

IPC-2591-Version 1.7

2023 - August

Connected Factory Exchange (CFX)

Supersedes IPC-2591-Version 1.6
March 2023

An international standard developed by IPC

IPC Mission

IPC is a global trade association dedicated to furthering the competitive excellence and financial success of its members, who are participants in the electronics industry.

In pursuit of these objectives, IPC will devote resources to management improvement and technology enhancement programs, the creation of relevant standards, protection of the environment, and pertinent government relations.

IPC encourages the active participation of all its members in these activities and commits to full cooperation with all related organizations.

About IPC Standards

IPC standards and publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for their particular need. Existence of such IPC standards and publications shall not in any respect preclude any entity from manufacturing or selling products not conforming to such IPC standards and publication, nor shall the existence of such IPC standards and publications preclude their voluntary use.

IPC standards and publications are approved by IPC committees without regard to whether the IPC standards or publications may involve patents on articles, materials or processes. By such action, IPC does not assume any liability to any patent owner, nor does IPC assume any obligation whatsoever to parties adopting an IPC standard or publication. Users are wholly responsible for protecting themselves against all claims of liabilities for patent infringement.

IPC Position Statement on Specification Revision Change

The use and implementation of IPC standards and publications are voluntary and part of a relationship entered into by customer and supplier. When an IPC standard or publication is revised or amended, the use of the latest revision or amendment as part of an existing relationship is not automatic unless required by the contract. IPC recommends the use of the latest revision or amendment.

Standards Improvement Recommendations

IPC welcomes comments for improvements to any standard in its library. All comments will be provided to the appropriate committee.

If a change to technical content is requested, data to support the request is recommended. Technical comments to include new technologies or make changes to published requirements should be accompanied by technical data to support the request. This information will be used by the committee to resolve the comment.

To submit your comments, visit the IPC Status of Standardization page at www.ipc.org/status.



IPC-2591-Version 1.7

Connected Factory Exchange (CFX)

Developed by the IPC-CFX Standard Task Group (2-17a) of the
Electronic Product Data Description Committee (2-10) of IPC

Supersedes:

IPC-2591-Version 1.6 -
March 2023
IPC-2591-Version 1.5 -
July 2022
IPC-2591-Version 1.4 -
December 2021
IPC-2591-Version 1.3 -
February 2021
IPC-2591-Version 1.2 -
September 2020
IPC-2591-Version 1.1 -
January 2020
IPC-2591 -
March 2019

Users of this publication are encouraged to participate in the
development of future revisions.

Contact:

IPC
3000 Lakeside Drive, Suite 105N
Bannockburn, Illinois
60015-1249
Tel 847 615.7100
Fax 847 615.7105



Documents

Document Sharing Hub

This Page Intentionally Left Blank

Acknowledgment

Any document involving a complex technology draws material from a vast number of sources across many continents. While the principal members of the IPC-CFX Standard Task Group (2-17a) of the Electronic Product Data Description Committee (2-10) are shown below, it is not possible to include all of those who assisted in the evolution of this standard. To each of them, the members of the IPC extend their gratitude.

Electronic Product Data Description Committee	IPC-CFX Standard Task Group	Technical Liaison of the IPC Board of Directors
Chair Michael Ford Aegis Software	Co-Chairs Thomas Marktscheffel ASMPT GmbH & Co. KG Michael Ford Aegis Software	Bob Neves Microtek (Changzhou) Laboratories

Plug & Players A-Team

IPC recognizes this A-Team for their exceptional leadership and effort in the development of this standard.

Tim Burke Arch Systems	Naim Kapadia Grinsty Rail Ltd	Bruno Muller Essemtec AG
Michael Ford Aegis Software	Wes Karpiak Celestica International L.P.	Dmitrii Pesterev Continental Automotive Hungary Kft.
Alexis Fouquet Europlacer	Thomas Marktscheffel ASMPT GmbH & Co. KG	John Walls Aegis Software

IPC-CFX Standard Task Group

Timon Ahlgrim ERSA GmbH	Julie Cliche-Dubois Cogiscan Inc.	Tom Hamelinckx ST Engineering iDirect
Nicolas Bartschat ERSA GmbH	Craig Dickinson Flextronics International	Richard Helou 3D Photonix
Jennifer Bennett IBM Corporation	Gruian Dinu-Tudor Continental Automotive Systems, S.R.L.	Pei Hung Hsieh Institute for Information Industry
Anastasia Bobkova SMH Technologies	Benoit Fillastre MBtech	Constantin Hudon East West Quebec
Frederic Bourdat MBtech	Michael Ford Aegis Software	Ramesh Kandasamy Getech Automation
Tim Burke Arch Systems	Steve Foster ITW EAE	Naim Kapadia Grinsty Rail Ltd
Javier Caraccioli Lockheed Martin Missiles & Fire Control	Alexis Fouquet Europlacer	Wes Karpiak Celestica International L.P.
YenChi Chang Mata Inventive	Mahendra Gandhi Northrop Grumman Space Systems	Noriyuki Kato Omron Manufacturing, Netherlands
Zhiman Chen ZHUSHOU CRRC TIMES ELECTRIC CO., LTD.	Feng Gao ZHUSHOU CRRC TIMES ELECTRIC CO., LTD.	Christy Kavanaugh Siemens Industry Inc.
Lin Tzu Chieh Institute for Information Industry	Stephen Golemme Go Let Me	Jeff Kennedy ZESTRON Americas
Sean Clancy HzO, Inc.	Ken Gonzales Schweitzer Engineering Laboratories, Inc.	Theresa Klemmt ERSA GmbH
Thomas Cleere BAE Systems	Helena Gurusamy CloudVidia	Nathan Knipe Lockheed Martin Missiles & Fire Control
		Caitlin Knowles AFOA

Michael Lange Northrop Grumman	John Neiderman Vitronics Soltec	Vladimir Sitko PBT Works s.r.o.
Minhao Lee ITRI	Arne Neiser SEHO Systems GmbH	Zakary Smith Mata Inventive
Geoffrey Leeds Insulectro	Nuno Nunes Critical Manufacturing	Jason Spera Aegis Software
Hugo Leite Critical Manufacturing	Catalina Pamatmat Temic Automotive (Phils.) Inc.	John Striker Vitronics Soltec
Chuck LePard DXC Technology	Marc Peo Heller Industries Inc.	Lauren Tori HGC Westshore LLC
Andrew Leslie BAE Systems	Dmitrii Pesterev Continental Automotive Hungary Kft.	Jesse Vaughan Summit Interconnect - Anaheim
Te San Liao Institute for Information Industry	Bart Piwowar DarwinAI	John Walls Aegis Software
Te-ming Liao Sunsda Technology Co., Ltd.	Sarumathi Ramasamy Getech Automation	Jeff Wayne NAS Electronics
Thomas Marktscheffel ASMPT GmbH & Co. KG	Michael Rauter Flextronics International GmbH	Jarrold Webb Lockheed Martin Missiles & Fire Control
Neil Martin The Manufacturing Technology Centre	Naveen Ravindran ZESTRON Americas	Perla Wehbe I.F. Engineering
Zohair Mehkri Flextronics International	Andreas Reinhardt SEHO Systems GmbH	Chia-ta Wu Sunsda Technology Co., Ltd.
Markus Mittermair Rehm Thermal Systems GmbH	Ivan Rinaldi SMH Technologies	Yu Yueh Delta Electronics
Miles Moreau KIC	Raquel Rodriguez Quintero INSYTE, S.A.	Yuson Zhang GBS Communication Technology Co., Ltd
Klaus Mueller 42Q	Robert Schmeisser Siemens AG	Paul Zutter U.S. Army Aviation & Missile Command
Bruno Muller Essemtec AG	Cameron Shearon Raytheon Company	

Table of Contents

1	SCOPE	1	2	APPLICABLE DOCUMENTS	4
1.1	Purpose	1	2.1	IPC	4
1.2	Application of This Standard	1	2.2	ECMA International	4
1.3	CFX and The Hermes Standard	1	2.3	International Organization for Standardization (ISO)	4
1.4	Updates to This Standard	1	2.4	SEMI	4
1.5	Definition of Requirements	2	3	GENERAL REQUIREMENTS	4
1.6	Order of Precedence	2	3.1	Guidance on the Use of This Standard	4
1.6.1	Conflict	2	3.1.1	Technical Reference	4
1.6.2	Clause References	2	3.1.2	Application Reference	4
1.7	Acronyms	2	3.2	Users of CFX	4
1.8	Terms and Definitions	2	3.2.1	Automated Assembly Processes	5
1.8.1	Activity	2	3.2.2	In-House Manufacturing Solution Development	5
1.8.2	Component	2	3.2.3	MES Software Solutions	5
1.8.3	Dashboard	2	3.3	Software Development Environment	5
1.8.4	Data Integrity	2	3.4	CFX Support Declaration	5
1.8.5	Endpoint	3	4	CFX STRUCTURAL OVERVIEW	5
1.8.6	Factory Resource	3	4.1	Primary Transport Layer: AMQP v1.0	5
1.8.7	Lane	3	4.1.1	CFX Message Channels	6
1.8.8	Lock	3	4.1.2	Channel Configuration	7
1.8.9	Material Carrier	3	4.1.3	CFX Message Types	7
1.8.10	Material Chain	3	4.1.4	CFX Compression	7
1.8.11	Material Location	3	4.1.5	CFX – AMQP Message Properties	7
1.8.12	Material Package	3	4.2	Encoding: JSON	7
1.8.13	Material Traceability	3	4.2.1	JSON Data Types	8
1.8.14	Materials	3	4.3	CFX-Defined Content	9
1.8.15	Operator	3	4.4	CFX Key Parameters	9
1.8.16	Process Endpoint (Station)	3	4.4.1	Endpoint Identification (CFX Handle)	9
1.8.17	Production Unit	3	4.4.2	TransactionID	9
1.8.18	Recipe	3	4.5	CFX Message Envelope	10
1.8.19	Root	3	4.6	Operator Information	10
1.8.20	Setup	3	4.7	CFX Endpoint Configuration	10
1.8.21	State (Production State)	3	4.7.1	Specific CFX Endpoint Configuration Addresses	11
1.8.22	Station (Process Endpoint)	3	4.8	Message Attachments	12
1.8.23	Stage	3	4.8.1	Specifying Message Attachments	12
1.8.24	Subassembly	3	4.8.2	In-Band Message Attachments	12
1.8.25	Symptom	3	4.8.3	Out-of-Band Attachments	13
1.8.26	Tool	4			
1.8.27	Transactional Endpoint	4			

4.8.4	Supported Tabular Data Formats for Message Attachments	13	7.2.4	CFX.InformationSystem.UnitValidation (Level 2).....	33
4.9	Supported Message Attachment Types.....	14	7.2.5	CFX.InformationSystem.WorkOrderManagement (Level 2).....	33
4.9.1	SolderPasteInspection.SolderPaste Measurement	14	7.3	CFX.Maintenance (Level 1).....	33
4.9.2	PCBInspection.OffsetMeasurement.....	14	7.4	CFX.Materials (Level 1)	34
4.10	Messages Supporting Attachments	15	7.4.1	CFX.Materials.Management (Level 2).....	34
4.10.1	CFX.Production.TestAndInspection.Units Inspected Attachments.....	15	7.4.1.1	CFX.Materials.Management.MSDManagement (Level 3).....	35
4.11	CFX Message Recorder	15	7.4.2	CFX.Materials.Storage (Level 2)	35
5	CFX OPERATIONAL MODELING	16	7.4.3	CFX.Materials.Transport (Level 2)	36
5.1	Equipment State Model	16	7.5	CFX.Production (Level 1).....	36
5.2	Station Fault Event Model.....	18	7.5.1	CFX.Production.Application (Level 2)	38
5.3	Production Unit Architecture.....	18	7.5.2	CFX.Production.Assembly (Level 2)	38
5.3.1	Production Unit Status	20	7.5.2.1	CFX.Production.Assembly.PressInsertion (Level 3).....	38
5.3.2	Production Endpoint Decision-Making Categories	20	7.5.3	CFX.Production.Hermes (Level 2).....	39
5.3.3	Examples of Production Unit Status	21	7.5.4	CFX.Production.LoadingAndUnloading (Level 2).....	39
5.3.3.1	Processing Station.....	21	7.5.5	CFX.Production.Processing (Level 2)	39
5.3.3.2	Processing Station Creating Scrap	21	7.5.6	CFX.Production.ReworkAndRepair (Level 2) .	39
5.3.3.3	Inspection Station Identifying Potential Defect	21	7.5.7	CFX.Production.TestAndInspection (Level 2) .	40
5.3.3.4	Review/Repair Station Resolving Failed Production Unit.....	22	7.6	CFX.ResourcePerformance (Level 1).....	40
5.3.3.5	Review/Repair Station Scrapping Failed Production Unit.....	23	7.6.1	CFX.ResourcePerformance.PressInsertion (Level 2).....	41
6	CFX TOPICS AND DYNAMIC STRUCTURES	23	7.6.2	CFX.ResourcePerformance.SMTPlacement (Level 2).....	42
6.1	Hierarchy of CFX Topics.....	24	7.6.3	CFX.ResourcePerformance.SolderPastePrinting (Level 2).....	42
6.1.1	CFX Topic Support Declaration.....	25	7.6.4	CFX.ResourcePerformance.THTInsertion (Level 2).....	42
6.2	CFX Message Names	25	7.7	CFX.Sensor (Level 1)	42
6.3	CFX Structures	25	7.7.1	CFX.Sensor.Identification (Level 2)	42
6.4	CFX Dynamic Structures.....	25	7.8	CFX Message Flow	43
6.5	CFX Messaging Requirements by Equipment Type	25	7.8.1	Production Endpoint (Station) Connection.....	43
7	CFX MESSAGES	31	7.8.2	Station State Transition.....	44
7.1	Root Level Messages.....	31	7.8.3	Station Processing	45
7.2	CFX.InformationSystem (Level 1).....	32	8	CFX TECHNICAL REFERENCE	46
7.2.1	CFX.InformationSystem.DataTransfer (Level 2).....	32	APPENDIX A	Version Updates	47
7.2.2	CFX.InformationSystem.OperatorValidation (Level 2).....	32	APPENDIX B	Index of Acronyms and Abbreviations	58
7.2.3	CFX.InformationSystem.ProductionScheduling (Level 2).....	32			

Figures	Tables
Figure 1-1 Version Change Tracking Example..... 2	Table 4-1 Types of CFX Messages..... 7
Figure 4-1 CFX Channels Between Endpoints 6	Table 4-2 CFX – AMQP Message Properties 8
Figure 4-2 The CFX TransactionID 9	Table 4-3 CFX Message Envelope 10
Figure 5-1 SEMI E10 Equipment State Model 15	Table 4-4 Format of Attachments Array Entries 12
Figure 5-2 Examples of Groupings of Production Units 19	Table 4-5 Columns for SolderPasteMeasurement Attachment Type..... 14
Figure 5-3 Panelized Printed Board (Top View) 20	Table 4-6 Columns for OffsetMeasurement Attachment Type..... 15
Figure 5-4 CFX Unit Locations Identified on Multiple-Board Panel (Top View)..... 20	Table 5-1 Station Event Fault Model..... 18
Figure 5-5 Panel Processing Example — No Defects Found 21	Table 6-1 CFX Capability Requirements by Equipment Type 26
Figure 5-6 Panel Processing Example – One Production Unit Scrapped 21	Table 6-2 Required Messages by Capability 28
Figure 5-7 Panel Inspection Example – One Production Unit Failure 22	Table 7-1 CFX.Root Messages 32
Figure 5-8 Panel Review/Repair Example – One Production Unit Repaired 22	Table 7-2 CFX.InformationSystem.DataTransfer Messages..... 32
Figure 5-9 Panel Review/Repair Example – Whole Panel Scrapped..... 22	Table 7-3 CFX.InformationSystem.OperatorValidation Messages..... 32
Figure 5-10 CFX Production Station Process Model ... 23	Table 7-4 CFX.InformationSystem.Production Scheduling Messages..... 32
Figure 7-1 CFX Station Connection Example Message Flow 43	Table 7-5 CFX.InformationSystem.UnitValidation Messages..... 33
Figure 7-2 CFX Station State Transition Example Message Flow 44	Table 7-6 CFX.InformationSystem.WorkOrder Management Messages..... 33
Figure 7-3 CFX Station Processing Example Message Flow 45	Table 7-7 CFX.Maintenance Messages 34
	Table 7-8 CFX.Materials.Management Messages 35
	Table 7-9 CFX.Materials.Management.MSD Management Messages..... 35
	Table 7-10 CFX.Materials.Storage Messages 36
	Table 7-11 CFX.Materials.Transport Messages 36
	Table 7-12 CFX.Production Messages 37
	Table 7-13 CFX.Production.Application Messages 38
	Table 7-14 CFX.Production.Assembly Messages 38
	Table 7-15 CFX.Production.Assembly.PressInsertion Messages..... 38
	Table 7-16 CFX.Production.Hermes Messages 39
	Table 7-17 CFX.Production.LoadingAndUnloading Messages 39
	Table 7-18 CFX.Production.Processing Messages 39
	Table 7-19 CFX.Production.ReworkAndRepair Messages..... 40

Table 7-20 CFX.Production.TestAndInspection
Messages.....40

Table 7-21 CFX.ResourcePerformance Messages 41

Table 7-22 CFX.ResourcePerformance.Press
Insertion Messages 41

Table 7-23 CFX.ResourcePerformance.SMTPlacement
Messages.....42

Table 7-24 CFX.ResourcePerformance.SolderPaste
Printing Messages 42

Table 7-25 CFX.ResourcePerformance.THTInsertion
Messages.....42

Table 7-26 CFX.Sensor.Identification Messages 43

IPC-2591, Version 1.7

Connected Factory Exchange (CFX)

1 SCOPE

This standard establishes the requirements for the omnidirectional exchange of information between manufacturing processes and associated host systems for assembly manufacturing. This standard applies to communication between all executable processes in the manufacture of printed board assemblies – automated, semiautomated and manual – and is applicable to related mechanical assembly and transactional processes.

1.1 Purpose With the growth and acceptance of digital modeling and practices in manufacturing, the lack of a holistic Industrial Internet of Things (IIoT) standard for the transfer of information between machines, systems and processes has become a severe limitation to the growth of digitization and computerization in the electronics manufacturing industry, inhibiting technology innovations such as Industry 4.0 and Smart Factories being available to all companies in the industry, regardless of size, sector and location.

This Connected Factory Exchange (CFX) standard provides a true “plug and play” Internet of Things (IoT) communication environment throughout manufacturing, where all equipment, manufacturing processes and transactional stations can communicate with each other without the need for the development and use of bespoke interfaces. CFX-enabled equipment and solutions from different vendors work seamlessly together.

There are many types of users of this CFX standard, including equipment vendors, solution providers, in-house information technology (IT) groups, etc. The many types of data included in CFX are used in different ways depending on the application; for example, closed-loop feedback systems, live production dashboards, traceability (IPC-1782), manufacturing execution systems (MES) control, lean supply chain management, active quality management, production control, etc.

As CFX data is fully omnidirectional, any CFX endpoint connection can consume data as well as create it. As an illustration, consider the scenario in which a single machine from a certain vendor is connected in-line with other machines from different vendors. CFX messages are sent from the single machine to other machines in the line, and to host systems such as MES. The single machine can also receive CFX messages from all other machines in the line, as well as from the host systems in order to optimize the machine operation and enable the vendor of the machines to create added-value functionality, such as to support machine-specific Industry 4.0. In this way, a smart, digital, Industry 4.0 factory will be comprised of many different Industry 4.0 computerization applications, each of which can be provided by different suppliers, at the machine, line, site and even enterprise levels, all working together, sharing data seamlessly through CFX.

This CFX standard supports the concept of big data by including data of different types from across the factory, including performance, materials, resources, users, quality events, product tracking, etc., all of which can be combined to create a big-data environment. CFX, therefore, provides many kinds of added value opportunities to the whole manufacturing operation, including, for example, improving operational efficiency and productivity, quality and reliability, agility and responsiveness. This CFX standard helps organizations ensure that end users/consumers will receive products and services that meet or exceed their expectations and in the timeliest and most economically viable method.

1.2 Application of This Standard This standard defines the communication protocol and content across all assembly production processes, irrespective of type or method of operation. It can also be applied to transactional operations. There are no restrictions in terms of product classification sector, size of operation or location. Surface-mount technology (SMT) production is not required to be a part of the factory. Though intended to support all aspects of printed board production, the use of CFX can be extended downstream to include, for example, mechanical assembly, personalization, packing and shipping, as well as upstream to include, for example, electrical and mechanical subassemblies.

1.3 CFX and The Hermes Standard This CFX standard is complementary to IPC-HERMES-9852. The Hermes Standard, as an advanced, intelligent Surface Mount Equipment Manufacturers Association (SMEMA) standard replacement, provides near-instant line control, passing information about production units as they pass down the line. CFX provides vertical messaging that is complementary to Hermes.

1.4 Updates to This Standard The IPC-CFX Standard Task Group intends to make frequent incremental revisions to this standard to support additional machines and processes. Version updates are identified by version number and the change (added, removed, etc.), so the reader can easily identify changes in each version. See Figure 1-1 for an example of this version change tracking using a portion of the CFX.Production.TestAndInspection table of this standard as an example.

Appendix A also provides an itemized version history.

All messages in this standard apply to IPC-2591, Version 1.0 and subsequent versions, unless otherwise stated.

Table 7-20 CFX.Production.TestAndInspection Messages

Name	Type	Description
GetInspectionInfo	RR	This message requests from an endpoint solder paste inspection results for a list of reference designators. – Added v1.7
UnitsInspected	E	Sent by a station when one or more units have been inspected.
UnitsTested	E	Sent by a station when one or more units undergo a series of tests.

Figure 1-1 Version Change Tracking Example

1.5 Definition of Requirements The words **shall** or **shall not** are used in the text of this document wherever there is a requirement for materials, preparation, process control or acceptance.

The word “should” reflects recommendations and is used to reflect general industry practices and procedures for guidance only.

Line drawings and illustrations are depicted herein to assist in the interpretation of the written requirements of this Standard.

The text takes precedence over the figures.

1.6 Order of Precedence The contract **shall** take precedence over this standard, referenced standards and drawings.

In the event of conflict, the following order of precedence applies:

- 1) Procurement as agreed and documented between user and supplier.
- 2) Master drawing, design brief or tech pack reflecting the user’s detailed requirements.
- 3) When invoked by the customer or per contractual agreement, this standard.

When documents other than this standard are cited, the order of precedence **shall** be defined in the procurement documents. The user can specify alternate acceptance criteria.

1.6.1 Conflict In the event of conflict between the requirements of this standard and the applicable drawing(s) and documentation, the applicable user-approved drawing(s) and documentation govern.

Some examples of documentation include the contract, purchase order, technical data package, engineering specification or performance specification. In the event of a conflict between the text of this standard and the applicable documents cited herein, the text of this standard takes precedence. In the event of conflict between the requirements of this standard and drawing(s) and documentation that has not been user approved, this standard governs.

1.6.2 Clause References When a clause in this document is referenced, its subordinate clauses apply, unless the requirement references specific subordinate clauses.

1.7 Acronyms See Appendix B for a list of acronyms used in this standard.

1.8 Terms and Definitions Other than those terms listed below, the definitions of terms used in this standard are in accordance with IPC-T-50.

1.8.1 Activity An action performed by an endpoint in the course of performing its function. There are many different types of activities that may be performed, depending on the specific function of the endpoint. Activities may be value added (such as placing components on a printed board), or non-value added (such as a tool change by a pick-and-place machine).

1.8.2 Component The component is a single instance of a bulk material, such as a resistor, capacitor, integrated circuit (IC), etc. Bulk components are normally supplied in material packages.

1.8.3 Dashboard An information management tool used to track key performance indicators (KPIs), metrics and other key data points relevant to a business, department or specific process. Through the use of data visualizations, dashboards simplify complex data sets to provide users with at-a-glance awareness of current performance.

1.8.4 Data Integrity Maintaining and ensuring accuracy and consistency of data over the data’s entire lifecycle. Data integrity is a critical aspect to the design, implementation and usage of any system which stores, processes or retrieves data.

1.8.5 Endpoint A participant in a CFX network, which is a software instance that can both send and receive CFX messages. Each endpoint is a software program associated with a particular machine or process, used to exchange information related to the operation.

1.8.6 Factory Resource A physical location, machine, tool or device within the factory that performs any sort of function.

1.8.7 Lane A corridor within an endpoint down which production units flow through a series of production stages (see 1.8.23). Many endpoints only have one singular lane, but multiple production lanes on a production line are possible and supported by CFX.

1.8.8 Lock A process lock, or locked process, is one in which the endpoint is not allowed to continue any active operation.

1.8.9 Material Carrier The material carrier is the mechanism by which a material package is located for use at an endpoint. In the case of SMT, for example, a feeder is a common example of the material carrier.

1.8.10 Material Chain A set of two or more material packages connected together (i.e., by splicing them together).

1.8.11 Material Location Identifies a specific location on an endpoint where material packages may be loaded or otherwise mounted (perhaps employing a material carrier to do so). For example, a single feeder slot on a feeder bank of a pick-and-place machine is a single material location.

1.8.12 Material Package Material package refers to the physical mechanism in which bulk materials are supplied (e.g., on a reel, in a box or bag or on a tray). A single material package represents one singular package of material (one reel, one box, etc.) and is identified by a unique identifier that is typically labeled on to the package.

1.8.13 Material Traceability The ability to verify the history, location or application of an item by means of documented recorded identification.

1.8.14 Materials For purposes of this standard, all physical and nonphysical items which are combined together to create a final assembly.

1.8.15 Operator A production operator who can actively work at an endpoint or is responsible for its operation.

1.8.16 Process Endpoint (Station) An endpoint associated with a production process of any kind, whether automated or manual. Normally referred to as a station.

1.8.17 Production Unit A singular instance of an item that is being produced via a production process. For printed board assembly, a single printed board would be considered a production unit. A single panelized printed board layout may contain multiple printed boards/production units on one single panel. For final assembly, the uppermost assembly would also be considered a production unit itself, perhaps containing one or more subassemblies, which are themselves production units produced by a different, previously executed process.

1.8.18 Recipe A set of instructions, data or documentation used to control or guide the activity at an endpoint. For an SMT machine, this would be the set of files that form the machine program.

1.8.19 Root The highest level in the structure, above level 1.

1.8.20 Setup A list of materials and tools that are required to be in place in specific locations at a station.

1.8.21 State (Production State) This refers to the state or mode of operation of a station. Possible states vary by process type, identified by the process state model.

1.8.22 Station (Process Endpoint) An endpoint associated with a production process of any kind, whether automated or manual.

1.8.23 Stage An area or zone within a Production Station endpoint in which added-value work is performed.

1.8.24 Subassembly An assembly that will become a material of another assembly, whether they are made in the same environment as their peers in the hierarchical product bill of materials (BOM) or transferred from outside as materials.

1.8.25 Symptom An indication that something may be wrong with a production unit under test.

1.8.26 Tool An item used by a person or automated endpoint in the course of performing a particular function. Examples of tools used by humans would be wrench, hammer, screwdriver, etc. Examples of tools used by automated endpoints would be stencil or squeegee (for a stencil printer), head or nozzle (for a pick-and-place machine).

1.8.27 Transactional Endpoint An endpoint associated with a production process that manages or transacts, rather than being associated with an operational process. A transactional endpoint is usually a piece of software that performs some type of visibility or control function, such as a material logistics terminal.

2 APPLICABLE DOCUMENTS

2.1 IPC¹

IPC-T-50 Terms and Definitions for Interconnecting and Packaging Electronic Circuits

IPC-1782 Standard for Manufacturing and Supply Chain Traceability of Electronic Products

IPC-2581 Generic Requirements for Printed Board Assembly Products Manufacturing Description Data and Transfer Methodology

IPC-HERMES-9852 The Global Standard for Machine-to-Machine Communication in SMT Assembly

2.2 ECMA International²

Standard ECMA-262 3rd Edition ECMAScript Language Specification

2.3 International Organization for Standardization (ISO)³

ISO 8601:2004 Data elements and interchange formats—Information interchange—Representation of dates and times

2.4 SEMI⁴

SEMI E58-0703 Automated Reliability, Availability, and Maintainability Standard (ARAMS): Concepts, Behavior, and Services

SEMI E10-0814E Specification for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM) and Utilization

3 GENERAL REQUIREMENTS

3.1 Guidance on the Use of This Standard This standard is comprised of two parts:

- 1) The first part is this document.
- 2) The second part is available online, which describes the detailed technical specification of messages and data structures.

The combination of both parts forms the complete documentation of this standard.

3.1.1 Technical Reference The link to the technical reference part of the standard is www.connectedfactoryexchange.com/CFXDemo/sdk.

This link can also be used to access the CFX software development kit (SDK).

3.1.2 Application Reference This CFX standard has a dedicated website that provides up to date information about the standard, applications and supporters. Information is also posted about any updates that may be made to the CFX messaging as the scope expands to cover additional and new technologies. Visit www.ipc-cfx.org.

3.2 Users of CFX There are many kinds of CFX endpoints that perform different purposes and roles within the overall CFX network. Each of them is able to create and receive CFX messages.

3.2.1 Automated Assembly Processes A CFX endpoint can be created as part of the software package that exists as part of or together with an automated machine process. This can be applied as a standard part of the machine software or as an optional extra. CFX native support is expected to be built into new machines that appear on the market and provided where practical for machines currently supported in the market. In certain environments, where multiple physical machines are managed within a specific single-vendor-provided software environment, the vendor software can provide a single access point for CFX

1. www.ipc.org

2. www.ecma-international.org

3. www.iso.ch

4. www.semi.org

communication, through which messages will flow as if directly to the machines. This emulation facility allows machine vendors to utilize CFX in line applications as well as machine applications.

The native CFX interface represents a single data exchange mechanism to provide information about the machine operation to users of the machines, replacing the need for additional proprietary interfaces.

Where there is no practical way to provide CFX native support, machine vendors and third parties may consider the use of a simple inexpensive external hardware which can act as a CFX endpoint. It is connected to the machine through a combination of legacy native interface support and additional sensors and switches as required.

3.2.2 In-House Manufacturing Solution Development CFX provides a single interface for any specialist or bespoke internal manufacturing IT projects, related to enterprise resource planning (ERP), MES, product lifecycle management (PLM), etc. Multiple-point solutions can exist on the same CFX network, each providing niche functionality in a common data environment. CFX messaging promotes the ability for point-solutions to communicate effectively through the option of inclusion of complete information sets in messages.

3.2.3 MES Software Solutions Many areas of CFX work in context within a defined factory environment. MES systems provide the infrastructure around the various aspects of CFX messaging in a single environment, to gain the maximum value from CFX. This does not exclude the use of in-house developed point solutions, which can coexist with MES solutions on the CFX network.

3.3 Software Development Environment CFX endpoint software is created in a choice of development environments. Any language and platform may be used. The default environment is based on Windows, for which an SDK is provided. It is updated in line with the latest CFX specification updates. To access the SDK, visit www.connectedfactoryexchange.com/CFXDemo/sdk.

SDKs for other software development environments are encouraged, supported by members of the IPC Connected Factory Initiative Subcommittee. News and information about these are announced on the CFX website (www.ipc-cfx.org) when available.

3.4 CFX Support Declaration This CFX specification allows the CFX developer to choose the areas of the CFX specification that are appropriate for support based on the nature and capabilities of the machine, process or solution that is being developed. It is essential that the level of support for CFX topics is clearly shown to end-users, as they select different CFX endpoints as part of their overall CFX-based operation. These capabilities should also be detailed when requested by the CFX messaging system. See 6.1.1 for guidance on support declaration and 6.5 for CFX messaging requirements by equipment type.

4 CFX STRUCTURAL OVERVIEW CFX communication consists of three key elements:

- Content – the definition and meaning of data to be encoded
- Encoding – the way in which data are encoded ready for transportation
- Transport – the messaging mechanism and infrastructure

CFX combines all three of these elements to create a true plug-and-play standard.

4.1 Primary Transport Layer: AMQP v1.0 The primary transport layer for CFX messages is Advanced Message Queuing Protocol (AMQP) v1.0 (www.amqp.org). All CFX endpoints **shall** implement support for the primary CFX primary transport layer. Versions of AMQP prior to v1.0 are not compatible with CFX.

AMQP was originally developed by the financial industry to facilitate the reliable processing of secure financial transactions across broad geographies. Over the years, its footprint in the computerized world and IoT space has grown well beyond the bounds of the financial industry, and its natural qualities of high-reliability and secure message transmission make it an ideal fit for CFX.

The use of AMQP as the primary transport protocol also greatly eases the burden of equipment makers and factory-level system providers as they implement their CFX support by eliminating most of the development effort needed to send and receive CFX messages. Equipment providers need not understand, nor worry, about the complexities of message routing and subscription management. Routing architectures can be as simple or complicated as needed.

This CFX standard does not dictate any specific message AMQP v1.0 routing infrastructure, such as specific message brokers, queues, exchanges, pub/sub subscription management, etc. The specifics of message routing are at the discretion of the factory-level system provider. Numerous open-source AMQP software development toolkits are freely available. Additionally, there are numerous message routing systems (message brokers, cloud IoT message buses, message-oriented middleware, etc.) that

support the AMQP v1.0 wire protocol, both easing the burden of factory-level system providers and providing them greater flexibility in their implementations.

The use of alternative transport layers is technically possible and may be applied in special cases, though this requires implementation in addition to the primary transport layer for all parties concerned.

4.1.1 CFX Message Channels CFX messages are exchanged through AMQP 1.0 protocol using one or more channels. A single endpoint may be configured to exchange messages through multiple channels. There are three types of channels:

- **Event Channels**

CFX endpoints publish event messages to these event channels. In most cases, the recipient of event channel messages is an AMQP v1.0 broker or host software, such that the messages can be received by any subscribed endpoints. Event channels are used to publish all event type messages, as well as broadcast requests messages. The network address (URL) and AMQP target (typically identifying an exchange on a message broker) of event channels **shall** be preconfigured as part of the initial CFX configuration for each endpoint. Event channels remain active at all times while the endpoint is active and participating in a CFX network. Each CFX endpoint should be preconfigured to publish events to at least one event channel. Additional event channels may be configured as required.

- **Subscription Channels**

CFX endpoints may also be configured to receive unsolicited events and broadcast responses from one or more sources. The network address (URL) and AMQP source (typically identifying a queue on a message broker) of subscription channels **shall** be preconfigured as part of the initial CFX configuration for each endpoint. Subscription channels remain active at all times while the endpoint is active and participating in a CFX network. Each CFX endpoint should be preconfigured to receive broadcast requests from at least one subscription channel. Additional subscription channels may also be configured as may be required.

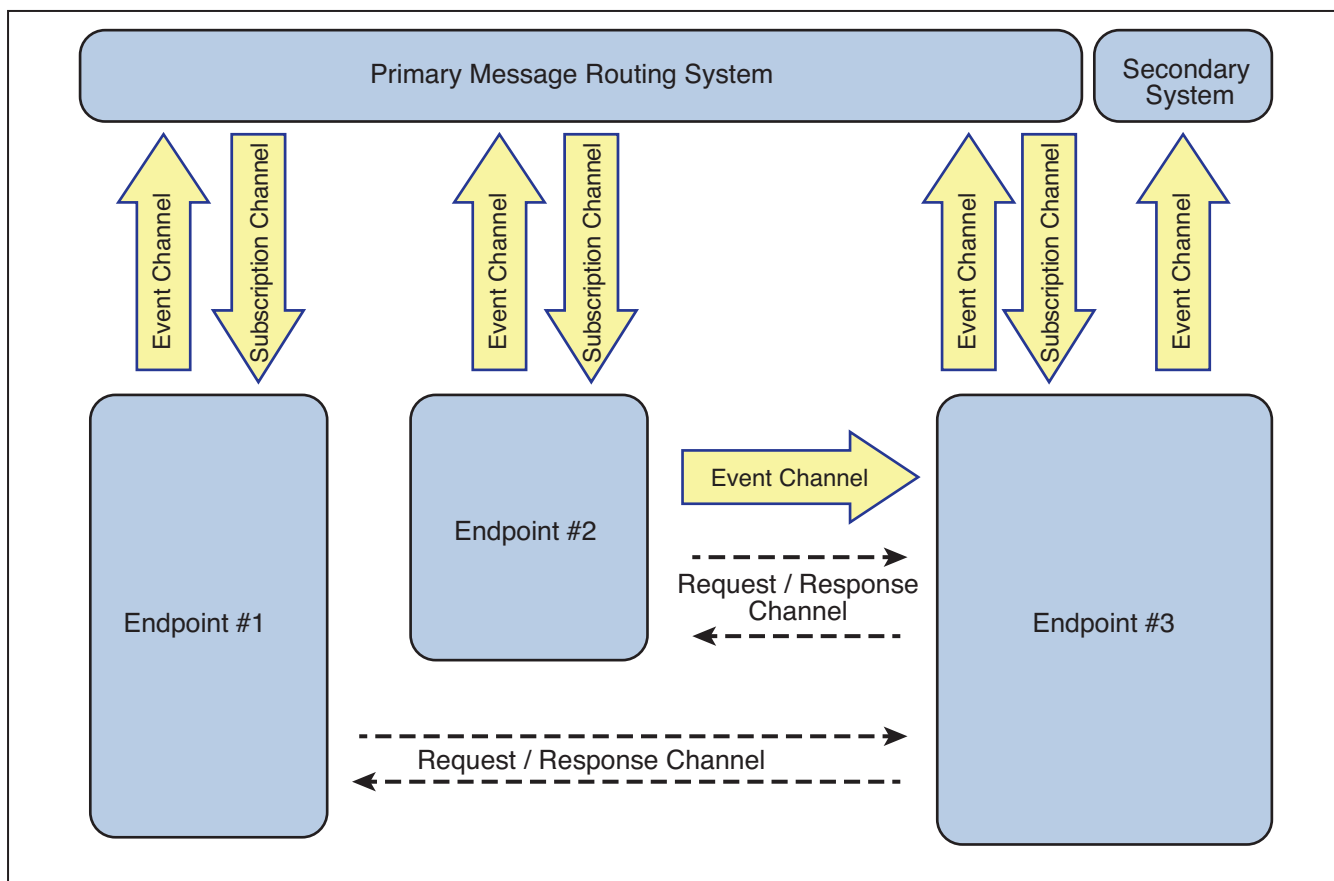


Figure 4-1 CFX Channels Between Endpoints

- **Request/Response Channels**

CFX endpoints may dynamically create request/response channels with which to communicate directly with another CFX endpoint without the need to go through a broker or host. This mechanism is used for time-critical issues such as commands

where, unlike a broadcast message, a response from the specific targeted endpoint is a requirement. Request/Response channels are created on an on-demand basis as needed, when one CFX endpoint wishes to send a request type message to another. The channel remains open for the lifetime of a single request/response transaction and then terminates.

The diagram in Figure 4-1 illustrates the many ways in which a CFX network with a complex set of channels can be created.

In Figure 4-1, all three endpoints have been configured to publish their events to the same factory-level message routing system. Additionally, endpoint #2 has been configured to also publish its events directly to endpoint #3.

The two request/response channels shown in the diagram are ad-hoc. They need not be preconfigured and may be opened and closed by the requester as required. Any CFX endpoint may, at any time, solicit a direct request to any other CFX endpoint.

4.1.2 Channel Configuration Each CFX endpoint should permit end-users to configure any number of event and subscription channels. Additionally, endpoints **shall** permit end-users to enable or disable the transmission of specific CFX topics for each event channel, as well as specific messages within any CFX topics that are configured as enabled.

It might be that in a particular deployment, some of the event content is not needed by any consumers. Filtering messages at the source prevents unnecessary information from cluttering the network and message routing infrastructure.

4.1.3 CFX Message Types Table 4-1 summarizes the CFX message types.

Table 4-1 Types of CFX Messages

Type	Flow	Recipient
Event	Publish (Outbound Flow)	Any Number of Recipients
Request	Source -> Destination	One Targeted Recipient
Response	Destination -> Source	Originator of Request
Broadcast Request	Source -> Destination(s)	Any Number of Recipients
Broadcast Response	Destination(s) -> Source(s)	Any Number of Recipients

4.1.4 CFX Compression Beginning with IPC-2591, v 1.1 (SDK v 1.1.11 and later), the ability to apply compression to message data is supported. As of IPC-2591, v 1.2, compression is applied by default. Compression reduces the size of message data. The amount of compression varies by data type and content. This is enabled by default when using the CFX SDK.

4.1.5 CFX – AMQP Message Properties Beginning with IPC-2591, v 1.2, all CFX messages transmitted via AMQP 1.0 protocol **shall** include the information in Table 4-2 in the AMQP message and message application properties.

4.2 Encoding: JSON The encoding of data in CFX refers to the format in which data fields and structures are written in CFX messages.

JavaScript Object Notation (JSON) (www.json.org) is a lightweight data-interchange format. It is easy for humans to read and write, and it is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition.

All encoding within CFX **shall** comply with UTF-8 as defined by the Unicode Standard (www.unicode.org/standard/standard.html).

Table 4-2 CFX – AMQP Message Properties

Type	Name	CFX Population
AMQP Property	message-id	CFXEnvelope.UniqueID
	user-id	<< Not Populated >>
	to	CFXEnvelope.Target
	subject	CFXEnvelope.Source + " " + CFXEnvelope.MessageName *Default value. Can be customized if desired.
	reply-to	CFXEnvelope.Source
	correlation-id	CFXEnvelope.RequestID
	content-type	application/json; charset="utf-8"
	(no message attachments)	
	content-type	multipart/mixed; boundary="CHOSEN_ASCII_BOUNDARY_CHARACTERS"
	(message attachments)	
	content-encoding	gzip (If using gzip compression, otherwise not populated)
	absolute-expiry-time	<< Not Populated >>
	creation-time	CFXEnvelope.TimeStamp
AMQP Application Property	group-id	<< Not Populated >>
	group-sequence	<< Not Populated >>
	reply-to-group-id	<< Not Populated >>
	cfx-topic	Fully qualified topic name of message contained in CFXEnvelope. (e.g., If MessageName is CFX.Production.WorkStarted, cfxtopic = CFX.Production)
	cfx-message	CFXEnvelope.MessageName
	cfx-handle	CFXEnvelope.Source

4.2.1 JSON Data Types

The following data types are used with JSON data structure and messaging:

- **Enum:** The data element **shall** be one of the allowable choices as defined in an associated list.
Note: The English language will be used for enumeration data.
- **Integer:** A whole number; for example, “123.”
- **Numeric:** A number, represented by, for example, “1.23” or “12.65E3” (no commas, just one decimal point as required); maximum precision is 15 digits.
- **String:** A sequence of characters; for example, “Hello CFX.” The maximum length of a string is 2,000 characters, except where noted.
- **Date/Time:** Represented in ISO 8601 format, with mandatory time zone included. Fractional seconds are optional (i.e., “2018-09-15T15:53:08+05:00”).
- **String Lists:** RFC 4180 compliant comma delimited list of multiple strings. Strings containing commas **shall** have them enclosed by parentheses.
- **Boolean** “1” (True) or “0” (False)
- **GUID:** A system-unique 128-bit integer.
- **MIME:** Encoded binary, where the MIME type and data size are specified.

4.3 CFX-Defined Content The messages used in CFX have defined data content, as defined in the detailed on-line specification (as referred to in 3.1.1). When possible, parameters, values and options have been defined to represent technology-based information, rather than being vendor-dependent or proprietary.

4.4 CFX Key Parameters The key parameters in 4.4.1 and 4.4.2 are used throughout CFX messages.

4.4.1 Endpoint Identification (CFX Handle) Each endpoint is identified through the use of a CFX handle, which is represented by a string of text that is unique within each CFX network instance. Each CFX endpoint is expected to allow end-users to configure the associated CFX handle, rather than it be hard-coded. It is recommended that CFX handles have a human-readable significance. Characters chosen as part of a CFX handle **shall** be able to be safely encoded in UTF-8/JSON format. The length of a CFX handle **shall not** exceed 50 characters.

A CFX handle does not match necessarily to a singular network address or host, as a single network address might service more than one CFX endpoint participant. For example, an SMT machine line controller might provide CFX support for multiple machines. Within CFX, each machine is regarded as a separate endpoint with its own unique CFX handle, all serviced through a single network point.

In order to avoid duplication within the industry, the recommended CFX Handle format is:

<maker>.<model>.<serial>

For example: “Acme.Rabbit.00001”

4.4.2 TransactionID Many CFX messages in the CFX.Production topic share TransactionID as a common parameter. This parameter is used to tie events associated with a production unit as it is processed by a CFX endpoint. The TransactionID is created by the CFX.Production.WorkStarted message. Any message that then uses the TransactionID can be interpreted as being relevant to a specific production unit during the entire processing operation. Messages may also be sent after the CFX.Production.WorkCompleted message using the TransactionID, in the case, for example, time was taken by the endpoint to prepare data about the production unit without blocking the operation of the machine. See Figure 4-2.

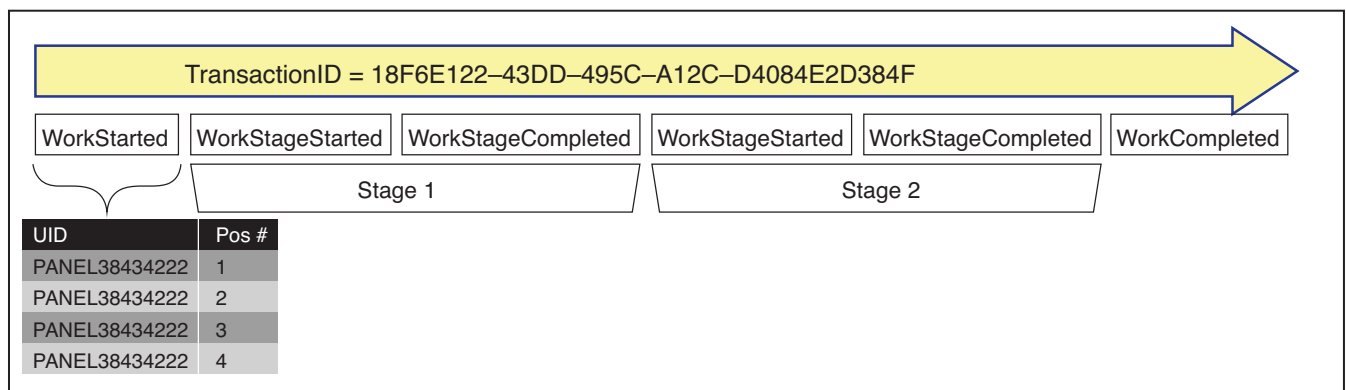


Figure 4-2 The CFX TransactionID

4.5 CFX Message Envelope All CFX messages are transmitted together with a formatted envelope that is in addition to the message itself. The envelope contains information regarding the message, including the sender and intended receiver of the message.

Table 4-3 describes the information contained within a CFX envelope.

Table 4-3 CFX Message Envelope

Field	Type	Description
MessageName	String	The full name of the CFX message contained within the transmission (i.e., CFX.Production.WorkStarted).
Version	String	The version of the CFX message contained within the payload of this transmission. This is the version number (revision) of this published standard.
Timestamp	Date-Time	The date and time when the event associated with this CFX message took place, rather than the time the transmission is sent.
UniqueID	GUID	A globally unique ID used to identify the message. This ID is used to ensure messages are not processed more than once. It is also used in cases where the receiver needs to identify the specific message that is being replied to.
Source	String	The CFX Handle of the creator/publisher of this message.
Target	String	For request/response type messages, this is the CFX Handle of the endpoint to which the message is intended to be received.
RequestID	GUID	For request/response type messages, this is a globally unique ID which identifies a specific request/response pairing. Each request/response pairing will have a unique and matching RequestID.
Attachments	Array	When message attachments are present, they shall be described in this attachments array. If no attachments are present the array may either be empty or the field may be omitted entirely.

4.6 Operator Information Many CFX messages and data structures contain operator information. Depending on local legislation and customer company policy, restrictions may be placed on the level of detail of personal information that may be captured and/or exchanged using CFX messaging. All endpoints are responsible for enabling through configuration the level of personal details such that any legal or policy requirements can be met.

4.7 CFX Endpoint Configuration There are several elements that require configuration for any CFX endpoint. Ideally, an intuitive user interface should be provided at the endpoint, which allows the user of the endpoint to enter and modify CFX configuration data. Configuration **shall** include:

- 1) The CFX handle of the endpoint; example: Acme.Rabbit.0001
- 2) FriendlyName: A short name for easy reference to the endpoint; example: SMD Printer 5
- 3) RequestNetworkURI: The network address/URI to be used for requests to this endpoint; example: amqp://host55:5688/
- 4) RequestTargetAddress: The AMQP 1.0 target address to be used for requests to this endpoint; example: /queue/SN23123
- 5) UniqueIdentifier: The barcode, radio-frequency identification (RFID) or other unique identifier that is used to identify this machine/endpoint; example: Line1-4.Acme.Rabbit
- 6) Restriction of operator personal information in line with local legislative and customer policy.
- 7) For each event channel to which the endpoint will publish events and broadcast requests, the configuration **shall** include:
 - a. The network address (URI) of the target broker or system. For secure, encrypted communications over AMQP 1.0, the URI will typically use the amqps:// protocol prefix, and TCP port 5671 (instead of the amqp:// prefix, and TCP port 5672, which is used for unencrypted communication).
 - b. The AMQP 1.0 target address to which messages will be directed (typically identifies an exchange on the target broker, though other types of targets may also be used).
 - c. If authentication is required to communicate with the target system, the end-user needs to be able to configure the username and password of an account authorized on the target system to publish messages.

Note: When using authentication, encrypted AMQPS (TLS/SSL Encrypted AMQP) should be utilized as well. Otherwise, passwords may be transmitted across the network in plain text format.

- d. The CFX topics that **shall** be published to this channel (end user may enable/disable each supported topic selectively). Additionally, for all enabled topics, end users may also enable/disable the publication of specific CFX messages within those topics.
 - e. The selection of whether Level 4 exact traceability data (as defined by IPC-1782) is sent by the machine. Where capable, this selection can result in a far greater amount of data to be created and sent by the machines, the overhead of which may not be required in all cases.
- 8) For each subscription channel from which the endpoint will receive broadcast responses (and perhaps events published by other endpoints), the configuration **shall** include:
- a. The network address (URI) of the source broker or system. For secure, encrypted communications over AMQP 1.0, the URI will typically use the amqps:// protocol prefix, and TCP port 5671 (instead of the amqp:// prefix, and TCP port 5672, which is used for unencrypted communication).
 - b. The AMQP 1.0 source address from which messages are to be received (typically identifies a queue on the source broker, though other types of sources may also be used).
 - c. If authentication is required to communicate with the target system, the end user needs to be able to configure the username and password of an account authorized on the target system to publish messages.

Note: When using authentication, encrypted AMQPS (TLS/SSL Encrypted AMQP) should be utilized as well. Otherwise, passwords may be transmitted across the network in plain text format.

4.7.1 Specific CFX Endpoint Configuration Addresses CFX also includes direct send (request) and response messages that can be targeted specifically at another CFX endpoint without the use of the broker. An example of this case is where a closed-loop communication is set up between specific machines on the line, where data are exchanged between machines directly. Perhaps an endpoint performing solder paste inspection (SPI) wants to direct a command to the upstream stencil printer to perform a stencil clean operation. To accommodate this, the SPI machine would need to allow end-users to configure the Request URI (network address) and Request AMQP Target Address of the appropriate stencil printer to which it should direct the clean stencil request/command.

Another example would be the case in which an endpoint wishes to validate production units with an upper-level factory control system before commencing work on a particular unit. The upper-level system may determine that this unit has skipped a critical step in the process and should not be processed at the endpoint at this time. In this case, the endpoint would need to allow end-users to configure the Request URI (network address) and Request AMQP Target Address of the upper-level factory control system to which validation requests should be made.

If an endpoint is capable of making different types of requests, it may be necessary to configure multiple targets for each type of request. For example, stencil clean requests go to one target (the stencil printer), while unit validation requests go to a different target (the upper-level control system).

For each endpoint to which request/response messages will be sent, the configuration **shall** include the following information:

- 1) The CFX handle of the target endpoint to which requests **shall** be directed.
- 2) The network address (URI) of the target endpoint. For secure, encrypted communications over AMQP 1.0, the URI will typically use the amqps:// protocol prefix, and TCP port 5671 (instead of the amqp:// prefix, and TCP port 5672, which is used for unencrypted communication).
- 3) The AMQP 1.0 target address to which requests will be directed.
- 4) If authentication is required to communicate with the target system, the end-user needs to be able to configure the username and password of an account authorized on the target system to publish messages.

Note: When using authentication, encrypted AMQPS (TLS/SSL Encrypted AMQP) should be utilized as well. Otherwise, passwords may be transmitted across the network in plain text format.

4.8 Message Attachments Starting with IPC-2591, Version 1.5, messages are allowed to have attachments. An attachment is a data file containing additional information relevant to the CFX message to which it is attached but not transmitted inside of the JSON message itself. The primary intended purpose of message attachments is to permit the transfer of large amounts of tabular data along with CFX messages without requiring that the data be converted to JSON format. In keeping with the CFX concept of fully standardizing all messages, message attachments **shall not** contain arbitrary or vendor-defined content. Instead, each attachment **shall** conform to a defined schema.

4.8.1 Specifying Message Attachments When files are attached to a CFX message, the Attachments property of the CFX envelope **shall** be present and contain a nonempty array, in which each entry in the array describes one attachment. The format of these entries **shall** be as defined in Table 4-4.

Table 4-4 Format of Attachments Array Entries

Field	Type	Description
MimeType	String	The mime type of the attachment file. Only supported mime types as specified in this standard may be attached.
AttachmentType	String	The standardized attachment type that is being sent as defined in this standard and supporting materials. Only supported attachment types shall be sent. The attachment type is equivalent to a dynamic structure in the CFX format in that it describes the contents of the attachment in a fully standardized way.
TransferMode	String	The literal string "InBand" or "OutOfBand". InBand signifies that the attachments directly follow the main CFX message envelope in a multipart/mixed mime format as described in 4.8.2. OutOfBand signifies that the attachment is not physically transferred with the CFX message and may be downloaded from the location specified in AttachmentURL.
AttachmentURL	String	This shall be null unless TransferMode equals "OutOfBand", in which case it shall be a URL that can be used to download the attachment file.

As described in Table 4-4, attachments may be sent either in-band or out-of-band. Implementations should use in-band transfer unless the attachment file size is so large that in-band transfer is not feasible.

4.8.2 In-Band Message Attachments In-band message attachments are sent in a manner similar to attachments in electronic mail (email). They are included by encoding the overall CFX message in *multipart/mixed* mime format as described in RFC-1341 with the first part of the multipart message being the main CFX message body in JSON format and any attachments following as subsequent parts of the multipart message. The following requirements apply:

- 1) Unlike RFC-1341, the mime type of an attachment **shall not** itself be a multipart format (i.e., nested multipart messages are not allowed).
- 2) Every part of the multipart message **shall** have a content-type header set with the same string value as the corresponding MimeType property in the Attachments array.
- 3) All attachment parts of the multipart message **shall** additionally have an x-cfx-attachment-type header set to the same string value as the corresponding AttachmentType property in the Attachments array.
- 4) The order of in-band attachments **shall** match the order of entries in the Attachments array.
- 5) If out-of-band attachments are present they **shall** occur after any in-band entries in the Attachments array.
- 6) The first part of the multipart message **shall** be the CFX message envelope and **shall** have its content-type header set to: application/json; charset="utf-8".
- 7) The content-type header of the AMQP message itself **shall** be multipart/mixed with a boundary as described in Table 4-4.

Listing 4-1 provides an example of a multipart CFX message with two in-band attachments. The first attachment contains component offset measurements encoded in CSV format, and the second attachment contains solder paste measurements encoded in Apache Parquet format.