

A Tool for Testing Compliance with CIP4's Interoperability Conformance Specifications

Claes Buckwalter
ITN, Linköping University
Norrköping, Sweden

Abstract

The *Job Definition Format (JDF)* is a relatively young industry standard developed by *The International Cooperation for the Integration of Processes in Prepress, Press and Postpress (CIP4)*. JDF specifies a data format for describing print jobs and exchanging production information between systems in the print production workflow. Most vendors in the printing industry today offer "JDF-enabled" products, but the industry lacks a well-defined test protocol and specialized tools for testing and verifying the level of JDF-compliance that products have.

This paper presents an open source test tool for testing the level of JDF compliance as specified by the CIP4 organization's *Interoperability Conformance Specification (ICS)* documents. The test tool is semi-automated and validates system behavior and output of JDF instances and JMF messages both syntactically and based on context.

Introduction

Although most vendors in the printing industry offer some level of JDF-support in their products there is unfortunately no guarantee that JDF-enabled products from one vendor will be interoperable with those from another vendor. To alleviate this problem the CIP4 organization has developed the *Interoperability Conformance Specification (ICS)* documents. Each ICS document defines a subset of the JDF specification¹ that is required by a certain class of product. The Base ICS² is the least common denominator that all JDF-enabled products must comply with. On top of the Base ICS there are at the time of writing six other ICS documents³ for different product classes.

Testing JDF Compliance

Testing and verifying the behavior of a JDF compliant product is a complex task. For example, testing that a printing press interprets a JDF job ticket and produces the correct output would involve analyzing the printed sheets and in some way attempting to verify that they match what the JDF job ticket specified. Obviously, there are many factors that have nothing to do with a product's ability to interpret JDF data that can influence the physical output produced by the product. Functional testing of this kind is beyond the scope of the test tool

presented in this paper. Instead, the test tool presented here is concerned only with testing a product's ability to consume and produce JDF instances and JMF messages as specified by the ICS's that the product claims to comply with.

The Test Tool

The test tool implements the role of a *manager*; a piece of software that sends JDF instances and JMF messages to a *worker*. The worker role is implemented by a piece of software that processes received JDF instances and JMF messages and after processing may send information back to the manager. The worker is the software under testing by the test tool presented here. An example of a product with the manager role would be a piece of prepress controller software that sends jobs to a RIP, the RIP being a worker.

Test Procedure

The test tool sends JDF/JMF data to a worker and tests that the worker responds correctly by producing valid output. The following tests are performed on JDF instances and JMF messages produced by the worker:

1. The syntax of the XML structure is validated against the JDF XML Schema⁴
2. The presence and correctness of all required XML elements and attributes specified by the relevant ICS documents is checked
3. The XML data is examined and it is checked that it is correct according to the current context

Validation against the JDF XML Schema is straightforward and performed using CIP4's JDF library JDFLib-J.⁵ The presence of the XML elements and attributes required by the ICS documents are tested by test cases configured with XPath expressions. One test case with a set of XPath expressions is available for each ICS document. Verifying that JDF instances and JMF messages that the worker produces are correct according to the current context is performed by maintaining a record of all JDF and JMF data previously consumed and produced by the worker. Any new data produced by the worker is compared with the previously recorded data and validated using specifically built test cases.

All test results are logged in an XML file. It is left to other tools beyond the scope of this paper to analyze and visualize the information in the XML log file. For example, the XML log file could be transformed using

XSLT to an XHTML document for viewing in a web browser.

Architecture

The test tool is implemented as a Java application that consists of three main modules, as shown in Figure 1. The client module sends JDF/JMF data to the worker under test; the server module receives JDF/JMF data from the worker; and the test module coordinates the test data sent to the worker and validates all data produced by the worker. The outgoing test data used to probe the worker consists of a set of manually written JDF and JMF files; all validated using CIP4's CheckJDF tool.⁵ These files can easily be modified or replaced if necessary. The test cases are dynamically loaded at startup of the test tool and are configured using configuration files. New ICS support can be added by creating a new test case.

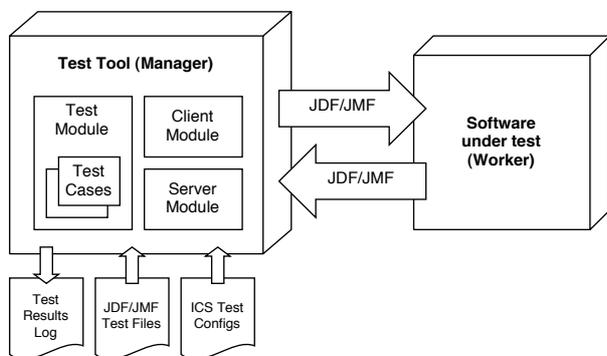


Figure 1. Architecture

Future Work

The test tool has so far been used for testing CIP4's internal software tools but more extensive testing against third-party products must be conducted in order to thoroughly evaluate the test tool. Currently, there is only a

test case for the Base ICS. Additional test cases for other ICS's must be implemented for testing more than the minimal level of JDF functionality. Nonetheless, the test tool is still useful since many vendors still lack full support for even the Base ICS.

The complete source code and up to date information regarding the development of the test tool presented in this paper is available at Ref. [6].

References

1. JDF Specification Version 1.2 [Online]. Available: http://www.cip4.org/documents/jdf_specifications/JDF1.2.pdf [2005, February 4]
2. Base Interoperability Conformance Specification (Base ICS) [Online]. Available: http://www.cip4.org/document_archive/documents/ICS-Base-1.0.pdf [2005, February 4]
3. ICS Registry [Online]. Available: http://www.cip4.org/document_archive/ics.php [2005, February 4]
4. XML Schema for JDF Version 1.2 [Online]. Available: http://www.cip4.org/Schema/JDFSchema_1_2/JDF.xsd [2005, February 4]
5. JDF Open Source [Online]. Available: http://www.cip4.org/open_source/index.html [2005, February 4]
6. JDF ICS Test Tool [Online]. Available: <http://www.itn.liu.se/~clabu/dpp05-51> [2005, February 4]

Biography

Claes Buckwalter has a Master's degree in Media Technology and Engineering from Linköping University, Sweden, where he is currently a Ph.D. student. His Ph.D. studies focus on process automation and information interchange between processes in the print production workflow, mainly within prepress. Before committing to an academic career, Claes co-founded Swedish software company Frog Software that develops Internet-based print job delivery solutions.