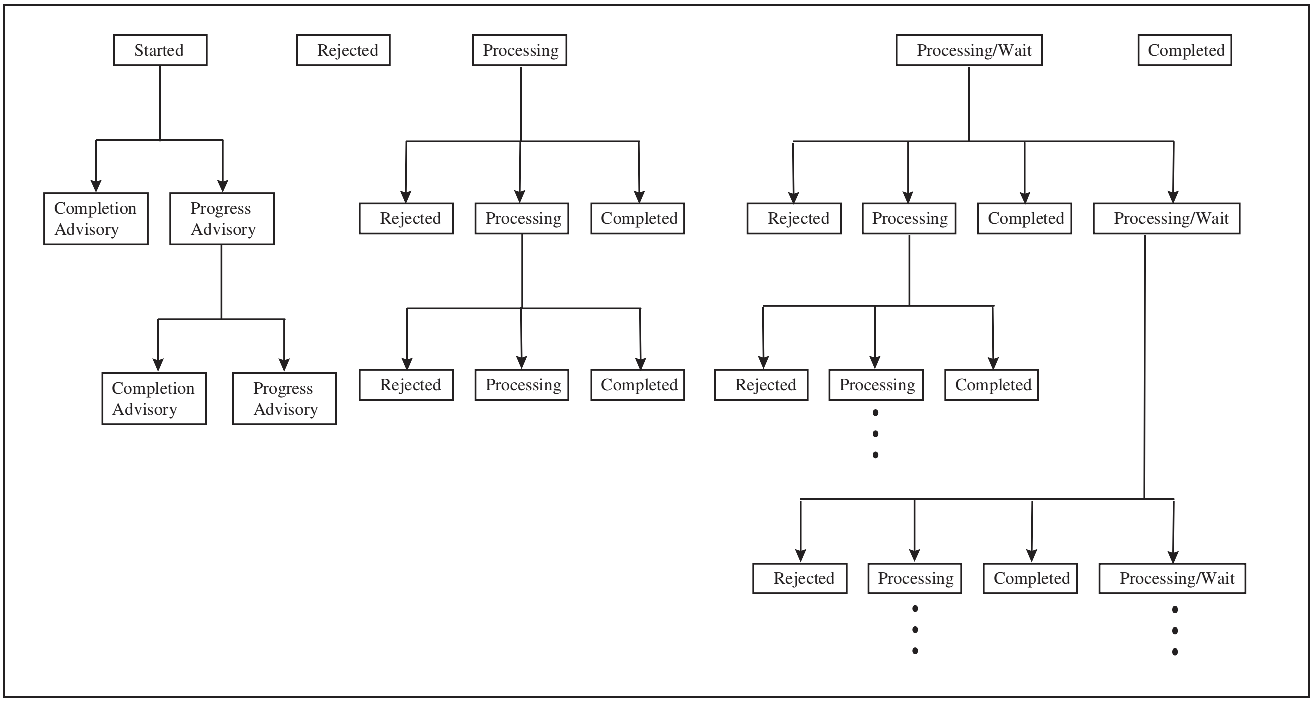

Figure 4-1. Directive Response Sequences

<u>Completed</u>

If a subsystem can execute a directive within one second of receiving it, then the subsystem **shall** send a Completed response after executing the directive. The one second applies when Completed is the first directive response in a directive/response exchange.

Upon receiving a Completed response, the exchange is complete. The NMC may then send a new directive immediately, beginning a new exchange.

Completed responses can also be sent as a second response following a Processing [or Processing/Wait] response. Refer to the Processing and Processing/Wait paragraphs below.

<u>Processing</u>

If the subsystem can expect to execute the directive within one to five seconds after receiving it, and if during this time it can receive additional directives, the subsystem **shall** send a Processing response within one second of receiving the directive. This type of response signifies that the subsystem is processing the directive.

Upon receiving a Processing response, the NMC may send a new directive immediately to begin another exchange. This second directive **shall** be processed concurrently. Processing of concurrent exchanges **shall** follow the protocol standards independently.

The subsystem **shall** send a second response within a total of five seconds from the time it received the original directive. This second response could also be another Processing response. The subsystem **shall** continue to send responses until either a Rejected response or a Completed response is sent. Transmission of a Rejected response or a Completed response completes the exchange for the original directive.

Upon receiving the Rejected or Completed response, the NMC may send a new directive to begin a new exchange.

Processing/Wait

If, while executing a directive, the subsystem is unable to receive a second directive, the Processing/Wait response **shall** be sent within one second of receiving the first directive.

Upon receiving the Processing/Wait response, the NMC will not send another directive to the particular subsystem until receiving a Rejected or Completed response. The NMC will queue any subsequent directive(s) for the subsystem until a Rejected or Completed response is received from the subsystem. Once a Rejected or a Completed response is received, the NMC will send the next queued directive to the subsystem.

Within 15 seconds of receiving the directive, the subsystem **shall** send a second directive response. This second response could also be another Processing/Wait response. The subsystem **shall** continue to send responses until either a Rejected response or a Completed response is sent. Transmission of a Rejected response or a Completed response completes the exchange for the original directive.

Started

If the subsystem takes more than five seconds to execute the directive (and the Wait condition does not apply), the Started response **shall** be sent to the NMC within one second of receiving the directive. In the case of the Started directive response, the response message should include an estimate of the length of time that will be required to execute the directive.

Upon receiving the Started response the exchange is complete, and the NMC may send a new directive to begin a new exchange.

After executing the directive, the subsystem **shall** send a Completion Advisory (see Section 5.2.1.2.2) to the NMC.

The subsystem **shall** send Progress Advisories (see Section 5.2.1.2.3) to the NMC while the subsystem executes the directive. These advisories should include any updated information on the estimate of completion time.

In cases where the directive is not executed successfully, the subsystem **shall** send a Deviation Advisory (see Section 5.2.1.2.1) or a Warning Alarm, as appropriate (see Section 5.2.1.1.3).

Rejected

If the subsystem cannot accept a directive, the Rejected response message **shall** be sent to the NMC and clearly state the reason for rejecting the directive (e.g., syntax, semantics, parameter validity check, out of sequence, etc.). The subsystem **shall** send the Rejected response message within one second of receiving the directive; the one second applies when Rejected is the first directive response in a directive/response exchange.

Upon receiving the Rejected response the exchange is complete, and the NMC may send a new directive to begin a new exchange.

Rejected responses can also be sent as a second response following a Processing [or Processing/Wait] response. Refer to the Processing and Processing/Wait paragraphs above.

4.2.3 *Timing Considerations*

This standard defines three time limits. The timing period for each limit begins when the subsystem receives the first directive to begin an exchange. Limits of one second, five seconds, or fifteen seconds thereafter **shall** be satisfied when the subsystem transmits an appropriate directive response.

Subsystem designs **shall** accommodate directive response time limits as follows:

(1) In the absence of any other I/O activity to or from the LAN(s), the response time limit **is** met.

(2) During normal I/O activity, no more than 5% of the responses are transmitted later than the nominal response time limit.

(3) The absolute maximum response time limits associated with each of the nominal response time limits (referred to Section 4.2.2 above) are provided below. Regardless of LAN I/O activity (but assuming that all acknowledged transmissions are successful on first attempt), the response is transmitted within the absolute maximum response time limit defined as follows:

Targeted Response Time	**Maximum Response Time**
1 second	3 seconds
5 seconds	8 seconds
15 seconds	18 seconds

If a directive response is not received within the required time limit, the MCIS completes the directive/response exchange by generating a Timeout directive response to the NMC. (Note that Timeout directive responses are not issued by subsystems.)

If a response is received at a time other than as specified in this protocol, the MCIS treats the directive response as if an event had been received from the subsystem, by issuing a Late Response event notification (see Section 5.2.1.4) to the NMC.

4.2.4 *Standard Directives/Responses*

MON-3 lists standard directives and describes directive syntax and directive naming conventions. MON-3 also establishes guidelines for providing feedback to operators; these guidelines are applicable to the message content of directive responses.

4.2.5 *Directive Help Facility*

MON-3 establishes a requirement for an on-line help facility for directives and sets general guidelines for developing a help facility. MON-7 also specifies the standard file names and location of the HTML files for access through UDS subsystem displays.

4.3 *Subsystem-Provided Information*

This section describes the information associated with directives that must be made available in order for NMC to interface with the subsystems. It addresses directive descriptions, and the content and format of the directives file needed by the NMC.

In cases where positive closed-loop control relationships exist (i.e., a directive is associated with monitor data or specific events), such relationships **shall** be identified in the NMC subsystem interface agreement.

4.3.1 NMC Directive List File

The NMC Directive List File contains the name of all directives recognized by the subsystem controller together with the NMC-required and optional attributes for the directives. The NMC uses this file to present a menu of directives for the selected subsystem. This is also used by NMC, for safety reasons, to prevent control of remote equipment.

The NMC requires that a NMC Directive List File (aka a subsystem's *directives.dat* file) be delivered for each subsystem software version. It is the responsibility of the subsystem CDE to produce and deliver this file using the test/delivery procedures defined in Section 12. The subsystem CDE **shall** provide the information listed in Table 4-1 for each subsystem directive in the subsystem's *directives.dat* file.

Table 4-1. Directive Entry Keywords

Keyword	Valid Options	Description
OD=	The name/mnemonic of the directive in uppercase.	The mnemonic of the directive. This keyword should precede any other keywords for the directive.
ODSS=	The category of this directive, in uppercase. (See Appendix E to identify the category(s) that apply to your subsystem)	This categorization is used by NMC to organize various display elements including icons and menus. For example, a Receiver (RCV) or Transmitter (TXR) directive.
ODREMOTE=	TRUE, FALSE (Defaults to FALSE, if not provided)	Set to "FALSE" when the directive should not be executed from a remote site, e.g., for safety concerns.
ODMODE=	OP, TEST, MAINT, DBUG	Defines the execution mode(s) for which the directive is intended, i.e., the directive is intended for use in the operational, test, maintenance and/or debug mode(s). List all of the modes that apply to this directive, separated by commas. (A white space can be added after the comma, if desired.)

Keyword	Valid Options	Description
ODPHASE=	PRETK, INTK, POSTTK, NOTK, CONFIG, CALIB, SUPDAT, STSPER, ACTEND, DIAG.	Defines the execution phase(s) of a connection for which the directive is intended, i.e., the directive is intended for use in the pre-track, track, post-track, configuration, calibration, support data, status/performance, end of activity or diagnostics phase(s). List all of the phases that apply to this directive, separated by commas. (NOTK relates to a directive that is not connection-specific.) (A white space can be added after the comma, if desired.)
ODMOU=	YES, NO	Defines whether the directive is used frequently by an operator. (The intent is to identify the most commonly used directives of an assembly.) For a directive that is used often, set to YES, otherwise NO.
ODDESC=		A short, one-sentence description of the directive. (White spaces are allowed.)
END		Ends the parameters for the described directive. The directive statements should always end with this statement.

Sample entries in a subsystem's `directives.dat` file are shown below in Figure 4-2. The following guidelines apply to the file:

(1) Keywords (and associated values) listed in Table 4-1 are used to describe each directive.

(2) Each line begins with a directive entry keyword and ends with a carriage return. (Note that lines beginning with "#" are comments.) There will only be one keyword per line.

(3) The values associated with each keyword are in uppercase, except for those associated with ODDESC and the comments, which can use uppercase and lowercase.

(4) White spaces are not allowed, unless otherwise noted.

(5) If an optional keyword is not needed, it should not be entered.

(6) Required keywords are OD, ODSS (with parameters), and END.

The following is an example of a `directives.dat` file for two directives:

```
#*****************************************************************
#*
```

```
#*  This NMC Directive List File created

#*  on Thu Jan 29 09:29:51 PST 1998

#*
# COLLECTION_NAME : JeaneMON-2
# PROGRAM_ID : JGS-0000-OP
# ACRONYM : FYI
# VERSION_NUMBER : DV1.0.0
# DATA_TYPE : test
# MDSPECS_EXISTS : TRUE
# DISPS_EXISTS : TRUE
# DIRECTIVES_EXISTS : TRUE
#**************************************************************

OD=DIAG
ODSS=TRK
ODMOU=NO
ODMODE=TEST, MAINT, DEBUG
ODPHASE=DIAG
ODREMOTE=FALSE
ODDESC=Perform Level-2 Diagnostics
END

OD=HI
ODSS=TRK
ODMOU=YES
ODMODE=OP
ODPHASE=PRETK, INTK, POSTTK
ODREMOTE=TRUE
ODDESC=Version and MDA Active Query
END
```

Figure 4-2. Sample Entries in a `directives.dat` File

This page is intentionally left blank.

Section 5
Event Notifications

5.1 Introduction

Event notifications are short text messages issued by subsystems to provide feedback, typically to an operator, regarding an event or occurrence in the subsystem. This section describes the different types of event notifications and their usage, i.e., alarm, advisory, and prompt.

5.2 Practices

5.2.1 Event Notification Classifications and Associated Behaviors

Event notifications can be classified into three categories: alarms, advisories and prompts. (Note that an additional event type is described in Section 5.2.1.4. This Late Response event type is not generated by subsystems, however.) Figure 5-1 shows the criteria for generating an event notification.

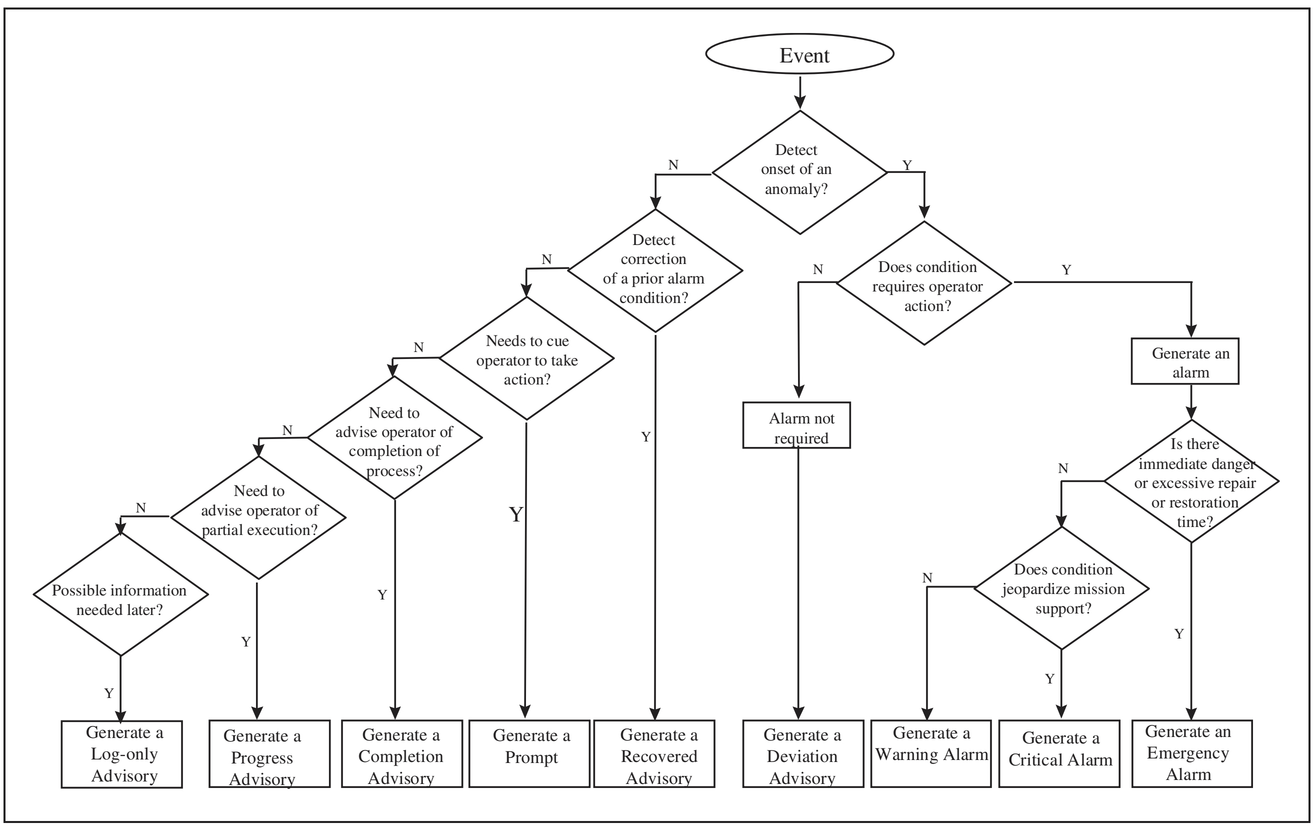

Figure 5-1. Criteria for Generating an Event Notification

5.2.1.1 Alarms

There are three levels of alarms: emergency, critical, and warning. Each of these is described in subsections below.

Subsystems **shall** report, via an alarm, a subsystem anomaly for which the operator is:

(1) expected to take some corrective action, or

(2) to be aware that some corrective action is being carried out automatically.

Subsystems **shall** issue the same alarm only once.

5.2.1.1.1 Emergency Alarm

Subsystems **shall** issue an emergency alarm to report :

(1) any condition that could result in immediate danger to personnel, major equipment, or station facilities, or

(2) an event that could result in costly repairs or excessive restoration time, if not addressed.

An example of an event that should result in an emergency alarm is the loss of the oil film on an antenna-bearing surface.

5.2.1.1.2 Critical Alarm

Subsystems **shall** issue a critical alarm to report any problem that, if uncorrected, will interrupt one or more data streams or otherwise jeopardize mission support.

An example of an event that should result in a critical alarm is the loss of a critical data flow interface.

5.2.1.1.3 Warning Alarm

Subsystems **shall** issue a warning alarm to report any problem that is not critical but requires operator action to correct.

Subsystems **shall** issue a warning alarm if the subsystem observes a series of minor malfunctions foreshadowing a condition that would result in significant loss of data.

As an example, a subsystem should issue a warning alarm when marginally performing equipment needs to be replaced by backup equipment. Another example that should result in a warning alarm is the loss of a backup unit currently not in use.

5.2.1.2 Advisories

Subsystems **shall** report minor malfunctions, changes in status, routine progress, etc., by issuing advisories. Operators need not act upon an advisory. There are five types of advisories: deviation, completion, progress, log-only and recovery.

Subsystems **shall** issue the same advisory only once.

5.2.1.2.1 Deviation Advisory

Subsystems **shall** issue a deviation advisory to report any anomalous condition that, by itself, requires no action by the operator.

An example is a minor malfunction that in itself requires no direct action by the operator. Deviation advisories may show a trend toward trouble that might ultimately jeopardize mission support.

Subsystems **shall** issue an information advisory when a deviation in performance or a transient is observed. The transient need not have affected the performance of the subsystem but should be a noteworthy part of the overall performance record. An example of a performance deviation that results in an information advisory would be reporting an intermittent error in writing on tape.

5.2.1.2.2 Completion Advisory

Subsystems **shall** issue a completion advisory to report the completion of some expected (and therefore non-anomalous) event. A subsystem would typically use a completion advisory to tell an operator that the subsystem has completed a lengthy process that was initiated by the operator.

An example of a completion advisory would be a message informing the operator that an antenna has been repositioned to an elevation previously requested by the operator via a pointing directive.

Also refer to Section 4.2.2 regarding the Started directive response and completion advisories.

5.2.1.2.3 Progress Advisory

Subsystems **shall** issue a progress advisory to report an ongoing, routine (i.e., non-anomalous) activity.

Subsystems **shall** issue progress advisories that clearly indicate whether the action is actually in process or merely in a queue.

Subsystems **shall** issue progress advisories that include completion times (e.g., seconds remaining), when the completion time is accurate and readily available.

Also refer to Section 4.2.2 regarding the Started directive response and progress advisories.

5.2.1.2.4 Log-Only Advisory

Subsystems **shall** issue a log-only advisory to report performance information which is not necessarily needed at the time it is reported, but needs to be available for review at a later time.

A log-only advisory provides an audit trail capability for post-pass analysis and is not normally displayed to the operator in real time.

5.2.1.2.5 Recovered Advisory

Subsystems **shall** issue a recovered advisory to inform the operator when the condition triggering an alarm no longer exists.[1]

[1] The NMC (NSM-5620-OP-C) and the MCIS (DSI-6000-OP-C) do not currently support the Recovered Advisory.

5.2.1.3 Prompts

Subsystems **shall** issue a prompt to gain the operator's attention to inform the operator of a condition that either requires or permits some procedural step.

Prompts are triggered by non-anomalous events occurring within a system or subsystem. An example of a prompt is *Mount Next Tape*.

5.2.1.4 Late Responses

Another category of event notification is the Late Response. This event type, however, is <u>not</u> initiated by the subsystems. Rather, the MCIS issues this type of an event to the NMC when a directive response is received after the time-out period has been exceeded, or when the context in the directive response received by the MCIS is not valid.

5.2.2 Protocol and API

MON-1 describes the protocol and API for event notifications.

5.2.3 Interface Agreement Negotiations

In the cases where positive closed-loop control relationships exist (i.e., an event is associated with monitor data, or a particular directive or display), such relationships **shall** be identified in the NMC–subsystem interface agreement.

5.2.4 Destination of Subsystem Events

Events are sent point-to-point from a subsystem to the target subsystem controller. In general, MON-2 subsystems will send events to the NMC if events must be displayed to the operator or used by Automation, but the subsystems can also send events to other subsystems, if needed.

Connection assignable subsystems **shall** send events to the NMC Complex Supervisor (CS) permanent functional address /fa/<domain>[2]/cs when unassigned, and to the NMC Connection Engine (CE) functional address provided in the configuration control notification (CCN) when assigned.

Non-Assignable subsystems **shall** send events only to the NMC CS permanent functional address.

5.2.5 Availability of Subsystem Events

Any subsystem can subscribe to an event from another subsystem by using the MCIS Event Notification Service (ENS). Refer to the MCIS User's Guide for a description of how the ENS is used.

---

[2] <domain> is the functional domain name assigned to each facility (see DFL-1-7, Sec. 5).

Section 6
Monitor Data

6.1 Introduction

Monitor data are produced by DSN subsystems to describe their status, configuration, and performance. This section establishes the monitor and control standards for monitor data.

The MON-1 monitor data protocol reflects a fundamental shift in paradigm away from that of the DFL-1-2/890-132 protocols, whereby subsystems transmit monitor data, which is largely unsolicited, to a specific target(s). Instead, in the MON-1 environment, a *providing* subsystem makes its data available through the act of publishing, and a *using* subsystem only requests the specific data actually needed through the act of subscribing. Additionally, the concept of polling, which is prevalent in the DFL-1-2/890-132 world, no longer applies in the MON-1 environment. (There are, however, vestiges of polling implemented in MON-1 to allow MON-2 subsystems to obtain data that are available only on-poll from DFL-1-2/890-132 subsystems.)

In this new environment, published monitor data are used not only to support NMC monitor and control, but also subsystem UDS displays and subsystem automation.

6.2 Practices

6.2.1 Monitor Data Classifications.

Subsystem controllers **shall** produce monitor data describing the subsystem's status, configuration, and performance.

Status denotes the operational readiness or the internal functioning of equipment and software. It may apply to an entire subsystem, assembly, subassembly, subsystem function, or other component.

Configuration denotes the internal arrangement, processing mode, or state of an assembly, subassembly, resource, or other component. Configuration is generally expressed as the current settings of variables over which the operator has control.

Performance reflects the functioning of an assembly, subassembly, subsystem function, or other component with respect to the environment in which it operates. Performance will contain measurements of both internal and external quantities as well as variables derived from those measurements.

6.2.2 Content and Format

MON-2 monitor data formats are defined in Appendix B, Section B.1 of this document. (Section B.2 defines the 890-132 monitor data formats; these definitions supersede the less complete list in 890-132.)

6.2.3 Monitor Data Formats

This section describes the acceptable data formats for MON-2 status, configuration and performance monitor data.

Monitor data parameters that report status information **shall** be expressed using the `FORMAT_STATUS` data format, which is described in detail in Appendix B, Section B.1.

Monitor data parameters that report configuration information **shall** be reported as ASCII strings, using the `FORMAT_STRING` data format, with the exception of configuration parameters that contain numeric information. In such latter cases, any of the standard numeric data formats may be used. Configuration information **shall** be encoded such that meaningful human interpretation does not require conversion by the NMC. For example, parameters which are intended to communicate one of a few possible configuration states should generally be encoded as mnemonic ASCII strings.

Monitor data parameters that report performance information as numeric values **shall** use any of the standard numeric data formats. Performance parameters that do not contain numeric information, however, may use any of the other standard data formats.

6.2.4 Design of Monitor Data

The monitor data provided by a subsystem **shall** be designed such that:

(1) overall subsystem status can be described by a single monitor data item. The purpose of doing so is to feed high-level NMC displays, whereby the overall status of a subsystem (including all of its assemblies and subassemblies) can be depicted by a single icon.

(2) visibility into a subsystem can be achieved by successive and ordered views which are increasingly detailed. The objective is to provide diagnostics to accommodate an exceptions-handling strategy for monitor and control. Thus, the general rule is that monitor data should be available to the lowest meaningful component of the subsystem with respect to what can be swapped out or calibrated. Status, configuration, and performance monitor data **shall** be provided to the extent necessary to describe the health and state of the subsystem to this component level.

(3) closed-loop control can be accomplished via monitor data. For example, when a directive is issued to a subsystem, the ‘result’ of this directive (or failure to achieve the result) should be reflected in monitor data. Such closed-loop relationships between directives and monitor data **shall** be described in the “Description” of the monitor data.

(4) critical subsystem information is produced as monitor data.

The MCIS manages monitor data publishing and subscribing using the segments defined in the subsystem Monitor Data Specification File (*MDS mdspecs.dat*). These segments are used by the MCIS to index and distribute subsystem monitor data. As such, the size of a subsystem’s monitor data segments can affect the efficiency with which MCIS handles the monitor data for the subsystem. As a general rule, MCIS manages smaller monitor data segments more efficiently than larger segments. For instance, MCIS will manage monitor data defined in 50 segments of 100 data items each more efficiently than monitor date defined in 5 segments of 1000 items each. (However, there is an overhead associated with each segment so that 5000 separate segments would not be the most efficient arrangement in this example.)

Segments also provide a means for organizing and synchronizing subsystem monitor data. For instance, it may be useful to organize monitor data based upon its source, e.g., organize weather data and equipment status data into separate segments. In order to insure synchronization, associated monitor data items should be included in common segments. For example, position and associated error data should be included in the same segment to insure MCIS synchronizes the update of these data items. Similarly, monitor data with a common time-stamp or monitor data items that are valid for a common time should be included in the same segment.

Finally, the manner in which monitor data is organized into segments can influence the efficiency with which the monitor data is processed by MCIS. For example, grouping monitor data items that change frequently into separate segments from those that change infrequently is more efficient. Therefore, when organizing subsystem monitor data into segments also consider the frequency with which the data will change.

6.2.5 Subsystem Status

Subsystems **shall** accurately report (via monitor data) their current status and state at all times. (See Section 6.3.2.) Thus, all subsystem status or state changes **shall** result in the publishing of the associated monitor data by the subsystem. When uncertain as to the status of the subsystem component, the more critical status value **shall** be reported.

6.2.6 Program/Version ID Standard

In the MON-1 environment, a subsystem's controller program and version IDs provide the basis by which subsystem-specific files are organized and retrieved by the NMC. This also identifies the subsystem when delivering to SPMC.[3] Allowed SPMC program and version ID formats are defined in 813-024, *SPMC Configuration Management Plan*.

The standard defined in Figure 6-1 **shall** be used by a subsystem when publishing program and version IDs for the NMC.

6.3 Required Subsystem Information

This section defines the specific standard monitor data items that a subsystem must produce in order for subsystem to interface with the NMC.

6.3.1 Monitor Data Descriptions

6.3.1.1 MDS Monitor Data Specification File

The MDS Monitor Data Specification file (*MDS mdspecs.dat*) defines the monitor data items published by a subsystem. The NMC and the MDS use this file to accomplish publication of and subscription to subsystem monitor data.

The NMC requires that a *MDS mdspecs.dat* file be delivered for each new subsystem software version. It is the responsibility of the subsystem CDE to produce and deliver this file using the test/delivery procedures defined in Section 12. The subsystem CDE **shall** identify each monitor data item to be published by the subsystem and the format of the data item in the subsystem's *MDS mdspecs.dat* file.

[3] As background, SPMC is responsible for issuing program IDs. A sample program ID might be: NSM-5621-OP. The first release of software is A; thus, the software would be identified by NSM-5621-OP-A. From the SPMC perspective, "NSM-5621-OP-A" constitutes the program ID. However, from the NMC and MDDS perspective, the program ID is "NSM-5621-OP." The "A" is considered part of the version ID.

Program ID and version ID general format:

XXX-####-OP-op#V##.##.##

The program ID — specifically, the PGMID monitor data item (XXX-####-OP) — is the program unique identification provided by SPMC, e.g., DOA-5556-OP.

The version ID — specifically, the VersionID monitor data item (op#V##.##.##) — is a concatenation of the operational revision identifier and the delivered program version numbers, e.g., BV3.0.7 or EV10.14.2. It is provided by the subsystem CDE. The version ID is composed as follows:

<op>[<version>]V<major>.<minor>[.<bugfix>]

where,

`<op>[<version>]`	=	operational revision identifier, e.g., B or E
`V<major>.<minor>[.<bugfix>]`	=	delivered program version, e.g., 3.0.7, 2.0, or 10.14.2

and where,

`<op>`=a capital letter, in the range A–U
`<major>`=major version change, decimal digits
`<minor>`=minor version change, decimal digits
`<bugfix>`=same version, but with bug fixes only, decimal digits

Note that parameters contained within brackets [ ] are optional.

Figure 6-1. Program/Version ID Standard

6.3.1.2 Interface Agreements

All subsystem monitor data subscribed to by the NMC[4] and the functional address(s) under which this data is published **shall** be defined in an interface agreement between the subsystem and the NMC. The monitor data descriptions in the interface agreement **shall** consist of the following information:

(1) data item identifier (name). All data items published under the same functional address must have unique names.

(2) item description

  (a) brief description of the data item.

  (b) identification of any positive closed-loop control relationship(s) (e.g., between the monitor data item and subsystem directives or events).

  (c) identification of any relationship(s) to predicts and/or standards and limits, in the case of performance data.

(3) format.

(4) units/precision.

(5) range of values.

The interface agreement with the NMC **shall** identify all subsystem assemblies and provide the

[4] Note that the NMC may only subscribe to a small subset of all of the monitor data published by a subsystem.

monitor data defined in Section 6.3.2 for the subsystem and each assembly.

6.3.2 Subsystem/Assembly Standard Monitor Data

This section describes the standard monitor data to be published by each subsystem. Note that although monitor data items are grouped into data sets, there is no implication that the data must be published together. Rather, the data sets and their names are strictly used as a logical grouping to facilitate description of the data.

The standard monitor data that should be published by each subsystem are identified in Tables 6-1 and 6-2. The addresses and circumstances under which these data must be published are defined in Table 6-3. (Note that the data identifiers (Data Item Names, in these cases) should be used exactly as shown in Tables 6-1 and 6-2 with upper and lower case letters.)

If a subsystem does not have separately identifiable assemblies, the ***AssemblyList*** data item in Table 6-1 and the data items in Table 6-2 should not be published by the subsystem.

If a subsystem delivers its UDS displays under a separate program ID (i.e., a program ID different than the one specified in Table 6-1), the *DisplayExePath* data item in Table 6-1 should specify the full path to where the UDS display executables are stored on the NMCFS. If a subsystem delivers its UDS displays under the program ID specified in Table 6-1, the *DisplayExePath* data item should not be published by the subsystem.

Connection assignable and antenna group subsystems **shall** publish the monitor data defined in Tables 6-1 and 6-2, as appropriate, under the permanent functional addresses shown in Table 6-3, independent of the assigned mode of the subsystem.

Connection assignable and antenna group subsystems while in an assigned mode **shall** also publish the monitor data defined in Tables 6-1 and 6-2, as appropriate, under the functional addresses provided in the assigned CCN. The general format of these functional addresses is shown in Table 6-3.

A non-assignable subsystem **shall** publish the monitor data described in Tables 6-1 and 6-2, as appropriate, under the permanent functional addresses defined in Table 6-3.

6.3.2.1 Monitor Data Standard for Communicating with DFL-1-2/890-132 Subsystems

For MON-2 subsystems needing to provide monitor data to DFL-1-2/890-132 subsystems, refer to Appendix C, Section C.6 for monitor data required formats that must be followed by the MON-2 subsystem.

6.4 NMC Connection Monitor Data

The NMC will publish the monitor data needed by each MON-2 subsystem to self-configure for a connection. By default, the NMC will publish the standard connection data defined in Table 6-4 for each MON-2 subsystem in a connection. Any additional data or modifications needed to this standard connection data by a subsystem should be specified in the NMC-SS interface agreement.

The standard connection monitor data in Table 6-4 will be published by the NMC at connection start-up and on-change. The NMC will insure that these monitor data are published at connection start-up before any Assign CCNs are issued to connection MON-2 subsystems. Data items 2-6 of Table 6-4 may also be updated by the NMC at <u>any</u> time during the course of a connection. For example, data items ***DssList***, ***ConnSubsystemsList*** and ***SubsystemsFAList*** will be updated whenever an antenna is added to or removed from the connection. (Additionally, it is possible that all of the monitor data items in Table 6-4 will temporarily become ‘uninitialized’ during certain NMC recovery and reconfiguration activities. However, once the NMC recovery/reconfiguration activity has completed, these data items will again become ‘valid’.)

The NMC will publish the standard connection data in Table 6-4 under the NMC functional address specified in the Assign CCNs sent to connection subsystems. Therefore, a connection subsystem does not need a priori knowledge of the NMC functional address under which this data will be published.

A connection subsystem should use the NMC functional address specified in its Assigned CCN to subscribe to the monitor data shown in Table 6-4.

Table 6-1. Subsystem Standard Monitor Data: Controller Identification, State, and Status

Data Set Name: Controller Status and Configuration Information
Published by: Each DSN subsystem controller monitored by the NMC, and running the MON-1 protocol
Under addresses: Appropriate subsystem addresses for the controller (see Table 6-3)
Published at: Start-up; on change

Item #	Item Description	Format	Units/ Precision	Range
1	**PGMID**. Subsystem controller program identification number. (See Figure 6-1) **	STRING	N/A	N/A
2	**VersionID**. Subsystem controller version number. (See Figure 6-1) **	STRING	N/A	N/A
3	**PGMSTA.** Reflects communications mode of the subsystem controller. Possible values are: 1 = unassigned, 2 = assigned, and are set by the CCN.	INTEGER	N/A	1 to 2
4	**Status**. Overall status of the subsystem. Refer to Table B-2 in Appendix B for all possible status for a subsystem, and list in the interface agreements with NMC the ones that apply to the subsystem.	STATUS	N/A	As appropriate
5	**AssemblyList**.* A comma-separated list of subsystem assembly acronyms (in lower case) as follows: • Under the subsystem controller functional address (when assigned), list all subsystems that are configured to be controlled by this subsystem controller. • Under the subsystem controller permanent functional address, list all the subsystem assemblies independent of the assignment state.	STRING	N/A	N/A
6	**DisplayExePath**.* The full NMCFS path to the subsystem's UDS display executables. (See Appendix H.)	STRING	N/A	N/A

* Optional data item. Publish only if appropriate for the subsystem. See Section 6.3.2.

** If a separate program identification is not available for the subsystem controller, use the subsystem program identification instead.

Table 6-2. Subsystem Standard Monitor Data: Assemblies Identification

Data Set Name: Assemblies Identification and status
Published by: The subsystem controller on behalf of the subsystem's assemblies
Under address: Appropriate subsystem address for each assembly (see Table 6-3)
Published at: Start-up; on change

Item #	Item Description	Format	Units/ Precision	Range
1	**ProgramVersionID**. program/versionID of one the assemblies identified in AssemblyList defined in Table 6-1. Concatenate the program ID and the version ID ** (see Figure 6-1) (example: MHR-0000-OPBV3.0.7)	STRING	N/A	N/A
2	**Status**. Overall status of this assembly.	STATUS	N/A	As appropriate

** If a separate program identification is not available for the assembly, use the subsystem controller ID, if available, or the subsystem identification.

Note: The same Monitor Data Specification file can be used to publish these data for all assemblies.

Table 6-3. Subsystem's Standard Functional Addresses

Publishing Address	When Used
Functional Addresses	
/fa/<domain>/con<n>-<resource name> and /fa/<domain>/con<n>-<resource name>/<assembly name>	*Connection assignable* and *multi-connection subsystem*: The subsystem controllers that are not hard wired to the Antenna publish monitor data under these addresses, when they are assigned to a connection, for themselves and on behalf of their assemblies.
/fa/<domain>/dss<mm>-<resource name> and /fa/<domain>/dss<mm>-<resource name>/<assembly name>	*Antenna Group subsystems*: Assigned Antenna Group subsystem's controllers publish monitor data under these addresses, when they are assigned to a connection, for themselves and on behalf of their assemblies.
Permanent Functional Addresses	
/fa/<domain>/<resource name><i> and /fa/<domain>/<resource name><i>/<assembly name>	*A connection assignable* and *multi-connection subsystem:* The subsystem's controllers publish monitor data under these permanent functional addresses at all times, for themselves and on behalf of their assemblies.
/fa/<domain>/<resource name><mm><i> and /fa/<domain>/<resource name><mm><i>/<assembly name>	*Antenna Group subsystem:* Antenna Group subsystem's controllers publish monitor data under these permanent functional addresses at all times, for themselves and on behalf of their assemblies.
/fa/<domain>/<resource name> and /fa/<domain>/<resource name>/<assembly name>	*Non-assignable subsystems:* These types of subsystem's controllers continuously publish monitor data under these permanent functional addresses, for themselves and on behalf of their assemblies.

Where:

<n>	is the connection number that the subsystem is supporting
<mm>	is the Deep Space Station (DSS) number used for the antenna
<domain>	is the functional domain name assigned to each facility (see DFL-1-7, Sec. 5)
<resource name>	is the official subsystem acronym by which it is known within DSN
<assembly name>	is the official assembly acronym by which it is known within DSN
<i>	is a letter or number used to distinguish between redundant assemblies

Table 6-4. NMC Standard Connection Data

Data Set Name: Standard Connection Data
Published by: NMC Connection Engine (CE)
Under functional address: As specified in the subsystem Assign CCN
Published at: Connection start-up; on change

Item #	Item Description	Format	Units/ Precision	Range
1	**LinkNumber**. The link number of this connection.	INTEGER	N/A	1 to 8
2	**ConnSubsystemsList**. A comma-separated list of the assigned subsystem controllers in the connection. Subsystem controllers are defined via their Directive Destination Code (DDC).	STRING	N/A	N/A
3	**SubsystemsFAList**. A comma-separated list of the functional addresses of the subsystem controllers in the connection. Same order as the ConnSubsystemList. Valid functional addresses are listed in Document 826, *DSMS/System Configuration Files*.	STRING	N/A	N/A
4	**DssList**. A comma-separated list of the DSS number(s) of the antenna(s) currently in the connection. Valid DSSs are listed in 810-047, Table 2-2.	STRING	N/A	Individual DSS number range: 0 to 99
5	**SpacecraftNumber**. A comma- separated list of the spacecraft number(s) in decimal, currently supported by the connection. Valid spacecraft numbers are listed in Document 826, *DSMS/System Configuration Files*	STRING	N/A	Individual spacecraft number range: 0 to 255
6	**PassNumber**. A comma-separated list of the pass numbers for the spacecraft in the connection. Same order as the SpacecraftNumber list.	STRING	N/A	Individual pass number range: 0 to 9999

Section 7
Subsystem Displays

7.1 Introduction

Subsystem displays constitute the primary mechanism for providing operators with visibility into the subsystem and its assemblies. This section describes the various types of subsystem displays that are supported by the NMC and how these displays are made available to the NMC.

7.2 Practices

7.2.1 Types of Subsystem Displays

Subsystem displays **shall** adhere to the MON-3 standard. To facilitate development of such displays, subsystem developers can use the Uniform Display Service (UDS) (described in MON-7 and the UDS User's Guide) which provides developers with basic X/Motif display capabilities, while providing some insulation from requiring X/Motif knowledge.

The UDS allows subsystem developers to create MON-3 compliant displays which execute on the NMC Workstation. The UDS provides an application programmer's interface (API) for creating subsystem displays, integrating the displays with the NMC Workstation software, and subscribing to and displaying subsystem monitor data. Each UDS display is a separate executable program built using the UDS library. When invoked on a NMC Workstation, a UDS display automatically subscribes to the subsystem monitor data needed to populate the display.

While the use of UDS is recommend, subsystem developers may still produce displays that make direct calls to the X library, particularly if a broader range of features is needed than is provided by the UDS.

7.2.1.1 Display Classifications

The NMC can support the following types of displays (see Section 7.3):

(1) "UDS" (Uniform Display Service)

(2) "PSI". Legacy subsystem displays refer to the 890-132/133 displays. The NMC still supports these displays, but from DFL-1-2/890-132 subsystems only. However, MON-3 displays can be built for DFL-1-2/890-132 subsystems, the only prerequisite being that the subsystem must be generating the monitor data needed to populate the displays.

(3) "URL." HTML displays are developed by subsystem personnel. They can be invoked from within or without a UDS display. HTML displays invoked from the Help Menu within UDS displays are the subsystem help files.

7.2.2 User Interface Standards

MON-3 establishes the standards for such topics as navigation and organization of displays, types of displays required [e.g., configuration (CNF)], display design, and standard display names. It also establishes the requirement for on-line help facility for displays, and sets general guidelines for a help facility.

MON-3 also sets forth a standard for subsystem displays wherein all subsystem displays can be accessed from any other subsystem display through the Displays menu.

Subsystem displays **shall** be associated with an identifier that is unique among displays for that particular subsystem. As referred to earlier, MON-3 contains a list of standard names for standard subsystem displays.

Subsystem displays are developed by subsystem personnel and are owned by the subsystems. Displays incorporating monitor data from multiple subsystems and developed by NMC personnel are owned by the NMC.

7.2.3 Display Help Facility

As previously mentioned, MON-3 establishes a requirement for an on-line help facility for directives and displays and sets general guidelines for developing a help facility.

7.3 Subsystem-Provided Information

This section describes the information associated with displays that must be made available in order for subsystems to interface with the NMC. As such, it addresses display descriptions and the content and format of the display file needed by the NMC.

7.3.1 NMC Display List File

The NMC Display List File contains the name of all displays recognized by the subsystem controller together with the NMC required and optional attributes for the displays. The NMC uses this file to present a menu of displays for the subsystem.

The NMC requires that a NMC Display List File (aka a subsystem's *disp.dat* file) be delivered for each subsystem software version. It is the responsibility of the subsystem CDE to produce and deliver this file using the test/delivery procedures defined in Section 12. The subsystem CDE **shall** provide the information listed in Table 7-1 for each subsystem display, as applicable.

Table 7-1. Display Entry Keywords

Keyword	Valid Options	Description
DSP=	The display name/mnemonic in upper case letters.	The mnemonic of the display.

Keyword	Valid Options	Description
DSPPARM=	List of parameters for the display, if any.	List of parameters separated by white spaces. The parameters can be anything as long as they are recognized by the subsystem/ assembly. If omitted, or nothing appears after the " = ", then the controller needs to enter the parameters in the command line before sending the directive, or the directive is presumed to have no parameters. The supplied parameters can also be changed by the operator before issuing the directive. Example: DSPPARM=-iconic -display :0.1 -bg blk
DSPPROG=	File name/program name	This is a UDS file name to execute when DSPTYPE=UDS (see MON-7 for more detail on where the binaries should be located).
DSPPROGPARM=	Additional list of parameter(s) for the program, if any.	Additional parameters supplied for the command line beyond -conn and -nmc, when invoking the UDS display.
DSPTYPE=	UDS, PSI, URL (Defaults to UDS, if not provided)	Defines the type of this display: UDS, PSI, HTML.
DSPSS=	The category of this display, in uppercase. (See Appendix E to identify the category(s) that apply to your subsystem)	This categorization is used by NMC to organize various display elements including icons and menus. For example, a Receiver (RCV) or Transmitter (TXR) display.
DSPCONN=	TRUE, FALSE (Defaults to TRUE, if not provided)	TRUE for connection-oriented displays, otherwise FALSE. An example of a non-connection-oriented display might be a facility-oriented one, such as a DSCC Media Calibration Subsystem (DMD) Display.
DSPREMOTE=	TRUE, FALSE (Defaults to FALSE, if not provided)	Set to "FALSE" when the display should not be invoked from a remote site.
DSPMODE=	OP, TEST, MAINT, DBUG	Defines the execution mode(s) for which the display is intended, i.e., the display is intended for use in the operational, test, maintenance and/or debug mode(s). List all of the modes that apply to this display, separated by commas. (A white space can be added after the comma, if desired.)

Keyword	Valid Options	Description
DSPPHASE=	PRETK, INTK, POSTTK, NOTK, CONFIG, CALIB, SUPDAT, STSPER, ACTEND, DIAG.	Defines the execution phase(s) of a connection for which the display is intended, i.e., the display is intended for use in the pre-track, track, post-track, configuration, calibration, support data, status/performance, end of activity or diagnostics phase(s). List all of the phases that apply to this display, separated by commas. NOTK relates to a subsystems display that is not connection-specific. (A white space can be added after the comma, if desired.)
DSPMOU=	YES, NO	Defines whether the display is frequently used by an operator. (The intent is to identify the most often used displays of a subsystem.) For a display that is often used set to YES, otherwise NO.
DSPDESC=	Description of display	Short, one sentence description of the display. (White spaces are allowed.)
DSPSSPEC	TRUE, FALSE (Defaults to TRUE, if not provided)	Identifies if the display is subsystem-specific, i.e., it displays data from only one subsystem. This value defaults to TRUE, and will probably only be FALSE for some NMC displays.
END		Ends the parameters for the described display.

Sample entries in a subsystem's display data, or `disp.dat` file are shown below in Figure 7-1. Note that the following guidelines apply to the file.

(1) Keywords (and associated values) listed in Table 7-1 are used to describe each subsystem display.

(2) Each line begins with a display entry keyword, and ends with a carriage return. (Note that lines beginning with "#" are comments.) There will be only one keyword per line.

(3) The values associated with each keyword are in uppercase, except for those associated with DSPDESC, DSPPARM, DSPPROGPARM, DSPPROG, and the comments, which can use uppercase and lowercase.

(4) White spaces are not allowed unless otherwise noted.

(5) If an optional keyword is not needed, it should not be entered.

(6) Keywords DSP, DSPTYPE, DSPSS, and END are required.

(7) The display categorization(s) (DSPSS) defined in a subsystem's `disp.dat` file must agree with the categorization(s) specified for the subsystem in the NMC Resource File (NRF) for the subsystem. (See the NMC SOM for a description of the NMC NRFs.)

The following is an example of a display data file:

```
#**************************************************************
#*
#*  This NMC Display List File created
#*  on Thu Jan 29 09:29:18 PST 1998
#*
# COLLECTION_NAME : JeaneMON-2
# PROGRAM_ID : JGS-0000-OP
# ACRONYM : FYI
# VERSION_NUMBER : AV1.0
# DATA_TYPE : test
# MDSPECS_EXISTS : TRUE
# DISPS_EXISTS : TRUE
# DIRECTIVES_EXISTS : TRUE
#**************************************************************

DSP=HDIR
DSPSS=RNG
DSPMOU=YES
DSPTYPE=PSI
DSPMODE=OP
DSPCONN=TRUE
DSPPARM=RNS3,1099
DSPPROGPARM=
DSPPHASE=PRETK, INTK, POSTTK
DSPDESC=DIRECTIVES QUICK REFERENCE HELP DISPLAY
DSPREMOTE=TRUE
DSPSPEC=TRUE
END

DSP=CNF
DSPSS=TRK
DSPMOU=YES
DSPTYPE=PSI
DSPMODE=OP
DSPCONN=TRUE
DSPPROG=
DSPPARM=
DSPPROGPARM=
DSPPHASE=PRETK, INTK, POSTTK
DSPDESC=CURRENT MDA CONFIGURATION DISPLAY
DSPREMOTE=TRUE
DSPSPEC=TRUE
END
```

Figure 7-1. A sample `disp.dat` File

Section 8
Support Data

8.1 Introduction

Subsystem support data consist of predictions files (commonly referred to as "predicts"), configuration tables, and standards and limits tables.

Predictions are used by DSCC subsystems to properly acquire and track a spacecraft. Predicts data will generally contain frequencies, pointing angles, and/or estimated signal strengths.

Configuration tables are used by DSCC subsystems for configuration purposes. They include data that may be spacecraft-specific, subsystem/spacecraft-specific, or site-specific, etc.

Standards and limits are used by DSN subsystems to assess both the performance of the equipment and the quality of the signals received by the equipment. Standards and limits data contain expected values and tolerances for various parameters.

8.2 Support Data Practices

The Network Support Subsystem (NSS) produces and/or configuration manages the support data needed by DSN subsystems. The specific support data produced and/or managed by the Network Support Subsystem (NSS) for a subsystem is defined in the individual interface agreement between the NSS and the subsystem.

For MON-2 subsystems, the NSS forwards support data as data files to the Support Products Provider Assembly (SPPA) located at each site. The SPPA automatically distributes each support data file to the appropriate MON-2 subsystem(s) upon its arrival at the site. The standard practices for the distribution of support data to MON-2 subsystems are defined in the following subsections.

8.2.1 Support Data Processing

A 'push' philosophy is used to distribute support data to MON-2 subsystems, i.e., support data files are pushed to MON-2 subsystems without solicitation from the subsystems. Under this approach, a support data file produced/managed by the NSS is sent to the SPPA at the appropriate site. The SPPA detects that a new file has arrived and determines what to do with the file based upon an internal configuration file. The SPPA then automatically distributes the file to MON-2 subsystem(s) as specified in its configuration file, records the distribution in its internal log, and archives the file.

A MON-2 subsystem **shall** receive support data files from the SPPA in a predefined input directory on a predefined host. It is the responsibility of the MON-2 subsystem CDE to coordinate with the SPPA to define the subsystem host (IP address) and the input directory structure into which the SPPA will distribute support files for the subsystem. Support data files should typically be loaded onto a subsystem's controller host.

All MON-2 subsystems **shall** automatically process support data files stored in their 'input' directory by the SPPA. This automatic processing **shall** include parsing of the file name, verification of the file contents and format, storage of the file in the appropriate 'user' directory(s), the handling of error conditions, and the periodic purging of the support files from both the subsystem ‘input’ and ‘user’ directories. Support files stored in a subsystem’s ‘input’ directory should only be opened with a read/write privilege to prevent the subsystem from attempting to open a file that is currently being written by the SPPA.

MON-2 subsystems **shall** automatically verify support data files to the extent possible upon their arrival at the subsystem. As a minimum, MON-2 subsystems **shall** verify that the file is readable and is complete, i.e., that the last line of the file is ‘*=END=*’. If an error is found during file verification, a MON-2 subsystem **shall** send a warning alarm event message to the NMC and delete the file from the subsystem ‘input’ directory.

The SPPA at each site maintains an archive of support data files previously distributed to the subsystems at the site. As a result, an operator can direct the SPPA, via the SPPA user interface, to retransmit one or more support data file(s) to a subsystem. This allows an operator, for example, to retransmit a file that was corrupted during the initial transmission or to ‘rebuild’ the files available at a subsystem due to a subsystem failure in which all of the original data files were lost.

If a required support data file is unavailable at connection configuration or if there is an ambiguity in which file to use (e.g., more that one version of a file is available), a MON-2 subsystem **shall** issue a critical alarm event message to the NMC. Any automated subsystem configuration should be halted at this point pending further direction from the NMC. MON-2 subsystems **shall** provide subsystem directives which allow an operator to manually identify/overwrite the specific support data file to be used by the subsystem.

MON-2 subsystems **shall** periodically purge support data files from their ‘input’ and ‘user’ directories based upon purge dates provided in each file’s file name.

8.2.2 File Naming Conventions

A standard file naming convention is used to construct the names of MON-2 subsystem support data files. This naming convention encodes within the file name itself sufficient information to allow a MON-2 subsystem to automatically process a support data file, e.g., information which defines the type of support data contained within the file, the intended use of the support data, and when the data should be purged. At the same time, the naming convention is sufficiently intuitive to also allow a human operator to interpret the encoded data. This file naming convention is used for both pass and non-pass related MON-2 support data files.

Each MON-2 support data file name is composed of ten (10) mandatory fields. The template used to construct the file names is as follows:

<1>_sn<2>_sr<3>_sc<4>_dss<5>_pn<6>_cl<7>_ty<8>_py<9>_pd<10>

where <> denotes a mandatory entry whose allowed values are defined in Table 8-1. All fields are required and remain in the order shown. All <> entries (except the last) are delimited by an ‘_’. All ASCII characters may be lower or upper case. Additionally, a <> entry may have an associated prefix which is defined below:

sn - Set Name
sr - Set Revision
sc - Spacecraft
dss - Deep Space Station
pn - Pass Number
cl - Support Data Class
ty - Support Data Type
py - Purge Year
pd - Purge Date

Table 8-1. Allowed Support Data File Name Entries

ENTRY < >	FIELD DESCRIPTION	FORMAT	RANGE
1	File Type. Type of data contained in the file.	1 ASCII alphabetic character	p = pass-related; n = non-pass related; a = administrative.
2	Set Name (sn). Unique identifier for the data set.	2-8 ASCII alphanumeric characters	For pass-related data: 2 ASCII character Franz Code (AA to YZ, ZZ reserved for testing). For non-pass data: Up to 8 ASCII characters.
3	Set Revision. Revision indicator for this data set. Subsequent revisions should be in ascending order. Examples: null – original version, A- revision 1, B-revision 2, etc.	1 ASCII alphabetic character	A-Z
4	Spacecraft Number. The decimal number of the spacecraft for which the data are applicable.	1-3 ASCII numeric characters	As defined in Document 826, *DSMS/System Configuration Files*. Will contain at least one ASCII character which will be the ASCII character zero for data that are not spacecraft specific.
5	DSS Number. Number of the antenna for which the data are applicable.	1-2 ASCII numeric characters	As defined in 810-047. Will contain at least one ASCII character which will be the ASCII character zero for data that are not antenna specific.
6	Pass Number. Spacecraft pass number for which the data are applicable.	1-4 ASCII numeric characters	0-9999. Will contain at least one ASCII character which will be the ASCII character zero for data that are not pass specific.

ENTRY < >	FIELD DESCRIPTION	FORMAT	RANGE
7	Support Data Class. Defines the type of support data.	2 ASCII alphanumeric characters	As defined in the subsystem to NSS interface agreement.
8	Support Data Type. Defines the subsystem for which the data is intended.	1-2 ASCII alphanumeric characters	As defined in the subsystem to NSS (SS-NSS) interface agreement.
9	Purge Year. Last year to retain the data set.	1 or 4 ASCII numeric characters	0-9999. Will be set to the ASCII character zero for data that should not be purged.
10	Purge Day-of-Year (DOY). Last day of year to retain the data set.	1-3 ASCII numeric characters	0-366. Will be set to the ASCII character zero for data that should not be purged.

8.2.3 Support Data Handling – Future Plans

While it is anticipated that the exact mechanisms for producing and for distributing support files to MON-2 subsystems will change and evolve with the introduction of the Service Preparation Subsystem (SPS), the ‘push’ approach for distributing support data files is planned to remain unchanged. Additionally, the file naming convention described above is planned to remain unchanged. As a result, it is anticipated that no MON-2 subsystem changes will be required as the means of production and distribution of support data files evolve with the introduction of the SPS, i.e., a MON-2 subsystem’s support files will still ‘show up’ in the same subsystem input directory with the same name.

Section 9
Printed Reports

9.1 Introduction

Some DSN subsystems produce reports available for printing.

9.2 Practices

9.2.1 Report Directives

Subsystems are to be able to print reports in response to a directive from the NMC.

9.2.2 User Interface Guidelines

Refer to MON-3 for report layout guidelines.

9.2.3 Printer Selection and Control

Printers, including a label printer, are available at each DSCC. Unix remote printer commands **shall** be issued by the subsystem software to select and control where and how hardcopy reports will be output.

9.2.4 Printing Requests Flow Diagram

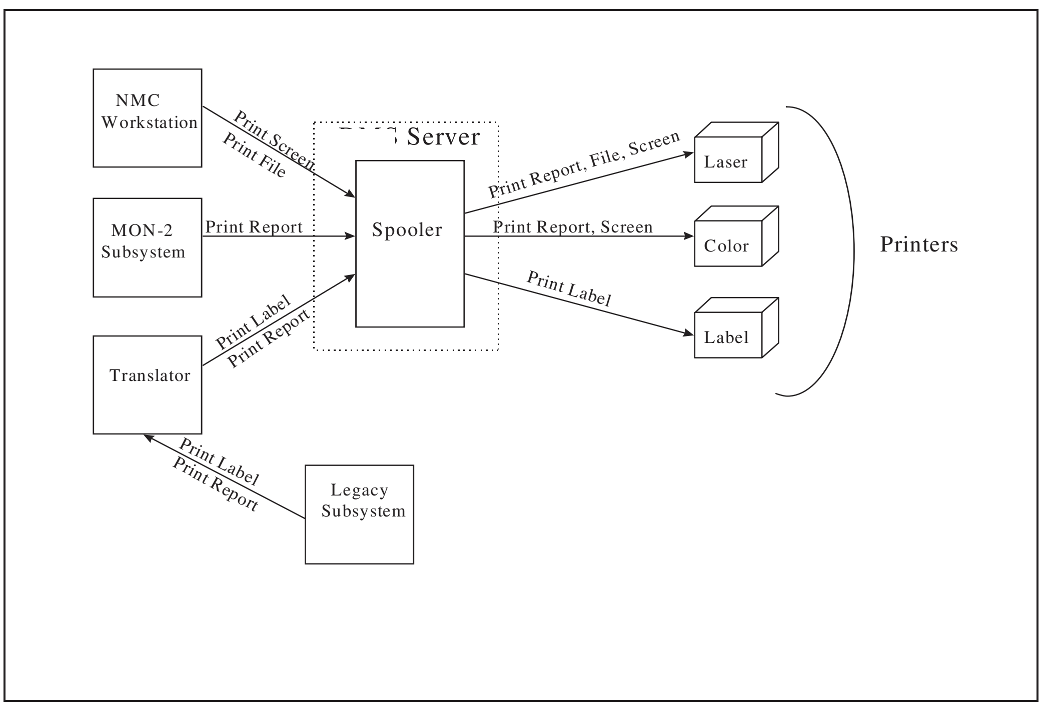

Figure 9-1. Printing Requests Flow Diagram

9.2.5 *Printer Naming Conventions*

Each printer and its page orientation will have a standard name that will not change from site-to-site.

The common names will be as follows:

Laser Printer Portrait	lp1-p
Laser Printer Landscape	lp1-l
Color Printer Portrait	lp2-p
Color Printer Landscape	lp2-l
Label Printer	lp3

Section 10
Documentation

10.1 Introduction

There are two subsystem software documents of interest to the NMC: (a) interface agreement, and (b) Software Operator's Manual (SOM). The content needed by NMC in the interface agreement is described below. The SOM preparation guide is document 813-110, *Preparation of Operator's Manuals for DSN Subsystems*.

10.2 NMC-Subsystem Interface Agreements

For the convenience of the reader, this section summarizes interface agreement practices that have been described elsewhere in this document:

(1) Section 3 Assignment

(2) Section 4 Directives/Responses

(3) Section 5 Event Notifications

(4) Section 6 Monitor Data

(5) Section 7 Subsystem Displays

Interface agreements are provided to the DSMS Cognizant Interface Engineer, and are available on-line on the DSMS Interface Server at URL: `http://jaguar.jpl.nasa.gov`. Guidelines for producing interface agreements are provided in 813-109 *Preparation Guidelines and Procedures for Deep Space Mission System (DSMS) Interface Specifications*.

10.2.1 General

Interface agreements **shall** match as-built software. Thus, interface agreements **shall** not list monitor data in the interface agreements, if the data are not actually published.

Interface agreements **shall** be identified with a specific program ID. The software program/version ID associated with an interface agreement is only tracked to that portion of the ID that precedes the "V". Any modifications to the interface agreement that are associated with software versions differentiated by the numbering following the "V" in the program/version ID should be identified within the body of the document.

10.2.2 Functional Addresses

The interface agreement **shall** identify all functional address(s) and permanent functional addresses by which the subsystem will be identified, and under what conditions each will be used (e.g., whether assigned or unassigned).

10.2.3 Assignment

Subsystems **shall** provide the following information in their interface agreement with the NMC:

(1) Identification of the subsystem's assignment type, i.e., connection assignable, multi-connection or non-assignable subsystem. (Refer to Section 3.)

(2) How swaps and reboots are to be performed for the subsystem (e.g., whether a backup needs to be assigned at the time of a connection or when the main controller fails).

10.2.4 Directives/Responses

For cases where a positive closed-loop control relationship exists (i.e., specific monitor data can be used to verify the successful execution of a directive), such relationships **shall** be identified in the NMC–subsystem interface agreement.

Subsystem directives utilized by NMC automation **shall** be identified in the subsystem interface agreement with the CDSCC Center For Automation.

10.2.5 Event Notifications

For cases where a positive closed-loop control relationship exist (i.e., an event is associated with monitor data, or a particular directive or display), such relationships **shall** be identified in the NMC–subsystem interface agreement.

Subsystem events utilized by NMC automation **shall** be identified in the subsystem interface agreement with the CDSCC Center For Automation.

10.2.6 Monitor Data

All subsystem monitor data published by an assembly and subscribed to by the NMC **shall** be defined in an interface agreement between the subsystem and the NMC. The monitor data descriptions in the interface agreement **shall** consist of the following information:

(1) Data item identifier

(2) Brief description

(3) Format

(4) Units/precision

(5) Range of values

(6) Identification of positive closed-loop control relationships (e.g., between monitor data and directives or events)

(7) Relationship to predicts and standards and limits (files), in the case of performance data

(8) Association between a Parameter Descriptor Word (PDW), if any, and specific monitor data items (for 890-132 subsystems)

10.2.7 Subsystem Displays

Displays utilized by NMC automation **shall** be identified in the subsystem interface agreement with the CDSCC Center For Automation.

10.2.8 Printed Reports/Tape Labels

No information on printed reports and tape labels are required in interface agreements.

Section 11
Testing Practices

11.1 Overview of DSN Testing

Typically, subsystem testing within the DSN proceeds through the following phases: unit testing, integration testing, pre-acceptance testing, and acceptance testing. Although each element and task varies, the phases are applied at the assembly, subsystem, and system levels. There is also the DSN SOAK test that is the responsibility of operations, but may involve support from element personnel in the development organization.

Unit testing includes subassembly and assembly testing. Pre-acceptance testing is done at the major assembly, subsystem, or system levels. It is typically held outside the development environment in the Development and Test Facility (DTF-21), or perhaps at a DSCC. Element personnel are responsible for running the pre-acceptance tests, which are considered to be a “dress rehearsal” for acceptance testing. In contrast, acceptance tests are run by operators in the DSCC environment.

Guidance on DSMS testing standards can be found in 813-112, *DSMS Testing Standards and Guidelines*.

11.2 Integration Testing with the NMC

Subsystem integration with the NMC often begins during a subsystem’s pre-acceptance testing, although it can begin as early as the subsystem’s own integration testing. Prior to integration testing with the NMC, the subsystem/assembly should have completed its own internal integration.

Prior to testing with the NMC, the subsystem CDE must produce/update the subsystem’s monitor, directive, and display data definition files for the specific software version interfacing with the NMC. The specific procedures for delivering and testing these data definition files with the NMC vary depending upon the location and type of testing being performed (See Section 12).

Subsystems **shall** specifically perform integration tests with the NMC and with NMC Automation for each subsystem re-delivery.

Subsystems **shall** demonstrate that subsystem test procedures address compliance with the “shall” statements in this document. A summary of all “shall” statements in this document is provided in Appendix F.

11.3 General Checklist for Testing

(1) CCNs

- (a) Transitions to the communication mode corresponding to the CCN type.
- (b) Subsystem behavior in each communication mode adheres to the constraints identified in Section 3.2.

(2) Directives/Responses

(a) Ensuring that safety measures are in place with regard to directives that have the potential of endangering human life or causing equipment damage.

(b) Messages in the directive responses adhere to the guidelines stated in Section 4.2.1.

(a) Documentation of all close-loop control relationships between directives, events and/or monitor data, i.e., the accomplishment of all directives controlling subsystem status and configuration are indicated in subsystem monitor data and/or event messages.

(c) Adherence to the conditions for issuing the various types of directive responses, and within the specified time constraints.

(d) Adherence to MON-3 standards, such as for directive syntax and directive naming conventions and content of the message.

(e) Complete documentation and description of directives in SOM and interface agreement.

(f) A subsystem *directives.dat* file has been produced/updated and is available for testing. (See Section 12.)

(3) Event Notifications

(b) Adherence to the conditions for issuing the different types of events.

(c) Documentation of all close-loop control relationships between events and monitor data.

(4) Monitor Data

(a) Monitor data have valid formats and are consistent with the monitor data classifications.

(b) Monitor data are available to accommodate the monitor by exception (see Section 2.3) capability (e.g., single monitor data item describing overall status of a subsystem) and self-evaluation, bubble-up displays, and positive closed-loop control (e.g., correlation of monitor data to alarms and directives).

(c) Monitor data consistently and accurately reflects the subsystem conditions.

(d) Status words are associated with the correct color, in accordance with MON-3.

(e) All monitor data listed in the interface agreement with the NMC are in fact published.

(f) All communications between the NMC and the subsystem are via the subsystem controller only.

(g) Program/version IDs adhere to the DSN standard.

(h) A MDS Monitor Data Specification file (*mdspec.dat* file) has been produced/updated and is available for testing. (See Section 12.)

(i) Standard subsystem monitor data defined in Section 6.3.2 are published.

(j) Adherence to the standards defined in Appendix C, if communicating with DFL-1-2/890-132 subsystems.

(5) Subsystem Displays

(a) Adherence to MON-3 display standards.

(b) Full documentation of displays in SOM.

(c) Subsystem NMC Display List file (*disp.dat* file) has been produced/updated and is available for testing. (See Section 12.)

(6) Printed Reports

(a) Printing of reports in accordance wth Section 9.

This page is intentionally left blank.

Section 12
SDD/UDS File Delivery Procedures

12.1 Introduction

The subsystem CDE is responsible for developing, testing, and delivering the subsystem data files needed to support the MON-1 publish/subscribe paradigm. This includes specification files needed to subscribe to subsystem published data, data files needed to support the NMC user interface, and subsystem UDS display files to be executed and displayed on the NMC Workstation. Since these data files must be accessible by other MON-2 subsystems, they are stored in a 'shared' directory structure on the NMC File System (NMCFS) (See Appendix H). This section describes procedures for delivering these data files.

12.2 Subsystem Data Definition Files

Each DSN subsystem that interfaces with the NMC is required to provide the following Subsystem Data Definition (SDD) files:

(1) MDS Monitor Data Specification File (*MDS mdspecs.dat*)

(2) Directive List File (*directives.dat*)

(3) Display List File (*disp.dat*)

(4) UDS Help File

The Subsystem Data Definition (SDD) files are subsystem-version sensitive and **shall** be re-delivered with each new version of subsystem software. This means that a subsystem's SDD files must be re-delivered with each new delivery of subsystem software, whether or not the subsystem's monitor, directive or display data has changed. It is the responsibility of the subsystem CDE to develop these data definition files for each software version of the subsystem using the procedures described below.

12.3 Subsystem UDS Display Files

Subsystems may produce UDS displays to be executed and displayed on the NMC Workstation. If a subsystem utilizes UDS displays, it is the responsibility of the Subsystem CDE to deliver the subsystem's UDS display files using the procedures described below.

12.4 SDD and UDS Files Delivery Procedures

A variety of means are available to the Subsystem CDE to move updated SDD and/or UDS files into a test or operational environment. The specific procedures to be used vary depending upon whether the files are needed for testing at DFT-21, for testing at a DSCC, or for a formal delivery to SPMC.

12.4.1 Procedures for DTF-21 Testing

The Subsystem CDE may directly move updated subsystem SDD and/or UDS files onto the NMCFS at DTF-21. The procedures to be used are illustrated in Figure 12-1 and are described below.

1. The Subsystem CDE places updated SDD and/or UDS files into a designated area of AFS.
2. The Subsystem CDE logs onto a subsystem host in DTF-21 and copies the updated files from AFS into a designated directory on the subsystem host. (The directory structure on the sub-system host must be the same as the subsystem directory structure on the NMCFS at DTF-21.)
3. Alternatively, the Subsystem CDE may use a physical media to transfer updated SDD/UDS files onto the subsystem host at DTF-21.
4. The Subsystem CDE copies the updated files into the appropriate directory structures on the NMCFS at DTF-21 by executing a NMC provided script.

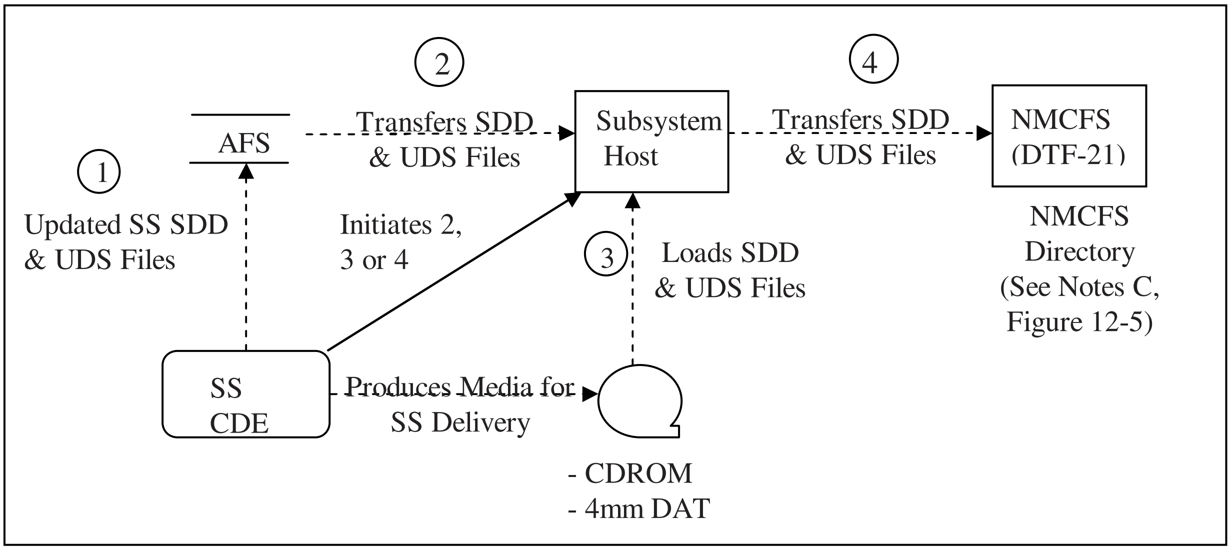

Figure 12-1. SDD/UDS Files Delivery Procedures – Test Delivery at DTF-21 Only

12.4.2 Procedures for DSCC Engineering Tests

The NMCFS at the DSCCs (and the NOCC) are under site configuration control, therefore the Subsystem CDE must coordinate the update of subsystem SDD and/or UDS files in order to perform engineering tests at these sites. The procedures to be used are illustrated in Figure 12-2 and are described below:

1. The Subsystem CDE places updated SDD and/or UDS files into a designated area of AFS.
2. Since the subsystem's SDD and/or UDS files will be loaded onto the operational NMCFS at the NOCC and at the test site, the Subsystem OE must approve the installation of these files on the NMCFS. Once the Subsystem OE concurs, the Subsystem CDE requests the NOCC NMCFS System Administrator (SA) to copy the updated files from AFS into the NMCFS at NOCC.
3. The updated files are then transferred from the NOCC NMCFS to the specific site(s) at which the testing will be performed. The updated files may be 'pushed' to the test site by the NOCC NMCFS SA or may be 'pulled' to the site by the Site NMCFS SA depending upon the

preference of the site.

4. Alternatively, if the subsystem employs a physical media for software delivery, the Subsystem CDE includes the updated SDD and UDS files with other subsystem software, produces the media, transfers the media to the site(s) involved, and requests the test site's NMCFS SA to load the subsystem's SDD/UDS files onto the site's NMCFS per directions in the subsystem RDD.

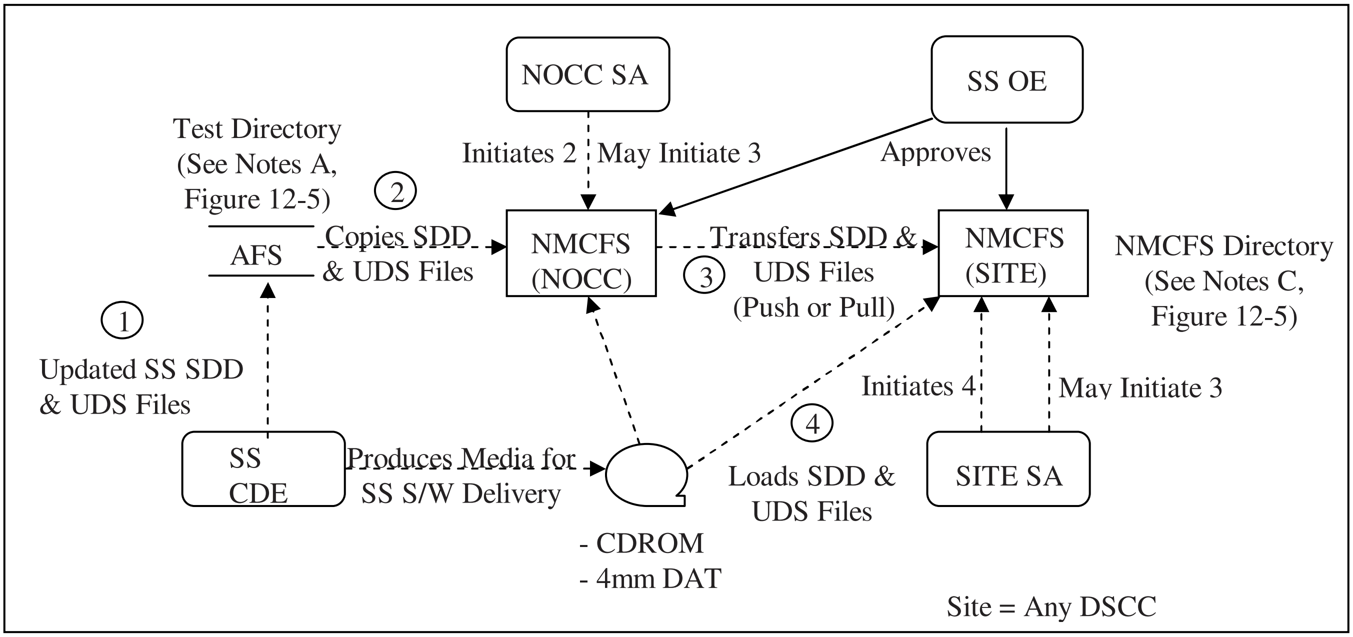

Figure 12-2. SDD/UDS Files Delivery Procedures –DSCC Engineering Tests

12.4.3 SPMC Delivery Procedures

The formal testing and delivery of subsystem software requires the involvement of SPMC to maintain configuration control of subsystem software, including subsystem SDD and UDS files. Typically, SPMC is responsible for building subsystem software from the pre-acceptance phase through subsystem delivery. The procedures to be used for the formal testing and delivery of subsystem SDD and UDS files are illustrated in Figures 12-4 and are described below:

1. The Subsystem CDE moves the updated subsystem's SDD and UDS files into the appropriate AFS area for a SPMC build for the subsystem. This will typically be done as part of the overall build process for the subsystem, i.e., the Subsystem CDE will place the subsystem's software, including the SDD and UDS files for the subsystem, into the AFS for a SPMC build.

2. The Subsystem CDE requests SPMC to build a version of the subsystem's software.

3. The SPMC builds the subsystem software and places the subsystem's updated SDD and UDS files in a SPMC controlled AFS area.

The specific means used to move updated SDD/UDS files into the NMC operational environment at a site will depend upon the prior agreement between SPMC, the subsystem and the specific site involved.. Each of the following procedures for the update of subsystem SDD/UDS files is supported by the NMC. (Since the subsystem files will be loaded onto the operational NMCFS at the NOCC/site, the Subsystem OE must approve the installation of these files on the NMCFS at these locations.)

4. If the NMCFS is used to deliver the subsystem's SDD and/or UDS files, the Subsystem CDE/OE requests the NOCC NMCFS SA to move the subsystem's updated SDD/UDS files from the SPMC AFS space onto the NOCC NMCFS per directions provided in the subsystem RDD.
5. Once on the NOCC NMCFS, the files may also be transferred to the site NMCFS per directions in the subsystem RDD. The updated files may be 'pushed' to the site by the NOCC NMCFS SA or may be 'pulled' to the site by the Site NMCFS SA depending upon the preference of the site.
6. If DOSL is used by SPMC for software delivery, the subsystem's compiled software, including SDD and UDS files, is moved to a site via DOSL.
7. The Site NMCFS SA is requested to install the subsystem's updated SDD and UDS files into the site's NMCFS from DOSL, as per directions in the subsystem RDD.
8. If the subsystem employs a physical media for software delivery, SPMC produces the media which is moved to the site/NOCC for installation. The Site/NOCC NMCFS SA is requested to load the subsystem's updated SDD/UDS files onto the site's NMCFS as per directions in the subsystem RDD.

Hybrids of the above procedures may also to supported depending upon arrangements between SPMC, the subsystem and the site, e.g., transferring updated files to a DSCC via a hardcopy media and updating the NOCC NMCFS via a transfer from SPMC AFS space.

12.5 Maintenance of 820-016 Interface Agreements

One, or more, 820-016 interface agreements may need to be updated depending upon the nature of a SDD file change. An interface agreement update is normally not required if a subsystem SDD file change only affects the subsystem, or if there are no subsystem-external subscribers for newly defined monitor data. However, if a change affects another subsystem, or if newly defined monitor data is subscribed to by another subsystem, the appropriate 820-016 interface agreement(s) should be updated. It is the responsibility of the Subsystem CDE to update all required 820-016 interface agreements.

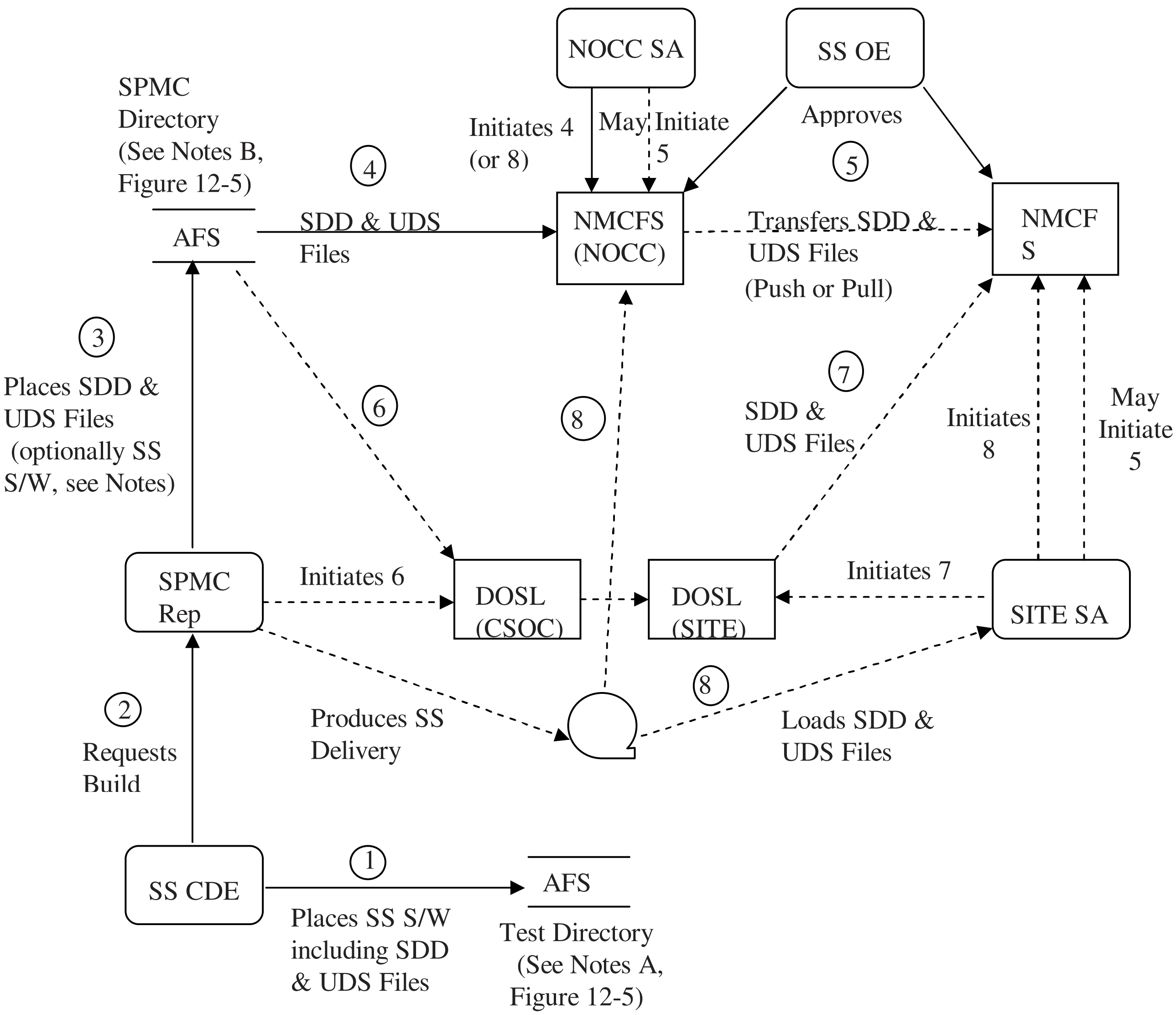

Figure 12-4. SDD/UDS Files Delivery Procedures – SPMC Delivery

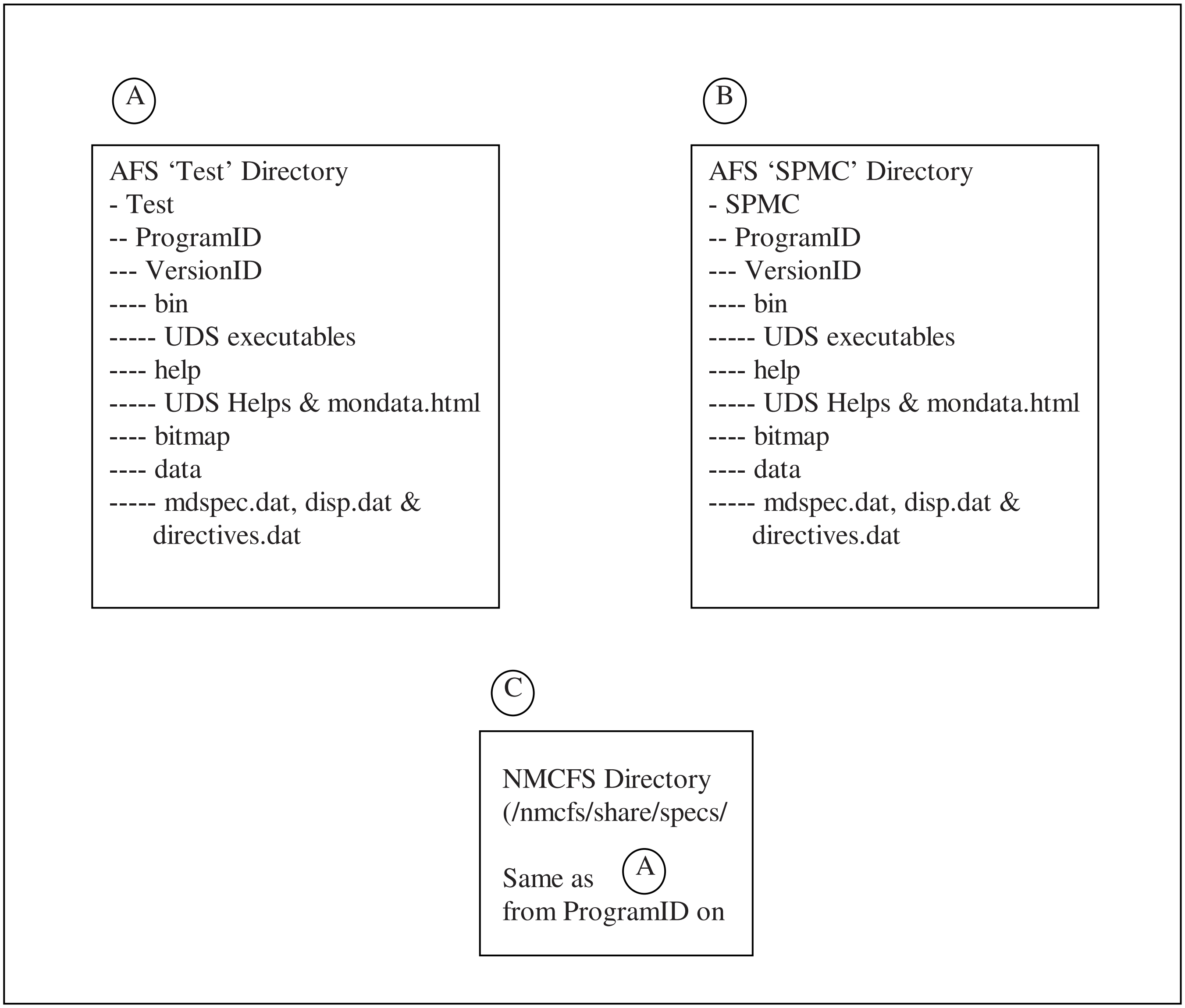

Figure 12-5. Notes for Figures 12-2 through 12-5.

Appendix A
Terminology

A.1 Acronyms and Abbreviations

AA	Automation Assembly (NMC assembly)
AFS	Andrew File System
aka	Also known as
ANSI	American National Standards Institute
ANT	DSCC Antenna Mechanical Subsystem
API	application programming interface
ASCII	American Standard Code for Information Interchange
ASP	Activity Support Package
ASR	Activity Support Request
CCN	Configuration Change Notice
CDE	Cognizant Development Engineer
CDT	Command Display Terminal
CE	Connection Engine (NMC assembly)
CMC	Complex Monitor and Control Processor
CMD	DSN Command System
CNF	Configuration (display identifier)
Comm	communications
COTS	commercial off the shelf
CPA	Command Processor Assembly
CS	Complex Supervisor
dB	decibel
DCC	Downlink Channel Controller
DDC	directive destination code
DDSO	DSN Data Services Office
DMC	DSCC Monitor and Control Subsystem
DMD	DSCC Media Calibration Subsystem
DMS	Data Management Services subsystem
DSCC	Deep Space Communications Complex
DSMS	Deep Space Mission System
DSN	Deep Space Network
DSS	Deep Space Station (synonymous with FEA)
DTF-21	Development and Test Facility
DTR	Detailed Technical Review
ENS	Event Notification Service
ETX	DSCC Exciter-Transmitter Subsystem
FAS	Functional Addressing Service
FEA	front-end area (synonymous with DSS)
FSP	Full Spectrum Processing subsystem
FSR	Full Spectrum Recorder
FTDD	fault tolerant data delivery

FTP	File Transfer Protocol
FTS	Frequency and Timing Subsystem
GCF	Ground Communications Facility
GCS	Ground Communications System
GDD	GCF Data Delivery Subsystem
H/W	hardware
HTML	Hyper-Text Markup Language
I/O	input/output
ID	identifier/identification
IEEE	Institute of Electrical and Electronic Engineers
ISB	inter-subsystem block
JPL	Jet Propulsion Laboratory
LAN	local area network
LMC	Link Monitor and Control Processor
MCIS	Monitor and Control Infrastructure Services
MCS	Monitor and Control Services
MDA	Metric Data Assembly
MDS	Monitor Data Service
MID	Message Identification
MSB	most significant bit
MVT	Mission Verification Test
NCP	Network Control Project
NMC	Network Monitor and Control subsystem
NMCFS	NMC File System
NMC/TSP	NMC Translator Plus
NOCC	Network Operations Control Center (JPL)
NRF	NMC Resource File (see NMC SOM)
NRT	NOCC Real-Time Subsystem
OCI	operator control input (old term for “directive”)
OD	operator directive
PDW	Parameter Descriptor Word
POSIX	Portable Operating System for UNIX
RCV	DSCC Receiver-Exciter Subsystem (Blocks III & IV)
RDD	Release Description Document
RPC	remote procedure call
RS	radio science
S&L	standards and limits
S/W	software
SC	spacecraft
SCP	Station Communications Processor
SDD	Subsystem Data Definition
SOM	Software Operator's Manual
SPPA	Support Products Provider Assembly
SPT	system performance test
SRA	Sequential Ranging Assembly
SSE	Subsystem Engineer
TDN	Temporal Dependency Network
TLM	DSN Telemetry System
TMOD	Telecommunications and Mission Operations Directorate

TRK	DSN Tracking System
TSP	Translator Plus (NMC assembly)
TSA2	Telemetry Simulation Assembly II
TSR	Technical Status Review
TXR	DSCC Transmitter Subsystem
UDS	Uniform Display Service
UGC	Generic Microwave Controller Software
UI	user interface
ULC	Uplink Controller
URL	Uniform Resource Locator
UTC	universal time (coordinated)
UWV	DSCC Antenna Microwave Subsystem
WAN	wide area network

This page is intentionally left blank.

Appendix B
Standard Monitor Data Formats

B.1 MON-2 Monitor Data Formats

Subsystem monitor data should be expressed in one of several standard formats defined in Table B-1 below. The standard formats are summarized in the table, and are described further in MON-1. (For MON-2 subsystems desiring to communicate with a DFL-1-2/890-132 subsystem, refer to Appendix C.) In addition to defining the standard data formats for monitor data in Table B-1, this section elaborates on the special data types FORMAT_STATUS, FORMAT_TIME, and FORMAT_PARAM_DESCRIPTOR. Section 6.2.4 defines the relationship between the monitor data classifications and monitor data formats.

Table B-1. MON-1 Monitor Data Formats

MON-1 Data Formats`	Data Type	Size [Byte]	Structure Members
FORMAT_STATUS	status_type	8	status_code_type code status_qual_type qualifier
FORMAT_INTEGER	short	2	
FORMAT_DOUBLE_INTEGER	long	4	
FORMAT_FLOATING	float	4	
FORMAT_DOUBLE_FLOATING	double	8	
FORMAT_TIME	timespec_type	12	long tv_sec long tv_nsec short tv leap
FORMAT_STRING	char *	4	
FORMAT_PARAM_DESCRIPTOR	md_pd_type	12	boolean valid md_analysis_type analysis char *src

B.1.1 FORMAT_STATUS

FORMAT_STATUS is a data type that represents the status of individual DSN resources. It contains two fields: a status code and a qualifier. The status code is an enumerated data type. The value for each of the status codes is provided below in Table B-2.

Table B-2. Status Codes

Status Code Value	Status Code Meaning
CODE_OUT_OF_SERVICE	Not immediately needed to support the mission
CODE_OPERATIONAL (i.e., normal)	Functioning normally (according to specifications) as far as can be determined
CODE_BUSY	Temporarily unavailable or in preparation but soon to be operational without operator intervention
CODE_DEVIATION	Functionally impaired but not warranting an alarm (see Section 5.2 for alarm criteria)
CODE_MARGINAL	Impaired but not critically so; mission support is probably not in jeopardy
CODE_CRITICAL	Seriously impaired; mission support may be in jeopardy
CODE_EMERGENCY	Immediate danger to personnel, major equipment, or station facilities; operator action is urgently required to correct the situation

The qualifier may be used to provide clarification on the status code. It is an enumerated data type and consists of the meanings indicated in Table B-3 below.

Table B-3. Status Code Qualifiers

QUAL_NONE
QUAL_OFF_LINE
QUAL_ON_LINE
QUAL_IN_SYNC
QUAL_OUT_OF_SYNC
QUAL_IN_LOCK
QUAL_OUT_OF_LOCK
QUAL_RECORDING
QUAL_READING
QUAL_AWAITING_OPERATOR (intervention)
QUAL_INPUT_MISSING
QUAL_ENABLED
QUAL_DISABLED
QUAL_ON_POINT
QUAL_OFF_POINT
QUAL_IDLE
QUAL_ACTIVE
QUAL_ACQUISTION

B.1.2 FORMAT_TIME

`timespec_type` is used for parameters that report time. It is defined as follows:

```
typedef struct  {
```

```
long tv_sec;
long tv_nsec;
short tv_leap;
}  timespec_type;
```

`tv_sec` is a value of time consistent with the POSIX definition of time_t; i.e., the number of seconds since the Epoch (1 January 1970 UTC) excluding leap seconds. `tv_nsec` is the number of nanoseconds since the value of `tv_sec`. `tv_leap` is the number of leap seconds since the Epoch.

B.1.3 FORMAT_PARAM_DESCRIPTOR

This monitor data is only generated by the Translator Plus (TSP) when publishing data on behalf of the 890-132 subsystems. MON-2 subsystems do not need to publish this information for their monitor data.

`md_pd_type` is a data type used to identify the validity and quality of other monitor data items. It is defined as follows:

```
typedef struct  {
boolean valid;
md_analysis_type  analysis;
[string, ptr] char *src;
}  md_pd_type;
```

The `valid` member identifies the validity of the data. It indicates whether the monitor parameter was actually observed prior to publication. If `valid` is TRUE, it indicates the subsystem has observed the parameter; the value may be used. If `valid` is FALSE, it indicates the subsystem either has not observed the parameter, or the parameter is not applicable to this mode of operation; the value should be disregarded.

The `analysis` member is used when an analysis is done of the monitor parameter against the associated standards and limits (S&L) prior to publication. It identifies the results of such an analysis. `analysis` is an enumerated type and is defined in Table B-4 below.

Table B-4. Parameter Descriptor Codes

PDW Analysis Code	**Meaning**
ANALYSIS_DISREGARD	The subsystem has not analyzed the parameter; disregard this field.
ANALYSIS_NORMAL	The parameter has a normal, reasonable, or expected value.
ANALYSIS_HIGH_WARNING	High warning limit has been exceeded.
ANALYSIS_HIGH_CRITICAL	High critical limit has been exceeded.
ANALYSIS_LOW_WARNING	Low warning limit has been exceeded.
ANALYSIS_LOW_CRITICAL	Low critical limit has been exceeded.

The `src` member is used in instances where the parameter is observed by a subsystem other than the one publishing the parameter. `src` identifies the functional address of the observing subsystem.

B.2 890-132 Monitor Data Formats

The list (see Table B-5) and definition of 890-132 monitor data formats described below supersede those in document 890-132. Definitions for each format are described in the following sections of this appendix. Note that Fixed Point Binary and Unsigned Integer are also described, but are not monitor data formats that are currently used by any existing DFL-1-2/890-132 subsystems. The data formats are used when describing header information (i.e., segment ID and segment length) in an interface agreement's description of monitor data transfer segments.

Table B-5. 890-132 Monitor Data Formats

```
STATUS
TIME
PDW
INTEGER
DOUBLE_INT
FLOAT
DOUBLE_FLOAT
IEEE_SINGLE
IEEE_DOUBLE
ASCII_FLOAT
ASCII_2
ASCII_4
ASCII_8
ASCII_12
ASCII_20
BINARY_POINT nnn*
UNFORMATTED
```

B.2.1 Status

The Status format is a data type that represents the status of individual DSN resources. It contains two fields: a status code and a qualifier. The status codes and their meanings are provided in Table B-6 below. The qualifier may be used to provide clarification on the status code. The qualifier codes and their meanings are provided in Table B-7 below. The Status format is as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
qualifier code													status code		

Table B-6. Status Codes

Status Code	Meaning
0	**Out of service**. Not immediately needed to support the mission
1	**Operational (or Normal)**. Functioning normally (according to specifications) as far as can be determined
2	**Busy**. Temporarily unavailable or in preparation but soon to be operational without operator intervention
3	**Deviation**. Functionally impaired but not warranting an alarm (see Section 5.2 for alarm criteria)
4	**Marginal**. Impaired but not critically so; mission support is probably not in jeopardy
5	**Critical**. Seriously impaired; mission support may be in jeopardy

6	**Emergency**. Immediate danger to personnel, major equipment, or station facilities; operator action is urgently required to correct the situation

Table B-7. Status Code Qualifiers

Status Code Qualifier	Meaning
0	None
1	Off line
2	On line
3	In sync
4	Out of sync
5	In lock
6	Out of lock
7	Recording
8	Reading
9	Awaiting operator intervention
10	Expected input missing
11	Enabled
12	Disabled
13	On point
14	Off point
15	Idle
16	Active
17	Acquisition

B.2.2 Time

The Time format (12 bytes) can be used to represent a time measurement to a resolution of one millisecond. Time **shall** be expressed as a string of 12 ASCII digits containing the following information:

Item	Bytes	Minimum Value	Maximum Value
Day of Year	3	001	366
Hour	2	00	23
Minute	2	00	59
Second	2	00	60
Millisecond	3	000	999

Each item **shall** be expressed with leading zeros and is arranged in six 16-bit words as follows:

1	2–8	9	10–16
0	day	0	day
0	day	0	hour
0	hour	0	minute
0	minute	0	second
0	second	0	millisecond
0	millisecond	0	millisecond

B.2.3 Parameter Descriptor Word (PDW)

The Parameter Descriptor Word (PDW) is used to describe a monitor data item(s) that it precedes in a monitor data transfer segment. The PDW format is a 16-bit word, and is as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
V	Analysis			Source											

The PDW consists of three fields: validity, analysis, and source. The validity field (labeled "V" in the layout diagram) indicates whether the monitor data item, with which a PDW is associated, was actually observed prior to transmission. It is occasionally necessary to transmit a monitor data transfer segment even though some included parameter may have no meaning in the particular mode of operation. The validity field **shall** be coded as follows:

Table B-8. PDW Validity Codes

Validity Code	Meaning
0	The parameter has been observed; the value may be used.
1	The parameter has not been observed, or the parameter is not applicable to this mode of operation; the value should be disregarded.

The analysis field is used when an analysis of the monitor data parameter against the associated standards and limits (S&L) is done prior to transmission. If a S&L check is done on the monitor data parameter, this field contains the results of the check. This field will have different interpretations depending upon the type of the monitor parameter (i.e., status, configuration, or performance). For status and configuration parameters, this field is not defined, and **shall** be coded as zero. For performance parameters, the analysis codes **shall** be used and their meanings are provided below.

Table B-9. PDW Analysis Codes

Analysis Code	Meaning
0	The subsystem has not analyzed the parameter; disregard this field.
1	The parameter has a normal, reasonable, or expected value.
2	High warning limit has been exceeded.
3	High critical limit has been exceeded.
4	Low warning limit has been exceeded.
5	Low critical limit has been exceeded.

The source field is used in instances where the parameter is observed by a subsystem other than the one transmitting the parameter. The source field contains the 890-131 process code of the observing subsystem.

B.2.4 Integer (Signed)

The Integer format (16 bits) can be used to express integral quantities in the range -32768 to 32767 with a resolution of unity. Integers are expressed in signed two's complement notation. The Integer format is as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S	2^{14}														2^0

B.2.5 Double Integer

The Double Integer format (32 bits) can be used to express integral quantities in the range –2,147,483,648 to +2,147,483,647 with a resolution of unity. Double integers **shall** be expressed in a signed two's complement notation [most significant byte (MSB) in the first word]. The Double Integer format is as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S	2^{30}														2^{16}
2^{15}															2^0

B.2.6 Floating Point

The Floating Point format (32 bits) can be used to express quantities in the approximate range $\pm 2^{255}$ (about $\pm 10^{76}$) with a precision of 22 bits (over six significant decimal digits). Floating point numbers **shall** be expressed with a sign, a nine-bit exponent, and a 22-bit mantissa. The Floating Point format is as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S	exponent									mantissa					
mantissa (continued)															

B.2.7 Double Floating Point

The Double Floating Point format (48 bits) can be used to express quantities in the approximate range $\pm 2^{255}$ (about $\pm 10^{76}$) with a precision of 38 bits (over ten significant decimal digits). Double floating point numbers **shall** be expressed with a sign, a nine-bit exponent, and a 22-bit mantissa. The Double Floating Point format is as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S	exponent									mantissa					
mantissa (continued)															
mantissa (continued)															

B.2.8 IEEE Floating Point

The IEEE Floating Point format is the IEEE-754 floating point values represented as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S	exponent								mantissa						

mantissa (continued)

B.2.9 IEEE Double Floating Point

The IEEE Double Floating Point format is the IEEE-754 floating point values represented as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S	exponent											mantissa			
mantissa (continued)															
mantissa (continued)															
mantissa (continued)															

B.2.10 ASCII Floating Point

An ASCII floating point format (12 bytes) can be used to express quantities in the range $\pm 10^{99}$ with a precision of seven decimal digits. The quantity **shall** be represented as a 12-character string in the following format:

+0.000000+00

where:

"+" is either a plus sign or a minus sign

"0" is a digit (0 to 9)

"." is a decimal point.

The string **shall** be interpreted as a signed decimal number with six fractional digits multiplied by ten raised to a two-decimal-digit signed power.

B.2.11 ASCII

ASCII data are ANSI standard, left-justified, with trailing spaces. Each ASCII character is represented by 8 bits consisting of a "0" bit followed by seven bits as prescribed in ANSI X3.4-1977 (ASCII). Strings of ASCII characters are left-justified with trailing blanks; these are represented in the format as "ASCII-" followed by the number of characters/bytes, e.g., ASCII-2, ASCII-4, ASCII-8, ASCII-12, ASCII-20.

B.2.11.1 ASCII-2

ASCII-2 (2 bytes) can be used to express 2-character ASCII strings. The ASCII-2 format is as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	character 1							0	character 2						

B.2.11.2 ASCII-4

ASCII-4 (4 bytes) can be used to express 4-character ASCII strings. The ASCII-4 format is as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

0	character 1	0	character 2
0	character 3	0	character 4

B.2.11.3 ASCII-8

ASCII-8 (8 bytes) can be used to express 8-character ASCII strings. The ASCII-8 format is as follows:

1	2–8	9	10–16
0	character 1	0	character 2
0	character 3	0	character 4
0	character 5	0	character 6
0	character 7	0	character 8

B.2.11.4 ASCII-12

ASCII-12 (12 bytes) can be used to express 12-character ASCII strings. The ASCII-12 format is as follows:

1	2–8	9	10–16
0	character 1	0	character 2
0	character 3	0	character 4
0	character 5	0	character 6
0	character 7	0	character 8
0	character 9	0	character 10
0	character 11	0	character 12

B.2.11.5 ASCII-20

ASCII-20 (20 bytes) can be used to express 20-character ASCII strings. The ASCII-20 format is as follows:

1	2–8	9	10–16
0	character 1	0	character 2

B.2.12 Unformatted

Unformatted messages **shall** be used to report anything that cannot be expressed in any of the other formats. The individual interface agreements **shall** explain how any unformatted messages are to be constructed and interpreted.

B.2.13 Binary Point

Binary point format can be used to express values which have a specific word length and relative binary point location. The format number specifies the location of the binary point, and takes the form "nxx," where:

n = number of words (maximum =7, minimum =1)
xx = number of bits to the left of the binary point (maximum =99, minimum =0)

Example: format number 208 indicates there is a binary point between bits 8 and 9 of a 32-bit (2-word) field.

B.2.14 Unsigned Integer

Unsigned Integer is not a monitor data format, but is included here because it often appears in interface agreements when describing the format for segment ID and segment length. Segment ID and segment length are contained in the header of monitor data transfer segments, and are unsigned binary data. Their format is identified as "U" followed by the number of bits, e.g., U7, U9.

B.3 Conversion from 890-132 to MON-2 Monitor Data Formats

The Table C-1 in Appendix C, Section C.6 describes the corresponding data format from the two protocols.

Appendix C
Interfacing With Legacy Subsystems

C.1 Introduction

This section is addressed to (a) personnel responsible for MON-2 subsystems that must communicate with a DFL-1-2/890-132 subsystem, or (b) personnel responsible for DFL-1-2/890-132 subsystems who may be wondering what the implications are for their subsystems under the MON-1 infrastructure environment.

C.2 NMC Translator Plus

The NMC Translator Plus (NMC/TSP) is the gateway that provides for communications between DSCC subsystems that run the DFL-1-2/890-132 protocol, and those that run the MON-2 (DFL-1-7/MON-1) protocols. Communications between the two different types of subsystems are accomplished in a manner that is transparent to both. The data types that travel between the two include configuration change notices (CCNs), directives/responses, event notifications, monitor data, legacy subsystem displays, and printed reports/tape labels.

To accomplish the translation of monitor data between the two protocols, and to support subsystem displays and directives, it is imperative that the correct program/versionID be provided in Segment 1 with each new redelivery of a DFL-1-2/890-132 subsystem. Therefore, DFL-1-2/890-132 subsystems **shall** ensure that the program/versionID transmitted (in Segment 1) actually corresponds to the software version producing the information.

C.3 Assignments

The NMC sends MON-1 CCNs intended for DFL-1-2/890-132 subsystems to the TSP. The TSP converts the received CCN to a CCN inter-subsystem block (ISB), and sends it to the destination subsystem.

MON-2 subsystems will receive Assign CCNs, Unassign CCNs, and Reassign CCNs. However, other existing CCN types (e.g., Quiescent, Dissolve, Swap In, and Swap Out) will continue to exist for the legacy DFL-1-2/890-132 subsystems.

C.4 Directives/Responses

The TSP provides for the transmission of directives from the MON-1 side (namely, NMC) to the DFL-1-2/890-132 side, and the transmission of responses in the DFL-1-2/890-132 to MON-1 direction. CDEs of DFL-1-2/890-132 subsystems provide the NMC Directives List File information via the procedures described in Section 4.3.1.

C.5 Event Notifications

The NMC/TSP provides for the transmission of events in both directions between the DFL-1-2/890-132 side and the MON-1 side.

C.6 Monitor Data

The NMC/TSP provides for the transmission of monitor data in both directions between DFL-1-2/890-132 and MON-1.

For MON-2 subsystems communicating with DFL-1-2/890-132 subsystems, these guidelines should be followed:

(1) Monitor data items should adhere to certain DFL-1-2/890-132 constraints as follows:

   (a) Monitor data item names should be no longer than eight characters and in upper case.

   (b) Monitor data types should conform to those specified in DFL-1-2/890-132, as defined in Appendix B, Section B.2 of this document. A conversion table of the monitor data formats used in the two protocols is provided in Table C-1.

   (c) All monitor data items being provided to a 890-132 subsystem should be collected in a segment.

   (d) Segment numbers for those segments to be provided to a 890-132 subsystem should be less than or equal to 127.

   (e) Monitor data items in the segment should use 890-132 formats and be only for a 890-132 subsystem.

   (f) The Destination field should have the “process code” of the subsystem receiving the monitor data.

Table C-1. Comparison of Monitor Data Formats

890-132 Monitor Data Format Standards	MON-1 Monitor Data Format Standards
ASCII 2	FORMAT STRING
ASCII 4	FORMAT STRING
ASCII 8	FORMAT STRING
ASCII 12	FORMAT STRING
ASCII 20	FORMAT STRING
ASCII FLOAT	FORMAT STRING
DOUBLE FLOAT	FORMAT DOUBLE FLOATING
IEEE DOUBLE	FORMAT DOUBLE FLOATING
FLOAT	FORMAT FLOATING
IEEE SINGLE	FORMAT FLOATING
DOUBLE INT	FORMAT DOUBLE INTEGER
INTEGER	FORMAT INTEGER
STATUS	FORMAT STATUS
TIME	FORMAT TIME
PDW	FORMAT PARAM DESCRIPTOR
BINARY POINT nnn*	FORMAT DOUBLE FLOATING
FIX POINT BINARY	FORMAT DOUBLE FLOATING
UNFORMATTED	FORMAT INTEGER

* Note that currently, binary point data greater than 2nn cannot be accommodated.

(2) The MCS provides a mon_poll() remote call procedure (RPC) to allow MON-2 subsystems to obtain DFL-1-2/890-132 monitor data that is available on-poll only. For MON-2 to MON-2 subsystem communications, the concept of polling is unnecessary.

(3) If a DFL-1-2/890-132 subsystem expects to poll for monitor data published by a MON-2 subsystem, the MON-2 subsystem **shall** ensure that the data are published under the functional address expected by the DFL-1-2/890-132 subsystem. For example, if a DFL-1-2/890-132 subsystem polls for a MON-2 subsystem's monitor data using a link process code, the MON-2 subsystem must ensure that the requested data are published under a connection functional address. Similarly, if the requesting subsystem uses a physical process code corresponding to the destination subsystem, that MON-2 subsystem must publish the data under a permanent functional address. The same is true for the cases of global process codes and antenna process codes.

CDEs of DFL-1-2/890-132 subsystems **shall** provide the MDS Monitor Data Definition information as described in Section 6.3.1.

C.7 Subsystem Displays

The NMC/TSP passes legacy subsystem displays on to the NMC user interface (UI) without processing them, using the isb_send() RPC.

DFL-1-2/890-132 subsystem personnel can generate X/Motif subsystem displays as well. These displays would not go through the NMC/TSP, since they are not transmitted by the subsystem in real time.

DFL-1-2/890-132 subsystem personnel can generate UDS subsystem displays as well. These displays would not go through the NMC/TSP, since they are executed at the NMC and use monitor data published by the subsystem.

CDEs of DFL-1-2/890-132 subsystems **shall** provide the NMC Display List File information as described in Section 7.3.1.

C.8 Support Data

Support data will continue to arrive at the subsystem as specified in the Subsystem-NSS interface agreement.

C.9 Printed Reports/Tape Labels

The NMC/TSP accommodates the sending of printed reports and tape labels from DFL-1-2/890-132 subsystems.

C.10 NRT-Destined Monitor Data

The NMC will provide the required monitor data to the NRT (until this system is retired).

This page is intentionally left blank.

Appendix D
Summary of Subsystem-Provided Information

This appendix summarizes the information that is required by the NMC for all subsystems, i.e., those running the MON-2 (DFL-1-7/MON-1) protocols, and those running the DFL-1-2/890-132 protocol.

Table D-1. Summary of Subsystem-Required Information

Subsystem Product	**Pointers to References**		**Notes**
	MON-2 Subsystems	**DFL-1-2/890-132 Subsystems**	
MDS Monitor Data Specification File	(1) MON-2, Section 6.3.1.1 and 6.3.1.2 (2) UG-DSI-6000-OP, MCIS User's Guide		It is the responsibility of the subsystem CDE to produce and deliver this file to the NMC for each subsystem software version using the procedures defined in this document.
NMC Directives List File	MON-2, Section 4.3.1		It is the responsibility of the subsystem CDE to produce and deliver this file to the NMC for each subsystem software version using the procedures defined in this document.
NMC Displays List File	MON-2, Section 7.3.1		It is the responsibility of the subsystem CDE to produce and deliver this file to the NMC for each subsystem software version using the procedures defined in this document.
Standard monitor data	MON-2, Section 6.3.2		Subsystems should adhere to the DSN standard for program/version IDs as defined in this document.
MDS Monitor Data Definitions File and `monhelp.dat`	MON-7		These products are also used by those subsystems building UDS displays. (DFL-1-2/890-132 subsystems can also develop UDS displays.)
NMC-Subsystem Interface Agreement (820-016 module)	MON-2, Section 10.2		
Automation-Subsystem Interface Agreement	MON-2, Sections 10.2.4 through 10.2.7		Subsystem CDE interfaces with the CDSCC Center For Automation for this interface agreement.

This page is intentionally left blank.

Appendix E
Subsystem Categorizations

The following categories are used by the NMC to organize various display elements including icons and menus. The appropriate category(s) should be selected and entered as parameters for ODSS and DSPSS, in the `directives.dat` and the `disp.dat` files, respectively.

Category	Description
ANT	Antenna Controllers, or any other antenna-specific subsystem with which NMC has direct contact.
CMD	Command Processors like the Command Processor Assembly (CPA) or Uplink Controller (ULC).
COMM	Communication subsystems, most notably Ground Communications System (GCS).
EXC	Exciters. Also includes subsystems, which include exciter-type activities along with other functions, like Exciter-Transmitter Subsystem (ETX).
PROJ	The projects. Usually the same as SC.
RCV	Receivers. Also includes subsystems which include receiving along with other functions, like receiver/exciter controller (REC).
RNG	Ranging Subsystems, like Metric Data Assembly (MDA), since it is used to control the Sequential Ranging Assembly (SRA). Note that MDA is also in the TRK category.
RS	Radio Science.
SC	Spacecraft, or any subsystem which has to do specifically with spacecraft.
TLM	Telemetry Controllers. Currently, the Downlink Channel Controller.
TRK	Tracking Subsystems like MDA. Note that MDA is also in the RNG category.
TST	Subsystems which provide test data to the connection in any way. These include Telemetry Simulation Assembly (TSA) and system performance test (SPT), even though they are otherwise quite different.
TXR	Transmitters. ETX and high powered radar transmitters (XH) are examples.
UWV	Microwave Controllers, like Generic Microwave Controller Software (UGC).
VLBI	Very Long Baseline Interferometry.

A single subsystem may allocate directives and displays in its `directives.dat` and `disp.dat` files, respectively, to multiple categories. An example of this might be MDA's `disp.dat` , where some displays would appropriately contain the line:

DSPSS=RNG

while others would contain the line:

DSPSS=TRK

depending on the purpose of the display, ranging or tracking.

Furthermore, a single directive or display entry may be seen as relating to multiple subsystems. To accomplish this the entire directive/display series must be repeated, but with the subsystem (and perhaps the description) changed.

Here is an example:

```
#
#  I want this display to appear in the Exciter menu/icons
#
DSP=MYDISP
DSPSS=EXC
DSPTYPE=UDS
DSPCONN=True
DSPMODE=OP,TEST,MAINT,DBUG
DSPPHASE=PRETK,INTK,POSTTK
DSPDESC=My Status display which includes Exciter parameters
END
#
#  I want the same display to also appear in the Transmitter menu/icons
#
DSP=MYDISP
DSPSS=TXR
DSPTYPE=UDS
DSPCONN=True
DSPMODE=OP,TEST,MAINT,DBUG
DSPPHASE=PRETK,INTK,POSTTK
DSPDESC=My Status display which includes Transmitter parameters
END
```

The display categorization(s) defined in a subsystem's `disp.dat` file must agree with the categorization(s) specified for the subsystem in the NMC Resource File (NRF) for the subsystem. (See the NMC SOM for a description of the NMC NRFs.)

Appendix F
Summary of MON-2 Shall Statements (Requirements)

Table F-1. MON-2 Requirements

Section	Requirement No.	Requirement Text
2.1 Automation	1	Any subsystem modification (including additions, deletions, or modifications to monitor data, directives, displays, and events) that affects automation shall be coordinated with the CDSCC Center For Automation.
2.1	2	Subsystems shall test with NMC Automation following any subsystem changes.
2.4 Self-Configuration	1	The subsystem shall wait indefinitely for the standard connection data to become available, i.e., the subsystem should not timeout.
2.4	2	If the subsystem cannot complete its configuration based upon available data, it shall issue an appropriate alarm notification to the NMC under its connection functional address.
2.4	3	The subsystem shall continue to accept and process directives appropriate for its state. For example, subsystem directive may be used to supply data needed to complete the subsystem's configuration.
2.4	4	A subsystem shall always accept an Unassign CCN from the NMC.
2.5 Self-Evaluation	1	Subsystems shall evaluate their performance and report deviations to the NMC. For example, a subsystem may determine a performance deviation by comparing actual performance data against subsystem 'Standards and Limits' or by comparing the delta of actual vs. predicted calculations against subsystem "Standards and Limits."
2.5	2	The results of these comparisons shall be made immediately available to operations personnel through event notifications and/or monitor data.
2.5	3	Subsystems shall monitor their output of all products, and alert operations personnel when the required outputs are either impaired or nonexistent.
2.6 Remote Restart	1	The specific mechanism to be used by the NMC to reset a subsystem controller shall be defined in the NMC-subsystem interface agreement.
2.7 Standard Deployment and Software Version	1	Subsystem files, particularly those used by the NMC, shall be identified with specific software versions.
3.1 Subsystem Types and Functional Addresses	1	A connection-assignable subsystem controller shall have either a connection functional address or an antenna group functional address and a permanent functional address.
3.1	2	A multi-connection subsystem controller shall be capable of having multiple connection functional addresses and a permanent functional address.
3.1	3	Non-assignable subsystem controllers shall only have permanent functional addresses.

Section	Requirement No.	Requirement Text
3.2.1.1 Unassigned	1	A well-behaved subsystem shall place itself into the unassigned mode following bootup or reset.
3.2.1.1	2	An connection-assignable subsystem in the unassigned mode shall register under its permanent functional address to receive CCNs.
3.2.1.1	3	While in an unassigned mode, an assignable subsystem shall use its permanent functional address for all external communications.
3.2.1.2	1	An assignable subsystem in the assigned mode shall use the functional address supplied in the CCN as its source address in subsequent connection-related communications.
3.2.2 Multi-Connection Subsystems	1	The interface agreements between the NMC and a multi-connection subsystems shall define the specific assignment protocol for these subsystems.
3.2.3 Non-Assignable subsystems	1	A non-assignable subsystem shall ensure that its permanent functional address is published in accordance with DFL-1-7.
3.2.3	2	The subsystem shall assume an assigned communications mode following a subsystem reset/restart, such subsystem will not receive CCNs.
3.3.3 Assigned to Assigned	1	A subsystem shall not allow a change to the connection number while in assign mode.
3.4 Subsystem-Provided Information	1	Subsystems shall provide the following information in their interface agreement with the NMC: identification of subsystem type with respect to assignability (multi-connection, non-assignable or connection assignable).
3.4	2	Subsystems shall provide the following information in their interface agreement with the NMC: identification of subsystem functional and permanent addresses.
4.2.1 Protocol and Content	1	All directives issued to subsystems shall be entered into the appropriate NMC log(s).
4.2.1	2	Subsystems shall provide text in the directive response message to further describe and clarify the meaning of the response.
4.2.1	3	Directives that control status or configuration parameters shall be associated with the monitor data reflecting the requested changes.
4.2.2 Directive Response Categories and Criteria for Use	1	Subsystems shall use an estimated-time-to-complete criterion in determining what response category to use when issuing the first response to a directive. (The estimated completion times are intended to be pre-execution estimates of the completion time of the process invoked by the directive.)
4.2.2	2	If a subsystem can execute a directive within one second of receiving it, then the subsystem shall send a Completed response after executing the directive.
4.2.2	3	If the subsystem can expect to execute the directive within one to five seconds after receiving it, and if during this time it can receive additional directives, the subsystem shall send a Processing response within one second of receiving the directive.

Section	Requirement No.	Requirement Text
4.2.2	4	Upon receiving a Processing response, the NMC may send a new directive immediately to begin another exchange. This second directive shall be processed concurrently.
4.2.2	5	Processing of current exchanges shall follow the protocol standards independently.
4.2.2	6	The subsystem shall send a second response within a total of five seconds from the time it received the original directive. This second response could also be another Processing response.
4.2.2	7	The subsystem shall continue to send responses until either a Rejected response or a Completed response is sent.
4.2.2	8	If, while executing a directive, the subsystem is unable to receive a second directive, the Processing/Wait response shall be sent within one second of receiving the first directive.
4.2.2	9	Within 15 seconds of receiving a second directive, the subsystem shall send a second directive response. This second response could also be another Processing/Wait response.
4.2.2	10	The subsystem shall continue to send responses until either a Rejected response or a Completed response is sent.
4.2.2	11	If the subsystem takes more than five seconds to execute the directive (and the Wait condition does not apply), the Started response shall be sent to the NMC within one second of receiving the directive.
4.2.2	12	After executing the directive, the subsystem shall send a Completion Advisory to the NMC.
4.2.2	13	The subsystem shall send Progress Advisories to the NMC while the subsystem executes the directive.
4.2.2	14	In cases where the directive is not executed successfully, the subsystem shall send a Deviation Advisory or a Warning Alarm.
4.2.2	15	If the subsystem cannot accept a directive, the Rejected response message shall be sent to the NMC and clearly state the reason for rejecting the directive (e.g., syntax, semantics, parameter validity check, etc.).
4.2.2	16	The subsystem shall send the Rejected response message within one second of receiving the directive; the one second applies when Rejected is the first directive response in a directive/response exchange.
4.2.3 Timing Considerations	1	Limits of one second, five seconds, or fifteen seconds thereafter shall be satisfied when the subsystem transmits an appropriate directive response.
4.2.3	2	Subsystem designs shall accommodate a directive response time limit as follows: In the absence of any other I/O activity to or from the LAN(s), the response time limit is met.
4.2.3	3	Subsystem designs shall accommodate a directive response time limit as follows: During normal I/O activity, no more than 5% of the responses are transmitted later than the nominal response time limit.

4.2.3	4	Subsystem designs shall accommodate a directive response time limit as follows: The absolute maximum response time limits associated with each of the nominal response time limits (1, 5 and 15 seconds) are 3, 8, and 18 seconds. Regardless of LAN I/O activity (but assuming that all acknowledged transmissions are successful on first attempt), the response is transmitted within these absolute maximum response time limits.
4.3 Subsystem-Provided Information	1	In cases where closed-loop control relationships exist (i.e., a directive is associated with monitor data or specific events), such relationships shall be identified in the NMC–subsystem interface agreement.
4.3.1 NMC Directives List File	1	The subsystem CDE shall provide the information listed in Table 4-1 for each subsystem directive in the subsystem's *directives.dat* file.
5.2.1.1 Alarms	1	Subsystems shall report, via an alarm, a subsystem anomaly for which the operator is expected to take some corrective action.
5.2.1.1	2	Subsystems shall report, via an alarm, a subsystem anomaly for which the operator is expected to be aware that some corrective action is being carried out automatically.
5.2.1.1	3	Subsystems shall issue the same alarm only once.
5.2.1.1.1 Emergency Alarm	1	Subsystems shall issue an emergency alarm to report any condition that could result in immediate danger to personnel, major equipment, or station facilities.
5.2.1.1.1	2	Subsystems shall issue an emergency alarm to report an event that could result in costly repairs or excessive restoration time, if not addressed.
5.2.1.1.2 Critical Alarm	1	Subsystems shall issue a critical alarm to report any problem that, if uncorrected, will interrupt one or more data streams or otherwise jeopardize mission support.
5.2.1.1.3 Warning Alarm	1	Subsystems shall issue a warning alarm to report any problem that is not critical but requires operator action to correct.
5.2.1.1.3	2	Subsystems shall issue a warning alarm if the subsystem observes a series of minor malfunctions foreshadowing a condition that would result in significant loss of data.
5.2.1.2 Advisories	1	Subsystems shall report minor malfunctions, changes in status, routine progress, etc., by issuing advisories. Operators need not act upon an advisory.
5.2.1.2	2	Subsystems shall issue the same advisory only once.
5.2.1.2.1 Deviation Advisory	1	Subsystems shall issue a deviation advisory to report any anomalous condition that, by itself, requires no action by the operator.
5.2.1.2.1	2	Subsystems shall issue an information advisory when a deviation in performance or a transient is observed.
5.2.1.2.2 Completion Advisory	1	Subsystems shall issue a completion advisory to report the completion of some expected (and therefore non-anomalous) event.
5.2.1.2.3 Progress Advisory	1	Subsystems shall issue a progress advisory to report an ongoing, routine (i.e., non-anomalous) activity.
5.2.1.2.3	2	Subsystems shall issue progress advisories that clearly indicate whether the action is actually in process or merely in a queue.

5.2.1.2.3	3	Subsystems shall issue progress advisories that should include completion times (e.g., seconds remaining), when the completion time is accurate and readily available.
5.2.1.2.4 Log-Only Advisory	1	Subsystems shall issue a log-only advisory to report performance information which is not necessarily needed at the time it is reported, but needs to be available for review at a later time.
5.2.1.2.5 Recovered Advisory	1	Subsystems shall issue a recovered advisory to inform the operator when the condition triggering an alarm no longer exists.
5.2.1.3 Prompts	1	Subsystems shall issue a prompt to gain the operator's attention to inform the operator of a condition that either requires or permits some procedural step.
5.2.3 Interface Agreement Negotiations	1	In the cases where closed-loop control relationships exist (i.e., an event is associated with monitor data, or a particular directive or display), such relationships shall be identified in the NMC–subsystem interface agreement.
5.2.4 Destination of Subsystem Events	1	Connection assignable subsystems shall send events to the NMC Complex Supervisor (CS) permanent functional address /fa/<domain>/cs when unassigned, and to the NMC Connection Engine (CE) functional address provided in the configuration control notification (CCN) when assigned.
5.2.4	2	Non-Assignable subsystems shall send events only to the NMC CS permanent functional address.
6.2.1 Monitor Data Classifications	1	Subsystem controllers shall produce monitor data describing the subsystem's status, configuration, and performance.
6.2.3 Monitor Data Formats	1	Monitor data parameters that report status information shall be expressed using the `FORMAT_STATUS` data format.
6.2.3	2	Monitor data parameters that report configuration information shall be reported as ASCII strings, using the `FORMAT_STRING` data format, with the exception of configuration parameters that contain numeric information.
6.2.3	3	Configuration information shall be encoded so that meaningful human interpretation does not require conversion by the NMC.
6.2.3	4	Monitor data parameters that report performance information as numeric values shall use any of the standard numeric data formats.
6.2.4 Design of Monitor Data	1	Monitor data of a subsystem shall be designed such that the overall subsystem status can be described by a single monitor data item.
6.2.4	2	Monitor data of a subsystem shall be designed such that visibility into a subsystem can be achieved by successive and ordered views which are increasingly detailed.
6.2.4	3	Monitor data of a subsystem shall be designed such that status, configuration, and performance monitor data shall be provided to the extent necessary to describe the health and state of a component.
6.2.4	4	Monitor data of a subsystem shall be designed such that closed-loop control can be accomplished via monitor data. For example, when a directive is issued to a subsystem, the 'result' of this directive (or failure to achieve the result) should be reflected in monitor data.

6.2.4	5	Closed-loop relationships between directives and monitor data shall be described in the “Description” of the monitor data.
6.2.4	6	Monitor data of a subsystem shall be designed such that critical subsystem information is produced as monitor data.
6.2.5 Subsystem State	1	Subsystems shall accurately report (via monitor data) their current status and state at all times.
6.2.5	2	All subsystem status or state changes shall result in the publishing of the associated monitor data by the subsystem.
6.2.5	3	When uncertain as to the status of the subsystem component, the more critical status value shall be reported.
6.2.6 Program/Version ID Standard	1	The following standard shall be adhered to when publishing program IDs: The program ID – specifically the PGMID monitor data item – is the program identification number provided by SPMC (e.g., DOA-5556-OP).
6.2.6	2	The following standard shall be adhered to when publishing version IDs: The version ID – specifically the VersionID monitor data item – is a concatenation of the operational revision identifier and the delivered program version numbers (e.g., BV3.0.7 or EV10.14.2.).
6.3.1.1 MDS Monitor Data Specification File	1	The subsystem CDE **shall** identify each monitor data item to be published by the subsystem and the format of the data item in the subsystem’s *MDS mdspecs.dat* file.
6.3.1.2 Interface Agreements	1	All subsystem monitor data subscribed to by the NMC and the functional address(s) under which this data is published shall be defined in an interface agreement between the subsystem and the NMC.
6.3.1.2	2	The monitor data descriptions in the interface agreement shall consist of the following information: (1) data item identifier (name). (All data items published under the same functional address must have unique names.) (2) item description (a) brief description of the data item. (b) identification of any positive closed-loop control relationship(s) (e.g., between the monitor data item and subsystem directives or events). (c) identification of any relationship(s) to predicts and/or standards and limits, in the case of performance data. (3) format. (4) units/precision. (5) range of values.
6.3.1.2	3	Interface agreements with the NMC shall identify all subsystem assemblies and provide the required monitor data for each assembly.
6.3.2 Subsystem/ Assembly Standard Monitor Data	1	Connection assignable and antenna group subsystems shall publish the monitor data defined in Tables 6-1 and 6-2, as appropriate, under the permanent functional addresses shown ion Table 6-3, independent of the assigned mode of the subsystem.

6.3.2	2	Connection assignable and antenna group subsystems while in assigned mode shall also publish the monitor data defined in Tables 6-1 and 6-2, as appropriate, under the functional addresses provided in the assigned CCN.
6.3.2	3	A non-assignable subsystem shall publish the monitor data described in Tables 6-1 and 6-2, as appropriate, under the permanent functional addresses defined in Table 6-3.
7.2.1 Types of Subsystem Displays	1	Subsystem displays shall adhere to the MON-3 standard.
7.2.2 User Interface Standards	1	Subsystem displays shall be associated with an identifier that is unique among displays for that particular subsystem. (MON-3 contains a list of standard names for standard subsystem displays.)
7.3.1 NMC Display Data File	1	The subsystem CDE **shall** provide the information listed in Table 7-1 for each subsystem display.
8.2.1 Support Data Processing	1	A MON-2 subsystem shall receive support data files from the SPPA in a predefined input directory on a predefined host.
8.2.1	2	All MON-2 subsystems shall automatically process support data files stored in their 'input' directory by the SPPA.
8.2.1	3	This automatic processing shall include parsing of the file name, verification of the file contents and format, storage of the file in the appropriate 'user' directory(s), the handling of error conditions, and the periodic purging of the support files from both the subsytem 'input' and 'user' directories.
8.2.1	4	MON-2 subsystems shall automatically verify support data files to the extent possible upon their arrival at the subsystem.
8.2.1	5	MON-2 subsystems shall verify that the file is readable and is complete, i.e., that the last line of the file is '*=END=*'.
8.2.1	6	If an error is found during file verification, a MON-2 subsystem shall send a warning alarm event message to the NMC and delete the file from the subsystem 'input' directory.
8.2.1	7	If a required support data file is unavailable at connection configuration or if there is an ambiguity in which file to use (e.g., more that one version of a file is available), a MON-2 subsystem shall issue a critical alarm event message to the NMC.
8.2.1	8	MON-2 subsystems shall provide subsystem directives which allow an operator to manually identify/overwrite the specific support date file to be used by the subsystem.
8.2.1	9	MON-2 subsystems shall periodically purge support data files from their 'input' and 'user' directories based upon purge dates provided in each file's file name.
9.2.3 Printer Selection and Control	1	Printers, including a label printer, are available at each DSCC. Unix remote printer commands shall be used by the subsystem software to select and control where and how hardcopy reports will be output.
10.2.1 General	1	Interface agreements shall match as-built software.
10.2.1	2	Interface agreements shall not list monitor data in the interface agreements if the data are not actually published.
10.2.1	3	Interface agreements shall be identified with a specific program ID
10.2.2 Functional Addresses	1	The interface agreement shall identify all functional address(s) and permanent functional addresses by which the subsystem will be identified, and under what conditions each will be used.

10.2.3 Assignment	1	Subsystems shall provide the following information in their interface agreement with the NMC: Identification of the subsystem's assignment type, i.e., connection assignable, multi-connection or non-assignable subsystem.
10.2.3	2	Subsystems shall provide the following information in their interface agreement with the NMC: how swaps and reboots are to be performed for the subsystem.
10.2.4 Directives/ Responses	1	For cases where a positive closed-loop control relationship exists (i.e., specific monitor data can be used to verify the successful execution of a directive), such relationships shall be identified in the NMC–subsystem interface agreement.
10.2.4	2	Subsystem directives utilized by NMC automation shall be identified in the subsystem interface agreement with the CDSCC Center For Automation.
10.2.5Event Notification	1	For cases where a positive closed-loop control relationship exist (i.e., an event is associated with monitor data, or a particular directive or display), such relationships shall be identified in the NMC–subsystem interface agreement.
10.2.5	2	Subsystem events utilized by NMC automation shall be identified in the subsystem interface agreement with the CDSCC Center For Automation.
10.2.6 Monitor Data	1	All subsystem monitor data published by an assembly and subscribed to by the NMC shall be defined in an interface agreement between the subsystem and the NMC.
10.2.6	2	The monitor data descriptions in the interface agreement shall consist of the following information: (1) Data item identifier (2) Brief description (3) Format (4) Units/precision (5) Range of values (6) Identification of positive closed-loop control relationships (e.g., between monitor data and directives or events) (7) Relationship to predicts and standards and limits (files), in the case of performance data (8) Association between a PDW, if any, and specific monitor data items (for 890-132 subsystems)
10.2.7 Subsystem Displays	1	Displays utilized by NMC automation shall be identified in the subsystem interface agreement with the CDSCC Center For Automation.
11.2 Integration Testing with the NMC	1	Subsystems shall specifically perform integration tests with the NMC and with NMC Automation for each subsystem re-delivery.
11.2	2	Subsystems shall demonstrate that subsystem test procedures address compliance with the "shall" statements in this document.
12.2 NMC-Required Data Definition Files	1	The subsystem Data Definition files are subsystem-version sensitive and shall be re-delivered with each new version of subsystem software.

B.2.2 Time	1	The Time format (12 bytes) can be used to represent a time measurement to a resolution of one millisecond. Time shall be expressed as a string of 12 ASCII digits containing the following information: Day of Year: 3 bytes, Min Value 001, Max Value 366 Hour: 2 bytes, Min Value 00, Max Value 23 Minute: 2 bytes, Min Value 00, Max Value 59 Second: 2 bytes, Min Value 00, Max Value 60 Millisecond: 3 bytes, Min Value 000, Max Value 999
B.2.2	2	Each item shall be expressed with leading zeros and is arranged in six 16-bit words.
B.2.3 Parameter Descriptor Word (PDW)	1	The validity field shall be coded as follows: 0 = The parameter has been observed; the value may be used. 1 = The parameter has not been observed, or the parameter is not applicable to this mode of operation; the value should be disregarded.
B.2.3	2	For status and configuration parameters, the PDW analysis field is not defined, and shall be coded as zero.
B.2.3	3	For performance parameters, the analysis codes shall be used and their meanings are as follows: 0 = The subsystem has not analyzed the parameter; disregard this field. 1 = The parameter has a normal, reasonable, or expected value. 2 = High warning limit has been exceeded. 3 = High critical limit has been exceeded. 4 = Low warning limit has been exceeded. 5 = Low critical limit has been exceeded.
B.2.5 Double Integer	1	The Double Integer format (32 bits) can be used to express integral quantities in the range –2,147,483,648 to +2,147,483,647 with a resolution of unity. Double integers shall be expressed in a signed two's complement notation (MSB in the first word).
B.2.6 Floating Point	1	The Floating Point format (32 bits) can be used to express quantities in the approximate range $\pm 2^{255}$ (about $\pm 10^{76}$) with a precision of 22 bits (over six significant decimal digits). Floating point numbers shall be expressed with a sign, a nine-bit exponent, and a 22-bit mantissa.
B.2.7 Double Floating Point	1	The Double Floating Point format (48 bits) can be used to express quantities in the approximate range $\pm 2^{255}$ (about $\pm 10^{76}$) with a precision of 38 bits (over ten significant decimal digits). Double floating point numbers shall be expressed with a sign, a nine-bit exponent, and a 22-bit mantissa.

B.2.10 ASCII Floating Point	1	An ASCII floating point format (12 bytes) can be used to express quantities in the range $\pm 10^{99}$ with a precision of seven decimal digits. The quantity shall be represented as a 12-character string in the following format: +0.000000+00 where: "+" is either a plus sign or a minus sign "0" is a digit (0 to 9) and "." is a decimal point. The string shall be interpreted as a signed decimal number with six fractional digits multiplied by ten raised to a two decimal digit signed power.
B.2.12 Unformatted	1	Unformatted messages shall be used to report anything that cannot be expressed in any of the other formats.
B.2.12	2	The individual interface agreements shall explain how any unformatted messages are to be constructed and interpreted.
C.2 NMC Translator Plus	1	DFL-1-2/890-132 subsystems shall ensure that the program/versionID transmitted (in Segment 1) actually corresponds to the software version producing the information.
C.6 Monitor Data	1	If a DFL-1-2/890-132 subsystem expects to poll for monitor data published by a MON-2 subsystem, the MON-2 subsystem shall ensure that the data are published under the functional address expected by the DFL-1-2/890-132 subsystem.
	2	CDEs of DFL-1-2/890-132 subsystems **shall** provide the MDS Monitor Data Definition information as described in Section 6.3.1.
C.7 Subsystem Displays	1	CDEs of DFL-1-2/890-132 subsystems **shall** provide the NMC Display List File information as described in Section 7.3.1.

Appendix G
Subsystem Failure Scenarios

G.1 Subsystem Controller Failure Scenario*s*

- Scenario 1: Dedicated Backup
- Scenario 2: Standby Backup
- Scenario 3: Selectable Backup
- Scenario 4: Recovery by Controller Reboot

G.2 Subsystem Controller Failure Scenarios

These scenarios specifically address new controllers, designed under 820-019, MON-2 - *DSN Monitor and Control Standard Practices*.

Terminology

Dedicated Backup: This is a dedicated 'hot' backup controller for the primary subsystem controller. It monitors the primary controller, detects the primary controller failure, and automatically takes over for the primary controller, i.e., no operator intervention is required for the second controller to assume the role of the primary controller.

Standby Backup: There is a dedicated 'warm' backup controller for the primary controller that needs to be commanded to assume the role of the primary controller.

Selectable Backup: There is/are other controller(s) able to replace the primary controller. The operator selects the appropriate controller to use at the time the primary failure occurs.

Primary Controller: The subsystem controller that is assigned to support the connection or, in the case of a non-assignable subsystem, the controller that is providing the required monitor data to the NMC.

Scenario 1: Dedicated Backup

A failure is experienced in the subsystem primary controller while the subsystem is assigned to a connection. The subsystem assemblies continue processing as required.

Subsystem setup:

1. NMC sends an assign CCN to the primary subsystem controller to support a connection.
2. Subsystem controller provides configuration information to its backup controller unit.
3. Subsystem backup controller configures itself and starts to monitor the primary subsystem controller.

Scenario after failure:

1. The backup controller detects a failure of the primary controller.
2. The subsystem backup controller retrieves the checkpoint data left by the failed controller and then takes over as the primary controller.
3. The subsystem backup controller starts communicating with the subsystem assemblies, and registers with MCIS under the failed controller functional address.

4. The subsystem sends an event message to the NMC that the backup controller has taken over for the failed controller.

Scenario 2: Standby Backup

A hardware failure is experienced in the subsystem primary controller while the subsystem is assigned to a connection. The subsystem assemblies continue processing as required.

Subsystem setup:

1. Operator adds the primary subsystem controller and the backup subsystem controller to the connection.
2. NMC sends an assign CCN to both the primary and the backup subsystem controllers. (The primary subsystem controller is aware that it is the primary controller, and the subsystem backup controller is aware that it is the backup controller.)
3. Subsystem configures itself and starts supporting the pass.

Scenario after hardware failure:

1. The NMC notifies the operator of the primary controller failure.
2. The operator directs the NMC to issue a Reassign CCN to the backup subsystem controller.
3. The Reassign CCN indicates to the backup controller that it needs to assume control of the subsystem, and to continue supporting the connection from where the other controller left off. Therefore, it retrieves the checkpoint data left by the failed controller.
4. The subsystem backup controller configures itself and takes over as the primary controller.
5. It then starts communicating with the assemblies, and registers with MCIS under the functional address provided by the NMC in the Reassign CCN.
6. The subsystem sends an event message to the NMC that the backup controller has taken over for the failed controller.

Scenario 3: Selectable Backup

A hardware (H/W) failure is experienced on the subsystem controller while the subsystem is assigned to a connection. The subsystem assemblies continue processing as required.

Subsystem setup:

1. NMC sends an assign CCN to the subsystem controller to support a connection.
2. All other subsystems in the connection have been configured.
3. Subsystem configures itself and starts supporting the pass.

Scenario after hardware failure:

1. The NMC notifies the operator of the primary controller failure.
2. The operator selects a unit to replace the failed controller and adds the backup controller to the connection.
3. The operator directs the NMC to issue a Reassign CCN to the backup controller.
4. The Reassign CCN indicates to the backup controller that it needs to continue supporting a connection from where the other controller left off. Therefore, it retrieves the checkpoint data left by the failed controller.
5. The subsystem backup controller configures itself and takes over as the primary unit.
6. It then starts communicating with the assemblies, and registers with MCIS under the functional address provided by the NMC in the Reassign CCN.
7. The controller publishes required monitor data.

Scenario 4: Recovery by Controller Reboot

A software failure is experienced by the subsystem controller while the subsystem is assigned to a connection. The subsystem assemblies continue performing as required.

Subsystem setup:

1. NMC sends an assign CCN to the subsystem controller to support a connection.
2. Subsystem configures itself and starts supporting the pass.

Scenario after software failure detection:

1. NMC detects and reports failure of subsystem controller.
2. The operator verifies that it is not a H/W failure.
3. The operator issues a remote reset directive to the subsystem.
4. The subsystem controller unregisters from MCIS for all previously registered services and resets itself.
5. If the subsystem controller does not respond in step 3 and 4 above, the operator performs a remote login to stop/start the subsystem controller application.
6. The subsystem controller comes up in unassigned mode.
7. NMC issues a Reassign CCN to the subsystem controller.
8. The Reassign CCN indicates to the controller that it needs to continue supporting a connection from where it left off. Therefore, it retrieves the checkpoint data left from before the reset/restart.
9. The subsystem controller configures itself.
10. It then starts communicating with the assemblies, and registers with MCIS under the functional address provided by the NMC in the CCN.
11. The controller publishes required monitor data.

This page is intentionally left blank.

Appendix H

NMC File System Access

The NMC and other MON-2 subsystems require that certain data files be available on the NMC File System (NMCFS) in order to operate. Data files that both the NMC and/or other MON-2 subsystems must have access to include the MCIS and UDS library files, Subsystem Data Definition (SDD) files, subsystem UDS-related display files, MCIS server lists files, and various other files. The location of these files within the NMCFS is described in this appendix. Additionally, the types of NMCFS access granted to MON-2 subsystems are also described in this appendix.

H.1 MON-2 Subsystem Access

MON-2 subsystems will have controlled access to the NMCFS. Specifically, MON-2 subsystems will be allowed to mount certain portions of the NMCFS and will be granted file transfer type access to other portions of the NMCFS. See Figure H-1 for a diagram of the NMCFS file space that can be accessed by MON-2 subsystems.

MON-2 subsystems will be granted mount access to all NMCFS directories and files at or below the NMCFS **share** directory. A MON-2 subsystem will be able to access the NMCFS **share** directory, and all of its subdirectories and files, as if the directory were located on it's local disk. The structure of the **share** directory is further described below.

MON-2 subsystems will also be allowed file transfer access to the file space allocated to them on the NMCFS. This file space will be found under the **subsystem_files** directory on the NMCFS. In this context, MON-2 subsystems will be allowed to copy files onto and retrieve files from the NMCFS via the NMCFS 'cmd' commands provided for this purpose. (See the NMCFS SOM for more information on these file transfer commands.)

H.2 NMCFS Share Directory

The NMCFS **share** directory contains two subdirectories; the **cs** (common services) subdirectory and the **specs** (specifications) subdirectory. All of the files stored under these two subdirectories may be read directly by MON-2 subsystems. However, files under these subdirectories may not be altered by the subsystems nor can any new files be written directly to this area by MON-2 subsystems.

H.2.1 Common Services (cs) Subdirectory

The common services (**cs**) subdirectory of the NMCFS contains common services files that a MON-2 subsystem may need to access. These files include MCIS and UDS files, Subsystem Controller Framework (SCF) files, CDT executables, MCIS server lists files, and various NCT script files. These files and the NMCFS directory structures into which these files will be stored are shown in Figure H-2 and described below.

- **MCIS**. The MCIS runtime libraries will be located at the root level of this directory. Additionally, other MCIS-related subdirectories will be provided in this directory to support standalone testing by MON-2 subsystems. See Section H.4 for a description of these subdirectories.

- **UDS**. The UDS runtime libraries will be located within this directory.

- **SCF**. This directory will contain the 'sitename' and 'getcellname' scripts (which allow a MCIS user to determine the name of the site at which it is operating or the name of the MCIS cell in which it is operating, respectively).

- **CDT**. The CDT executables will be located within this directory.

- **ServerLists**. This directory will contain the MCIS 'servers' list (which allows a MCIS user to find the MDS server(s) within the local cell), the ‘Local_Nameservers_Config_File’ (which allows a MCIS users to find the name server host(s) within the local cell), and the ‘Remote_Nameservers_Config_File’ (which allows a MCIS name server host to find other name server hosts across cells).

- **NMCFS Scripts**. This directory will contain the NMCFS ‘cmd’ scripts which allow a MON-2 subsystem to obtain a directory listing from the NMCFS (nmcfsls), copy file(s) from the NMCFS onto another host (nmcfsget), copy file(s) from another host into the NMCFS (nmcfsput), and remove a file from the NMCFS (nmcfsrm). (See the NMCFS User’s Guide.)

H.2.2 Specifications (specs) Subdirectory

As illustrated in Figure H-3, the **specs** subdirectory of the NMCFS contains a separate directory structure for each DSN subsystem (legacy and MON-2). For instance, the APC, MDA and UPL subsystems each have a separate directory structure in the **specs** subdirectory. A subsystem’s program identification number (Program ID) is used as the name for the subsystem’s directory. The Program ID naming standards defined in Section 6.2.6 of this document will be followed and the name assigned to the subsystem’s directory must be the same as the PGMID monitor data item published by (or for) the subsystem under its connection functional address.

Within each subsystem directory will be one or more software version subdirectories. The exact number of version sub-directories needed will depend upon the number of separate versions of subsystem software that are operational at a site. A separate version subdirectory is required for each operational version of the subsystem. For instance, if there are two operational versions of XYZ subsystem at a site, then two version directories are needed within the NMCFS for subsystem XYZ, one for each operational version. A subsystem’s Version ID is used as the name for the subsystem’s corresponding version subdirectory. The Version ID naming standards defined in Section 6.2.6 of this document will be followed and the name assigned to a subsystem’s version directory must be the same as the VersionID monitor data item published by (or for) the subsystem under its functional address.

(The appropriate directory structures will be created within the NMCFS for each DSN subsystem that exists at the time the NMCFS is initially delivered. As new subsystems are delivered and as new versions of existing subsystems are delivered, new subdirectories will need to be created within the NMCFS to contain the subsystem’s files. This is a function performed by the local site NMCFS system administrator.)

H.2.2.1 SDD Data Definition Files

Within each subsystem (Program ID) version directory are located **bin**, **help**, **bitmap** and **data** subdirectories as shown in Figure H-3. The SDD Data Definition Files for a MON-2 (or legacy) subsystem consist of the mdspecs.dat, directives.dat, disp.dat and mondata.html files. These files are produced by subsystem CDEs and will be stored within the NMCFS as indicated in Figure H-3. Procedures for moving these files into the NMCFS are provided in Section 12. (Note that a subsystem may not require all of the SDD Data Definition Files shown in Figure H-3. For instance, if a subsystem does not have a directive interface with the NMC, then a directives.dat file will not be required for the subsystem. Similarly, if a subsystem does not provide displays, then the disp.dat file will not be needed.)

(Available SDD Data Definition Files will be stored in the NMCFS for each DSN subsystem that exists at the time the NMCFS is initially delivered. Subsequent delivery(s) of a subsystem's SDD Data Defintion files will be the responsibility of the subsystem CDE, each time a new version of the subsystem is delivered. These files will be installed by the local NMCFS System Administrator based upon instructions in the subsystem's RDD.)

H.2.2.2 UDS-Related Files

A subsystem's UDS-related files (i.e., UDS display executables, UDS help and UDS icon/bitmap files) will be stored within the NMCFS as shown in Figure H-3. (The NMC currently assumes that a subsystem's UDS files are stored at the same relative directory path as the subsystem's mdspecs.dat file. Specifically, a subsystem's UDS files will be stored in the **bin**, **help** and **bitmap** directories, as indicated in Figure H-3, at the same level as the **data** directory containing the subsystem's mdspecs.dat file.) Procedures for moving these files into the NMCFS are provided in Section 12.

(Available UDS files will be stored in the NMCFS for each MON-2 subsystem that exists at the time the NMCFS is initially delivered. Subsequent delivery(s) of a subsystem's UDS files will be the responsibility of each subsystem CDE, each time a new version of the subsystem is delivered. These files will be delivered via SPMC and will be installed by the local NMCFS System Administrator based upon instructions in the subsystem's RDD.)

H.3 Subsystem Files

In addition to the (shared) files described above, MON-2 subsystems will also be allowed to store (and retrieve) subsystem-specific files on (and from) specially designated file space in the NMCFS. For example, MON-2 subsystems may want to store subsystem checkpoint files on the NMCFS, or other subsystem files that need to be made available to multiple subsystem hosts. Access to this file space will be via the NMCFS 'cmd' commands provided for this purpose. (See the NMCFS SOM for more information on these file transfer commands.)

Storage space for MON-2 subsystems will be located within the NMCFS **subsystem_files** directory. A separate subdirectory will be created for each requesting MON-2 subsystem as shown in Figure H-4. As above, the Program ID of a subsystem will be used as the name of the subsystem's directory. (The Program ID naming standards defined in Section 6.2.6 will be followed.) The content and organization of all subsystem directories/files below this level are at the discretion of the MON-2 subsystem.

Each requesting MON-2 subsystem will initially be allocated 10 MBs of storage space within the NMCFS. Any additional storage space required beyond this amount must be negotiated in the NMC interface agreement with the subsystem.

(A separate NMCFS account will be setup for each requesting MON-2 subsystem in order to control access to the subsystem's subdirectory and to control storage space usage. It is the responsibility of the

MON-2 subsystem CDE to initiate the request for this account with the NMC OE. The local NMCFS System Administrator will be responsible for setting up the NMCFS account and subsystem subdirectory, when requested to do so by the NMC OE.)

H.4 Standalone MON-2 Subsystem Testing (DTF-21 Only)

MON-2 subsystems may have need to test at DTF-21 with a MCIS infrastructure in a standalone mode, i.e., testing with a dedicated MCIS, but without the NMC subsystem. To facilitate this type of testing, the MCIS executables are provided on the NMCFS at DTF-21. However, since other MON-2 subsystems (including the NMC) may be concurrently testing on the network at DTF-21, the procedures described below must be followed to setup a standalone MCIS test environment to avoid conflicts with other MON-2 subsystem testing. **This type of standalone MCIS testing is only allowed at DTF-21 and is not authorized at the DSCCs or at other operational facilities**. (Standalone MCIS testing can also occur in MON-2 subsystem development environments. Since the NMCFS will not be available in these development environments, MCIS executables should be obtained from SPMC to support testing in these environments.)

Within the MCIS directory on the DTF-21 NMCFS (See Figure H-5) will be one or more MCIS version subdirectories. The exact number of version subdirectories available will depend upon the number of MCIS versions currently being tested at DTF-21. The Version ID naming standards defined in Section 6.2.6 of this document will be followed.

The MCIS executables and the Standalone_Nameserver_Script needed to setup a standalone MCIS will be provided in each MCIS version subdirectory. The Standalone_Nameserver_Script must be used to setup the standalone MCIS to avoid conflicts with other MON-2 subsystem testing at DTF-21. (If the Standalone_Nameserver_Script is not used, dual MCIS usage by another MON-2 subsystem at DTF-21 may cause difficult to diagnose MCIS errors in both test environments.) A standalone MCIS may be executed directly from the DTF-21 NMCFS or the MCIS executables and the Standalone_Nameserver_Script may be downloaded to a separate subsystem host and executed from that host.

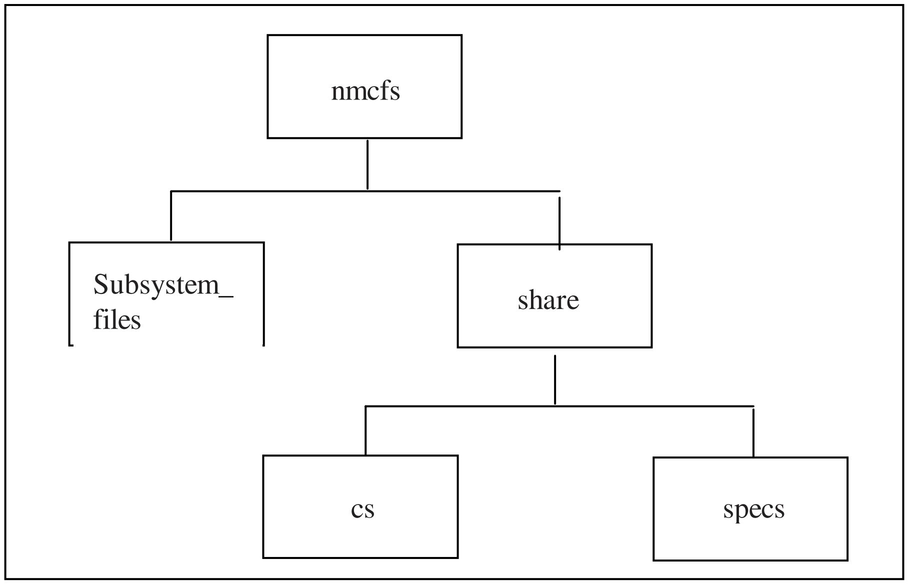

Figure H-1. **MON-2 Subsystem Access To The NMCFS**

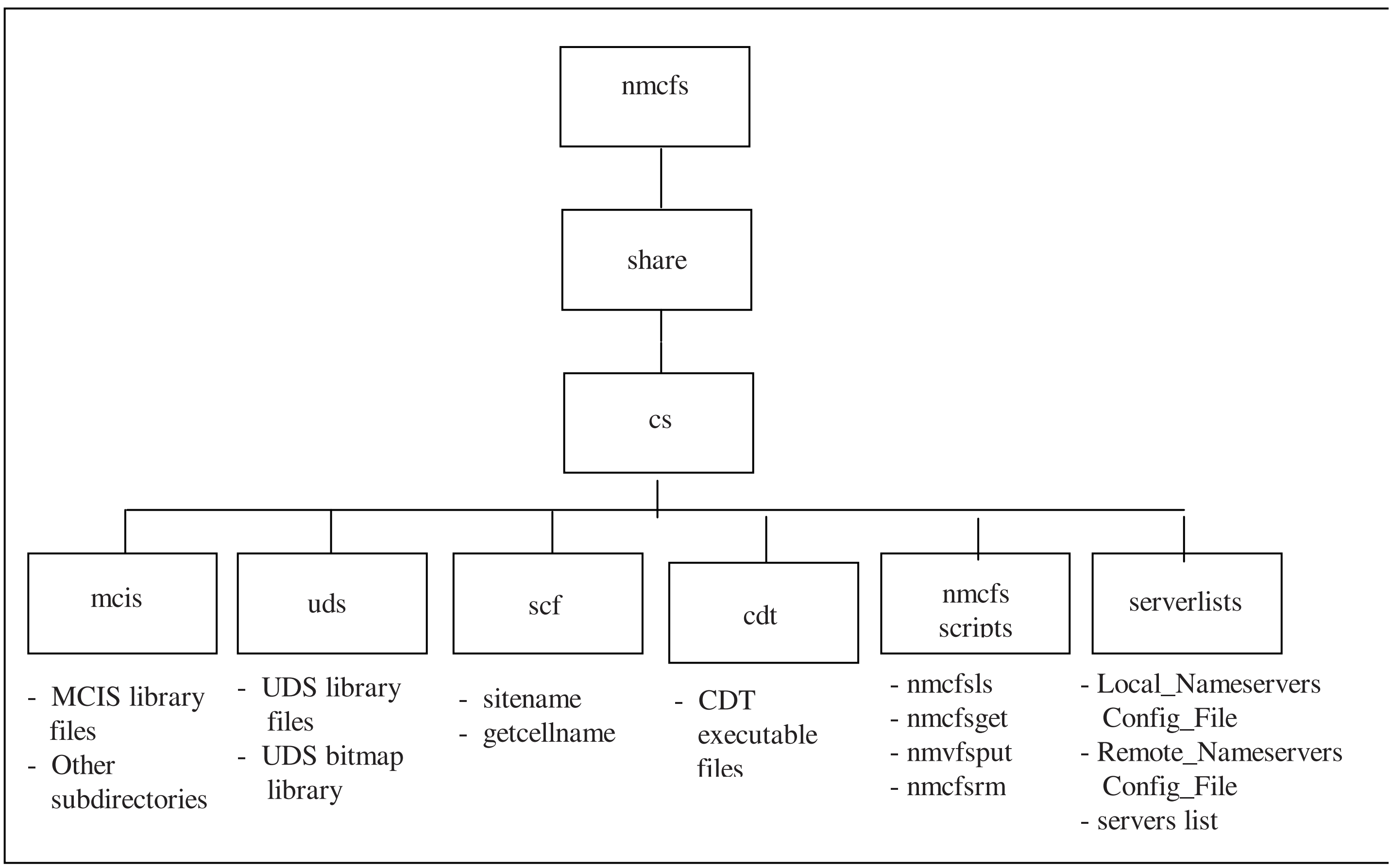

Figure H-2. NMCFS Common Services (CS) Directory Structure

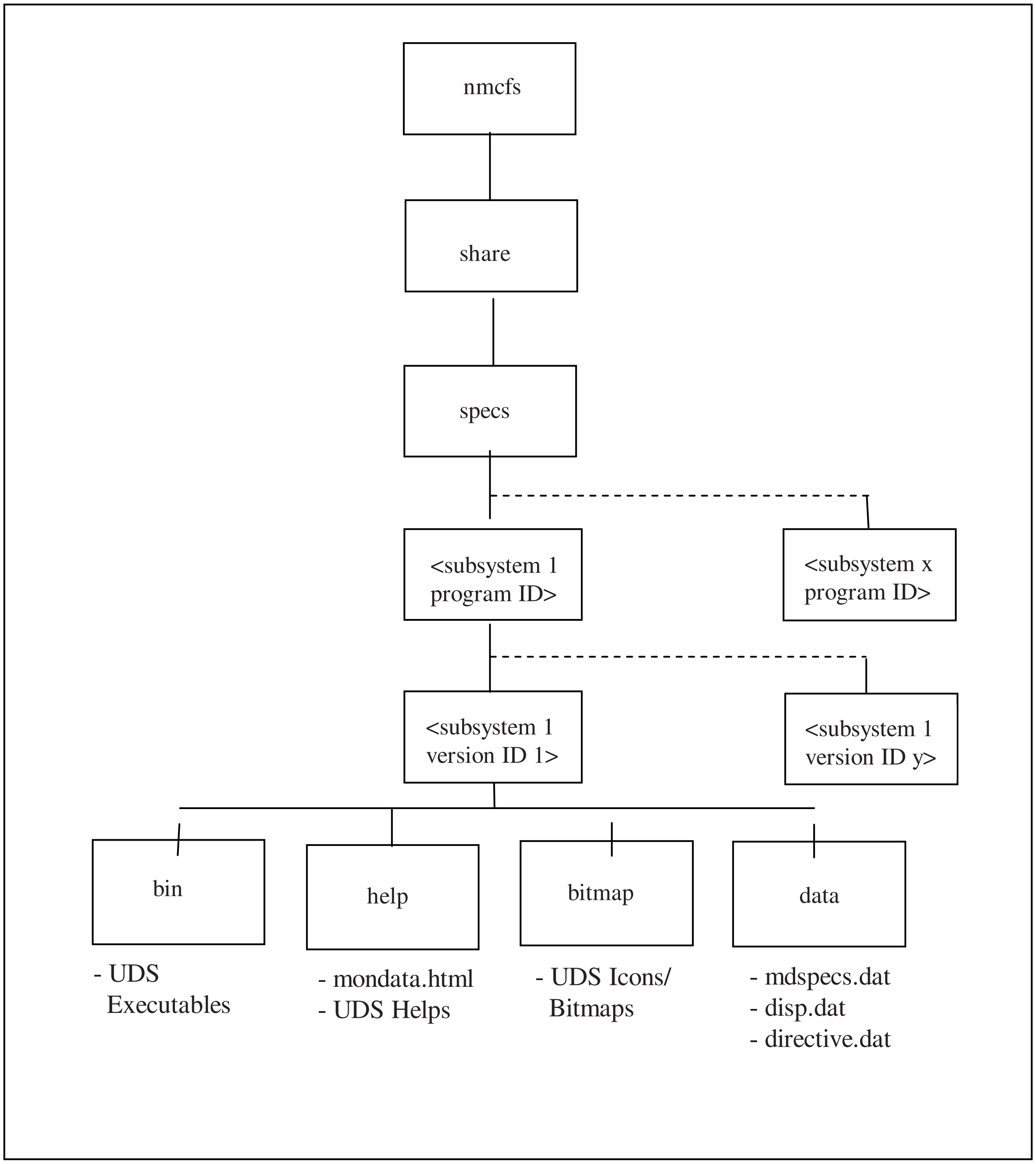

Figure H-3. NMCFS Specs Directory Structure

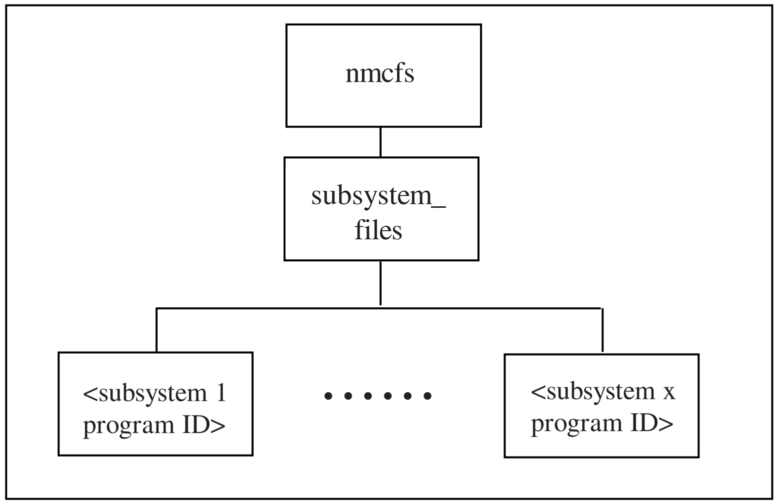

Figure H-4. NMCFS Subsystem_Files Directory Structure

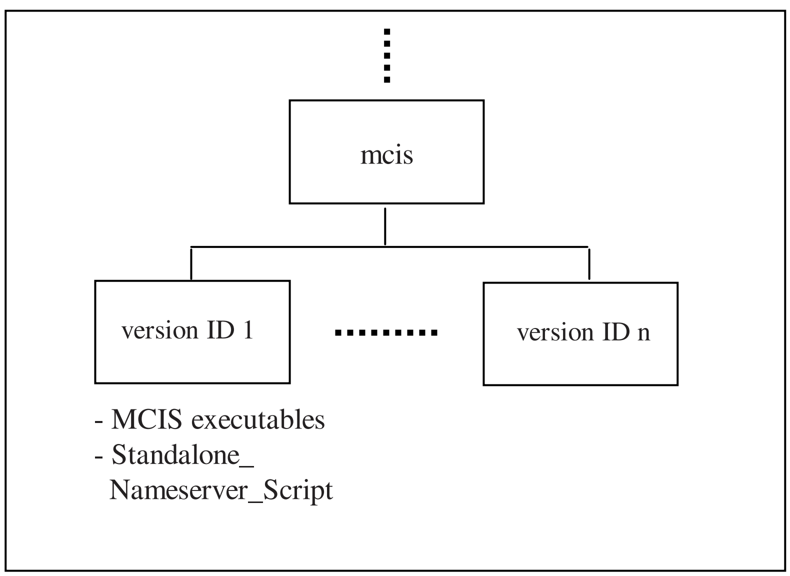

Figure H-5. NMCFS MCIS Directory Structure

This page is intentionally left blank.

