Interface Description

SAP AII-DC 1.0

Used by SAP Auto-ID Infrastructure 2.1

Document Version 0.95 – December 14, 2004

SAP AG
Neurottstraße 16
69190 Walldorf
Germany
T +49/18 05/34 34 24
F +49/18 05/34 34 20
www.sap.com

Contents

1 About this Document

In this document we describe the interface between *SAP Auto-ID Infrastructure (SAP AII)* and an RFID device controller. This interface consists of an *XML command message* sent from SAP AII to the device controller and an *XML notification message* sent from the device controller to SAP AII.

We include sample messages that support predefined SAP AII message processing rules. These sample messages are borrowed from an RFID enabled outbound delivery scenario.

The message interface we describe is Version 1.0 of the SAP AII-DC Interface (SAP AII-DC 1.0). SAP AII-DC 1.0 is implemented by version 2.1 of SAP AII.

The purpose of this document is to enable SAP partners to deliver RFID device controllers that are interoperable with SAP AII 2.1

2 Overview

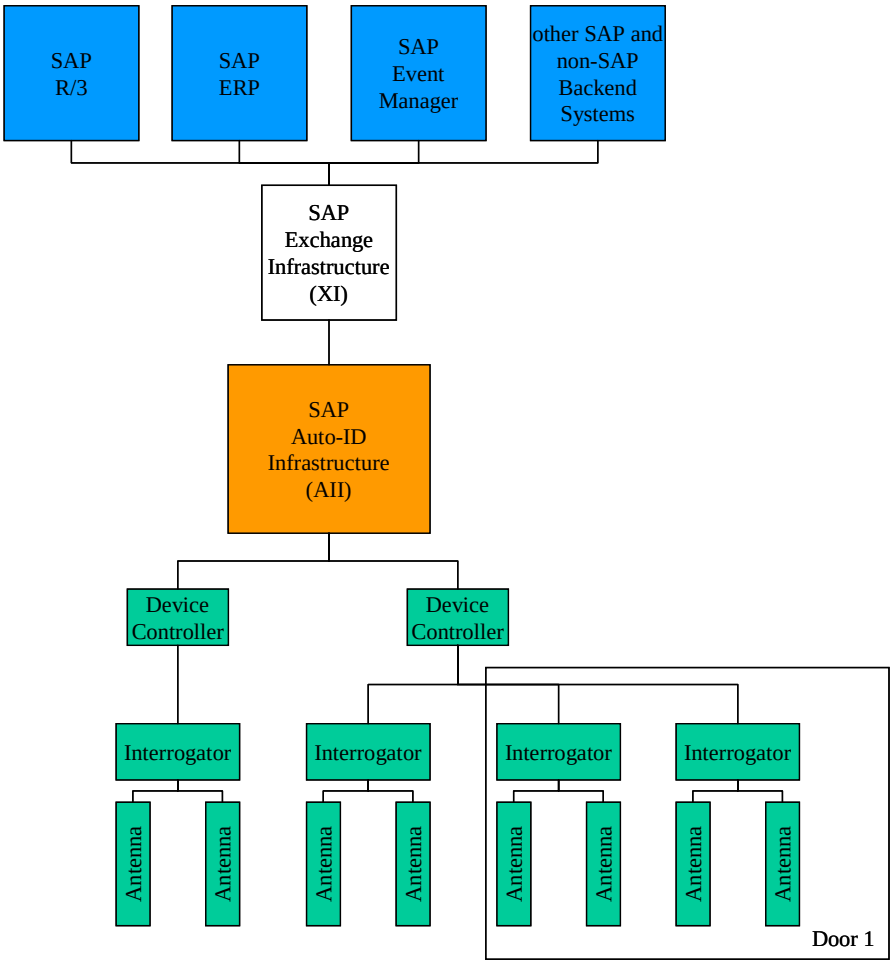
2.1 SAP Auto-ID Infrastructure

Radio frequency identification (RFID) is becoming an important method for automatic identification (Auto-ID) of goods, especially for supply chain visibility. Goods holding a little chip with an antenna (RFID tag) are detected at several points along the supply chain. In contrast to barcode strategies that typically identify only the material (for example, the EAN), most RFID applications focus on the identification of each object (pallet, case, or individual item), giving you much more detailed information than barcodes alone.

SAP Auto-ID Infrastructure (SAP AII) enables backend systems like mySAP ERP and supply chain visibility tools like SAP Event Manager to leverage RFID technology. SAP AII connects the physical world, as observed by RFID device controllers, with the business-oriented document view of an ERP system. Thus, the RFID detection of pallets and cases at a gate in the warehouse can confirm that an expected inbound delivery is complete.

2.2 Architecture of an RFID Enabled Landscape

This is an example of an RFID enabled landscape:



Device controllers are software components that link the SAP Auto-ID Infrastructure with interrogator hardware components. A device controller may run on the hardware of the interrogator or on different hardware. One device controller may manage more than one interrogator.

An interrogator is a hardware device that reads RFID tags. An interrogator may have one or more antennas. These antennas and additional equipment, like light sensors, can be used to determine direction of movement.

The components below the SAP Auto-ID Infrastructure shown in the *Example RFID Enabled Landscape* figure are provided by SAP partners. In SAP AII, these components are represented by groups of RFID devices managed by RFID device controllers. SAP AII RFID devices can represent interrogators, specific antennas, RFID printers, or non-RFID sensors (e.g., light sensors and barcode readers).

The frame in the lower right corner indicates the instrumentation in a single warehouse location. For example, this may be a door or gate where goods are brought out of the warehouse and loaded on a truck. For better reading accuracy and direction sensing, the device controller uses two interrogators to sense the goods issue. In this case, a single RFID device in SAP AII may be used to model these two interrogators.

2.3 Message Processing in the SAP Auto-ID Infrastructure

A notification message received by SAP AII is processed by a Rule. SAP AII is configured to select the appropriate rule based on information derived from the incoming message.

A rule is a sequence of Activities. Activities contain the detailed application logic that can include sending a command message to a device controller. SAP AII 2.1 includes many predefined rules and activities for a variety of scenarios.

Additional rules can be configured from the predefined activities. Custom activities can be developed.

2.4 Message Element Notation

The message elements (fields) and their parent – child relationships are described by figures and tables. Additional notes are provided where necessary for clarification.

The rectangles represent an element and show its name. A plus sign at the right edge of the element's symbol indicates that it includes further elements, which are shown in a separate figure. A minus symbol indicates that the child elements are included in the same figure.

The choice symbol () indicates, that the element on the left of it can only have one of the elements on the right as a child element.

The sequence symbol () indicates, that the element on the left has all the elements on the right as children and the order of the elements on the right must be maintained.

If one element can be a member of another element more than once, that is noted near the element. If the element can be omitted, the border of the rectangle is dotted.

3 Message Descriptions

The SAP AII-DC 1.0 describes two types of messages:

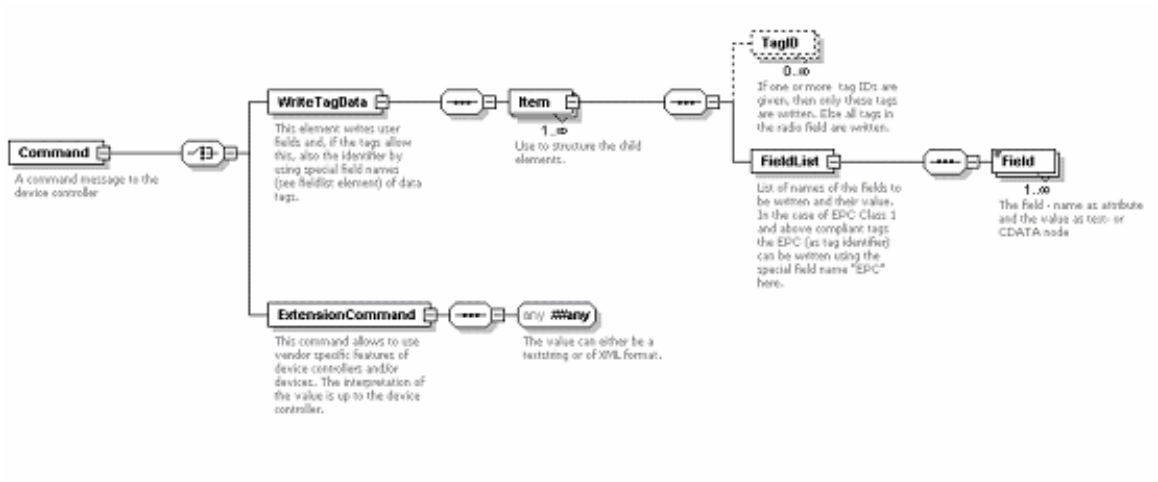
- Command messages sent to the Device Controller to control tag writing and light towers
- Notification messages received from the Device Controller to report tag observations and other events

A technical description of the XML structure of each message type is followed by sample messages that are used with predefined SAP AII activities.

3.1 Command Message

3.1.1 Elements of the Command Message

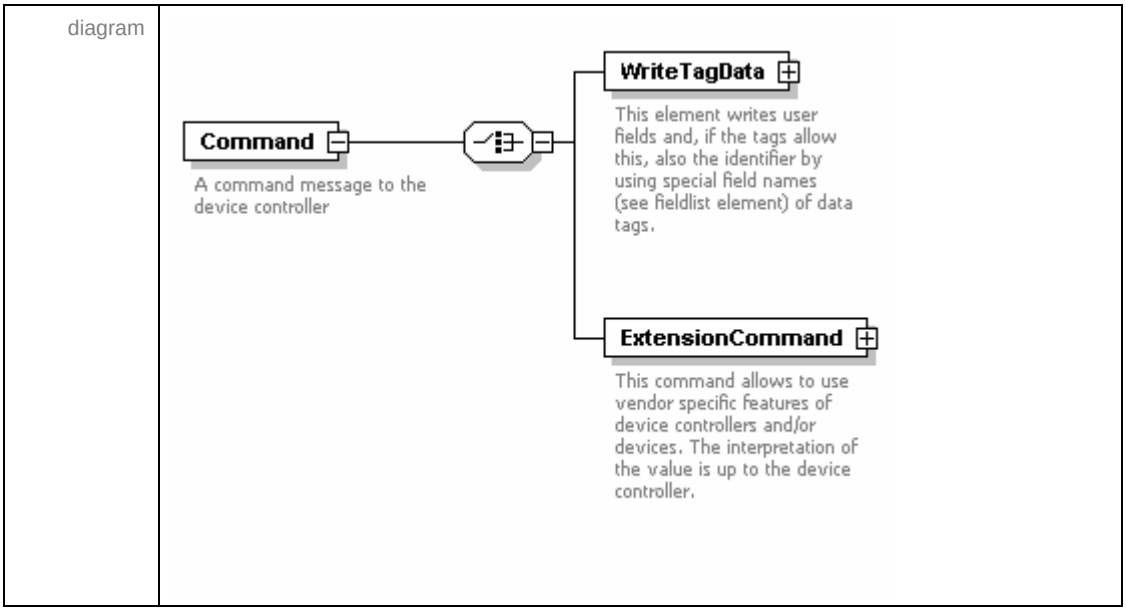
The following figure shows an overview of the command message element structure. The XML schema for the Command Message is provided in the appendix.



3.1.1.1 Details of the Command Element

Predefined SAP AII activities send two types of command messages, WriteTagData and ExtensionCommand. The optional id attribute of the Command element is not used.

element **Command**

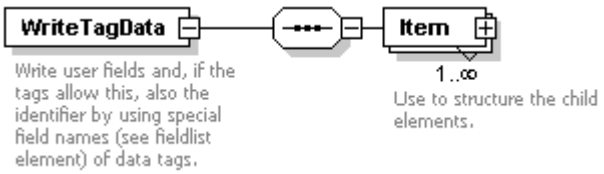


properties	content	complex		
children	**WriteTagData ExtensionCommand**			
attributes	Name	Type	Use	Annotation
	id	**xs:string**	optional	
annotation	documentation	A command message to the device controller		

3.1.1.2 Details of the WriteTagData Element

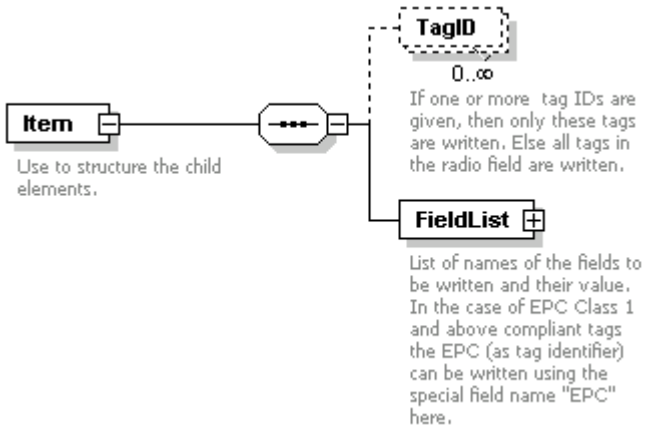
The readerID attribute of the WriteTagData element contains the RFID Device ID, as configured in SAP AII.

element **Command/WriteTagData**

diagram					
properties	isRef	0			
	content	complex			
children	**Item**				
attributes	Name	Type	Use	Annotation	
	readerID	**xs:string**	optional	documentation	Used if only a specific (physical) reader should be queried.
annotation	documentation	Write user fields and, if the tags allow this, also the identifier by using special field names (see fieldlist element) of data tags.			

3.1.1.3 Details of the WriteTagData/Item Element

element **Command/WriteTagData/Item**

diagram	
properties	isRef 0 content complex
children	TagID FieldList
annotation	documentation Use to structure the child elements.

3.1.1.4 Details of the WriteTagData/Item/TagID Element

The TagID element is not populated by the pre-delivered SAP AII 2.1 tag commissioning activity. The ID to be written to the tag is contained in the FieldList/Field name=EPC (see below). TagID would be relevant for re-writing a specific tag.

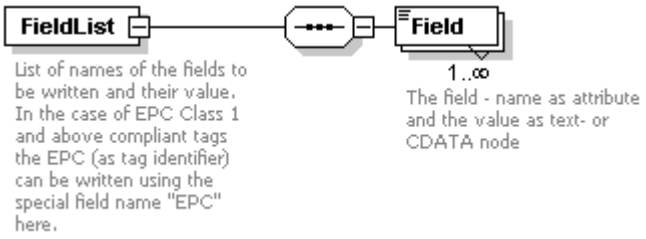
element **Command/WriteTagData/Item/TagID**

diagram	
properties	isRef 0
annotation	documentation If one or more tag IDs are given, then only these tags are written. Else all tags in the radio field are written.

3.1.1.5 Details of the Item/FieldList Element

For the FieldList element, SAP AII populates the jobName and quantity attributes only if the format attribute is populated. The quantity value is always 1. (If multiple tags are expected, separate item elements are provided). The jobName is constructed from the readerID and time.

element **Command/WriteTagData/Item/FieldList**

diagram	
properties	isRef 0 content complex
children	**Field**

attributes	Name	Type	Use	Annotation
	format	**xs:string**	optional	
	jobName	**xs:string**	optional	
	quantity	**xs:string**	optional	
	printerName	**xs:string**	optional	

annotation	
documentation	List of names of the fields to be written and their value. In the case of EPC Class 1 and above compliant tags the EPC (as tag identifier) can be written using the special field name "EPC" here.

3.1.1.6 Details of the Item/FieldList/Field Element

The device controller must support the field name=”EPC” in order to commission a tag.

Example of the Field element

```
<Field name="EPC">01002345678900FFABCD0001</Field>
```

In addition to a HEX representation of the binary EPC, the following fields can be included in the WriteTagData command with configuration.

EPC	Hexadecimal EPC
EPC_TYPE	EPC type (SSCC-96, SGTIN-96 …)
EPC_URN	EPC in uri format
EPC_URN_NO_HEADER	EPC uri format numeric portion

SSCC	Serialized Shipping Container Code
GTIN	Global Trade Identification Number
PRODUCT	Product associated with the GTIN in the EPC
PRODUCT_DESCRIPTION	Product Description
PRODUCT_QUANTITY	Quantity in the base unit of measure
BASE_UOM	Base unit of measure, i.e., PC
UOM	EPC unit of measure, i.e., CS
DOCUMENT_NO	Document ID
DOCUMENT_TYPE	Document Type Code
BUSINESS_PARTNER	Business Partner
BUSINESS_PARTNER_ADDRESS	Address of Business Partner

element **Command/WriteTagData/Item/FieldList/Field**

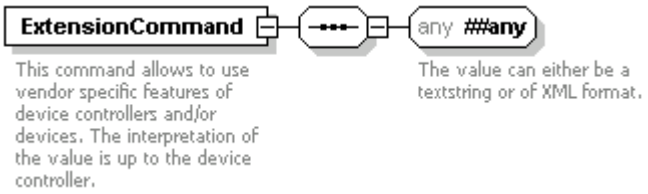
diagram				
properties	isRef 0 content complex mixed true			
attributes	Name	Type	Use	Annotation
	name	**xs:string**	required	documentation: Name of the data field on the tag
annotation	documentation: The field - name as attribute and the value as text- or CDATA node			

3.1.1.7 Details of the Command/ExtensionCommand Element

The ExtensionCommand element allows SAP AII to use special features delivered by the device controller. Currently, it is used to control a light tower.

The value of the ExtensionCommand element must be of type string or XML.

element **Command/ExtensionCommand**

diagram	
properties	isRef 0 content complex
annotation	documentation This command allows use of partner specific device specific features. The interpretation of the value is up to the device controller.

3.1.2 Samples of the WriteTagData Command

The WriteTagData command message is sent by SAP AII to instruct the device controller to write the ID on an RFID tag. In SAP AII 2.1, this is called tag commissioning. Optionally, WriteTagData supports RFID printers that can print information on a label as well as commission the embedded RFID tag. The predefined SAP AII Activity, DEVICE_PRINT_TAG, sends this WriteTagData command.

3.1.2.1 Sample 1: RFID Tag Commissioning

After sensing the presence of a product case, the Auto-ID Infrastructure sends a WriteTagData command to the device controller requesting that a specific EPC be written to the RFID tag embedded in the corrugate of the case by the Writer_Device.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="Command.xsd">
    <WriteTagData readerID="Writer_Device">
        <Item>
            <FieldList>
                <Field name="EPC">3074024220403B8000000008</Field>
            </FieldList>
        </Item>
    </WriteTagData>
</Command>
```

3.1.2.2 Sample 1: RFID Label Printing

After stretch wrapping the cases on a pallet, the Auto-ID Infrastructure sends a WriteTagData command to the device controller requesting that a pallet label be printed and that a specific EPC be

written to the RFID tag in the label. The command message includes additional information fields and field list attributes used by the printer (Writer_Device).

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="Command.xsd">
    <WriteTagData readerID="Writer_Device">
        <Item>
            <FieldList format="C:LABEL.PL"
                    jobName="Writer_Device20040929165746"
                    quantity="1">
                <Field name="EPC">3074024220403B8000000008</Field>
                <Field name="EPC_TYPE">SGTIN-96</Field>
                <Field name="EPC_URN">
                    urn:autoid:tag:sgtin-96:3.5.0037000.065774.8</Field>
                <Field name="PRODUCT">SGPROD</Field>
                <Field name="PRODUCT_DESCRIPTION">Test product</Field>
            </FieldList>
        </Item>
    </WriteTagData>
</Command>
```

The sequence of events leading to sending the WriteTagData command (step 5) is given below.

1. Auto-ID Infrastructure provides a master list of variable names, including “EPC”, which can be used during label design. (The values of these variables can be provided in the WriteTagData command sent by Auto-ID Infrastructure).

2. The label designer creates a named format with tools provided by the printer partner. The content of the format directs the printer to print a label that includes the value of several variables from the master list. The format also directs the printer to write the value of the “EPC” variable to the RFID tag.

3. An RFID printer is connected to the TCP/IP network and given a specific IP address. The printer listens for print commands on a TCP socket (or HTTP server) for a specific port. The printer must have access to the format created in step 2.

4. Auto-ID Infrastructure is configured with the address and port of the RFID printer, the name of the printing format and the associated list of variable names.

5. Auto-ID Infrastructure selects the printer and printer format and evaluates the associated data variables, including a unique EPC value. A WriteTagData command message is sent to the printer.

6. The RFID printer prints a label and writes the EPC to the RFID tag embedded in the label.

3.1.3 Sample of the ExtensionCommand

The predefined SAP AII Activity, DEVICE_SEND_LIGHT, used the ExtensionCommand message to tell the device controller to change the color of a light tower to RED or GREEN.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="Command.xsd">
        <ExtensionCommand>GREEN</ExtensionCommand>
</Command>
```

3.2 Notification Message

The notification messages sent by the device controller are processed in SAP AII as “fixed reader” messages. SAP AII configuration specifies the conditions used to route fixed reader messages to processing rules.

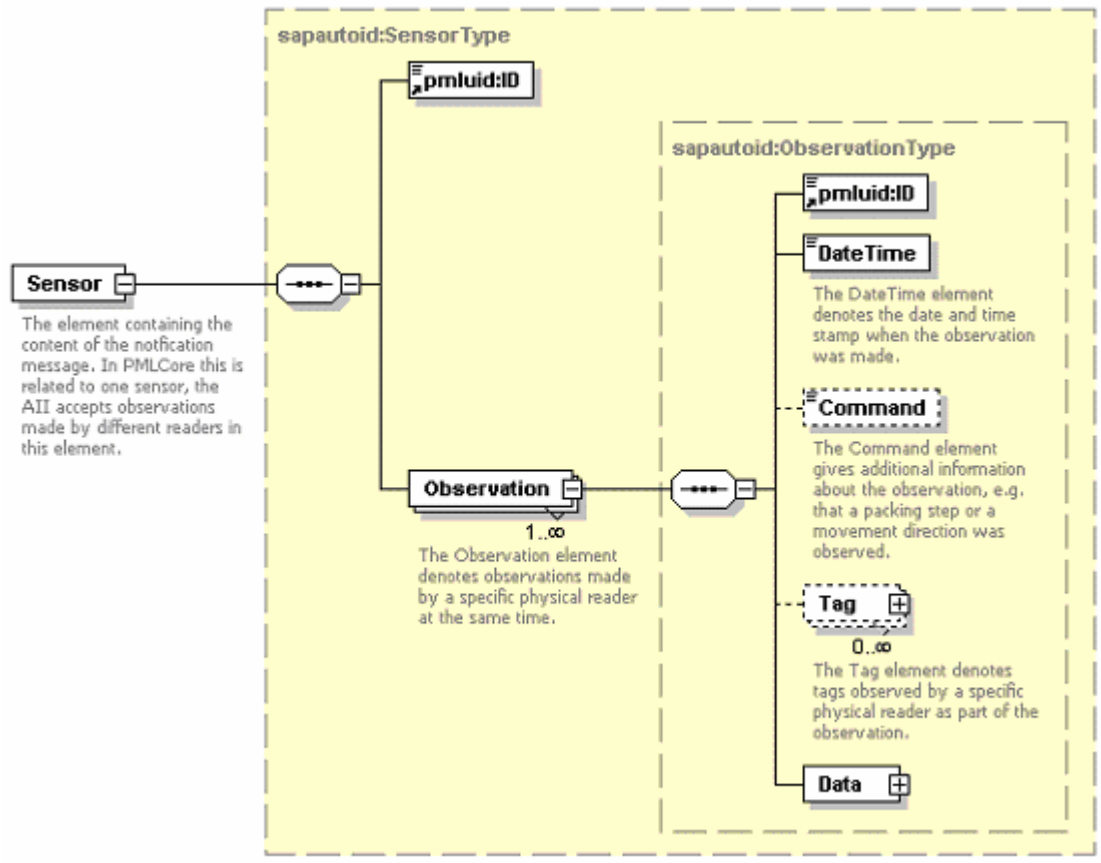
3.2.1 Elements of the Notification Message

The notification message in SAP AII-DC 1.0 is based on the Sensor message described in the PML Core Specification 1.0. The PML Specification, including an XML schema, can be found at the following link:

http://www.epcglobalinc.org/standards_technology/Secure/v1.0/PML_Core_Specification_v1.0.pdf

Exceptions and extensions used by SAP AII are noted below.

The following figure shows an overview of the complete notification message.



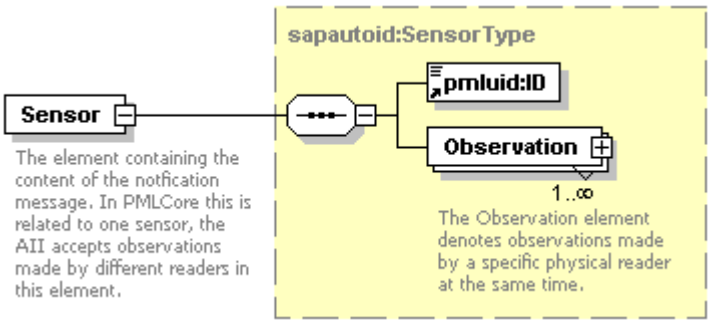
3.2.1.1 Details of the Sensor Element

All elements of the notification are included in the Sensor element.

The pmluid:ID element, which is the first child element of the Sensor element, identifies the device controller that sent the notification message.

A notification message may contain one or more observation elements, each from a different RFID device. Each observation is processed by the appropriate SAP AII rule logic; however, processing of the entire notification message is one logical transaction. SAP AII will commit all observations or none to the database.

element **Sensor**

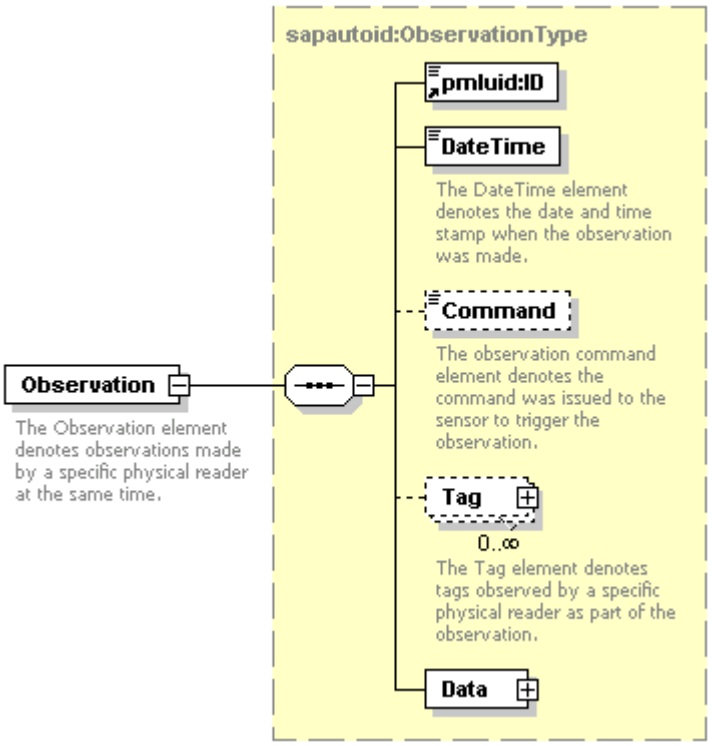
diagram	
namespace	urn:sap:autoID:notification:xml:schema:1
type	**sapautoid:SensorType**
properties	content complex
children	**pmluid:ID Observation**
annotation	documentation The element containing the content of the notification message. In PMLCore this is related to one sensor, the SAP AII accepts observations made by different readers in this element.

3.2.1.2 Details of the Observation Element

The pmluid:ID element, that identifies the observation, is optional in SAP AII.

The command element is required by SAP AII because it is used to select the message processing logic. There is a further description in the Observation/Command section below.

element **Sensor/Observation**

diagram	
namespace	urn:sap:autoID:notification:xml:schema:1
type	**sapautoid:ObservationType**
properties	isRef 0 content complex
children	**pmluid:ID DateTime Command Tag Data**
annotation	documentation The Observation element denotes observations made by a specific physical reader at the same time.

3.2.1.3 Details of the Observation/DateTime Element

The type “dateTime” is a standard XML data type. Please see the definition, currently at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#dateTime.

element **Observation/DateTime**

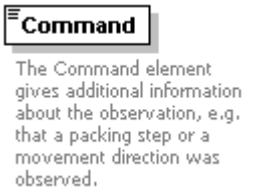
diagram	
namespace	urn:sap:autoID:notification:xml:schema:1
type	**dateTime**
properties	isRef 0 content simple
annotation	documentation The DateTime element denotes the date and time stamp when the observation was made.

3.2.1.4 Details of the Observation/Command Element

The values of the command can be specified by the partner. There are no predefined values in SAP AII. Typically, command values indicate the action associated with the observation. Examples include:

- Direction of movement (IN or OUT)
- Business action (PACK, UNPACK)

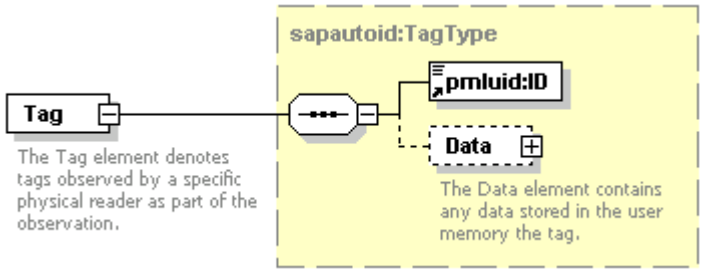
element **Observation/Command**

diagram	
namespace	urn:sap:autoID:notification:xml:schema:1
type	**string**
properties	isRef 0 content simple
annotation	documentation The Command element gives additional information about the observation, e.g. that a packing step or a movement direction was observed.

3.2.1.5 Details of the Observation/Tag Element

The pmluid:ID element is the unique identifier of the RFID tag. This element is omitted for observations of objects without RFID tags.

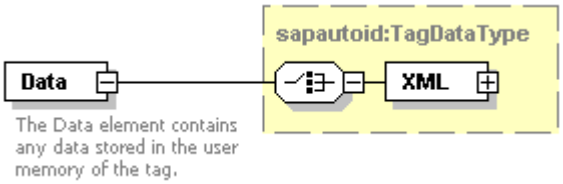
element **Observation/Tag**

diagram	
namespace	urn:sap:autoID:notification:xml:schema:1
type	**sapautoid:TagType**
properties	isRef 0 content complex
children	**pmluid:ID Data**
annotation	documentation The Tag element denotes tags observed by a specific physical reader as part of the observation.

3.2.1.6 Details of the Observation/Tag/Data Element

XMLis the only child element ofTag/Data used in predefined SAP AII activities.

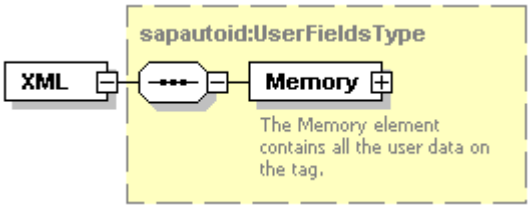
element **Observation/Tag /Data**

diagram	
namespace	urn:sap:autoID:notification:xml:schema:1
type	**sapautoid:TagDataType**
properties	isRef 0 content complex
children	**XML**
annotation	documentation The Data element contains any data stored in the user memory of the tag.

3.2.1.7 Details of the Observation/Tag/Data/XML Element

EPCStatus is the only child element of the Tag/Data/XML element used in a predefined AII activity. The value of EPCStatus can indicate that a tag was commissioned successfully.

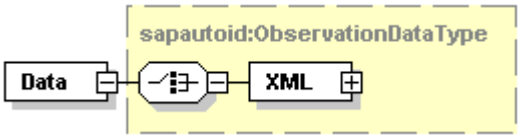
element **Observation/Tag /Data/XML**

diagram	
namespace	urn:sap:autoID:notification:xml:schema:1
type	**sapautoid:UserFieldsType**
properties	isRef 0 content complex
children	**EPCStatus**

3.2.1.8 Details of the Observation/Data Element

The XML element is the only child of the Observation/Data element used by SAP AII.

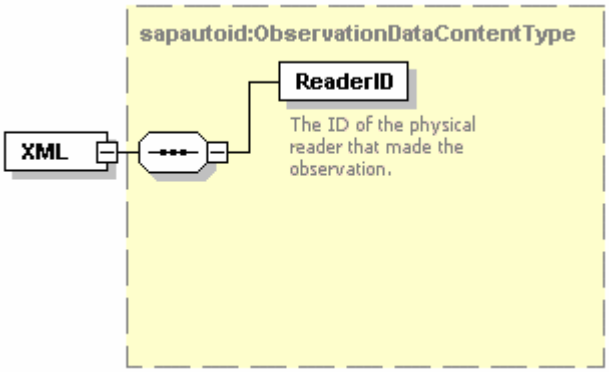
element **Observation/Data**

diagram	
namespace	urn:sap:autoID:notification:xml:schema:1
type	**sapautoid:ObservationDataType**
properties	isRef 0 content complex
children	**XML**

3.2.1.9 Details of the Observation/Data/XML Element

Besides the required ReaderID element, the Observation/Data/XML element has two optional children elements that are used by AII. The GTIN and SSCC elements are used by SAP AII to capture data from non-RFID observations.

element **Observation/Data /XML**

diagram	
namespace	urn:sap:autoID:notification:xml:schema:1
type	**sapautoid:ObservationDataContentType**
properties	isRef 0 content complex
children	**ReaderID GTIN SSCC**

3.2.1.10 Details of the Observation/Data/XML/ReaderID Element

The ReaderID tag must contain the SAP AII RFID Device ID that made the observation.

element **Observation/Data /XML/ReaderID**

diagram	**ReaderID** The ID of the physical reader that made the observation.
namespace	urn:sap:autoID:notification:xml:schema:1
properties	isRef 0
annotation	documentation The ID of the physical reader that made the observation.

3.2.2 Samples of the RFID Tag Notification Messages

3.2.2.1 Single Tag

One scenario for the notification message is reading a single EPC RFID tag and sending that information to SAP AII. A sample XML shows how the device controller sends the message to SAP AII. This message might be configured to trigger a Verify, Pack, Move or Load action.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <pmlcore:Sensor
     xmlns:pmlcore="urn:autoid:specification:interchange:PMLCore:xml:schema:1"
     xmlns:pmluid="urn:autoid:specification:universal:Identifier:xml:schema:1"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pmluid:ID>DEVICE_CONTROLLER_NAME</pmluid:ID>
     <pmlcore:Observation>
      <pmlcore:DateTime>2004-11-12T02:00:00.762+01:00</pmlcore:DateTime>
      <pmlcore:Command>IN</pmlcore:Command>
        <pmlcore:Tag>
          <pmluid:ID>A00300607800002000000000</pmluid:ID>
      </pmlcore:Tag>
        <pmlcore:Data>
           <pmlcore:XML>
            <ReaderID>PACKING_DEVICE_NAME</ReaderID>
          </pmlcore:XML>
      </pmlcore:Data>
    </pmlcore:Observation>
</pmlcore:Sensor>
```

With the *Tag commissioning-Verify Tag (TCVR)* rule, this message can be used to verify that a tag, commissioned by SAP AII, can be read.

With the *Conveyor Packing – Case Reading (CPCR)* rule, this message can be used to report the observation of a case that is being packed onto a pallet.

In the move scenario, this message can report the location of a pallet and, by implication, the associated cases. The location is determined from the location assigned to the RFID Device in SAP AII.

With a *Free Pack – Loading (FPL)* or *Pack to Stock – Loading (PSL)* rule, this message can confirm the loading of a pallet and, by implication, the associated cases.

3.2.2.2 Multiple Tags in One Observation

Observations of more than one tag can be used in all the single tag scenarios above, except conveyor packing. In addition, multi-tag observations can enable enhanced pack and load logic.

With a *Free Pack – Packing (FPP)* or *Pack to Stock – Packing (PSP)* rule, this message can be used to report the observation of a pallet and the cases that are being packed. The rule logic relies on the tag's "EPC Filter" value to determine which object is the pallet. Only one pallet can be included in the list of tags in a valid pack observation. Technically, these rules identify the tag with the lowest filter value as the parent object in the hierarchy. All other tags are registered as the parent's children.

In the pack scenario, this message can confirm the loading of a pallet, together with the associated cases. Some configurations require that all or some of the cases be observed, before a loading can be confirmed.

```
<?xml version="1.0" encoding="UTF-8" ?>
 <pmlcore:Sensor
    xmlns:pmlcore="urn:autoid:specification:interchange:PMLCore:xml:schema:1"
    xmlns:pmluid="urn:autoid:specification:universal:Identifier:xml:schema:1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <pmluid:ID>DEVICE_CONTROLLER_NAME</pmluid:ID>
   <pmlcore:Observation>
    <pmlcore:DateTime>2004-11-12T02:00:00.762+01:00</pmlcore:DateTime>
    <pmlcore:Command>IN</pmlcore:Command>
     <pmlcore:Tag>
      <pmluid:ID>A00300607800002000000000</pmluid:ID>
     </pmlcore:Tag>
     <pmlcore:Tag>
      <pmluid:ID>319402422000000005E9F014</pmluid:ID>
     </pmlcore:Tag>
     <pmlcore:Tag>
      <pmluid:ID>30740242204031400097999 9</pmluid:ID>
     </pmlcore:Tag>
     <pmlcore:Data>
       <pmlcore:XML>
        <ReaderID>PACKING_DEVICE_NAME</ReaderID>
```

```
        </pmlcore:XML>
      </pmlcore:Data>
    </pmlcore:Observation>
</pmlcore:Sensor>
```

3.2.2.3 Verification of External Tag Commissioning

This process reports that a tag was commissioned and indicates whether the tag was successfully read. The SAP AII rule *Tag commissioning-Verify Tag* (*TCVR) handles notification messages that inlcude the EPCStatus element*. The following sample XML shows how the device controller sends the message to SAP AII.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <pmlcore:Sensor
     xmlns:pmlcore="urn:autoid:specification:interchange:PMLCore:xml:schema:1"
     xmlns:pmluid="urn:autoid:specification:universal:Identifier:xml:schema:1"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pmluid:ID>DEVICE_CONTROLLER_NAME</pmluid:ID>
     <pmlcore:Observation>
      <pmlcore:DateTime>2004-11-12T02:00:00.762+01:00</pmlcore:DateTime>
      <pmlcore:Command>IN</pmlcore:Command>
       <pmlcore:Tag>
        <pmluid:ID>A00300607800002000000000</pmluid:ID>
        <pmlcore:Data>
            <pmlcore:XML>
             <EPCStatus>WS</EPCStatus>
           </pmlcore:XML>
         </pmlcore:Data>
       </pmlcore:Tag>
        <pmlcore:Data>
          <pmlcore:XML>
           <ReaderID>VERIFY_DEVICE_NAME</ReaderID>
         </pmlcore:XML>
       </pmlcore:Data>
    </pmlcore:Observation>
</pmlcore:Sensor>
```

The element EPCStatus may have following values:

- ‘WS’ to indicate that the tag was written and verified successfully
- ‘WU’ to indicate that the tag was written, but not verified successfully

3.2.3 Sample of Other Observations (Barcodes, Light Sensors, Etc.)

The notification message can be used to report events other than reading an RFID tag. In the following examples, such events are used by SAP AII to trigger tag commissioning. These might be used on a conveyer. The notification message is sent when a case is detected. SAP AII rule *Tag Commissioning – Print Tag (TCPR)* responds by sending a tag commission command WriteTagData to the RFID device that is downstream on the conveyer.

3.2.3.1 Reading a GTIN (from a Barcode) to Trigger Tag Commissioning

A fixed bar code reader reads the GTIN on a case. The device controller sends the following XML to SAP AII. This notification message triggers the TCPR rule which uses the GTIN to generate an SGTIN EPC and send the tag commission command WriteTagData to an RFID device.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <pmlcore:Sensor
     xmlns:pmlcore="urn:autoid:specification:interchange:PMLCore:xml:schema:1"
     xmlns:pmluid="urn:autoid:specification:universal:Identifier:xml:schema:1"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pmluid:ID>DEVICE_CONTROLLER_NAME</pmluid:ID>
     <pmlcore:Observation>
      <pmlcore:DateTime>2004-11-01T13:04:33.050+06:00</pmlcore:DateTime>
      <pmlcore:Command>PRNT</pmlcore:Command>
        <pmlcore:Data>
          <pmlcore:XML>
            <ReaderID>READER_DEVICE_NAME</ReaderID>
            <GTIN>00037000567394</GTIN>
         </pmlcore:XML>
       </pmlcore:Data>
    </pmlcore:Observation>
</pmlcore:Sensor>
```

3.2.3.2 Reading an SSCC to Trigger Printing

A fixed bar code reader reads the SSCC on a pallet. The device controller sends the following XML to SAP AII. This notification message triggers the TCPR rule which uses the SSCC to generate an SSCC EPC and send the tag commission command WriteTagData to an RFID device.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <pmlcore:Sensor
     xmlns:pmlcore="urn:autoid:specification:interchange:PMLCore:xml:schema:1"
     xmlns:pmluid="urn:autoid:specification:universal:Identifier:xml:schema:1"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pmluid:ID>DEVICE_CONTROLLER_NAME</pmluid:ID>
     <pmlcore:Observation>
```

```
      <pmlcore:DateTime>2002-11-06T13:04:33.050-06:00</pmlcore:DateTime>
      <pmlcore:Command>PRNT</pmlcore:Command>
          <pmlcore:Data>
          <pmlcore:XML>
            <ReaderID>READER_DEVICE_NAME</ReaderID>
            <SSCC>000370000000000266</SSCC>
        </pmlcore:XML>
      </pmlcore:Data>
    </pmlcore:Observation>
</pmlcore:Sensor>
```

3.2.3.3 Sending a Blank Message to Trigger Tag Commissioning

A light sensor detects a case on the conveyer. The device controller sends the following XML to SAP SAP AII. This notification message triggers the TCPR rule which generates an EPC and sends the tag commission command WriteTagData to an RFID device.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <pmlcore:Sensor
     xmlns:pmlcore="urn:autoid:specification:interchange:PMLCore:xml:schema:1"
     xmlns:pmluid="urn:autoid:specification:universal:Identifier:xml:schema:1"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <pmluid:ID>DEVICE_CONTROLLER_NAME</pmluid:ID>
     <pmlcore:Observation>
      <pmlcore:DateTime>2002-11-01T13:04:33.05-06:00</pmlcore:DateTime>
      <pmlcore:Command>PRNT</pmlcore:Command>
          <pmlcore:Data>
          <pmlcore:XML>
            <ReaderID>READER_DEVICE_NAME</ReaderID>
        </pmlcore:XML>
      </pmlcore:Data>
    </pmlcore:Observation>
</pmlcore:Sensor>
```

4 Communication Protocol

The format of all messages is XML. The header of the XML message identifies the encoding, for example, UTF-8.

The interface uses the HTTP 1.1 protocol (RFC 2.6.1.6, http://www.w3.org/protocol) for the messages from the Auto-ID Infrastructure to the device controller and from the device controller to the Auto-ID Infrastructure. Partners can deliver HTTPS as an extension.

The messages to and from SAP AII are sent via HTTP POST and must be answered synchronously by HTTP OK. This indicates successful receipt of the message and does not imply successful processing of the message by the application logic.

Alternately, the Auto-ID Infrastructure can be configured to send the WriteTagData Command message to a TCP socket on the device controller. Note: for SAP AII systems with Unicode enabled, XML sent to the socket has UTF-16 encoding.

5 APPENDIX

5.1 Command Message XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 4 U (http://www.xmlspy.com) by Mr. Uwe SchÃ¤fer (SAP AG) -->
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
	<xs:element name="Command">
		<xs:annotation>
			<xs:documentation>A command message to the device controller</xs:documentation>
		</xs:annotation>
		<xs:complexType>
			<xs:choice>
				<xs:element name="WriteTagData">
					<xs:annotation>
						<xs:documentation>This element writes user fields and, if the tags allow this, also the
identifier by using special field names (see fieldlist element) of data tags.</xs:documentation>
					</xs:annotation>
					<xs:complexType>
						<xs:sequence>
							<xs:element name="Item" maxOccurs="unbounded">
								<xs:annotation>
									<xs:documentation>Use to structure the child
elements.</xs:documentation>
								</xs:annotation>
								<xs:complexType>
									<xs:sequence>
										<xs:element name="TagID" minOccurs="0"
maxOccurs="unbounded">
											<xs:annotation>
												<xs:documentation>If one or more  tag IDs are given,
then only these tags are written. Else all tags in the radio field are written.</xs:documentation>
											</xs:annotation>
										</xs:element>
										<xs:element name="FieldList">
											<xs:annotation>
												<xs:documentation>List of names of the fields to be
written and their value.
In the case of EPC Class 1 and above compliant tags the EPC (as tag identifier) can be written using the special field
name "EPC" here.</xs:documentation>
											</xs:annotation>
											<xs:complexType>
												<xs:sequence>
													<xs:element name="Field"
maxOccurs="unbounded">
														<xs:annotation>
															<xs:documentation>The field - name as
```

```
attribute and the value as text- or  CDATA node</xs:documentation>
                                                                    </xs:annotation>
                                                                    <xs:complexType mixed="true">
                                                                        <xs:attribute name="name"
type="xs:string" use="required">
                                                                            <xs:annotation>
                                                                                <xs:documentation>Name of
the data field on the tag</xs:documentation>
                                                                            </xs:annotation>
                                                                        </xs:attribute>
                                                                    </xs:complexType>
                                                                </xs:element>
                                                            </xs:sequence>
                                                            <xs:attribute name="format" type="xs:string"
use="optional"/>
                                                            <xs:attribute name="jobName" type="xs:string"
use="optional"/>
                                                            <xs:attribute name="quantity" type="xs:string"
use="optional"/>
                                                            <xs:attribute name="printerName" type="xs:string"
use="optional"/>
                                                        </xs:complexType>
                                                    </xs:element>
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                    <xs:attribute name="readerID" type="xs:string" use="optional">
                                        <xs:annotation>
                                            <xs:documentation>Used if only a specific (physical) reader should be
queried.</xs:documentation>
                                        </xs:annotation>
                                    </xs:attribute>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="ExtensionCommand">
                                <xs:annotation>
                                    <xs:documentation>This command allows use of vendor specific features of device
controllers and/or devices. The interpretation of the value is up to the device controller.</xs:documentation>
                                </xs:annotation>
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:any namespace="##any">
                                            <xs:annotation>
                                                <xs:documentation>The value can either be a textstring or of XML
format.</xs:documentation>
                                            </xs:annotation>
                                        </xs:any>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:choice>
                        <xs:attribute name="id" type="xs:string" use="optional"/>
                    </xs:complexType>
                </xs:element>
</xs:schema>
```

5.2 Summary of Message Elements

In the command message sent by SAP AII, the fields marked “Mandatory” are always populated. Fields marked “Optional” are populated as described. For more information on configuration, see the SAP AII documentation.

	Element	Mandatory	Optional
Command	WriteTagData	readerID attribute is provided	
	WriteTagData/Item	x	
	Item/FieldList	x	Attributes for printing are configurable
	Item/FieldList/Field	Field named EPC is provided	Additional named fields are configurable
	ExtensionCommand	x	

The notification message received by SAP AII is expected to include the elements marked “Mandatory“. Optional fields apply only in specific scenarios.

	Element	Mandatory	Optional
Sensor (Notification)	Sensor/pmluid:ID	Device Controller ID	
	Observation/DateTime	x	
	Observation/Command	x	
	Observation/Tag		only for RDIF observations
	Tag/pmluid:ID		Required if Tag is present
	Tag/Data/XML/EPCStatus		write validation
	Observation/Data/ReaderID	Device ID	
	Observation/Data/XML/GTIN		Print/write signal
	Observation/Data/XML/SSCC		Print/write signal