JMF ICS

Version 1.5

Date: 2015-05-11

File: ICS-JMF-1.5-150511.docx, *.pdf*

MIS WG

Abstract

This document, the JMF ICS, defines the conformance requirements that relate to communication using JMF. This document specifies one ***Conformance Level*** of Conformance Requirements.

This version applies to interactions using [JDF1.5].

Copyright Notice

Licenses and Trademarks

Table of Contents

Tables

1 Introduction

This ICS builds upon the hot folder based workflow specified by the [Base-ICS]. This ICS adds communication using JMF messaging. JMF provides a command and control language that facilitates interaction between Managers and Workers.

This ICS is product sector independent, meaning that it can be used in any workflow where JMF messaging is desired.

Note: in this document, terms identified with ***bold-italics*** are defined in the [Base-ICS] if their definition does not appear immediately in this document.

Using [Base-ICS] terminology, the [JMF-ICS] specifies the generic parts of the ***Manager Interface*** when it communicates with the ***Worker Interface***. It also specifies the corresponding generic parts of the Worker Interface when it communicates with a Manager Interface.

This ICS includes specifications of JMF Messages used for:

- Determining Supported Messages and submission methods.
- Device and Controller discovery.
- Dealing with Job submission and return.
- General Queue entry handling such as requesting, aborting, holding, suspending, resuming, resubmitting, and removing Queue entries.
- Querying and signaling the status of employees, Devices, and Jobs.

This ICS defines exactly one conformance level which is summarized as follows:

Level 1: Defines requirements for:

- Creating and managing Persistent Channels.
- Using JMF Messages for Job submission and return.
- Bidirectional Support for a subset of JMF messaging between the Manager and Worker.
- Several Messages that MAY be Supported by the Manager and Worker.

2 Glossary

This section defines terminology used in this ICS. The terms appear in alphabetic order. If a word is in ***bold-italic***, it is defined in [Base-ICS] if it is not defined elsewhere in this section. Elsewhere in ICSs, the first letter of each word of these terms is capitalized.

Table 1: Glossary

Term	Definition
Signaler	A ***Manager*** or ***Worker*** in a role where it consumes a subscription request, i.e. reads and processes a Query or Registration containing a Subscription Element.
Subscriber	A ***Manager*** or ***Worker*** in a role where it writes a Query or Registration that includes a Subscription Element.

3 Conformance Levels

This ICS specifies exactly one ***Conformance Level*** of ***Conformance Requirements*** – a subset of [JDF1.5]. These Conformance Requirements are common to all Product-Sector ICSs. Table 2 below specifies the Conformance Requirements for Manager Interfaces and Worker.

See Appendix A "*How to Read ICS Documents*" in [Base-ICS] for an explanation of Conformance Tables.

To be conformant to this ICS, a Manager SHALL conform to the Manager part and a Worker SHALL conform to the Worker part of the ICSs and levels specified in Table 2 below.

Table 2: Conformance Levels

Level of this ICS	[Base-ICS]			Description
1	2			JMF used for Job submission, Job return and Queue management. A Manager SHALL be able to interpret MIME packages and access their parts. A Worker SHALL be able to create MIME packages.

4 Conformance Tables – Resources

The table in this section shows the Conformance Requirements for the **NodeInfo** Resource.

4.1 NodeInfo Resource

Table 3 specifies the conformance requirements for Attributes and Elements for the **NodeInfo** Resource, whether it is linked to a JDF Root Node or a JDF Subnode.

Table 3: NodeInfo

From: [JDF1.5] Table 8-165

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
TargetRoute	!w			r?			Both JMF/Query/QueueSubmissionParams/@ReturnJMF and JMF/QueueSubmissionParams/@ReturnURL SHALL be supplied, and they both supersede *@TargetRoute*. Workers SHALL use JMF/QueueSubmissionParams/@ReturnJMF or JMF/QueueSubmissionParams/@ReturnURL in lieu of *@TargetRoute*.

5 Conformance Tables – JMF Instances

This section contains Conformance Tables that specify conformance requirements for JMF Messages:

5.1 Manager/Worker versus Producer/Consumer

For sections 5.1 and 5.3, the Conformance Table columns for Manager and Worker are relabeled ***Producer*** and ***Consumer***. When a Manager sends a JMF Message to a Worker, the Manager is the Producer and the Worker is the Consumer. When a Worker sends a JMF Message to a Manager, the Worker is the Producer and the Manager is the Consumer.

5.2 JMF Root Node

This ICS specifies the JMF Messages that are common to all areas where JMF is used.

This ICS also describes the conformance requirements for all of the Messages identified in Table 14: List of JMF Messages.

This ICS describes both the JMF Message formats and the transfer protocol. Managers and Workers SHALL be able to use HTTP, and MAY use HTTPS, for sending and receiving Messages. For the remainder of this document, whenever HTTP is referenced, HTTPS MAY also be used.

This section contains the Conformance Table for the JMF Element that is the Root Node of any JMF Message.

Table 4: JMF

From: [JDF1.5] Table 5-1

Root Node of: JMF Instance

Name or Value	*Producer*			*Consumer*			Description
Level ➔	1	2	3	1	2	3	
DeviceID	w?			r			The value identifies the intended recipient of the JMF Element. If the Message is being sent to a Device, this Attribute indicates that Device's *@DeviceID*. If the Message is being sent to a Controller, this Attribute identifies that Controller's *@ControllerID*. **r-Test:** If the immediate Consumer or any subsequent recipient does not recognize the value of this Attribute, it SHALL reject the Message with *@ReturnCode* = "*121*", "Unknown DeviceID".
ICSVersions	w←			r?			Has the same semantics as JDF/@ICSVersions. See [Base-ICS].
JMF_L1-1.5	w			r?			Specifies that the JMF Element conforms to [JMF-ICS] level 1.
all remaining values	w←			r?			Values specified in other ICSs.
MaxVersion	w			r			
1.5	w			r			A value higher than 1.5 MAY be specified. **r-Test:** Consumer does not write any Elements or Attributes in the response that do not conform to the version specified.
ResponseURL	!w			r?			This ICS does not allow JMF messaging via hot folders.

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
							Note: deprecated in [JDF1.5].
SenderID	w			r?			Identifies the sender. If the sender is a Device, this Attribute SHALL contain the Device's *@DeviceID*. If the sender is a Controller, this Attribute SHALL contain the Controller's *@ControllerID*.
TimeStamp	w			r?			Date and time the JMF is sent.
Version	w			r?			
1.5	w			r?			
xmlns	w←			r?			The namespace for JDF may be the default namespace or any prefixed namespace.
http://www.CIP4.org/JDFSchema_1_1	w			r?			**Note:** that for all 1.x versions of [JDF1.5], the namespace URI is the same.
xmlns:xsi	w←			r?			SHALL be present in the JMF Root of a JMF Message.
http://www.w3.org/2001/XMLSchema-instance	w			r?			
Message	w			r			Abstract Element(s). See Table 14: List of JMF Messages. See Table 5: Abstract Message.

5.3 JMF Message Families

This section contains Conformance Tables that specify Conformance Requirements for the 6 JMF Message Families.

5.3.1 Abstract Message

Table 5: Abstract Message

From: [JDF1.5] Table 5-2

Referenced by: JMF

Superclass of: Query, Command, Response, Acknowledge, Signal

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
ID	w			r←			Unique identifier of Message Values SHALL be unique for all Messages initiated by the same sender. In other words, the combination of JMF/@SenderID and this *@ID* SHALL be unique.

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
							If the message is a Query or Command Message, Consumers SHALL read @*ID*. **r-Test:** The response to the Query or Command specifies a @*refID* which matches the @*ID* of the Query or Command.
Time	w←			r?			Time at which the Message was generated. This Attribute SHALL be specified unless the time is the same as JMF/@TimeStamp.
Type	w			r			**r-Test:** Any actions taken or response generated corresponds to the message type identified by this value.
xsi:type	w			r?			Helps JDF Schema-aware implementations to identify specific Message types.
<Message Family> <value of /JMF/Message/@Type>	w			r?			Example: "*CommandSubmitQueueEntry*". • "*Command*" is the Message Family Name, i.e. the actual Element name of the Message. See column 1 of Table 14: List of JMF Messages. • "*SubmitQueueEntry*" is the value of /JMF/Message/@Type. See Table 70: Command – SubmitQueueEntry.

5.3.2 Query

Table 6: Query

From: [JDF1.5] Table 5-4

Subclass of: Abstract Message

Class for: Query – KnownDevices, Query – KnownMessages, Query – KnownSubscriptions, Query – QueueStatus, Query – SubmissionMethods

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
AcknowledgeURL	w←			r?			If a Producer allows the Consumer to send an Acknowledge, the Producer SHALL supply this Attribute.
QueryTypeObj	w←			r			Abstract Element that is a placeholder for any descriptive Elements that provide details required for the Query. See [JDF1.5].
Subscription	w←			r←			A Consumer SHALL Support this Element for establishing a Persistent Channel. The Message types that SHALL support

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
							Subscriptions is defined in further ICS documents. See e.g. [MIS-ICS]. See Table 11: Subscription.

5.3.3 Response

In Table 7, the Producer is the producer of the Response and the Consumer is the consumer of the original Response. Thus the roles have been exchanged with respect to the original Command or Query tables.

Table 7: Response

From: [JDF1.5] Table 5-5

Subclass of: Abstract Message

Class for: Response – AbortQueueEntry, Response – HoldQueueEntry, Response – KnownDevices, Response – KnownMessages, Response – KnownSubscriptions, Response – QueueStatus, Response – RemoveQueueEntry, Response – RequestQueueEntry, Response – ResourcePull, Response – ResubmitQueueEntry, Response – ResumeQueueEntry, Response – ReturnQueueEntry, Response – SetQueueEntryPosition, Response – SetQueueEntryPriority, Response – StopPersistentChannel, Response – SubmissionMethods, Response – SubmitQueueEntry, Response – SuspendQueueEntry

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
Acknowledged	w←			r?			The Producer SHALL supply this Attribute with a value of "*true*" when it will send an asynchronous Acknowledge later.
refID	w←			r?			See [JDF1.5].
ReturnCode	w←			r?			If an error occurs, a Producer SHALL write a nonzero value. See [JDF1.5] Appendix D for a list of Supported values. A Consumer SHALL be able to detect nonzero values.
Subscribed	w←			r?			The Producer SHALL supply this Attribute if the Query contained a Subscription (see Table 6: Query).
true	w←			r?			A Producer SHALL accept Subscriptions for Persistent Channels in Queries.

5.3.4 Signal

Table 8: Signal

From: [JDF1.5] Table 5-6

Subclass of: Abstract Message

Name or Value	*Producer*			*Consumer*			Description
Level ➔	1	2	3	1	2	3	
refID	w←			r?			If the Signal is a result of a Subscription, the Producer SHALL supply the ID of the Subscription Query.
QueryTypeObj	w			r?			If the Signal is the result of a hardwired configuration, the Producer SHALL supply a QueryTypeObj that specifies the corresponding query parameters that apply to this Signal. If the Signal is the result of a subscription, the same QueryTypeObj that was in the subscription (or one with additional context added) SHALL be supplied here. See [JDF1.5].

5.3.5 Command

Table 9: Command

From: [JDF1.5] Table 5-9

Subclass of: Abstract Message

Class for: Command – AbortQueueEntry, Command – HoldQueueEntry, Command – RemoveQueueEntry, Command – RequestQueueEntry, Command – ResourcePull, Command – ResubmitQueueEntry, Command – ResumeQueueEntry, Command – ReturnQueueEntry, Command – SetQueueEntryPosition, Command – SetQueueEntryPriority, Command – StopPersistentChannel, Command – SubmitQueueEntry, Command – SuspendQueueEntry

Name or Value	*Producer*			*Consumer*			Description
Level ➔	1	2	3	1	2	3	
AcknowledgeURL	w←			r?			If a Producer allows the Consumer to send an Acknowledge, the Producer SHALL supply this Attribute.
CommandTypeObj	w←			r			Abstract Element that is a placeholder for any descriptive Elements that provide details required for the Command. See [JDF1.5].

5.3.6 Acknowledge

Table 10: Acknowledge

From: [JDF1.5] Table 5-10

Subclass of: Abstract Message

Class for: Acknowledge – AbortQueueEntry, Acknowledge – ResumeQueueEntry, Acknowledge – ReturnQueueEntry, Acknowledge – SubmitQueueEntry, Acknowledge – SuspendQueueEntry

Name or Value	*Producer*			*Consumer*			Description
Level ➔	1	2	3	1	2	3	
AcknowledgeType	w←			r?			
all values	w←			r?			
refID	w			r?			ID of the Message being Acknowledged.
ReturnCode	w←			r?			A non-zero return code indicates failure. If an error can occur, a Producer SHALL be capable of writing at least one nonzero value. See [JDF1.5] Appendix D for a list of Supported values. A Consumer SHALL be able to detect nonzero values. How a Consumer handles nonzero value is implementation dependent.

5.4 JMF Handshaking

5.4.1 Persistent Channels

5.4.1.1 Creating Persistent Channels

A Manager or Worker sends JMF Signals to another Controller or Device in what is called a Persistent Channel. Persistent Channels are created by sending a Query or Registration Message that includes a Subscription Element.

A ***Subscriber*** SHALL send the Query or Registration Message as a separate JMF Query or Registration Message via HTTP. A ***Signaler*** SHALL support Messages with subscriptions sent via HTTP.

5.4.1.2 Managing Persistent Channels

A subscriber SHOULD monitor the status of Persistent Channels by using the KnownSubscriptions Query. A Signaler SHALL respond to a KnownSubscriptions Query with a KnownSubscriptions Response that contains one SubscriptionInfo for every Persistent Channel that matches the Query. For details see Section 6.5.1 Query – KnownSubscriptions and Section 6.5.2 Response – KnownSubscriptions.

5.4.1.3 Closing Persistent Channels

Closing a Persistent Channel means that additional Messages related to the subscription SHALL NOT be created. Messages previously created MAY still be pending delivery. ***Subscriber*** SHALL be able to receive and process further messages at the URL defined in the original subscription.

A Persistent Channel SHALL be closed by sending a StopPersistentChannel Command.

All Subscription Producers and Consumers SHALL Support StopPersistentChannel Commands.

5.4.2 Asynchronous Acknowledges

A Consumer of a JMF Command or Query Message SHALL respond by returning a ResponseTypeObj by one of two methods: synchronously or asynchronously. With certain types of Command or Query Messages, the Producer of the Message SHALL Support the use of asynchronous acknowledges by the Consumer of the Message. These Messages can be identified by the following means:

- The Producer of the Message SHALL write the *@AcknowledgeURL* Attribute in the Command or Query Element
- The Producer of the Command or Query SHALL read and Support the Response Element's *@Acknowledged* Attribute with a value of "*true*" in the response to the Command or Query.
- The Producer of the Command or Query SHALL Support an Acknowledge Message of the same type.

If a Consumer responds synchronously, it SHALL:

- Include a ResponseTypeObj in a Response Element,
- send the JMF Message in the HTTP response channel.

If a Consumer responds asynchronously, it:

- SHALL include the Response Element, with the *@Acknowledged* Attribute set to true. The Response Element SHOULD NOT contain a ResponseTypeObj.
- SHALL send the JMF Message in the HTTP response channel.
- MAY include a ResponseTypeObj in an Acknowledge Element,
- SHALL send one or more Acknowledge Messages as separate JMF Messages. If this ICS requires the response to the command or query being acknowledged to include one or more ResponseTypeObj Elements, the final Acknowledge Message that is sent SHALL include all required ResponseTypeObj Elements.

For a message that does not require Support of asynchronous acknowledges, if the Producer does not Support asynchronous responses to the command or query, and has not provided an *@AcknowledgeURL*, the Consumer SHALL process the message using exactly one of the following methods:

- The message SHALL be fully processed and a full response sent within the timeframe of an http connection.

OR

- A response SHALL be returned which indicates *@ReturnCode* = "*11*", which is a new error code that will appear in JDF 1.5 that will be defined as "Synchronous mode not supported for message."

5.4.3 Subscription

Also see other ICSs for Subscription contents.

5.4.3.1 Subscription - Query

Table 11: Subscription - Query
From: [JDF1.5] Table 5-12
Referenced by: Query

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
MinDelayTime	w?			r			**r-Test:** Signals related to this Subscription are not sent any more frequently than this interval. Reliable signals SHALL NOT be retried more frequently than the interval specified by *MinDelayTime*.
RepeatTime	w?			r			*@RepeatTime* SHALL NOT be less than *@MinDelayTime*. **r-Test:** Signals are generated at the interval specified (+/- 10%).
URL	w			r			**r-Test:** Signals are delivered to the specified URL.
ObservationTarget	w?			r?			See Table 12: ObservationTarget.

5.4.3.1.1 ObservationTarget

Table 12: ObservationTarget
From: [JDF1.5] Table 5-13
Referenced by: Subscription - Query

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
ObservationPath	w			r			**r-Test:** When the value specified by the XPath changes a signal is sent.

5.4.3.2 Subscription - KnownSubscriptions Response

Table 13: Subscription - KnownSubscriptions Response
From: [JDF1.5] Table 5-12
Referenced by: SubscriptionInfo

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
RepeatTime	w←			r?			Producer SHALL supply if messages are

Name or Value	*Producer*			*Consumer*			Description
Level ➔	1	2	3	1	2	3	
							generated at regular time intervals.
URL	w			r?			

6 Conformance Tables – JMF Messages

For some Messages, the Conformance-Table columns for Manager and Worker are relabeled ***Producer*** and ***Consumer***. When a Manager sends a JMF Message to a Worker, the Manager is the Producer and the Worker is the Consumer. When a Worker sends a JMF Message to a Manager, the Worker is the Producer and the Manager is the Consumer.

- If the Manager column contains a "write" for a Message, then the Manager is a Producer and the Worker is a Consumer for that Message.
  For example, for a SubmitQueueEntry Command, the Manager that is acting as a Producer sends a Message to a Worker acting as a Consumer.
- If the Manager column contains a "read" for a Message, then the Worker is a Producer and the Manager is a Consumer for that Message.
  For example, for a SubmitQueueEntry Response, the Worker that is acting as a Producer sends a Message to a Manager acting as a Consumer.

Table 14 specifies the Conformance Requirements for pairs of Message Type (/JMF/Message/@Type Attribute) and Message Family.

This ICS does not limit other ICSs from specifying conformance for additional Message Types.

Note: Table 14 provides the mapping between the Manager/Worker and the Producer/Consumer. In the table below, no **r-Test**s are provided. The test for read conformance is that the Manager or Worker meets the requirements of the tables corresponding to each message.

Table 14: List of JMF Messages

From: [JDF1.5] Table 5-3

Message Type	Family	Manager			Worker			Description
	Level ➔	1	2	3	1	2	3	
AbortQueueEntry	Command	w			r			See Table 15: Command – AbortQueueEntry.
AbortQueueEntry	Response	r			w			See Table 17: Response – AbortQueueEntry.
AbortQueueEntry	Acknowledge	r?			w?			See Table 18: Acknowledge – AbortQueueEntry.
HoldQueueEntry	Command	w?			r?			See Table 19: Command – HoldQueueEntry.
HoldQueueEntry	Response	r?			w?			See Table 21: Response – HoldQueueEntry.
KnownDevices	Query	w r			r w?			See Table 22: Query –

Message Type	Family	Manager			Worker			Description
	Level ➔	1	2	3	1	2	3	
								KnownDevices.
KnownDevices	Response	r w			w r			See Table 24: Response – KnownDevices.
KnownMessages	Query	w← r			r w?			Managers SHALL, and Workers SHOULD, provide the ability for a user/operator to initiate sending a KnownMessages Query. See Table 28: Query – KnownMessages.
KnownMessages	Response	r w			w r			See Table 30: Response – KnownMessages.
KnownSubscriptions	Query	w←			r			Manager SHALL be capable of sending if it Supports a Query with Subscriptions. Worker SHALL Support if it accepts a Query with Subscriptions. See Table 32: Query – KnownSubscriptions
KnownSubscriptions	Response	r			w			Table 34: Response – KnownSubscriptions
QueueStatus	Query	w?			r?			See Table 36: Query – QueueStatus.
QueueStatus	Response	r?			w?			See Table 37: Response – QueueStatus.
RemoveQueueEntry	Command	w			r			See Table 38: Command – RemoveQueueEntry.
RemoveQueueEntry	Response	r			w			See Table 40: Response – RemoveQueueEntry.
RequestQueueEntry	Command	r?			w?			See Table 41: Command – RequestQueueEntry.
RequestQueueEntry	Response	w?			r?			See Table 43: Response – RequestQueueEntry.
ResourcePull	Command	r?			w?			See Table 44: Command – ResourcePull
ResourcePull	Response	w?			r?			See Table 46: Response – ResourcePull
ResubmitQueueEntry	Command	w?			r?			See Table 47: Command – ResubmitQueueEntry.
ResubmitQueueEntry	Response	r?			w?			See Table 49: Response – ResubmitQueueEntry.
ResumeQueueEntry	Command	w?			r←			If Worker Supports

Message Type	Family	Manager			Worker			Description
	Level ➔	1	2	3	1	2	3	
								HoldQueueEntry Message or QueueSubmissionParams/@Hold, the Worker SHALL Support the ResumeQueueEntry Message. See Table 50: Command – ResumeQueueEntry.
ResumeQueueEntry	Response	r?			w←			If Worker Supports HoldQueueEntry Message or QueueSubmissionParams/@Hold, the Worker SHALL Support the ResumeQueueEntry Message. See Table 52: Response – ResumeQueueEntry.
ResumeQueueEntry	Acknowledge	r?			w?			See Table 53: Acknowledge – ResumeQueueEntry.
ReturnQueueEntry	Command	r			w			See Table 54: Command – ReturnQueueEntry.
ReturnQueueEntry	Response	w			r			See Table 56: Response – ReturnQueueEntry.
ReturnQueueEntry	Acknowledge	w?			r?			See Table 57: Acknowledge – ReturnQueueEntry.
SetQueueEntryPosition	Command	w?			r?			See Table 58: Command – SetQueueEntryPosition.
SetQueueEntryPosition	Response	r?			w?			See Table 60: Response – SetQueueEntryPosition.
SetQueueEntryPriority	Command	w?			r←			If Worker Supports QueueSubmissionParams/@Priority in SubmitQueueEntry command, it SHALL Support the SetQueueEntryPriority Message. See Table 61: Command – SetQueueEntryPriority.
SetQueueEntryPriority	Response	r?			w←			If Worker Supports QueueSubmissionParams/@Priority in SubmitQueueEntry Command, it SHALL Support the SetQueueEntryPriority

Message Type	Family	Manager			Worker			Description
	Level ➔	1	2	3	1	2	3	
								Message. See Table 63: Response – SetQueueEntryPriority.
StopPersistentChannel	Command	w←			r			Manager SHALL be capable of sending if it Supports a Query with Subscriptions. Worker SHALL Support if it accepts a Query with Subscriptions. See Table 64: Command – StopPersistentChannel.
StopPersistentChannel	Response	r			w			See Table 66: Response – StopPersistentChannel.
SubmissionMethods	Query	w? r			r w?			See Table 67: Query – SubmissionMethods.
SubmissionMethods	Response	rw			wr			See Table 68: Response – SubmissionMethods.
SubmitQueueEntry	Command	w			r			See Table 70: Command – SubmitQueueEntry.
SubmitQueueEntry	Response	r			w			See Table 72: Response – SubmitQueueEntry.
SubmitQueueEntry	Acknowledge	r←			w?			Support is required only if the Manager includes a Command/ @AcknowledgeURL See Table 73: Acknowledge – SubmitQueueEntry.
SuspendQueueEntry	Command	w?			r?			See Table 74: Command – SuspendQueueEntry.
SuspendQueueEntry	Response	r?			w?			See Table 76: Response – SuspendQueueEntry.
SuspendQueueEntry	Acknowledge	r?			w?			See Table 77: Acknowledge – SuspendQueueEntry.

6.1 AbortQueueEntry

6.1.1 Command – AbortQueueEntry

Table 15: Command – AbortQueueEntry
From: [JDF1.5] Table 5-80
Instance of: Command

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
AcknowledgeURL	w←			r?			If a Manager allows the Worker to send an Acknowledge, the Manager SHALL supply this Attribute. A Worker MAY respond synchronously if it can abort Queue entries within the time frame of an http connection. If a Worker chooses to acknowledge an AbortQueueEntry, it SHALL send an Acknowledge Message once the Queue entry has been aborted.
ID	w			r			Required in [JDF1.5].
Type	w			r			
AbortQueueEntry	w			r			
xsi:type	w			r?			
CommandAbortQueueEntry	w			r?			
AbortQueueEntryParams	w			r			See Table 16: AbortQueueEntryParams.

6.1.1.1 AbortQueueEntryParams

Table 16: AbortQueueEntryParams
From: [JDF1.5] Table 5-81
Referenced by: Command – AbortQueueEntry

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
EndStatus	w			r			Required in [JDF1.5].
QueueFilter	w←			r			QueueFilter SHALL be specified to select specific QueueEntry. If QueueFilter is NOT provided, the Command SHALL be applied to all QueueEntry elements of the Queue. See Table 80: QueueFilter.

6.1.2 Response – AbortQueueEntry

Table 17: Response – AbortQueueEntry
From: [JDF1.5] Table 5-80
Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Acknowledged	r			w←			
true	r			w←			
false	r			w←			
ID	r			w			Required in [JDF1.5].
ReturnCode	r			w			
Type	r			w			
AbortQueueEntry	r			w			
xsi:type	r?			w			
ResponseAbortQueueEntry	r?			w			

6.1.3 Acknowledge – AbortQueueEntry

Table 18: Acknowledge – AbortQueueEntry
From: [JDF1.5] Table 5-80
Instance of: Acknowledge

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
AbortQueueEntry	r			w			
xsi:type	r?			w			
AcknowledgeAbortQueueEntry	r?			w			

6.2 HoldQueueEntry

6.2.1 Command – HoldQueueEntry

Table 19: Command – HoldQueueEntry
From: [JDF1.5] Table 5-82
Instance of: Command

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
HoldQueueEntry	w			r			
xsi:type	w			r?			
CommandHoldQueueEntry	w			r?			
HoldQueueEntryParams	w			r			SeeTable 20: HoldQueueEntryParams.

6.2.1.1 HoldQueueEntryParams

Table 20: HoldQueueEntryParams
From: [JDF1.5] Table 5-83
Referenced by: Command – HoldQueueEntry

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
QueueFilter	w			r			The Manager SHALL supply this Element. See Table 80: QueueFilter.

6.2.2 Response – HoldQueueEntry

Table 21: Response – HoldQueueEntry
From: [JDF1.5] Table 5-82
Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Type	r			w			
HoldQueueEntry	r			w			
xsi:type	r?			w			
ResponseHoldQueueEntry	r?			w			

6.3 KnownDevices

Note: In this section, the Conformance-Table columns for Manager and Worker are relabeled ***Producer*** and ***Consumer***. See section 5.1 Manager/Worker versus Producer/Consumer.

6.3.1 Query – KnownDevices

Table 22: Query – KnownDevices
From: [JDF1.5] Table 5-16
Instance of: Query

Name or Value	*Producer*			*Consumer*			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
KnownDevices	w			r			
xsi:type	w			r?			
QueryKnownDevices	w			r?			
DeviceFilter	w			r			See Table 23: DeviceFilter.

6.3.1.1 DeviceFilter

Table 23: DeviceFilter
From: [JDF1.5] Table 5-17
Referenced by: Query – KnownDevices

Name or Value	*Producer*			*Consumer*			Description
Level ➔	1	2	3	1	2	3	
DeviceDetails	w			r			**r-Test:** The information in the Response contains the information corresponding to the value of *@DeviceDetails*.
None	w←			r			**r-Test:** Provide only

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
							DeviceInfo/@DeviceID and DeviceInfo/@DeviceStatus.
Brief	w←			r			**r-Test: :** SHALL provide DeviceInfo/@DeviceID and DeviceInfo/@DeviceStatus and SHALL NOT include DeviceInfo/Device.
Details	w←			r			**r-Test:** SHALL include DeviceInfo/Device. See Table 27: Device

6.3.2 Response – KnownDevices

Note: the Producer in the following Response tables is returning the response to the Consumer. The Query Producer is the Response Consumer, and the Response Producer is the Query Consumer.

Table 24: Response – KnownDevices

From: [JDF1.5] Table 5-16

Instance of: Response

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
KnownDevices	w			r			
xsi:type	w			r?			
ResponseKnownDevices	w			r?			
DeviceList	w			r			See Table 25: DeviceList.

6.3.2.1 DeviceList

Table 25: DeviceList

From: [JDF1.5] Table 5-18

Referenced by: Response – KnownDevices

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
DeviceInfo	w			r			See Table 26: DeviceInfo.

6.3.2.1.1 DeviceInfo

Table 26: DeviceInfo
From: [JDF1.5] Table 5-60
Referenced by: DeviceList

Name or Value	*Producer*			*Consumer*			Description
Level ➔	1	2	3	1	2	3	
DeviceID	w			r?			
DeviceStatus	w			r?			
Device	w←			r?			If the KnownDevices Query contained DeviceFilter/@DeviceDetails = "*Details*", the Producer SHALL return at least one **Device** Element where Device/@DeviceID matches JMF/@SenderID for the KnownDevices Response. See Table 27: Device.

6.3.2.1.2 Device

Table 27: Device
From: [JDF1.5] Table 9-6
Referenced by: DeviceInfo

Name or Value	*Producer*			*Consumer*			Description
Level ➔	1	2	3	1	2	3	
DescriptiveName	w			r?			Any descriptive information that helps identify the Device/Controller.
DeviceID	w			r?			
JDFVersions	w			r?			
JMFSenderID	w			r?			If a Device sends its own JMF Messages, the value of *@JMFSenderID* SHALL match the value of *@DeviceID*. Controllers SHALL specify their *@ControllerID* value in this Attribute.
JMFURL	w←			r?			*@JMFURL* SHALL be written for all Devices that Support JMF, including the Device whose *@DeviceID* matches the *@SenderID* in the KnownDevices Response.

6.4 KnownMessages

Note: In this section, the Conformance-Table columns for Manager and Worker are relabeled ***Producer*** and ***Consumer***. See section 5.1 Manager/Worker versus Producer/Consumer.

6.4.1 Query – KnownMessages

The KnownMessages Query is the recommended way to "ping" another Device/Controller, even if you aren't really interested in the Messages that are Supported. It is a good way to check if configuration is correct.

Table 28: Query – KnownMessages
From: [JDF1.5] Table 5-19
Instance of: Query

Name or Value	*Producer*			*Consumer*			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
KnownMessages	w			r			
xsi:type	w			r?			
QueryKnownMessages	w			r?			
KnownMsgQuParams	w			r			See Table 29: KnownMsgQuParams.

6.4.1.1 KnownMsgQuParams

Table 29: KnownMsgQuParams
From: [JDF1.5] Table 5-20
Referenced by: Query – KnownMessages

Name or Value	*Producer*			*Consumer*			Description
Level ➔	1	2	3	1	2	3	
ChannelMode	w?			r			If *ChannelMode* is omitted, the Consumer SHALL respond with the values that are supported.
FireAndForget	w←			r			**r-Test:** Response SHALL include Fire and Forget Signals, if any.
Reliable	w←			r			**r-Test:** Response SHALL include Reliable Signals, if any.
Exact	w←			r			
false	w←			r			**r-Test:** Response SHALL NOT contain DevCaps for the Messages.
true	w?			r?			Implementing Capabilities in

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
							KnownMessages is optional at all levels.
ListCommands	w?			r			**r-Test:** If *@ListCommands* = "*true*", Response SHALL include Supported commands.
ListQueries	w?			r			**r-Test:** If *@ListQueries* = "*true*", Response SHALL include Supported queries.
ListRegistrations	w?			r			**r-Test:** If *@ListRegistrations* = "*true*", Response SHALL include Supported registrations.
ListSignals	w?			r			**r-Test:** If *@ListSignals* = "*true*", Response SHALL include Supported signals.
Persistent	w?			r			**r-Test:** If *@Persistent* = "*true*", Response SHALL include only those messages that can use Persistent Channels.

6.4.2 Response – KnownMessages

Note: the Producer in the following Response tables is returning the response to the Consumer. The Query Producer is the Response Consumer, and the Response Producer is the Query Consumer.

Table 30: Response – KnownMessages

From: [JDF1.5] Table 5-19

Instance of: Response

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
KnownMessages	w			r			
xsi:type	w			r?			
ResponseKnownMessages	w			r?			
MessageService	w			r			See Table 31: MessageService.

6.4.2.1 MessageService

Table 31: MessageService
From: [JDF1.5] Table 5-21
Referenced by: Response – KnownMessages

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
Acknowledge	w			r?			
ChannelMode	w			r?			
Command	w			r?			
JMFRole	w			r?			
Persistent	w			r?			
Query	w			r?			
Registration	w			r?			
Signal	w			r?			
Type	w			r?			
URLSchemes	w			r?			

6.5 KnownSubscriptions

6.5.1 Query – KnownSubscriptions

Table 32: Query – KnownSubscriptions
From: [JDF1.5] Table 5-22
Instance of: Query

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
KnownSubscriptions	w			r			
xsi:type	w			r?			
QueryKnownSubscriptions	w			r?			
SubscriptionFilter	w?			r			See Table 33: SubscriptionFilter

6.5.1.1 SubscriptionFilter

Table 33: SubscriptionFilter
From: [JDF1.5] Table 5-23
Referenced by: Query – KnownSubscriptions

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
ChannelID	w?			r			**r-Test:** Consumer SHALL respond with matching SubscriptionInfo Elements that match *ChannelID*
DeviceID	w?			r			**r-Test:** Consumer SHALL respond with matching SubscriptionInfo Elements that match *DeviceID*
URL	w?			r			**r-Test:** Consumer SHALL respond with matching SubscriptionInfo Elements that match *URL*

6.5.2 Response – KnownSubscriptions

Note: the Producer in the following Response tables is returning the response to the Consumer. The Query Producer is the Response Consumer, and the Response Producer is the Query Consumer.

Table 34: Response – KnownSubscriptions
From: [JDF1.5] Table 5-22
Instance of: Response

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
KnownSubscriptions	w			r			
xsi:type	w			r?			
ResponseKnownSubscriptions	w			r?			
SubscriptionInfo	w←			r			**r-Test:** If there are matching subscriptions, the Producer SHALL supply. See Table 35: SubscriptionInfo.

6.5.2.1 SubscriptionInfo

Table 35: SubscriptionInfo
From: [JDF1.5] Table 5-24
Referenced by: Response – KnownSubscriptions

Name or Value	*Producer*			*Consumer*			Description
Level ➔	1	2	3	1	2	3	
ChannelID	w			r?			
Family	w			r?			
MessageType	w			r			**r-Test:** The Producer SHALL supply the message type of the subscription. See [JDF1.5].
SenderID	w			r?			
Subscription	w			r?			See Table 13: Subscription - KnownSubscriptions Response.

6.6 QueueStatus

6.6.1 Query – QueueStatus

Table 36: Query – QueueStatus
From: [JDF1.5] Table 5-111
Instance of: Query

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
QueueStatus	w			r			
xsi:type	w			r?			
QueryQueueStatus	w			r?			
QueueFilter – Status	w			r			See Table 81: QueueFilter – Status.

6.6.2 Response – QueueStatus

Table 37: Response – QueueStatus
From: [JDF1.5] Table 5-111
Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
QueueStatus	r			w			
xsi:type	r?			w			
ResponseQueueStatus	r?			w			
Queue	r			w			See Table 78: Queue.

6.7 RemoveQueueEntry

6.7.1 Command – RemoveQueueEntry

Table 38: Command – RemoveQueueEntry
From: [JDF1.5] Table 5-84
Instance of: Command

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
RemoveQueueEntry	w			r			
xsi:type	w			r?			
CommandRemoveQueueEntry	w			r?			
RemoveQueueEntryParams	w			r			See Table 39: RemoveQueueEntryParams.

6.7.1.1 RemoveQueueEntryParams

Table 39: RemoveQueueEntryParams
From: [JDF1.5] Table 5-85
Referenced by: Command – RemoveQueueEntry

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
QueueFilter	w←			r			QueueFilter SHALL be specified to select specific QueueEntry. If QueueFilter is NOT provided, the Command SHALL be applied to all QueueEntry elements of the Queue. See Table 80: QueueFilter.

6.7.2 Response – RemoveQueueEntry

Table 40: Response – RemoveQueueEntry
From: [JDF1.5] Table 5-84
Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
RemoveQueueEntry	r			w			
xsi:type	r?			w			
ResponseRemoveQueueEntry	r?			w			

6.8 RequestQueueEntry

6.8.1 Command – RequestQueueEntry

Table 41: Command – RequestQueueEntry
From: [JDF1.5] Table 5-86
Instance of: Command

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
RequestQueueEntry	r			w			
xsi:type	r?			w			
CommandRequestQueueEntry	r?			w			
RequestQueueEntryParams	r			w			See Table 42: RequestQueueEntryParams.

6.8.1.1 RequestQueueEntryParams

Table 42: RequestQueueEntryParams
From: [JDF1.5] Table 5-87
Referenced by: Command – RequestQueueEntry

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
JobID	r			w			**r-Test:** If the Response to the RequestQueueEntry specifies *@ReturnCode* = "*0*", the Manager SHALL initiate a SubmitQueueEntry Message for the Job specified by *@JobID*.
JobPartID	r			w?			**r-Test:** If the Response to the RequestQueueEntry specifies *@ReturnCode* = "*0*", the Manager SHALL initiate a SubmitQueueEntry Message for the Node specified by *@JobPartID*.
QueueURL	r			w			**r-Test:** If the Response to the RequestQueueEntry specifies *@ReturnCode* = "*0*", the Manager SHALL initiate a SubmitQueueEntry Message to the URL specified by *@QueueURL*.

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Part	r			w?			**r-Test:** If the Response to the RequestQueueEntry specifies *@ReturnCode* = "*0*", the Manager SHALL initiate a SubmitQueueEntry Message for the Partition of the Node identified by the Part Elements. See [JDF1.5].

6.8.2 Response – RequestQueueEntry

Table 43: Response – RequestQueueEntry

From: [JDF1.5] Table 5-86

Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
RequestQueueEntry	w			r			
xsi:type	w			r?			
ResponseRequestQueueEntry	w			r?			

6.9 ResourcePull

The OPTIONAL ResourcePull Command is provided to allow the Producer (Manager or Worker) to increase the Amount of an Output Resource to be produced by the Consumer. For instance, a press controller can request a new plate from a prepress system or from an MIS that is the Manager of a Prepress System.
If the ResourcePull Command is received after the process has been deemed completed by the Consumer (i.e. a ProcessRun has been written in the AuditPool or the JDF has been returned to the Consumer's Manager) the Consumer MAY send a RequestQueueEntry Command to its Manager. The Consumer SHALL create the required resource and fill the AuditPool with all required PhaseTime, Resource and ProcessRun Audits. After Completion the Consumer SHALL return the JDF to its respective Manager. Note that if no new RequestQueueEntry is sent by the Consumer, that this MAY lead to multiple ReturnQueueEntry Messages for one SubmitQueueEntry.

6.9.1 Command – ResourcePull

Table 44: Command – ResourcePull
From: [JDF1.5] Table 5-54
Instance of: Command

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
ResourcePull	w			r			
xsi:type	w			r?			
CommandResourcePull	w			r?			
ResourcePullParams	w			r			See Table 45: ResourcePullParams.

6.9.1.1 ResourcePullParams

Table 45: ResourcePullParams
From: [JDF1.5] Table 5-55
Referenced by: Command – ResourcePull

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
Amount	w			r			**r-Test:** The additional amount of Resource to be produced.
JobID	w			r			The *@JobID* of the process that creates the Resource.
ResourceID	w			r			**r-Test:** The *@ID* Attribute of the Resource requested.
Part	w?			r			**r-Test:** The Part Element of the part(s) of a Partitioned Output Resource being pulled.

6.9.2 Response – ResourcePull

Table 46: Response – ResourcePull
From: [JDF1.5] Table 5-54
Instance of: Response

Name or Value	Producer			Consumer			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
ResourcePull	r			w			
xsi:type	r?			w			
ResponseResourcePull	r?			w			
QueueEntry	r			w			See Table 79: QueueEntry.

6.10 ResubmitQueueEntry

6.10.1 Command – ResubmitQueueEntry

Table 47: Command – ResubmitQueueEntry
From: [JDF1.5] Table 5-88
Instance of: Command

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
AcknowledgeURL	w			r?			A Worker MAY respond synchronously if it accepts Queue entries and parses the JDF within the time frame of an http connection. If a Worker chooses to acknowledge a ResubmitQueueEntry, it SHOULD send an Acknowledge Message in a timely fashion and not wait until it executes the Node.
ID	w			r			Required in [JDF1.5].
Type	w			r			
ResubmitQueueEntry	w			r			
xsi:type	w			r?			
CommandResubmitQueueEntry	w			r?			
Comment	w?			r			See Table 83: Comment.

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ResubmissionParams	w			r			See Table 48: ResubmissionParams.

6.10.1.1 ResubmissionParams

Table 48: ResubmissionParams

From: [JDF1.5] Table 5-89

Referenced by: Command – ResubmitQueueEntry

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
QueueEntryID	w			r			**r-Test:** The QueueEntry identified by *@QueueEntryID* (and no other) is updated.
URL	w			r			**r-Test:** The QueueEntry is updated based on the referenced JDF Instance.

6.10.2 Response – ResubmitQueueEntry

Table 49: Response – ResubmitQueueEntry

From: [JDF1.5] Table 5-88

Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
ResubmitQueueEntry	r			w			
xsi:type	r?			w			
ResponseResubmitQueueEntry	r?			w			

6.11 ResumeQueueEntry

6.11.1 Command – ResumeQueueEntry

Table 50: Command – ResumeQueueEntry
From: [JDF1.5] Table 5-90
Instance of: Command

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
AcknowledgeURL	w←			r?			If a Manager allows the Worker to send an Acknowledge, the Manager SHALL supply this Attribute. A Worker MAY respond synchronously if it can resume Queue entries within the time frame of an http connection. If a Worker chooses to acknowledge a ResumeQueueEntry, it SHALL send an Acknowledge Message once the Queue entry has been resumed.
ID	w			r			Required in [JDF1.5].
Type	w			r			
ResumeQueueEntry	w			r			
xsi:type	w			r?			
CommandResumeQueueEntry	w			r?			
ResumeQueueEntryParams	w			r			See Table 51: ResumeQueueEntryParams.

6.11.1.1 ResumeQueueEntryParams

Table 51: ResumeQueueEntryParams
From: [JDF1.5] Table 5-91
Referenced by: Command – ResumeQueueEntry

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
QueueFilter	w←			r			QueueFilter SHALL be specified to select specific QueueEntry. If QueueFilter is NOT provided, the Command SHALL be applied to all QueueEntry elements of the Queue. See Table 80: QueueFilter.

6.11.2 Response – ResumeQueueEntry

Table 52: Response – ResumeQueueEntry
From: [JDF1.5] Table 5-90
Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Acknowledged	r?			w←			
true	r?			w←			
false	r?			w←			
ID	r			w			Required in [JDF1.5].
Type	r			w			
ResumeQueueEntry	r			w			
xsi:type	r?			w			
ResponseResumeQueueEntry	r?			w			

6.11.3 Acknowledge – ResumeQueueEntry

Table 53: Acknowledge – ResumeQueueEntry
From: [JDF1.5] Table 5-90
Instance of: Acknowledge

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
ResumeQueueEntry	r			w			
xsi:type	r?			w			
AcknowledgeResumeQueueEntry	r?			w			

6.12 ReturnQueueEntry

6.12.1 Command – ReturnQueueEntry

Table 54: Command – ReturnQueueEntry

From: [JDF1.5] Table 5-92

Instance of: Command

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
AcknowledgeURL	r?			w?			A Manager MAY respond synchronously if it accepts returned Queue entries and parses the JDF within the time frame of an http connection. If a Manager chooses to acknowledge a ResumeQueueEntry, it SHOULD send an Acknowledge Message in a timely fashion.
ID	r			w			Required in [JDF1.5].
Type	r			w			
ReturnQueueEntry	r			w			
xsi:type	r?			w			
CommandReturnQueueEntry	r?			w			
ReturnQueueEntryParams	r			w			See Table 55: ReturnQueueEntryParams.

6.12.1.1 ReturnQueueEntryParams

Table 55: ReturnQueueEntryParams

From: [JDF1.5] Table 5-93

Referenced by: Command – ReturnQueueEntry

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Aborted	r?			w←			
Completed	r?			w←			
QueueEntryID	r?			w			
URL	r?			w			References JDF Instance.
cid:…	r			w			URL whose scheme is "*cid*".

6.12.2 Response – ReturnQueueEntry

Table 56: Response – ReturnQueueEntry

From: [JDF1.5] Table 5-92

Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Acknowledged	w←			r?			
true	w←			r?			
false	w←			r?			
ID	w			r			Required in [JDF1.5].
Type	w			r			
ReturnQueueEntry	w			r			
xsi:type	w			r?			
ResponseReturnQueueEntry	w			r?			

6.12.3 Acknowledge – ReturnQueueEntry

Table 57: Acknowledge – ReturnQueueEntry

From: [JDF1.5] Table 5-92

Instance of: Acknowledge

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
ReturnQueueEntry	w			r			
xsi:type	w			r?			
AcknowledgeReturnQueueEntry	w			r?			

6.13 SetQueueEntryPosition

6.13.1 Command –SetQueueEntryPosition

Table 58: Command – SetQueueEntryPosition
From: [JDF1.5] Table 5-94
Instance of: Command

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
SetQueueEntryPosition	w			r			
xsi:type	w			r?			
CommandSetQueueEntryPosition	w			r?			
QueueEntryPosParams	w			r			See Table 59: QueueEntryPosParams.

6.13.1.1 QueueEntryPosParams

Table 59: QueueEntryPosParams
From: [JDF1.5] Table 5-95
Referenced by: Command – SetQueueEntryPosition

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
NextQueueEntryID	w←			r			Exactly one of *@NextQueueEntryID*, *@PrevQueueEntryID*, or *@Position* SHALL be specified. **r-Test:** The QueueEntry identified by *@QueueEntryID* is positioned immediately prior to the QueueEntry identified by *@NextQueueEntryID*.
Position	w←			r			Exactly one of *@NextQueueEntryID*, *@PrevQueueEntryID*, or *@Position* SHALL be specified. **r-Test:** The QueueEntry identified by *@QueueEntryID* is in the Queue in the position specified by *@Position*, where the first QueueEntry is in position 0.
PrevQueueEntryID	w←			r			Exactly one of *@NextQueueEntryID*, *@PrevQueueEntryID*, or *@Position*

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
							SHALL be specified. **r-Test:** The QueueEntry identified by *@QueueEntryID* is positioned immediately after to the QueueEntry identified by *@PrevQueueEntryID*.
QueueEntryID	w			r			**r-Test:** The Queue returned in the response has moved the QueueEntry identified by *@QueueEntryID* according to the other Attributes.

6.13.2 Response – SetQueueEntryPosition

Table 60: Response – SetQueueEntryPosition

From: [JDF1.5] Table 5-94

Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
SetQueueEntryPosition	r			w			
xsi:type	r?			w			
ResponseSetQueueEntryPosition	r?			w			

6.14 SetQueueEntryPriority

6.14.1 Command –SetQueueEntryPriority

Table 61: Command – SetQueueEntryPriority

From: [JDF1.5] Table 5-96

Instance of: Command

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5]..
Type	w			r			
SetQueueEntryPriority	w			r			

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
xsi:type	w			r?			
CommandSetQueueEntryPriority	w			r?			
QueueEntryPriParams	w			r			See Table 62: QueueEntryPriParams.

6.14.1.1 QueueEntryPriParams

Table 62: QueueEntryPriParams

From: [JDF1.5] Table 5-97

Referenced by: Command – SetQueueEntryPriority

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Priority	w			r			**r-Test:** In the Queue returned in the response, the QueueEntry identified by *@QueueEntryID* has the priority indicated.

6.14.2 Response – SetQueueEntryPriority

Table 63: Response – SetQueueEntryPriority

From: [JDF1.5] Table 5-4

Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
SetQueueEntryPriority	r			w			
xsi:type	r?			w			
ResponseSetQueueEntryPriority	r?			w			

6.15 StopPersistentChannel

6.15.1 Command – StopPersistentChannel

Table 64: Command – StopPersistentChannel
From: [JDF1.5] Table 5-33
Instance of: Command

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
StopPersistentChannel	w			r			
xsi:type	w			r?			
CommandStopPersistentChannel	w			r?			
StopPersChParams	w			r			See Table 65: StopPersChParams.

6.15.1.1 StopPersChParams

Table 65: StopPersChParams
From: [JDF1.5] Table 5-34
Referenced by: Command – StopPersistentChannel

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ChannelID	w?			r			The /JMF/Query/@ID in the Query Message that the Manager sent to create a Persistent Channel. If *@ChannelID* is specified, *@DeviceID* SHOULD also be specified. **r-Test:** The Worker does not create any further Commands or Signals whose /JMF/Signal/@refID matches *@ChannelID*. See Section 5.4.1.3 for more information.
DeviceID	w?			r			**r-Test:** The Worker does not create any further Commands or Signals whose /JMF/@SenderID matches *@DeviceID*. See Section 5.4.1.3 for more information.
URL	w			r			URL of the receiver of the Messages. **r-Test:** The Worker does not send any additional Commands or Signals to the specified

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
							URL. See Section 5.4.1.3 for more information.
http:…	w			r			SHALL match the original subscription URL.
all remaining values	!w			r?			

6.15.2 Response – StopPersistentChannel

Table 66: Response – StopPersistentChannel

From: [JDF1.5] Table 5-33

Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
StopPersistentChannel	r			w			
xsi:type	r?			w			
ResponseStopPersistentChannel	r?			w			

6.16 SubmissionMethods

6.16.1 Query – SubmissionMethods

Table 67: Query – SubmissionMethods

From: [JDF1.5] Table 5-113

Instance of: Query

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	w			r			Required in [JDF1.5].
Type	w			r			
SubmissionMethods	w			r			
xsi:type	w			r?			
QuerySubmissionMethods	w			r?			

6.16.2 Response – SubmissionMethods

Table 68: Response – SubmissionMethods

From: [JDF1.5] Table 5-113

Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
SubmissionMethods	r			w			
xsi:type	r?			w			
ResponseSubmissionMethods	r?			w			
SubmissionMethods	r			w			See Table 69: SubmissionMethods.

6.16.2.1 SubmissionMethods

Table 69: SubmissionMethods

From: [JDF1.5] Table 5-114

Referenced by: Response – SubmissionMethods

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Packaging	r?			w			If no types of packaging are Supported, an empty @*Packaging* Attribute SHALL be written.
URLSchemes	r?			w			If retrieving referenced files is not Supported, an empty @*URLSchemes* Attribute SHALL be written.

6.17 SubmitQueueEntry

6.17.1 Command – SubmitQueueEntry

Table 70: Command – SubmitQueueEntry
From: [JDF1.5] Table 5-98
Instance of: Command

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
AcknowledgeURL	w			r?			A Worker MAY respond synchronously if it accepts Queue entries and parses the JDF within the time frame of an http connection. If a Worker chooses to acknowledge a SubmitQueueEntry, it SHOULD send an Acknowledge Message in a timely fashion and not wait until it executes the Node.
ID	w			r			Required in [JDF1.5].
Type	w			r			
SubmitQueueEntry	w			r			
xsi:type	w			r?			
CommandSubmitQueueEntry	w			r?			
QueueSubmissionParams	w			r			See Table 71: QueueSubmissionParams.

6.17.1.1 QueueSubmissionParams

Table 71: QueueSubmissionParams
From: [JDF1.5] Table 5-99
Referenced by: Command – SubmitQueueEntry

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Hold	w?			r←			If the Worker Supports the HoldQueueEntry Message, the Worker SHALL Support this *@Hold* Attribute. **r-Test:** If QueueSubmissionParams/@Hold = "*true*", then immediately after submission QueueEntry/@Status SHALL be "*Held*".
Priority	w?			r←			If the Worker Supports the SetQueueEntryPriority Message, the

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
							Worker SHALL Support this *@Priority* Attribute. **r-Test:** If this Attribute is provided, then immediately after submission QueueEntry/@Priority SHALL equal QueueSubmissionParams/@Priority.
ReturnJMF	w			r			**r-Test:** When the QueueEntry is complete, the Worker sends the ReturnQueueEntry to the URL specified by *@ReturnJMF*.
http:…	w			r			URL whose scheme is "*http*" (or "*https*" - see section 5.2).
ReturnURL	!w			r?			
URL	w			r			References a JDF Instance. The Manager SHOULD use a scheme of "*cid*". If the scheme is not "*cid*" for the value of this Attribute, the Manager SHALL keep the JDF available for the Worker to retrieve until the Worker completes or aborts the Job as indicated by sending the updated JDF with a ReturnQueueEntry Command. **r-Test:** The Worker successfully retrieves the JDF instance via the URL.
cid:…	w←			r			URL whose scheme is "*cid*". This is the recommended URL scheme for Managers to use.
file:…	w←			r?			URL whose scheme is "*file*". File scheme NEED NOT be supported in an environment that does not support hot folders.
ftp:…	w?			r?			URL whose scheme is "*ftp*".
http:...	w←			r			URL whose scheme is "*http*" (or "*https*" - see section 5.2).
all remaining values	!w			r?			
Comment	w?			r			See Table 83: Comment.

6.17.2 Response – SubmitQueueEntry

The Worker SHALL return a SubmitQueueEntry Response before the HTTP connection would time out. In addition, the Worker SHALL parse the JDF supplied in the SubmitQueueEntry Command. If the Worker is unable to parse the JDF before returning the SubmitQueueEntry Response, the Worker SHALL return the Response followed by a SubmitQueueEntry Acknowledge after the Worker has parsed the JDF (see section 5.4.2 “Asynchronous Acknowledges” and Table 73: Acknowledge – SubmitQueueEntry).

Table 72: Response – SubmitQueueEntry

From: [JDF1.5] Table 5-98

Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Acknowledged	r			w←			
true	r			w←			
false	r			w←			
ID	r			w			Required in [JDF1.5].
Type	r			w			
SubmitQueueEntry	r			w			
xsi:type	r?			w			
ResponseSubmitQueueEntry	r?			w			
QueueEntry	r			w←			The Worker SHALL write this Element unless the Command is being acknowledged (@Acknowledged = "*true*"), or the Command failed (@ReturnCode != "*0*"). See Table 79: QueueEntry.

6.17.3 Acknowledge – SubmitQueueEntry

Table 73: Acknowledge – SubmitQueueEntry

From: [JDF1.5] Table 5-98

Instance of: Acknowledge

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
SubmitQueueEntry	r			w			
xsi:type	r?			w			
AcknowledgeSubmitQueueEntry	r?			w			
QueueEntry	r			w←			The Worker SHALL NOT supply this Element if the Command fails. See Table 79: QueueEntry.

6.18 SuspendQueueEntry

6.18.1 Command – SuspendQueueEntry

Table 74: Command – SuspendQueueEntry

From: [JDF1.5] Table 5-100

Instance of: Command

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
AcknowledgeURL	w←			r?			If a Manager allows the Worker to send an Acknowledge, the Manager SHALL supply this Attribute. A Worker MAY respond synchronously if it can suspend Queue entries within the time frame of an http connection. If a Worker chooses to acknowledge a SuspendQueueEntry, it SHALL send an Acknowledge Message once the Queue entry has been suspended.
ID	w			r			Required in [JDF1.5].
Type	w			r			
SuspendQueueEntry	w			r			
xsi:type	w			r?			
CommandSuspendQueueEntry	w			r?			
SuspendQueueEntryParams	w			r			SeeTable 75: SuspendQueueEntryParams.

6.18.1.1 SuspendQueueEntryParams

Table 75: SuspendQueueEntryParams

From: [JDF1.5] Table 5-101

Referenced by: Command – SuspendQueueEntry

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
QueueFilter	w←			r			QueueFilter SHALL be specified to select specific QueueEntry. If QueueFilter is NOT provided, the Command SHALL be applied to all QueueEntry elements of the Queue. See Table 80: QueueFilter.

6.18.2 Response – SuspendQueueEntry

Table 76: Response – SuspendQueueEntry
From: [JDF1.5] Table 5-100
Instance of: Response

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Acknowledged	r?			w←			
true	r?			w←			
false	r?			w←			
ID	r			w			Required in [JDF1.5].
Type	r			w			
SuspendQueueEntry	r			w			
xsi:type	r?			w			
ResponseSuspendQueueEntry	r?			w			

6.18.3 Acknowledge – SuspendQueueEntry

Table 77: Acknowledge – SuspendQueueEntry
From: [JDF1.5] Table 5-100
Instance of: Acknowledge

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
ID	r			w			Required in [JDF1.5].
Type	r			w			
SuspendQueueEntry	r			w			
xsi:type	r?			w			
AcknowledgeSuspendQueueEntry	r?			w			

6.19 Queue Elements

6.19.1 Queue

Table 78: Queue

From: [JDF1.5] Table 5-115

Referenced by: Response – QueueStatus

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
DeviceID	r?			w			
Status	r?			w			
QueueEntry	r?			w←			See [JDF1.5] Section 5.14.2 “QueueEntry”. See Table 79: QueueEntry.

6.19.2 QueueEntry

Table 79: QueueEntry

From: [JDF1.5] Table 5-116

Referenced by: Queue, Response – ResourcePull, Response – SubmitQueueEntry, Acknowledge – SubmitQueueEntry

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
JobID	r?			w←			If the QueueEntry is being returned in a Response to a Message other than SubmitQueueEntry, *@JobID* SHALL be specified.
JobPartID	r?			w←			If the QueueEntry is being returned in a response to a Message other than SubmitQueueEntry, *@JobPartID* from the JDF Root Node SHALL be specified.
Priority	r?			w←			If the Worker Supports priority of Queue entries, it SHALL write this value.
QueueEntryID	r?			w			
Status	r?			w			
SubmissionTime	r?			w←			If the QueueEntry is being written in the Response to the QueueStatus Message, the Worker SHALL write the *@SubmissionTime*.
Part	r?			w←			If there is an AncestorPool in the Root Node

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
							of the submitted JDF, and Part Elements appear within AncestorPool, the QueueEntry/Part Elements SHALL be copies of AncestorPool/Part of the Root Node of the submitted JDF. See [JDF1.5].

6.19.3 QueueFilter

6.19.3.1 QueueFilter – Params

Used to select the QueueEntrys to be manipulated by the Command it is referenced by.

Table 80: QueueFilter – Params

From: [JDF1.5] Table 5-118

Referenced by: AbortQueueEntryParams, HoldQueueEntryParams, RemoveQueueEntryParams, ResumeQueueEntryParams, SuspendQueueEntryParams

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
QueueEntryDef	w?			r			The QueueEntryDef Element can be used to filter the response down to a single Queue entry. See Table 82: QueueEntryDef.

6.19.3.2 QueueFilter – Status

Used to filter the information that is returned in the response to a QueueStatus query.

Table 81: QueueFilter – Status

From: [JDF1.5] Table 5-118

Referenced by: Query – QueueStatus

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
MaxEntries	w			r			When using QueueFilter with any Message other than QueueStatus, the Manager SHOULD write *@MaxEntries* = "*0*". When using QueueFilter with the QueueStatus Query, the Manager SHOULD write any appropriate value. **r-Test:** The Response to the Message contains no more than the number of Queue entries specified in *@MaxEntries*.

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
QueueEntryDetails	w			r			**r-Test:** The QueueEntry Elements in the Response contain the information corresponding to the value of *@QueueEntryDetails*.
None	w←			r			
Brief	w?			r			
all remaining values	!w			r?			

6.19.4 QueueEntryDef

Table 82: QueueEntryDef
From: [JDF1.5] Table 5-117
Referenced by: QueueFilter

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
QueueEntryID	w			r			**r-Test:** The Queue entry referenced by *@QueueEntryID* (and no other) is acted upon (or included in the response for queries) based on the type of the message.

6.20 Other Elements

6.20.1 Comment

Table 83: Comment
From: [JDF1.5] Table 3-5
Referenced by: Command – ResubmitQueueEntry, QueueSubmissionParams

Name or Value	Manager			Worker			Description
Level ➔	1	2	3	1	2	3	
Name	w			r?			
Instruction	w←			r?			Additional information provided from the Manager to the Worker about the Queue entry being acted upon (e.g. explanation for why it is being aborted).
all remaining values	w?			r?			
<content of Element>	w			r?			

7 Conformance Rules – Job Submission

7.1 Plain JDF versus JMF – SubmitQueueEntry

7.1.1 SubmitQueueEntry, QueueEntry, ReturnQueueEntry Relationship

A JDF Job submission via the SubmitQueueEntry Command SHALL result in a single QueueEntry being created by the Worker, and the Worker SHALL send at least one ReturnQueueEntry Message when the QueueEntry is completed or aborted. The Worker MAY send multiple ReturnQueueEntry Messages.

When a Worker determines that further work needs to be done on a Completed QueueEntry, the Worker SHALL set the QueueEntry/@Status to "*Waiting*" or "*Running*", and SHALL send any Signals associated with this change of status.

The Manager SHALL Support receiving multiple ReturnQueueEntry Messages related to the same QueueEntry from the Worker. If the Manager needs to prevent the Worker from sending further ReturnQueueEntry Messages related to the same QueueEntry, the Manager SHALL send a RemoveQueueEntry to remove the related QueueEntry.

If a Manager uses the RemoveQueueEntry Command to remove a QueueEntry while in the "*Waiting*" or "*Held*" state, the Worker NEED NOT send a ReturnQueueEntry Message.

If a Manager uses the RemoveQueueEntry Command to remove a QueueEntry while in the "*Completed*" or "*Aborted*" state, the Worker SHALL NOT send further ReturnQueueEntry Messages. This ICS enforces a restriction of a single SubmitQueueEntry Command per JMF Message. A JMF Element containing a SubmitQueueEntry Command SHALL contain one and only one SubmitQueueEntry Command Message.

7.1.2 URL External Reference versus MIME Encoded

When a Manager submits a JDF Instance via a JMF SubmitQueueEntry Command Message, the Manager SHALL use one of these options:

A. The JDF Instance and the JMF Message are packaged together using MIME. The JMF Message SHALL be the first part of the MIME package, and SHALL use a URL with the "*cid*" scheme to reference the JDF part of the MIME package. This is the recommended option. Note that further URL references, e.g. PDL content or preview images NEED NOT be in the CID scheme.
B. The JDF Instance is separate from the JMF Message. The JMF Message uses a URL to reference the JDF Instance.

To be conformant to this ICS, Managers and Workers SHALL Support both option A and option B.

8 References

8.1 Normative References

[Base-ICS] Base ICS, Version 1.5, published April 2015. Available at: http://www.cip4.org.

[JDF1.5] JDF Specification, Version 1.5, published December 31, 2013. Available at: http://www.cip4.org.

[MIS-ICS] MIS ICS, Version 1.5, published April 2015, Available at: http://www.cip4.org.

Appendix A: Changes from JMF ICS 1.4

This appendix lists the changes made to JMF ICS 1.4 to make JMF ICS 1.5:

Table 84: Changes from JMF ICS 1.4

Location	Description
1. Table 8: Signal	Deleted *@LastRepeat*
2. Section 5.4 JMF Handshaking	Reworded
3. Table 14: List of JMF Messages	Deleted KnownControllers, Query and Response, Occupation Query and Response.
4. Table 15: Command – AbortQueueEntry	Deleted QueueEntryDef and QueueFilter. Added AbortQueueEntryParams.
5. Section 6.1.1.1 AbortQueueEntryParams	Added this section with new table.
6. Table 17: Response – AbortQueueEntry	Added *@ReturnCode*, and deleted Queue.
7. Table 18: Acknowledge – AbortQueueEntry	Deleted Queue.
8. Table 19: Command – HoldQueueEntry	Deleted QueueEntryDef and QueueFilter. Added HoldQueueEntryParams.
9. Section 6.2.1.1 HoldQueueEntryParams	Added this section with new table.
10. Table 38: Command – RemoveQueueEntry	Deleted QueueEntryDef and QueueFilter. Added RemoveQueueEntryParams.
11. Section 6.7.1.1 RemoveQueueEntryParams	Added this section with new table.
12. Table 40: Response – RemoveQueueEntry	Deleted Queue.
13. Table 44: Command – ResourcePull	Deleted QueueFilter
14. Table 46: Response – ResourcePull	Deleted Queue.
15. Table 47: Command – ResubmitQueueEntry	Deleted. QueueFilter
16. Table 49: Response – ResubmitQueueEntry	Deleted Queue.
17. Table 50: Command – ResumeQueueEntry	Deleted QueueEntryDef and QueueFilter. Added ResumeQueueEntryParams.
18. Section 6.11.1.1 ResumeQueueEntryParams	Added this section with new table.
19. Table 52: Response – ResumeQueueEntry	Deleted Queue.
20. Table 53: Acknowledge – ResumeQueueEntry	Deleted Queue.
21. Table 58: Command – SetQueueEntryPosition	Deleted. QueueFilter
22. Table 60: Response – SetQueueEntryPosition	Deleted Queue.
23. Table 61: Command – SetQueueEntryPriority	Deleted QueueFilter
24. Table 62: QueueEntryPriParams	Deleted *@QueueEntryID.*

Location	Description
25. Table 63: Response – SetQueueEntryPriority	Deleted Queue.
26. Table 70: Command – SubmitQueueEntry	Deleted. QueueFilter
27. Table 72: Response – SubmitQueueEntry	Deleted Queue.
28. Table 73: Acknowledge – SubmitQueueEntry	Deleted Queue.
29. Table 74: Command – SuspendQueueEntry	Deleted QueueEntryDef and QueueFilter. Added SuspendQueueEntryParams.
30. Section 6.18.1.1 SuspendQueueEntryParams	Added this section with new table.
31. Table 76: Response – SuspendQueueEntry	Deleted Queue.
32. Table 77: Acknowledge – SuspendQueueEntry	Deleted Queue.
33. Section 6.19.3 QueueFilter	Split QueueFilter into two tables.
34. Table 82: QueueEntryDef	Deleted Comment.