

JMF ICS

Version 1.4

Date: 2009-12-22

File: ICS-JMF-1.4.doc, .pdf

Device Messaging / Job Tracking 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 [JDF1.4a].



CIP4 THANKS ITS PARTNER LEVEL MEMBERS



Copyright Notice

Copyright © 2000-2009, International Cooperation for Integration of Processes in Prepress, Press and Postpress, hereinafter referred to as CIP4. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Specification and associated documentation files (the “Specification”) to deal in the Specification, including without limitation the rights to use, copy, publish, distribute, and/or sublicense copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the following conditions. The above copyright notice and this permission notice must be included in all copies or substantial portions of the Specification.

The Specification is provided “as is”, without warranty of any kind, express, implied, or otherwise, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event will CIP4 be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of, or in connection with the Specification or the use or other dealings in the Specification.

Except as contained in this notice or as allowed by membership in CIP4, the name of CIP4 must not be used in advertising or otherwise to promote the use or other dealings in this Specification without prior written authorization from CIP4.

Licenses and Trademarks

International Cooperation for Integration of Processes in Prepress, Press and Postpress, CIP4, Job Description Format, JDF and the CIP4 logo are trademarks of CIP4.

Rather than put a trademark symbol in every occurrence of other trademarked names, we state that we are using the names only in an editorial fashion, and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Table of Contents

1	Introduction	7
2	Glossary	7
3	Conformance Levels	8
4	Conformance Tables – Resources	8
4.1	NodeInfo Resource	8
5	Conformance Tables – JMF Instances	9
5.1	Manager/Worker versus Producer/Consumer	9
5.2	JMF Root Node	9
5.3	JMF Message Families	10
5.3.1	Abstract Message	11
5.3.2	Query	12
5.3.3	Response	12
5.3.4	Signal	13
5.3.5	Command	14
5.3.6	Acknowledge	14
5.4	JMF Handshaking	15
5.4.1	Persistent Channels	15
5.4.1.1	Persistent Channels Created via JMF Embedded in NodeInfo Resources	15
5.4.1.2	Creating Persistent Channels	15
5.4.1.3	Closing Persistent Channels	15
5.4.2	Asynchronous Acknowledges	15
5.4.3	Subscription	16
5.4.3.1	Subscription - Query	16
5.4.3.1.1	ObservationTarget	17
5.4.3.2	Subscription - KnownSubscriptions Response	17
6	Conformance Tables – JMF Messages	17
6.1	AbortQueueEntry	21
6.1.1	Command – AbortQueueEntry	21
6.1.2	Response – AbortQueueEntry	22
6.1.3	Acknowledge – AbortQueueEntry	23
6.2	HoldQueueEntry	23
6.2.1	Command – HoldQueueEntry	23
6.2.2	Response – HoldQueueEntry	24
6.3	KnownControllers	24
6.3.1	Query – KnownControllers	24
6.3.2	Response – KnownControllers	24
6.3.2.1	JDFController	25
6.4	KnownDevices	25
6.4.1	Query – KnownDevices	26
6.4.1.1	DeviceFilter	26
6.4.2	Response – KnownDevices	26
6.4.2.1	DeviceList	27
6.4.2.1.1	DeviceInfo	27
6.4.2.1.2	Device	28
6.5	KnownMessages	28
6.5.1	Query – KnownMessages	28
6.5.1.1	KnownMsgQuParams	29
6.5.2	Response – KnownMessages	29
6.5.2.1	MessageService	30
6.6	KnownSubscriptions	31
6.6.1	Query – KnownSubscriptions	31
6.6.1.1	SubscriptionFilter	31
6.6.2	Response – KnownSubscriptions	31

- 6.6.2.1 SubscriptionInfo 32
- 6.7 Occupation 33
 - 6.7.1 Query – Occupation 33
 - 6.7.1.1 EmployeeDef 33
 - 6.7.2 Response – Occupation 33
 - 6.7.2.1 Occupation 34
 - 6.7.2.1.1 Employee 35
 - 6.7.2.1.2 CostCenter 35
 - 6.7.2.1.3 Person 35
- 6.8 QueueStatus 36
 - 6.8.1 Query – QueueStatus 36
 - 6.8.2 Response – QueueStatus 36
- 6.9 RemoveQueueEntry 37
 - 6.9.1 Command – RemoveQueueEntry 37
 - 6.9.2 Response – RemoveQueueEntry 37
- 6.10 RequestQueueEntry 38
 - 6.10.1 Command – RequestQueueEntry 38
 - 6.10.1.1 RequestQueueEntryParams 38
 - 6.10.2 Response – RequestQueueEntry 39
- 6.11 ResourcePull 39
 - 6.11.1 Command – ResourcePull 40
 - 6.11.1.1 ResourcePullParams 40
 - 6.11.2 Response – ResourcePull 41
- 6.12 ResubmitQueueEntry 41
 - 6.12.1 Command – ResubmitQueueEntry 41
 - 6.12.1.1 ResubmissionParams 42
 - 6.12.2 Response – ResubmitQueueEntry 42
- 6.13 ResumeQueueEntry 43
 - 6.13.1 Command – ResumeQueueEntry 43
 - 6.13.2 Response – ResumeQueueEntry 43
 - 6.13.3 Acknowledge – ResumeQueueEntry 44
- 6.14 ReturnQueueEntry 45
 - 6.14.1 Command – ReturnQueueEntry 45
 - 6.14.1.1 ReturnQueueEntryParams 45
 - 6.14.2 Response – ReturnQueueEntry 46
 - 6.14.3 Acknowledge – ReturnQueueEntry 46
- 6.15 SetQueueEntryPosition 47
 - 6.15.1 Command – SetQueueEntryPosition 47
 - 6.15.1.1 QueueEntryPosParams 47
 - 6.15.2 Response – SetQueueEntryPosition 48
- 6.16 SetQueueEntryPriority 48
 - 6.16.1 Command – SetQueueEntryPriority 48
 - 6.16.1.1 QueueEntryPriParams 49
 - 6.16.2 Response – SetQueueEntryPriority 49
- 6.17 StopPersistentChannel 50
 - 6.17.1 Command – StopPersistentChannel 50
 - 6.17.1.1 StopPersChParams 50
 - 6.17.2 Response – StopPersistentChannel 51
- 6.18 SubmissionMethods 51
 - 6.18.1 Query – SubmissionMethods 51
 - 6.18.2 Response – SubmissionMethods 52
 - 6.18.2.1 SubmissionMethods 52
- 6.19 SubmitQueueEntry 53
 - 6.19.1 Command – SubmitQueueEntry 53
 - 6.19.1.1 QueueSubmissionParams 53
 - 6.19.2 Response – SubmitQueueEntry 54

6.19.3	Acknowledge – SubmitQueueEntry	55
6.20	SuspendQueueEntry	56
6.20.1	Command – SuspendQueueEntry	56
6.20.2	Response – SuspendQueueEntry	57
6.20.3	Acknowledge – SuspendQueueEntry.....	57
6.21	Queue Elements	58
6.21.1	Queue.....	58
6.21.2	QueueEntry	58
6.21.3	QueueEntryDef	59
6.21.4	QueueFilter	60
6.22	Other Elements	60
6.22.1	Comment	60
7	Conformance Rules – Job Submission	61
7.1	Plain JDF versus JMF – SubmitQueueEntry	61
7.1.1	SubmitQueueEntry, QueueEntry, ReturnQueueEntry Relationship	61
7.1.2	URL External Reference versus MIME Encoded.....	61
8	References	62
8.1	Normative References	62
Appendix A: Changes from JMF ICS 1.3.....		63

Tables

Table 1: Glossary.....	7
Table 2: Conformance Levels.....	8
Table 3: NodeInfo.....	8
Table 4: JMF.....	9
Table 5: Abstract Message.....	11
Table 6: Query.....	12
Table 7: Response.....	12
Table 8: Signal.....	13
Table 9: Command	14
Table 10: Acknowledge.....	14
Table 11: Subscription - Query.....	16
Table 12: ObservationTarget	17
Table 13: Subscription - KnownSubscriptions Response	17
Table 14: List of JMF Messages.....	18
Table 15: Command – AbortQueueEntry	21
Table 16: Response – AbortQueueEntry	22
Table 17: Acknowledge – AbortQueueEntry	23
Table 18: Command – HoldQueueEntry	23
Table 19: Response – HoldQueueEntry.....	24
Table 20: Query – KnownControllers	24
Table 21: Response – KnownControllers	25
Table 22: JDFController.....	25
Table 23: Query – KnownDevices.....	26
Table 24: DeviceFilter	26
Table 25: Response – KnownDevices	27
Table 26: DeviceList	27
Table 27: DeviceInfo	27
Table 28: Device.....	28
Table 29: Query – KnownMessages.....	28
Table 30: KnownMsgQuParams.....	29
Table 31: Response – KnownMessages.....	30
Table 32: MessageService	30
Table 33: Query – KnownSubscriptions.....	31

Table 34: SubscriptionFilter	31
Table 35: Response – KnownSubscriptions	32
Table 36: SubscriptionInfo	32
Table 37: Query – Occupation.....	33
Table 38: EmployeeDef.....	33
Table 39: Response – Occupation	34
Table 40: Occupation.....	34
Table 41: Employee.....	35
Table 42: CostCenter	35
Table 43: Person	35
Table 44: Query – QueueStatus	36
Table 45: Response – QueueStatus.....	36
Table 46: Command – RemoveQueueEntry	37
Table 47: Response – RemoveQueueEntry	37
Table 48: Command – RequestQueueEntry	38
Table 49: RequestQueueEntryParams	38
Table 50: Response – RequestQueueEntry.....	39
Table 51: Command – ResourcePull	40
Table 52: ResourcePullParams	40
Table 53: Response – ResourcePull.....	41
Table 54: Command – ResubmitQueueEntry	41
Table 55: ResubmissionParams	42
Table 56: Response – ResubmitQueueEntry	42
Table 57: Command – ResumeQueueEntry	43
Table 58: Response – ResumeQueueEntry	43
Table 59: Acknowledge – ResumeQueueEntry	44
Table 60: Command – ReturnQueueEntry	45
Table 61: ReturnQueueEntryParams	45
Table 62: Response – ReturnQueueEntry.....	46
Table 63: Acknowledge – ReturnQueueEntry	46
Table 64: Command – SetQueueEntryPosition	47
Table 65: QueueEntryPosParams	47
Table 66: Response – SetQueueEntryPosition	48
Table 67: Command – SetQueueEntryPriority	48
Table 68: QueueEntryPriParams	49
Table 69: Response – SetQueueEntryPriority	49
Table 70: Command – StopPersistentChannel	50
Table 71: StopPersChParams	50
Table 72: Response – StopPersistentChannel.....	51
Table 73: Query – SubmissionMethods.....	51
Table 74: Response – SubmissionMethods	52
Table 75: SubmissionMethods	52
Table 76: Command – SubmitQueueEntry.....	53
Table 77: QueueSubmissionParams	53
Table 78: Response – SubmitQueueEntry	55
Table 79: Acknowledge – SubmitQueueEntry	55
Table 80: Command – SuspendQueueEntry	56
Table 81: Response – SuspendQueueEntry	57
Table 82: Acknowledge – SuspendQueueEntry	57
Table 83: Queue.....	58
Table 84: QueueEntry.....	58
Table 85: QueueEntryDef.....	59
Table 86: QueueFilter	60
Table 87: Comment	60
Table 88: Changes from JMF ICS 1.3	63

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
<i>Subscriber</i>	A Manager or Worker in a role where it writes a Query or Registration that includes a Subscription Element.
<i>Signaler</i>	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.

3 Conformance Levels

This ICS specifies exactly one *Conformance Level* of *Conformance Requirements* – a subset of [JDF1.4a]. 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 MUST conform to the Manager part and a Worker MUST 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 MUST be able to interpret MIME packages and access their parts. A Worker MUST 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

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

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 **MUST** 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
Root Node of: JMF Instance
Referenced by: NodeInfo**

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>DeviceID</i>	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 <i>DeviceID</i> . If the Message is being sent to a Controller, this Attribute identifies that Controller's <i>ControllerID</i> . r-Test: If the immediate Consumer or any subsequent recipient does not recognize the value of this Attribute, it MUST reject the Message with <i>ReturnCode</i> = " 121", "Unknown <i>DeviceID</i> ".
<i>ICSVersions</i>	w←			r?			Has the same semantics as JDF/@ICSVersions. See [[Base-ICS]].
<i>JMF_L1-1.4</i>	w			r?			Specifies that the JMF Element conforms to [JMF-ICS] level 1.
<i>all remaining values</i>	w←			r?			Values specified in other ICSs.
<i>MaxVersion</i>	w			r			

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
1.4	w			r			A value higher than 1.4 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.
SenderID	w			r?			Identifies the sender. If the sender is a Device, this Attribute MUST contain the Device's <i>DeviceID</i> . If the sender is a Controller, this Attribute MUST contain the Controller's <i>ControllerID</i> .
TimeStamp	w			r?			Date and time the JMF is sent.
Version	w			r?			
1.4	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 [JDF1.4a], the namespace URI is the same.
xmlns:xsi	w←			r?			MUST 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

Referenced by: JMF

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

Name or Value Level →	Producer			Consumer			Description
	1	2	3	1	2	3	
<i>ID</i>	w			r←			<p>Unique identifier of Message Values MUST be unique for all Messages initiated by the same sender. In other words, the combination of JMF/@SenderID and this <i>ID</i> MUST be unique.</p> <p>If the message is a Query or Command Message, Consumers MUST read <i>ID</i>.</p> <p>r-Test: The response to the Query or Command specifies a <i>refID</i> which matches the <i>ID</i> of the Query or Command.</p>
<i>Time</i>	w←			r?			<p>Time at which the Message was generated. This Attribute MUST be specified unless the time is the same as JMF/@TimeStamp.</p>
<i>Type</i>	w			r			<p>r-Test: Any actions taken or response generated corresponds to the message type identified by this value.</p>
<i>xsi:type</i>	w			r?			<p>Helps JDF Schema-aware implementations to identify specific Message types.</p>
<Message Family> <value of /JMF/Message/@Type>	w			r?			<p>Example: <i>CommandSubmitQueueEntry</i>.</p> <ul style="list-style-type: none"> “<i>Command</i>” is the Message Family Name, i.e. the actual Element name of the Message. See column 1 of Table 14: List of JMF Messages. “<i>SubmitQueueEntry</i>” is the value of /JMF/Message/@Type. See Table 76: Command – SubmitQueueEntry.

5.3.2 Query

Table 6: Query

Subclass of: Abstract Message

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

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>AcknowledgeURL</i>	w←			r?			If a Producer allows the Consumer to send an Acknowledge, the Producer MUST supply this Attribute.
QueryTypeObj	w←			r			Abstract Element that is a placeholder for any descriptive Elements that provide details required for the Query. See [JDF1.4a].
Subscription	w←			r←			A Consumer MUST Support this Element for establishing a Persistent Channel. The Message types that MUST support Subscriptions is defined in further ICS documents. See e.g. [MIS ICS 1.4] 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

Subclass of: Abstract Message

Class for: Response – AbortQueueEntry, Response – HoldQueueEntry, Response – KnownControllers, Response – KnownDevices, Response – KnownMessages, Response – KnownSubscriptions, Response – Occupation, 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	
<i>Acknowledged</i>	w←			r?			The Producer MUST supply this Attribute with a value of "true" when it will send an

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
							asynchronous Acknowledge later.
<i>refID</i>	w←			r?			See [JDF1.4a].
<i>ReturnCode</i>	w←			r?			If an error occurs, a Producer MUST write a nonzero value. See [JDF1.4a] Appendix D for a list of Supported values. A Consumer MUST be able to detect nonzero values.
<i>Subscribed</i>	w←			r?			The Producer MUST supply this Attribute if the Query contained a Subscription (see Table 6: Query).
<i>true</i>	w←			r?			A Producer MUST accept Subscriptions for Persistent Channels in Queries.

5.3.4 Signal

Table 8: Signal
Subclass of: Abstract Message

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>LastRepeat</i>	w←			r?			If the Producer has closed the Persistent Channel the value of this Attribute MUST be " <i>true</i> ".
<i>refID</i>	w←			r?			If the Signal is a result of a Subscription, the Producer MUST supply the ID of the Subscription Query.
<i>QueryTypeObj</i>	w			r?			If the Signal is the result of a hardwired configuration, the Producer MUST supply a <i>QueryTypeObj</i> that specifies the corresponding query parameters that apply to this Signal. If the Signal is the result of a subscription, the same <i>QueryTypeObj</i> that was in the subscription (or one with additional context added) MUST be supplied here. See [JDF1.4a].

5.3.5 Command

Table 9: Command

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	
<i>AcknowledgeURL</i>	w←			r?			If a Producer allows the Consumer to send an Acknowledge, the Producer MUST supply this Attribute.
<i>CommandTypeObj</i>	w←			r			Abstract Element that is a placeholder for any descriptive Elements that provide details required for the Command. See [JDF1.4a].

5.3.6 Acknowledge

Table 10: Acknowledge

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	
<i>AcknowledgeType</i>	w←			r?			
<i>all values</i>	w←			r?			
<i>refID</i>	w			r?			ID of the Message being Acknowledged.
<i>ReturnCode</i>	w←			r?			A non-zero return code indicates failure. If an error can occur, a Producer MUST be capable of writing at least one nonzero value. See [JDF1.4a] Appendix D for a list of Supported values. A Consumer MUST 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 Persistent Channels Created via JMF Embedded in NodeInfo Resources

Workers MUST search the submitted JDF Instance for embedded JMF subscriptions in JDF Nodes identified as executable, the parents of the executable JDF Nodes, and the Root Node. JMF subscriptions embedded in other Nodes MAY be ignored by Workers.

5.4.1.2 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* MAY send the Query or Registration Message either as a Subelement of a **NodeInfo** Resource or as a separate JMF Query or Registration Message via HTTP. A *Signaler* MUST Support both Messages with subscriptions in linked **NodeInfo** Resources, as well as Messages with subscriptions sent via HTTP.

5.4.1.3 Closing Persistent Channels

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

Persistent Channels that relate to a particular Job, Node, or Queue entry that are created with Messages sent via HTTP MUST be closed when the corresponding Job Node completes or aborts or the related QueueEntry/@Status changes to "PendingReturn".

If a Persistent Channel does *not* relate to a particular Job, Node, or Queue entry, it MUST be closed by sending a StopPersistentChannel Command.

If a *Subscriber* is capable of sending a Query with Subscription Elements via HTTP, the *Subscriber* MUST also be capable of sending a StopPersistentChannel Command. All Subscription Consumers MUST Support StopPersistentChannel Commands.

5.4.2 Asynchronous Acknowledges

A Consumer of a JMF Command or Query Message MUST 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 MUST 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 MUST write the *AcknowledgeURL* Attribute in the Command or Query Element
- The Producer of the Command or Query MUST 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 MUST Support an Acknowledge Message of the same type.

If a Consumer responds synchronously, it MUST:

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

If a Consumer responds asynchronously, it:

- MUST include the Response Element, with the *Acknowledged* Attribute set to true. The Response Element MAY contain a ResponseTypeObj.
- MUST send the JMF Message in the HTTP response channel.
- MAY include a ResponseTypeObj in an Acknowledge Element,
- MUST 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 MUST 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 MUST process the message using exactly one of the following methods:

- The message must be fully processed and a full response sent within the timeframe of an http connection.
- OR**
- A response MUST be returned which indicates *ReturnCode* = 11, which is a new error code that will appear in JDF 1.4 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

Referenced by: Query

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

5.4.3.1.1 ObservationTarget

Table 12: ObservationTarget
Referenced by: Subscription - Query

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
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
Referenced by: SubscriptionInfo

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
RepeatTime	w←			r?			Producer MUST supply if messages are 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-Tests** 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

Message Type	Family	Manager			Worker			Description
		Level →	1	2	3	1	2	
AbortQueueEntry	Command	w			r			See Table 15: Command – AbortQueueEntry.
AbortQueueEntry	Response	r			w			See Table 16: Response – AbortQueueEntry.
AbortQueueEntry	Acknowledge	r?			w?			See Table 17: Acknowledge – AbortQueueEntry.
HoldQueueEntry	Command	w?			r?			See Table 18: Command – HoldQueueEntry.
HoldQueueEntry	Response	r?			w?			See Table 19: Response – HoldQueueEntry.
KnownControllers	Query	w? r?			r? w?			Controller/Device Registration Servers SHOULD Support receiving and processing the KnownControllers Query Message. See [JDF1.4a] Section 5.8.2 for more information. See Table 20: Query – KnownControllers.
KnownControllers	Response	r? w?			w? r?			Controller/Device Registration Servers SHOULD Support producing the KnownControllers Response Message. See [JDF1.4a] section 5.8.2 for more information. See Table 21: Response – KnownControllers.
KnownDevices	Query	w r			r w?			See Table 23: Query – KnownDevices.
KnownDevices	Response	r w			w r			See Table 25: Response – KnownDevices.
KnownMessages	Query	w← r			r w?			Managers MUST, and Workers SHOULD, provide the ability for a user/operator to initiate sending a KnownMessages Query. See Table 29: Query – KnownMessages.
KnownMessages	Response	r w			w r			See Table 31: Response – KnownMessages.
KnownSubscriptions	Query	w←			r			Manager MUST be capable of sending if it Supports a Query

Message Type	Family	Manager			Worker			Description
		Level →	1	2	3	1	2	
								with Subscriptions. Worker MUST Support if it accepts a Query with Subscriptions. See Table 33: Query – KnownSubscriptions
KnownSubscriptions	Response	r			w			Table 35: Response – KnownSubscriptions
Occupation	Query	w? r?			r? w?			See Table 37: Query – Occupation.
Occupation	Response	r? w?			w? r?			See Table 39: Response – Occupation.
QueueStatus	Query	w?			r?			See Table 44: Query – QueueStatus.
QueueStatus	Response	r?			w?			See Table 45: Response – QueueStatus.
RemoveQueueEntry	Command	w			r			See Table 46: Command – RemoveQueueEntry.
RemoveQueueEntry	Response	r			w			See Table 47: Response – RemoveQueueEntry.
RequestQueueEntry	Command	r?			w?			See Table 48: Command – RequestQueueEntry.
RequestQueueEntry	Response	w?			r?			See Table 50: Response – RequestQueueEntry.
ResourcePull	Command	r?			w?			See Table 51: Command – ResourcePull
ResourcePull	Response	w?			r?			See Table 53: Response – ResourcePull
ResubmitQueueEntry	Command	w?			r?			See Table 54: Command – ResubmitQueueEntry.
ResubmitQueueEntry	Response	r?			w?			See Table 56: Response – ResubmitQueueEntry.
ResumeQueueEntry	Command	w?			r←			If Worker Supports HoldQueueEntry Message or QueueSubmissionParams/@Hold, the Worker MUST Support the ResumeQueueEntry Message. See Table 57: Command – ResumeQueueEntry.

Message Type	Family	Manager			Worker			Description
		Level →	1	2	3	1	2	
ResumeQueueEntry	Response	r?			w←			If Worker Supports HoldQueueEntry Message or QueueSubmissionParams/@Hold, the Worker MUST Support the ResumeQueueEntry Message. See Table 58: Response – ResumeQueueEntry.
ResumeQueueEntry	Acknowledge	r?			w?			See Table 59: Acknowledge – ResumeQueueEntry.
ReturnQueueEntry	Command	r			w			See Table 60: Command – ReturnQueueEntry.
ReturnQueueEntry	Response	w			r			See Table 62: Response – ReturnQueueEntry.
ReturnQueueEntry	Acknowledge	w?			r?			See Table 63: Acknowledge – ReturnQueueEntry.
SetQueueEntryPosition	Command	w?			r?			See Table 64: Command – SetQueueEntryPosition.
SetQueueEntryPosition	Response	r?			w?			See Table 66: Response – SetQueueEntryPosition.
SetQueueEntryPriority	Command	w?			r←			If Worker Supports QueueSubmissionParams/@Priority in SubmitQueueEntry command, it must Support the SetQueueEntryPriority Message. See Table 67: Command – SetQueueEntryPriority.
SetQueueEntryPriority	Response	r?			w←			If Worker Supports QueueSubmissionParams/@Priority in SubmitQueueEntry Command, it must Support the SetQueueEntryPriority Message. See Table 69: Response – SetQueueEntryPriority.
StopPersistentChannel	Command	w←			r			Manager MUST be capable of sending if it Supports a Query with Subscriptions. Worker MUST Support if it accepts a

Message Type	Family	Manager			Worker			Description
		Level →	1	2	3	1	2	
								Query with Subscriptions. See Table 70: Command – StopPersistentChannel.
StopPersistentChannel	Response	r			w			See Table 72: Response – StopPersistentChannel.
SubmissionMethods	Query	w? r			r w?			See Table 73: Query – SubmissionMethods.
SubmissionMethods	Response	rw			wr			See Table 74: Response – SubmissionMethods.
SubmitQueueEntry	Command	w			r			See Table 76: Command – SubmitQueueEntry.
SubmitQueueEntry	Response	r			w			See Table 78: Response – SubmitQueueEntry.
SubmitQueueEntry	Acknowledge	r			w?			See Table 79: Acknowledge – SubmitQueueEntry.
SuspendQueueEntry	Command	w?			r?			See Table 80: Command – SuspendQueueEntry.
SuspendQueueEntry	Response	r?			w?			See Table 81: Response – SuspendQueueEntry.
SuspendQueueEntry	Acknowledge	r?			w?			See Table 82: Acknowledge – SuspendQueueEntry.

6.1 AbortQueueEntry

6.1.1 Command – AbortQueueEntry

Table 15: Command – AbortQueueEntry

Instance of: Command

In: List of JMF Messages

Name or Value	Manager			Worker			Description	
	Level →	1	2	3	1	2		3
<i>AcknowledgeURL</i>	w←				r?			If a Manager allows the Worker to send an Acknowledge, the Manager MUST 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

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
							AbortQueueEntry, it MUST send an Acknowledge Message once the Queue entry has been aborted.
Type	w			r			
<i>AbortQueueEntry</i>	w			r			
xsi:type	w			r?			
<i>CommandAbortQueueEntry</i>	w			r?			
QueueEntryDef	w			r			See Table 85: QueueEntryDef.
QueueFilter	w?			r			The Manager SHOULD supply this Element. See Table 86: QueueFilter.

6.1.2 Response – AbortQueueEntry

Table 16: Response – AbortQueueEntry

Instance of: Response

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Acknowledged</i>		r			w←		
<i>true</i>		r			w←		
<i>false</i>		r			w←		
Type		r			w		
<i>AbortQueueEntry</i>		r			w		
xsi:type		r?			w		
<i>ResponseAbortQueueEntry</i>		r?			w		
Queue		r			w←		If Response/@Acknowledged = "false", Queue MUST be written. Otherwise Queue MAY be written. See Table 83: Queue.

6.1.3 Acknowledge – AbortQueueEntry

Table 17: Acknowledge – AbortQueueEntry

Instance of: Acknowledge

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
Type	r			w			
<i>AbortQueueEntry</i>	r			w			
xsi:type	r?			w			
<i>AcknowledgeAbortQueueEntry</i>	r?			w			
Queue	r			w←			If the Acknowledge Message is the final Acknowledge Message to the AbortQueueEntry Command, Queue MUST be written. Otherwise Queue MAY be written. See Table 83: Queue.

6.2 HoldQueueEntry

6.2.1 Command – HoldQueueEntry

Table 18: Command – HoldQueueEntry

Instance of: Command

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
Type	w			r			
<i>HoldQueueEntry</i>	w			r			
xsi:type	w			r?			
<i>CommandHoldQueueEntry</i>	w			r?			
QueueEntryDef	w			r			See Table 85: QueueEntryDef.
QueueFilter	w←			r			The Manager SHOULD supply this Element. See Table 86: QueueFilter.

6.2.2 Response – HoldQueueEntry

Table 19: Response – HoldQueueEntry

Instance of: Response

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>	r			w			
<i>HoldQueueEntry</i>	r			w			
<i>xsi:type</i>	r?			w			
<i>ResponseHoldQueueEntry</i>	r?			w			

6.3 KnownControllers

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 – KnownControllers

Table 20: Query – KnownControllers

Instance of: Query

In: List of JMF Messages

Name or Value	<i>Producer</i>			<i>Consumer</i>			Description
	Level →	1	2	3	1	2	
<i>Type</i>	w			r			
<i>KnownControllers</i>	w			r			
<i>xsi:type</i>	w			r?			
<i>QueryKnownControllers</i>	w			r?			

6.3.2 Response – KnownControllers

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

Table 21: Response – KnownControllers

Instance of: Response

In: List of JMF Messages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
Type		w			r		
<i>KnownControllers</i>		w			r		
xsi:type		w			r?		
<i>ResponseKnownControllers</i>		w			r?		
JDFController		w			r?		See Table 22: JDFController.

6.3.2.1 JDFController

Table 22: JDFController

Referenced by: Response – KnownControllers

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>ControllerID</i>		w			r?		Identifies the Controller. Each Controller MUST use a fixed <i>ControllerID</i> . This value MUST be used as JMF/@SenderID in Messages that the Controller sends, and SHOULD be used as JMF/@DeviceID for Messages intended for this Controller.
<i>URL</i>		w			r?		
<i>DescriptiveName</i>		w			r?		Any descriptive information that could be used by a person to help determine which Controller a new Device should be associated with.

6.4 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.

6.4.1 Query – KnownDevices

Table 23: Query – KnownDevices

Instance of: Query

In: List of JMF Messages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
Type		w			r		
KnownDevices		w			r		
xsi:type		w			r?		
QueryKnownDevices		w			r?		
DeviceFilter		w			r		See Table 24: DeviceFilter.

6.4.1.1 DeviceFilter

Table 24: DeviceFilter

Referenced by: Query – KnownDevices

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
DeviceDetails		w			r		r-Test: The information in the Response contains the information corresponding to the value of <i>DeviceDetails</i> .
None		w←			r		r-Test: Provide only DeviceInfo/@DeviceID and DeviceInfo/@DeviceStatus.
Brief		w←			r		r-Test: : MUST provide DeviceInfo/@DeviceID and DeviceInfo/@DeviceStatus and MUST NOT include DeviceInfo/Device.
Details		w←			r		r-Test: MUST include DeviceInfo/Device. See Table 28: Device

6.4.2 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 25: Response – KnownDevices

Instance of: Response

In: List of JMF Messages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
Type		w			r		
<i>KnownDevices</i>		w			r		
xsi:type		w			r?		
<i>ResponseKnownDevices</i>		w			r?		
DeviceList		w			r		See Table 26: DeviceList.

6.4.2.1 DeviceList

Table 26: DeviceList

Referenced by: Response – KnownDevices

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
DeviceInfo		w			r		See Table 27: DeviceInfo.

6.4.2.1.1 DeviceInfo

Table 27: DeviceInfo

Referenced by: DeviceList

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
DeviceID		w			r?		
DeviceStatus		w			r?		
Device		w←			r?		If the KnownDevices Query contained DeviceFilter/@DeviceDetails = "Details" , the Producer MUST return at least one Device Element where Device/@DeviceID matches JMF/@SenderID for the KnownDevices Response. See Table 28: Device.

6.4.2.1.2 Device

Table 28: Device
Referenced by: DeviceInfo

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

6.5 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.

6.5.1 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 29: Query – KnownMessages
Instance of: Query
In: List of JMF Messages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>Type</i>	w			r			
<i>KnownMessages</i>	w			r			
<i>xsi:type</i>	w			r?			
<i>QueryKnownMessages</i>	w			r?			

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
KnownMsgQuParams	w			r			See Table 30: KnownMsgQuParams.

6.5.1.1 KnownMsgQuParams

Table 30: KnownMsgQuParams
 Referenced by: Query – KnownMessages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>ChannelMode</i>	w?			r			If <i>ChannelMode</i> is omitted, the Consumer MUST respond with the values that are supported.
<i>FireAndForget</i>	w←			r			r-Test: Response MUST include Fire and Forget Signals, if any.
<i>Reliable</i>	w←			r			r-Test: Response MUST include Reliable Signals, if any.
<i>Exact</i>	w←			r			
<i>false</i>	w←			r			r-Test: Response MUST NOT contain DevCaps for the Messages.
<i>true</i>	w?			r?			Implementing Capabilities in KnownMessages is optional at all levels.
<i>ListCommands</i>	w?			r			r-Test: If <i>ListCommands</i> = "true", Response MUST include Supported commands.
<i>ListQueries</i>	w?			r			r-Test: If <i>ListQueries</i> = "true", Response MUST include Supported queries.
<i>ListRegistrations</i>	w?			r			r-Test: If <i>ListRegistrations</i> = "true", Response MUST include Supported registrations.
<i>ListSignals</i>	w?			r			r-Test: If <i>ListSignals</i> = "true", Response MUST include Supported signals.
<i>Persistent</i>	w?			r			r-Test: If <i>Persistent</i> = "true", Response MUST include only those messages that can use Persistent Channels.

6.5.2 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 31: Response – KnownMessages

Instance of: Response

In: List of JMF Messages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
Type	w			r			
<i>KnownMessages</i>	w			r			
xsi:type	w			r?			
<i>ResponseKnownMessages</i>	w			r?			
MessageService	w			r			See Table 32: MessageService.

6.5.2.1 MessageService

Table 32: MessageService

Referenced by: Response – KnownMessages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>Acknowledge</i>	w			r?			
<i>ChannelMode</i>	w			r?			
<i>Command</i>	w			r?			
<i>JMFRole</i>	w			r?			
<i>Persistent</i>	w			r?			
<i>Query</i>	w			r?			
<i>Registration</i>	w			r?			
<i>Signal</i>	w			r?			
Type	w			r?			
<i>URLSchemes</i>	w			r?			

6.6 KnownSubscriptions

6.6.1 Query – KnownSubscriptions

Table 33: Query – KnownSubscriptions

Instance of: Query

In: List of JMF Messages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
Type		w			r		
<i>KnownSubscriptions</i>		w			r		
<i>xsi:type</i>		w			r?		
<i>QueryKnownSubscriptions</i>		w			r?		
SubscriptionFilter		w?			r		See Table 34: SubscriptionFilter

6.6.1.1 SubscriptionFilter

Table 34: SubscriptionFilter

Referenced by: Query – KnownSubscriptions

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>ChannelID</i>		w?			r		r-Test: Consumer MUST respond with matching SubscriptionInfo Elements that match <i>ChannelID</i>
<i>DeviceID</i>		w?			r		r-Test: Consumer MUST respond with matching SubscriptionInfo Elements that match <i>DeviceID</i>
<i>URL</i>		w?			r		r-Test: Consumer MUST respond with matching SubscriptionInfo Elements that match <i>URL</i>

6.6.2 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 35: Response – KnownSubscriptions

Instance of: Response

In: List of JMF Messages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
Type		w			r		
<i>KnownSubscriptions</i>		w			r		
xsi:type		w			r?		
<i>ResponseKnownSubscriptions</i>		w			r?		
SubscriptionInfo		w←			r		r-Test: If there are matching subscriptions, the Producer MUST supply. See Table 36: SubscriptionInfo.

6.6.2.1 SubscriptionInfo

Table 36: SubscriptionInfo

Referenced by: Response – KnownSubscriptions

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
ChannelID		w			r?		
<i>all values</i>							See [JDF1.4a].
SenderID		w			r?		
<i>all values</i>							See [JDF1.4a].
Family		w			r?		
<i>all values</i>							See [JDF1.4a].
MessageType		w			r		
<i>all values</i>		w			r		r-Test: The Producer MUST supply the message type of the subscription. See [JDF1.4a].
Subscription		w			r?		See Table 13: Subscription - KnownSubscriptions Response.

6.7 Occupation

6.7.1 Query – Occupation

Table 37: Query – Occupation

Instance of: Query

In: List of JMF Messages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
Type	w			r			
Occupation	w			r			
xsi:type	w			r?			
QueryOccupation	w			r?			
EmployeeDef	w?			r			See Table 38: EmployeeDef.

6.7.1.1 EmployeeDef

Table 38: EmployeeDef

Referenced by: Query – Occupation

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
PersonalID	w			r			r-Test: Response MUST include only information about the employee identified by <i>PersonalID</i> .

6.7.2 Response – Occupation

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 39: Response – Occupation

Instance of: Response

In: List of JMF Messages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
Type		w			r		
Occupation		w			r		
xsi:type		w			r?		
ResponseOccupation		w			r?		
Occupation		w			r		See Table 40: Occupation.

6.7.2.1 Occupation

Table 40: Occupation

Referenced by: Response – Occupation

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
Busy		w			r?		
JobID		w←			r?		If the employee referenced in the Employee Element is working on a specific Job, <i>JobID</i> MUST be specified.
JobPartID		w←			r?		If the employee referenced in the Employee Element is working on a specific Node of a Job, <i>JobPartID</i> MUST be specified.
QueueEntryID		w←			r?		If the employee referenced in the Employee Element is working on a specific Queue entry, <i>QueueEntryID</i> MUST be specified.
Employee		w			r?		See Table 41: Employee.
Part		w←			r?		If the employee is working on one, some, but not all Partitions of a Node of a Job, the Part Element MUST be specified with appropriate Partition Keys specified to identify the Partition(s) that the employee is working on. See [JDF1.4a].

6.7.2.1.1 Employee

Table 41: Employee
Referenced by: Occupation

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>PersonalID</i>	w			r?			
<i>Roles</i>	w			r?			
<i>Shift</i>	w←			r?			If the shift is known for the employee, it MUST be specified.
<i>CostCenter</i>	w←			r?			If the MIS Cost Center is known for the employee, it MUST be specified. See Table 42: CostCenter.
<i>Person</i>	w?			r?			See Table 43: Person.

6.7.2.1.2 CostCenter

Table 42: CostCenter
Referenced by: Employee

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>CostCenterID</i>	w			r?			

6.7.2.1.3 Person

Table 43: Person
Referenced by: Employee

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>FamilyName</i>	w←			r?			If the employee's family name is known, it MUST be specified.
<i>FirstName</i>	w←			r?			If the employee's first name is known, it MUST be specified.

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>DescriptiveName</i>	w←			r?			If <i>FamilyName</i> and <i>FirstName</i> are not specified, the employee's full name MUST be specified in the <i>DescriptiveName</i> Attribute.

6.8 QueueStatus

6.8.1 Query – QueueStatus

Table 44: Query – QueueStatus

Instance of: Query

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>	w			r			
<i>QueueStatus</i>	w			r			
<i>xsi:type</i>	w			r?			
<i>QueryQueueStatus</i>	w			r?			
QueueFilter	w			r			See Table 86: QueueFilter.

6.8.2 Response – QueueStatus

Table 45: Response – QueueStatus

Instance of: Response

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>	r			w			
<i>QueueStatus</i>	r			w			
<i>xsi:type</i>	r?			w			
<i>ResponseQueueStatus</i>	r?			w			
Queue	r			w			See Table 83: Queue.

6.9 RemoveQueueEntry

6.9.1 Command – RemoveQueueEntry

Table 46: Command – RemoveQueueEntry

Instance of: Command

In: List of JMF Messages

Name or Value Level →	Manager			Worker			Description
	1	2	3	1	2	3	
<i>Type</i>	w			r			
<i>RemoveQueueEntry</i>	w			r			
<i>xsi:type</i>	w			r?			
<i>CommandRemoveQueueEntry</i>	w			r?			
QueueEntryDef	w			r			See Table 85: QueueEntryDef.
QueueFilter	w			r			See Table 86: QueueFilter.

6.9.2 Response – RemoveQueueEntry

Table 47: Response – RemoveQueueEntry

Instance of: Response

In: List of JMF Messages

Name or Value Level →	Manager			Worker			Description
	1	2	3	1	2	3	
<i>Type</i>	r			w			
<i>RemoveQueueEntry</i>	r			w			
<i>xsi:type</i>	r?			w			
<i>ResponseRemoveQueueEntry</i>	r?			w			
Queue	r			w			See Table 83: Queue.

6.10 RequestQueueEntry

6.10.1 Command – RequestQueueEntry

Table 48: Command – RequestQueueEntry

Instance of: Command

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>	r			w			
<i>RequestQueueEntry</i>	r			w			
<i>xsi:type</i>	r?			w			
<i>CommandRequestQueueEntry</i>	r?			w			
RequestQueueEntryParams	r			w			See Table 49: RequestQueueEntryParams.

6.10.1.1 RequestQueueEntryParams

Table 49: RequestQueueEntryParams

Referenced by: Command – RequestQueueEntry

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

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
Part	r			w?			r-Test: If the Response to the RequestQueueEntry specifies <i>ReturnCode</i> = "0", the Manager MUST initiate a SubmitQueueEntry Message for the Partition of the Node identified by the Part Elements. See [JDF1.4a].

6.10.2 Response – RequestQueueEntry

Table 50: Response – RequestQueueEntry

Instance of: Response

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>		w			r		
<i>RequestQueueEntry</i>		w			r		
<i>xsi:type</i>		w			r?		
<i>ResponseRequestQueueEntry</i>		w			r?		

6.11 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 MUST create the required resource and fill the AuditPool with all required PhaseTime, Resource and ProcessRun Audits. After Completion the Consumer MUST 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.

6.11.1 Command – ResourcePull

Table 51: Command – ResourcePull

Instance of: Command

In: List of JMF Messages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>Type</i>	w			r			
<i>ResourcePull</i>	w			r			
<i>xsi:type</i>	w			r?			
<i>CommandResourcePull</i>	w			r?			
ResourcePullParams	w			r			See Table 52: ResourcePullParams.
QueueFilter	w			r			See Table 86: QueueFilter.

6.11.1.1 ResourcePullParams

Table 52: ResourcePullParams

Referenced by: Command – ResourcePull

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

6.11.2 Response – ResourcePull

Table 53: Response – ResourcePull

Instance of: Response

In: List of JMF Messages

Name or Value	Producer			Consumer			Description
	Level →	1	2	3	1	2	
<i>Type</i>		r			w		
<i>ResourcePull</i>		r			w		
<i>xsi:type</i>		r?			w		
<i>ResponseResourcePull</i>		r?			w		
QueueEntry		r			w		See Table 84: QueueEntry.
Queue		r?			w		See Table 83: Queue.

6.12 ResubmitQueueEntry

6.12.1 Command – ResubmitQueueEntry

Table 54: Command – ResubmitQueueEntry

Instance of: Command

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>AcknowledgeURL</i>		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.
<i>Type</i>		w			r		
<i>ResubmitQueueEntry</i>		w			r		
<i>xsi:type</i>		w			r?		
<i>CommandResubmitQueueEntry</i>		w			r?		

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
Comment	w?			r			See Table 87: Comment.
ResubmissionParams	w			r			See Table 55: ResubmissionParams.
QueueFilter	w			r			See Table 86: QueueFilter.

6.12.1.1 ResubmissionParams

Table 55: ResubmissionParams

Referenced by: Command – ResubmitQueueEntry

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
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.

6.12.2 Response – ResubmitQueueEntry

Table 56: Response – ResubmitQueueEntry

Instance of: Response

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
Type		r			w		
ResubmitQueueEntry		r			w		
xsi:type		r?			w		
ResponseResubmitQueueEntry		r?			w		
Queue		r			w		See Table 83: Queue.

6.13 ResumeQueueEntry

6.13.1 Command – ResumeQueueEntry

Table 57: Command – ResumeQueueEntry

Instance of: Command

In: List of JMF Messages

Name or Value Level →	Manager			Worker			Description
	1	2	3	1	2	3	
<i>AcknowledgeURL</i>	w←			r?			If a Manager allows the Worker to send an Acknowledge, the Manager MUST 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 MUST send an Acknowledge Message once the Queue entry has been resumed.
<i>Type</i>	w			r			
<i>ResumeQueueEntry</i>	w			r			
<i>xsi:type</i>	w			r?			
<i>CommandResumeQueueEntry</i>	w			r?			
<i>QueueEntryDef</i>	w			r			See Table 85: QueueEntryDef.
<i>QueueFilter</i>	w			r			See Table 86: QueueFilter.

6.13.2 Response – ResumeQueueEntry

Table 58: Response – ResumeQueueEntry

Instance of: Response

In: List of JMF Messages

Name or Value Level →	Manager			Worker			Description
	1	2	3	1	2	3	
<i>Acknowledged</i>	r?			w←			
<i>true</i>	r?			w←			
<i>false</i>	r?			w←			
<i>Type</i>	r			w			

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>ResumeQueueEntry</i>	r			w			
<i>xsi:type</i>	r?			w			
<i>ResponseResumeQueueEntry</i>	r?			w			
Queue	r			w←			If Response/@Acknowledged = "false", Queue MUST be written. Otherwise Queue MAY be written. See Table 83: Queue.

6.13.3 Acknowledge – ResumeQueueEntry

Table 59: Acknowledge – ResumeQueueEntry

Instance of: Acknowledge

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>	r			w			
<i>ResumeQueueEntry</i>	r			w			
<i>xsi:type</i>	r?			w			
<i>AcknowledgeResumeQueueEntry</i>	r?			w			
Queue	r			w←			If the Acknowledge Message is the final Acknowledge Message to the ResumeQueueEntry Command, Queue MUST be written. Otherwise Queue MAY be written. See Table 83: Queue.

6.14 ReturnQueueEntry

6.14.1 Command – ReturnQueueEntry

Table 60: Command – ReturnQueueEntry

Instance of: Command

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>AcknowledgeURL</i>	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.
<i>Type</i>	r			w			
<i>ReturnQueueEntry</i>	r			w			
<i>xsi:type</i>	r?			w			
<i>CommandReturnQueueEntry</i>	r?			w			
<i>ReturnQueueEntryParams</i>	r			w			See Table 61: ReturnQueueEntryParams.

6.14.1.1 ReturnQueueEntryParams

Table 61: ReturnQueueEntryParams

Referenced by: Command – ReturnQueueEntry

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Aborted</i>	r?			w←			
<i>Completed</i>	r?			w←			
<i>QueueEntryID</i>	r?			w			
<i>URL</i>	r?			w			References JDF Instance.
<i>cid:...</i>	r			w			URL whose scheme is "cid".

6.14.2 Response – ReturnQueueEntry

Table 62: Response – ReturnQueueEntry

Instance of: Response

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Acknowledged</i>	W←			r?			
<i>true</i>	W←			r?			
<i>false</i>	W←			r?			
<i>Type</i>	W			r			
<i>ReturnQueueEntry</i>	W			r			
<i>xsi:type</i>	W			r?			
<i>ResponseReturnQueueEntry</i>	W			r?			

6.14.3 Acknowledge – ReturnQueueEntry

Table 63: Acknowledge – ReturnQueueEntry

Instance of: Acknowledge

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>	W			r			
<i>ReturnQueueEntry</i>	W			r			
<i>xsi:type</i>	W			r?			
<i>AcknowledgeReturnQueueEntry</i>	W			r?			

6.15 SetQueueEntryPosition

6.15.1 Command –SetQueueEntryPosition

Table 64: Command – SetQueueEntryPosition

Instance of: Command

In: List of JMF Messages

Name or Value Level →	Manager			Worker			Description
	1	2	3	1	2	3	
Type	w			r			
<i>SetQueueEntryPosition</i>	w			r			
<i>xsi:type</i>	w			r?			
<i>CommandSetQueueEntryPosition</i>	w			r?			
QueueEntryPosParams	w			r			See Table 65: QueueEntryPosParams.
QueueFilter	w			r			See Table 86: QueueFilter.

6.15.1.1 QueueEntryPosParams

Table 65: QueueEntryPosParams

Referenced by: Command – SetQueueEntryPosition

Name or Value Level →	Manager			Worker			Description
	1	2	3	1	2	3	
<i>NextQueueEntryID</i>	w←			r			Exactly one of <i>NextQueueEntryID</i> , <i>PrevQueueEntryID</i> , or <i>Position</i> MUST be specified. r-Test: The <i>QueueEntry</i> identified by <i>QueueEntryID</i> is positioned immediately prior to the <i>QueueEntry</i> identified by <i>NextQueueEntryID</i> .
<i>QueueEntryID</i>	w			r			r-Test: The <i>Queue</i> returned in the response has moved the <i>QueueEntry</i> identified by <i>QueueEntryID</i> according to the other Attributes.
<i>PrevQueueEntryID</i>	w←			r			Exactly one of <i>NextQueueEntryID</i> , <i>PrevQueueEntryID</i> , or <i>Position</i> MUST be specified. r-Test: The <i>QueueEntry</i> identified by

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
							<i>QueueEntryID</i> is positioned immediately after to the <i>QueueEntry</i> identified by <i>PrevQueueEntryID</i> .
<i>Position</i>	w←			r			Exactly one of <i>NextQueueEntryID</i> , <i>PrevQueueEntryID</i> , or <i>Position</i> MUST be specified. r-Test: The <i>QueueEntry</i> identified by <i>QueueEntryID</i> is in the <i>Queue</i> in the position specified by <i>Position</i> , where the first <i>QueueEntry</i> is in position 0.

6.15.2 Response – SetQueueEntryPosition

Table 66: Response – SetQueueEntryPosition

Instance of: Response

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>		r			w		
<i>SetQueueEntryPosition</i>		r			w		
<i>xsi:type</i>		r?			w		
<i>ResponseSetQueueEntryPosition</i>		r?			w		
Queue		r			w		See Table 83: Queue.

6.16 SetQueueEntryPriority

6.16.1 Command –SetQueueEntryPriority

Table 67: Command – SetQueueEntryPriority

Instance of: Command

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>		w			r		

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>SetQueueEntryPriority</i>	w			r			
<i>xsi:type</i>	w			r?			
<i>CommandSetQueueEntryPriority</i>	w			r?			
QueueEntryPriParams	w			r			See Table 68: QueueEntryPriParams.
QueueFilter	w			r			See Table 86: QueueFilter.

6.16.1.1 QueueEntryPriParams

Table 68: QueueEntryPriParams

Referenced by: Command – SetQueueEntryPriority

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Priority</i>	w			r			r-Test: In the Queue returned in the response, the <i>QueueEntry</i> identified by <i>QueueEntryID</i> has the priority indicated.
<i>QueueEntryID</i>	w			r			r-Test: In the Queue returned in the response, the <i>QueueEntry</i> identified by <i>QueueEntryID</i> has the priority indicated.

6.16.2 Response – SetQueueEntryPriority

Table 69: Response – SetQueueEntryPriority

Instance of: Response

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>				r			
<i>SetQueueEntryPriority</i>				r			
<i>xsi:type</i>				r?			
<i>ResponseSetQueueEntryPriority</i>				r?			
Queue				r			See Table 83: Queue.

6.17 StopPersistentChannel

6.17.1 Command – StopPersistentChannel

Table 70: Command – StopPersistentChannel

Instance of: Command

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
Type		w			r		
<i>StopPersistentChannel</i>		w			r		
xsi:type		w			r?		
<i>CommandStopPersistentChannel</i>		w			r?		
StopPersChParams		w			r		See Table 71: StopPersChParams.

6.17.1.1 StopPersChParams

Table 71: StopPersChParams

Referenced by: Command – StopPersistentChannel

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

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
							URL. See Section 5.4.1.3 for more information.
<i>http:...</i>	w			r			MUST match the original subscription URL.
<i>all remaining values</i>	!w			r?			

6.17.2 Response – StopPersistentChannel

Table 72: Response – StopPersistentChannel

Instance of: Response

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>				r			
<i>StopPersistentChannel</i>				r			
<i>xsi:type</i>				r?			
<i>ResponseStopPersistentChannel</i>				r?			

6.18 SubmissionMethods

6.18.1 Query – SubmissionMethods

Table 73: Query – SubmissionMethods

Instance of: Query

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>	w			r			
<i>SubmissionMethods</i>	w			r			
<i>xsi:type</i>	w			r?			
<i>QuerySubmissionMethods</i>	w			r?			

6.18.2 Response – SubmissionMethods

Table 74: Response – SubmissionMethods

Instance of: Response

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>	r			w			
<i>SubmissionMethods</i>	r			w			
<i>xsi:type</i>	r?			w			
<i>ResponseSubmissionMethods</i>	r?			w			
SubmissionMethods	r			w			See Table 75: SubmissionMethods.

6.18.2.1 SubmissionMethods

Table 75: SubmissionMethods

Referenced by: Response – SubmissionMethods

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Packaging</i>	r?			w			If no types of packaging are Supported, an empty <i>Packaging</i> Attribute MUST be written.
<i>URLSchemes</i>	r?			w			If retrieving referenced files is not Supported, an empty <i>URLSchemes</i> Attribute MUST be written.

6.19 SubmitQueueEntry

6.19.1 Command – SubmitQueueEntry

Table 76: Command – SubmitQueueEntry

Instance of: Command

In: List of JMF Messages

Name or Value	Manager			Worker			Description	
	Level →	1	2	3	1	2		3
<i>AcknowledgeURL</i>		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.
<i>Type</i>		w			r			
<i>SubmitQueueEntry</i>		w			r			
<i>xsi:type</i>		w			r?			
<i>CommandSubmitQueueEntry</i>		w			r?			
<i>QueueSubmissionParams</i>		w			r			See Table 77: QueueSubmissionParams.
<i>QueueFilter</i>		w←			r			It is highly recommended that the Manager supply this Element. See Table 86: QueueFilter.

6.19.1.1 QueueSubmissionParams

Table 77: QueueSubmissionParams

Referenced by: Command – SubmitQueueEntry

Name or Value	Manager			Worker			Description	
	Level →	1	2	3	1	2		3
<i>Hold</i>		w?			r←			If the Worker Supports the HoldQueueEntry Message, the Worker MUST Support this <i>Hold</i> Attribute. r-Test: If QueueSubmissionParams/@Hold = "true", then immediately after submission QueueEntry/@Status MUST be

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
							"Held".
Priority	w?			r←			If the Worker Supports the SetQueueEntryPriority Message, the Worker MUST Support this <i>Priority</i> Attribute. r-Test: If this Attribute is provided, then immediately after submission QueueEntry/@Priority MUST equal QueueSubmissionParams/@Priority.
ReturnJMF	w			r			r-Test: When the QueueEntry is complete, the Worker sends the ReturnQueueEntry to the URL specified by <i>ReturnJMF</i> .
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 MUST 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).
<i>all remaining values</i>	!w			r?			
Comment	w?			r			See Table 87: Comment.

6.19.2 Response – SubmitQueueEntry

The Worker MUST return a SubmitQueueEntry Response before the HTTP connection would time out. In addition, the Worker MUST parse the JDF supplied in the SubmitQueueEntry Command. If the Worker is unable to parse the JDF before returning the SubmitQueueEntry Response, the Worker MUST return the

Response followed by a SubmitQueueEntry Acknowledge after the Worker has parsed the JDF (see section 5.4.2 “Asynchronous Acknowledges” and Table 79: Acknowledge – SubmitQueueEntry).

Table 78: Response – SubmitQueueEntry

Instance of: Response

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Acknowledged</i>		r			w←		
<i>true</i>		r			w←		
<i>false</i>		r			w←		
<i>Type</i>		r			w		
<i>SubmitQueueEntry</i>		r			w		
<i>xsi:type</i>		r?			w		
<i>ResponseSubmitQueueEntry</i>		r?			w		
<i>QueueEntry</i>		r			w←		The Worker MUST write this Element unless the Command is being acknowledged (@Acknowledged = "true"), or the Command failed (@ReturnCode != "0"). See Table 84: QueueEntry.
<i>Queue</i>		r?			w←		The Worker MUST write this Element unless the Command is being acknowledged (@Acknowledged = "true"). See Table 83: Queue.

6.19.3 Acknowledge – SubmitQueueEntry

Table 79: Acknowledge – SubmitQueueEntry

Instance of: Acknowledge

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>		r			w		
<i>SubmitQueueEntry</i>		r			w		
<i>xsi:type</i>		r?			w		

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>AcknowledgeSubmitQueueEntry</i>	r?				w		
QueueEntry	r				w←		The Worker MUST NOT supply this Element if the Command fails. See Table 84: QueueEntry.
Queue	r?				w		See Table 83: Queue.

6.20 SuspendQueueEntry

6.20.1 Command – SuspendQueueEntry

Table 80: Command – SuspendQueueEntry

Instance of: Command

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>AcknowledgeURL</i>	w←				r?		If a Manager allows the Worker to send an Acknowledge, the Manager MUST 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 MUST send an Acknowledge Message once the Queue entry has been suspended.
<i>Type</i>	w				r		
<i>SuspendQueueEntry</i>	w				r		
<i>xsi:type</i>	w				r?		
<i>CommandSuspendQueueEntry</i>	w				r?		
QueueEntryDef	w				r		See Table 85: QueueEntryDef.
QueueFilter	w				r		See Table 86: QueueFilter.

6.20.2 Response – SuspendQueueEntry

Table 81: Response – SuspendQueueEntry

Instance of: Response

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Acknowledged</i>		r?			w←		
<i>true</i>		r?			w←		
<i>false</i>		r?			w←		
<i>Type</i>		r			w		
<i>SuspendQueueEntry</i>		r			w		
<i>xsi:type</i>		r?			w		
<i>ResponseSuspendQueueEntry</i>		r?			w		
Queue		r			w←		If Response/@Acknowledged = "false", Queue MUST be written. Otherwise Queue MAY be written. See Table 83: Queue.

6.20.3 Acknowledge – SuspendQueueEntry

Table 82: Acknowledge – SuspendQueueEntry

Instance of: Acknowledge

In: List of JMF Messages

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Type</i>		r			w		
<i>SuspendQueueEntry</i>		r			w		
<i>xsi:type</i>		r?			w		
<i>AcknowledgeSuspendQueueEntry</i>		r?			w		
Queue		r			w←		If the Acknowledge Message is the final Acknowledge Message to the SuspendQueueEntry Command, Queue MUST be written. Otherwise Queue MAY be written. See Table 83: Queue.

6.21 Queue Elements

6.21.1 Queue

Table 83: Queue

Referenced by: Response – AbortQueueEntry, Acknowledge – AbortQueueEntry, Response – QueueStatus, Response – RemoveQueueEntry, Response – ResourcePull, Response – ResubmitQueueEntry, Response – ResumeQueueEntry, Acknowledge – ResumeQueueEntry, Response – SetQueueEntryPosition, Response – SetQueueEntryPriority, Response – SubmitQueueEntry, Acknowledge – SubmitQueueEntry, Response – SuspendQueueEntry, Acknowledge – SuspendQueueEntry

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>DeviceID</i>	r?			w			
<i>Status</i>	r?			w			
<i>QueueEntry</i>	r?			w←			See [JDF1.4a] Section 5.14.2 “QueueEntry”. See Table 84: QueueEntry.

6.21.2 QueueEntry

Table 84: QueueEntry

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

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>JobID</i>	r?			w←			If the QueueEntry is being returned in a Response to a Message other than SubmitQueueEntry, <i>JobID</i> MUST be specified.
<i>JobPartID</i>	r?			w←			If the QueueEntry is being returned in a response to a Message other than SubmitQueueEntry, <i>JobPartID</i> from the JDF Root Node MUST be specified.
<i>Priority</i>	r?			w←			If the Worker Supports priority of Queue entries, it MUST write this value.
<i>QueueEntryID</i>	r?			w			

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
Status	r?			w			
SubmissionTime	r?			w←			If the QueueEntry is being written in the Response to the QueueStatus Message, the Worker MUST write the <i>SubmissionTime</i> .
Part	r?			w←			If there is an AncestorPool in the Root Node of the submitted JDF, and Part Elements appear within AncestorPool, the QueueEntry/Part Elements MUST be copies of AncestorPool/Part of the Root Node of the submitted JDF. See [JDF1.4a].

6.21.3 QueueEntryDef

Table 85: QueueEntryDef

Referenced by: QueueFilter, Command – AbortQueueEntry, Command – HoldQueueEntry, Command – RemoveQueueEntry, Command – ResumeQueueEntry, Command – SuspendQueueEntry

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
QueueEntryID	w			r			r-Test: The Queue entry referenced by <i>QueueEntryID</i> (and no other) is acted upon (or included in the response for queries) based on the type of the message.
Comment	w?			r?			See Table 87: Comment.

6.21.4 QueueFilter

Table 86: QueueFilter

Referenced by: Command – AbortQueueEntry, Command – HoldQueueEntry, Query – QueueStatus, Command – RemoveQueueEntry, Command – ResourcePull, Command – ResubmitQueueEntry, Command – ResumeQueueEntry, Command – SetQueueEntryPosition, Command – SetQueueEntryPriority, Command – SubmitQueueEntry, Command – SuspendQueueEntry

Name or Value	Manager			Worker			Description	
	Level →	1	2	3	1	2		3
<i>MaxEntries</i>		w			r			When using QueueFilter with any Message other than QueueStatus, the Manager SHOULD write <i>MaxEntries</i> = "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 <i>MaxEntries</i> .
<i>QueueEntryDetails</i>		w			r			r-Test: The QueueEntry Elements in the Response contain the information corresponding to the value of <i>QueueEntryDetails</i> .
<i>None</i>		w←			r			
<i>Brief</i>		w?			r			
<i>all remaining values</i>		!w			r?			
QueueEntryDef		w?			r			The QueueEntryDef Element can be used to filter the response down to a single Queue entry. See Table 85: QueueEntryDef.

6.22 Other Elements

6.22.1 Comment

Table 87: Comment

Referenced by: QueueEntryDef, Command – ResubmitQueueEntry, QueueSubmissionParams

Name or Value	Manager			Worker			Description	
	Level →	1	2	3	1	2		3
<i>Name</i>		w			r?			

Name or Value	Manager			Worker			Description
	Level →	1	2	3	1	2	
<i>Instruction</i>	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).
<i>all remaining values</i>	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 MUST result in a single `QueueEntry` being created by the Worker, and the Worker MUST 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 MUST set the `QueueEntry/@Status` to `"Waiting"` or `"Running"`, and MUST send any Signals associated with this change of status.

The Manager MUST 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 MUST 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 MUST 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 MUST 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 MUST use one of these options:

- A. The JDF Instance and the JMF Message are packaged together using MIME. The JMF Message MUST be the first part of the MIME package, and MUST 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 MUST Support both option A and option B.

8 References

8.1 Normative References

[Base-ICS] Base ICS, Version 1.4, published December 2009. Available at: <http://www.cip4.org>.

[JDF1.4a] JDF Specification, Version 1.4a, published December 17, 2009. Available at: <http://www.cip4.org>.

Appendix A: Changes from JMF ICS 1.3

This appendix lists the changes made to JMF ICS 1.3 to make JMF ICS 1.4:

Table 88: Changes from JMF ICS 1.3

Location	Description
1. Table 2: Conformance Levels, Table 3: NodeInfo, and throughout ICS	Removed Level 1 (NodeInfo/JMF method in <code>SubmitQueueEntry</code> to create a Persistent Channel for that <code>QueueEntry</code> only) and moved all Level 2 conformance to Level 1. An HTTP Manager with a Hot Folder Worker no longer seems a useful combination. Note: Base ICS 1.4 Level 0 and 1 (Manager and Worker Hot Folder) remains in the Base ICS 1.4 (but made OPTIONAL in Base ICS 1.4 Level 2).
2. Table 14: List of JMF Messages	For Producers and Consumers: Added REQUIRED <code>KnownSubscriptions</code> Query and Response and OPTIONAL <code>ResourcePull</code> Command and Response.
3. Table 84: <code>QueueEntry</code>	Added the clarification that the <code>QueueEntry/@JobPartID</code> MUST be the <code>JobPartID</code> from the JDF root.
4. Table 24: <code>DeviceFilter</code>	Clarified the r-Test for <code>DeviceDetails</code> = " <code>None</code> ", " <code>Brief</code> ", and " <code>Details</code> ".
5. Table 27: <code>DeviceInfo</code>	Clarified Producer condition for Device .
6. Table 30: <code>KnownMsgQuParams</code>	Added <code>ChannelMode</code> and values " <code>FireAndForget</code> " and " <code>Reliable</code> ".
7. Table 32: <code>MessageService</code>	Added <code>ChannelMode</code> .
8. Table 13: Subscription - <code>KnownSubscriptions</code> Response, Table 33: Query - <code>KnownSubscriptions</code> , Table 34: <code>SubscriptionFilter</code> , Table 35: Response - <code>KnownSubscriptions</code> , and Table 36: <code>SubscriptionInfo</code>	Added these tables for the REQUIRED <code>KnownSubscriptions</code> Query and Response for Producers and Consumers.
9. Table 51: Command - <code>ResourcePull</code> , Table 52: <code>ResourcePullParams</code> , and Table 53: Response - <code>ResourcePull</code>	Added these tables for the OPTIONAL <code>ResourcePull</code> Command and Response for Producers and Consumers to able to request an additional number of output resources to be produced.
10. Table 54: Command - <code>ResubmitQueueEntry</code>	Added <code>AcknowledgeURL</code> and <code>Comment</code> to correspond to <code>SubmitQueueEntry</code> .
11. Table 77: <code>QueueSubmissionParams</code>	Clarified <code>ReturnJMF</code> and <code>URL</code> can have " <code>https</code> " or " <code>http</code> ".

CIP4 THANKS ITS PARTNER LEVEL MEMBERS

