

SEMI T13.2-1104 (Reapproved 0710)

SPECIFICATION FOR XML PROTOCOL FOR DEVICE TRACKING

This standard was technically approved by the global Traceability Committee. This edition was approved for publication by the global Audits & Reviews Subcommittee on April 30, 2010. Initially available at www.semi.org in June 2010. Originally published November 2004.

1 Purpose

1.1 The purpose of this document is to map Services and data in SEMI T13 into XML protocol.

2 Scope

2.1 This document is applied to XML implementation of Jigs and Implements traceability with some interoperability with Surface Mount Technology (SMT) equipment.

NOTICE: This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory or other limitations prior to use.

3 Limitations

3.1 This protocol mapping has a consideration being interoperable with SMT manufacturing it may be used in such area, using this specification on such manufacturing area is not guaranteed because this specification is originally planned to be working on semiconductor manufacturing systems including equipment.

4 Referenced Standards and Documents

4.1 SEMI Standards

SEMI T12 — Specification for Tracing Jigs and Implements

SEMI T13 — Specification of Device Tracking: Concepts, Behavior and Services

4.2 Association Connecting Electronics Industries (IPC)¹

IPC-2501 — Definition for Web-Based Exchange of XML data; (July, 2003)

IPC-2541 — Generic Requirements for Electronics Manufacturing Shop-Floor Equipment Communication (CAMX); (October, 2001/ANSI Approved November, 2001)

4.3 Internet Engineering Task Force²

RFC2045 — Multipurpose Internet Message Extensions (MIME) Part 1: Format of Internet Message Bodies

RFC2616 — Hyper Text Transfer Protocol – HTTP/1.1

4.4 World Wide Web Consortium (W3C)³

REC-xml-20001006 — eXtensible Markup Language (XML) 1.0 (Second Edition)

NOTE-SOAP-20000508 — Simple Object Access Protocol (SOAP) 1.1

NOTICE: Unless otherwise indicated, all documents cited shall be the latest published versions.

¹ <http://www.ipc.org>; The original developer is IPC with NEMI cooperation, ANSI ends up with authorizing approval.

² IETF Secretariat c/o Corporation for National Research Initiatives, 1895 Preston White Drive, Suite 100, Reston, VA 20191-5434, USA. Telephone: 703.620.8990; Fax: 703.620.9071; <http://www.ietf.org>

³ World Wide Web Consortium, 32 Vassar St. Room 32-G515, Cambridge, MA 02139, USA. Telephone: 617.253.2613; Fax: 617.258.5999; <http://www.w3c.org>

5 Terminology

5.1 Abbreviations and Acronyms

5.1.1 *MIME* — Multipurpose Internet Message Extensions

5.1.2 *SOAP* — Simple Object Access Protocol

5.1.3 *XML* — eXtensible Markup Language

6 Base Requirements

6.1 *Base Protocol Mapping* — This XML protocol specification adopts Simple Object Access Protocol (SOAP) over Hyper Text Transfer Protocol (HTTP) on Transmission Control Protocol/Internet Protocol (TCP/IP). Some equipment users produce not only semiconductor devices but also Print Circuit Boards (PCBs). It is convenient for such users to have common messaging protocol with standard messaging for Surface Mount Technology (SMT). For this purpose this protocol mapping extension standard specifies the messaging mechanism conformant to IPC-2501 “Definition for Web-Based Exchange of XML Data” as one of mapping means. Also this extension standard document follows IPC-2541 “Generic Requirements for Electronics Manufacturing Shop-Floor Equipment Communication Messages” and IPC-2551 “Sectional Requirements for Manufacturing Execution System” for that reason. The following subsections describe a part of these specifications for introduction. Refer to these documents for detail reference.

6.2 *SOAP Structure* — A HTTP message in this specification has an envelope to contain a couple of MIME blocks. One is for SOAP Envelope and the other is for message detail to be able to mix with such non-XML information as binary data. The service message specified in SEMI T12 is contained in latter one. SOAP standard envelope MIME Block contains SOAP Header which has message information and SOAP Body which contains SOAP Faults. Outlined diagram is shown in Related Information. Official definitions are given in referenced documents.

6.3 *Messaging Mechanism* — Message Broker, that is logical middle ware server to handle and relay messages, is assumed somewhere on communication network. While the broker behaves as communication server, the other logical nodes are message clients. Every message starts with a client and responded by the server. Messages communicated between clients are exchanged through the server. The server keeps posted messages from clients in a queue for each individual client. Expecting recipient client asks the server for messages and a topmost message in the queue for the client is responded if the queue is not empty. The recipient client posts acknowledge back to the server. When the recipient replies back a message to the original sender client, same thing happens with switched roles between sender and recipient clients. Explanatory diagrams for this outline are shown in Related Information. Official specification is given in referenced documents.

6.4 *Issues Specific to this Messaging* — A few issues for IPC based communication protocol for Jigs and Implements Tracking are introduced here.

6.4.1 *Domain Configuration* — Message Broker has domain configuration information to make sure all acting entities, possible messages with related publisher, recipient and subscriber clients. Also it has other detail characteristics of the domain of communication. Every client needs to look up the information before the first messaging.

6.4.2 *Message Information* — This is an element contained in SOAP Header. This information provides time of transaction, sender, destination, message identification and schema type used for service message detail in the second MIME block.

6.4.3 *Attributes in Service Message Element* — The top element of each service message has date/time of the message, session ID or session reference, request ID or request reference, and optional values. Some of the optional values are extended for Jigs and Implements Tracking communication: linkId, linkCnt and linkExp. The extension may be used for delayed reply message and multiple divided replies to keep conformance of related messages.

7 Requirements

7.1 *Service Mapping* — Services defined in SEMI T12 are mapped on the following table.

7.1.1 *Capitalized Service Message Element Name Base* — Each service message is mapped into an XML element with capitalized heading letter to prevent collision with XML or XML Schema native words.

7.1.2 *Service Message Element Name Suffix* — Because of symmetry of message broker communication and for XML element naming convention, suffix of ‘Request’ or ‘Response’ is added to each service message element name for request or response respectively; e.g., AddEventRequest to request service and AddEventResponse for its response message.

Table 1 Service Instruction Mapping

<i>Service Message Name</i>	<i>Supporting Class</i>	<i>Service Message Element Name Base</i>	<i>Remarks</i>
acquireDTD	DieTraceMachine	AcquireDTD	
admit	DieTraceManager	Admit	
catchDTD	DieTracer	CatchDTD	
collectDTD	DieTracer	CollectDTD	
devolve	DieTraceManager	Devolve	
enroll	DieTracer	Enroll	
flush	DieTraceMachine	Flush	
logException	DieTraceManager	LogException	
planDTD	DieTracer	PlanDTD	
registerTracer	DieTraceMachine	RegisterTracer	
resign	DieTraceManager	Resign	
setAlert	DieTraceClient	SetAlert	
setForward	DieTracer	SetForward	
setPlanningItems	DieTraceMachine	SetPlanningItems	
showData	DieTraceData	ShowData	
showDaughter	CircuitModule	ShowDaughter	
showDTD	DieTracer	ShowDTD	
showPhenomena	DieTraceClient	ShowPhenomena	
showProperty	CircuitModule, DieTraceData	ShowProperty	
takeDTD	DieTracer	TakeDTD	

7.2 *Service Parameter Mapping* — Service parameter type mapping is shown in the following table.

Table 2 Service Parameter Type Mapping

<i>Parameter Name</i>	<i>Attribute/Element</i>	<i>Type</i>	<i>Remarks</i>
Alert	Element	anyType	Formatted Text
Clarifier	Element	string	
DataName	Element	string	
DataNameList	Element	string	Repeating DataName
DataValue	Element	anyType	Formatted Text
Designation	Element	string	
Exceptions	Element	anyType	Formatted Text
MachineType	Element	string	
ManagingTracer	Element	string	
Phenomena	Element	anyType	Formatted Text
Property	Element	anyType	Formatted Text
ServiceStatus	Element	complexType	status+Rejection

Parameter Name	Attribute/Element	Type	Remarks
Target	Element	string	
TraceData	Element	anyType	Formatted Text
Tracer	Element	string	
Tracers	Element	string	Repeating Tracer

7.3 Additional Parameters for Service Messages — Because this protocol mapping works for asynchronous HTTP server-client communication on message broker, such additional parameters to give complementary information as issued time and transaction identification are required to be attached to each message. The information may be dependent to the role of the message. Type mapping of them are shown in the following table.

Table 3 Type Mapping of Additional Parameters

Parameter Name	Attribute/Element	Type	Remarks
dateTime	Attribute	dateTime	
sessionId	Attribute	string	
sessionRef	Attribute	string	
requestId	Attribute	string	
requestRef	Attribute	string	
linkId	Attribute	string	extended for delayed response
linkCnt	Attribute	integer	extended for delayed response
linkExp	Attribute	duration	extended for delayed response

7.4 Example XML Message Fragments — The following subsections give examples of some fragments of service messages in XML. A full message is given with envelopes and headers.

7.4.1 showProperty Request — The following XML description is an example of the principal part of service request message for showProperty provided by CircuitModule objects. Because the Designation of parameters for a CircuitModule is dependent to the circuit module and tracing objectives, they should be discussed between trace data provider and consumer.

```
<ShowPropertyRequest dateTime="2004-03-29T13:05:25.000-08:00" SessionRef="SemiDtQs-9876543210" RequestId="123456">
  <Designation>Id</Designation>
  <Designation>Type</Designation>
  <Designation>ProductName</Designation>
  <Designation>Lot</Designation>
</History>
</ShowPropertyRequest>
```

7.4.2 showPhenomena Response — The following XML description is an example of the principal part of service response message for showPhenomena provided by DieTraceClient objects. Detail information of argument “Phenomena” is not specified in the standard because it’s dependent to target product. Additional definition should be exchanged in the system.

```
<ShowPhenomenaResponse dateTime="2004-03-29T09:07:59.000-08:00" SessionRef="SemiDtQs-9876543123" RequestRef="123443" status="GRANTED">
  <Phenomena>
    <Registered>2004-03-26T18:07:51.000-08:00</Registered>
    <Product>meson689-4A</Product>
    <Problem>Noise sensitive</Problem>
    <Problem>Restarted Occasionally</Problem>
    <Application>Office Computers</Application>
  </Phenomena>
</ShowPhenomenaResponse>
```

```
</Phenomena>  
<ServiceStatus>  
  <SvcAck>Successful</SvcAck>  
</ServiceStatus>  
</ShowPhenomenaResponse>
```

INACTIVE

APPENDIX 1

NOTICE: The material in this appendix is an official part of SEMI T13.2 and was approved by full letter ballot procedures on July 23, 2004.

A1-1 XML Message Schema

A1-1.1 *XML Schema* — XML Schema is a kind of template for documents. The following XML Schemas shows how messages in this XML protocol are represented.

A1-1.1.1 *Message Schema* — Each message in T12 has one XML Schema. Parsing with the schema, application can understand what elements and attributes mean. Also it can check syntax easily.

A1-1.1.2 *Common Definitions* — Common part of the schema is separated and it is included in each message schema.

A1-1.2 *Enumeration* — Attachment specific or equipment specific enumeration may not be addressed here. Corresponding part of the schema should be extended or restricted with 'redefine' element, for example, in application schema.

XML Message Schema Name: T13ShowDaughterRequestCmo.xsd for CircuitModule Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
  targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
  elementFormDefault="qualified" attributeFormDefault="qualified">

  <xsd:include
    schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

  <!-- - - - - - Show Daughter Request - - - - - -->
  <xsd:element name="ShowDaughterRequest" type="ShowDaughterRequestType"/>

  <xsd:complexType name="ShowDaughterRequestType">
    <xsd:attributeGroup ref="RegRequestAttr"/>
  </xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowDaughterResponseCmo.xsd for Circuit Module Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
  targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
  elementFormDefault="qualified" attributeFormDefault="qualified">

  <xsd:include
    schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

  <!-- - - - - - Show Daughter Response - - - - - -->
  <xsd:element name="ShowDaughterResponse" type="ShowDaughterResponseType"/>

  <xsd:complexType name="ShowDaughterResponseType">
    <xsd:sequence>
      <xsd:element name="Clarifier" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="RegResponseAttr"/>
  </xsd:complexType>
```

```
</xsd:schema>
```

XML Message Schema Name: T13ShowPropertyRequestCmo.xsd for CircuitModule Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
  targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
  elementFormDefault="qualified" attributeFormDefault="qualified">

  <xsd:include
    schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

  <!-- - - - - - Show Property Request - - - - - -->
  <xsd:element name="ShowPropertyRequest" type="ShowPropertyRequestType"/>

  <xsd:complexType name="ShowPropertyRequestType">
    <xsd:sequence>
      <xsd:element name="Designation" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="RegRequestAttr"/>
  </xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowPropertyResponseCmo.xsd for CircuitModule Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
  targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
  elementFormDefault="qualified" attributeFormDefault="qualified">

  <xsd:include
    schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

  <!-- - - - - - Show Property Response - - - - - -->
  <xsd:element name="ShowPropertyResponse" type="ShowPropertyResponseType"/>

  <xsd:complexType name="ShowPropertyResponseType">
    <xsd:sequence>
      <xsd:element name="Property" type="xsd:anyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="RegResponseAttr"/>
  </xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowPhenomenaRequestDtc.xsd for DieTraceClient Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
  targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
  elementFormDefault="qualified" attributeFormDefault="qualified">

  <xsd:include
    schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>
```

```
<!-- - - - - - Show Phenomena Request - - - - - -->
<xsd:element name="ShowPhenomenaRequest" type="ShowPhenomenaRequestType"/>

<xsd:complexType name="ShowPhenomenaRequestType">
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowPhenomenaResponseDtc.xsd for DieTraceClient Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
  targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
  elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
  schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - Show Phenomena Response - - - - - -->
<xsd:element name="ShowPhenomenaResponse" type="ShowPhenomenaResponseType"/>

<xsd:complexType name="ShowPhenomenaResponseType">
  <xsd:sequence>
    <xsd:element name="Phenomena" type="xsd:anyType" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13SetAlertRequestDtc.xsd for DieTraceClient Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
  targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
  elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
  schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - Set Alert Request - - - - - -->
<xsd:element name="SetAlertRequest" type="SetAlertRequestType"/>

<xsd:complexType name="SetAlertRequestType">
  <xsd:sequence>
    <xsd:element name="Alert" type="xsd:anyType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13SetAlertResponseDtc.xsd for DieTraceClient Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
  targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
```