

# SIEMENS

## SIMATIC Ident

### RFID systems XML programming for SIMATIC Ident

#### Application Manual

<u>Introduction</u>	<b>1</b>
<u>API description</u>	<b>2</b>
<u>Programming</u>	<b>3</b>
<u>Service &amp; Support</u>	<b>A</b>

## Legal information

### Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

<b>⚠ DANGER</b>
indicates that death or severe personal injury <b>will</b> result if proper precautions are not taken.
<b>⚠ WARNING</b>
indicates that death or severe personal injury <b>may</b> result if proper precautions are not taken.
<b>⚠ CAUTION</b>
indicates that minor personal injury can result if proper precautions are not taken.
<b>NOTICE</b>
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

### Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

### Proper use of Siemens products

Note the following:

<b>⚠ WARNING</b>
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

### Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

### Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

# Table of contents

<b>1</b>	<b>Introduction</b> .....	<b>5</b>
<b>2</b>	<b>API description</b> .....	<b>7</b>
2.1	Properties and area of application .....	7
2.2	Principle of operation .....	8
2.3	Demo application .....	9
2.3.1	Startup and structure of the demo application .....	9
2.3.2	User interface of the demo application .....	11
2.3.3	Working with the demo application .....	12
2.4	Compatibility .....	14
<b>3</b>	<b>Programming</b> .....	<b>15</b>
3.1	Command execution and responses .....	15
3.2	Command/response overview .....	15
3.3	XML commands/responses .....	18
3.3.1	Connections .....	19
3.3.1.1	hostGreetings.....	20
3.3.1.2	hostGoodbye.....	22
3.3.1.3	heartBeat .....	23
3.3.1.4	setIPConfig .....	24
3.3.1.5	getIPConfig .....	25
3.3.2	Device settings.....	27
3.3.2.1	setConfiguration.....	27
3.3.2.2	getConfiguration.....	28
3.3.2.3	getConfigVersion .....	29
3.3.2.4	getActiveConfiguration.....	30
3.3.2.5	getLogfile .....	32
3.3.2.6	resetLogfile .....	36
3.3.2.7	setParameter .....	38
3.3.2.8	getParameter .....	40
3.3.2.9	setTime .....	43
3.3.2.10	getTime .....	44
3.3.2.11	setIO.....	46
3.3.2.12	getIO .....	48
3.3.2.13	resetReader .....	49
3.3.2.14	resetDevice .....	51
3.3.2.15	getReaderStatus .....	52
3.3.2.16	getDeviceStatus.....	54
3.3.2.17	getCMReaderStatus .....	55
3.3.2.18	getAllSources.....	57
3.3.3	Transponder processing .....	59
3.3.3.1	editBlackList.....	59
3.3.3.2	getBlackList.....	61
3.3.3.3	triggerSource .....	62

3.3.3.4	readTagIDs .....	64
3.3.3.5	writeTagID .....	67
3.3.3.6	getObservedTagIDs .....	71
3.3.3.7	readTagMemory .....	75
3.3.3.8	writeTagMemory .....	79
3.3.3.9	readTagField .....	84
3.3.3.10	writeTagField .....	88
3.3.3.11	killTag .....	92
3.3.3.12	lockTagBank .....	95
3.3.3.13	getTagStatus .....	101
3.3.3.14	stopCommand .....	103
3.3.4	Error messages (negative XML replies) .....	104
3.4	XML reports .....	109
3.4.1	Events .....	109
3.4.1.1	tagEventReport .....	109
3.4.1.2	rssEventReport .....	113
3.4.1.3	ioEventReport .....	116
3.4.1.4	logEventReport .....	117
3.4.1.5	presenceEventReport .....	119
3.4.1.6	lastAccessEventReport .....	120
3.4.2	alarm .....	125
<b>A</b>	<b>Service &amp; Support .....</b>	<b>129</b>

# Introduction

## Purpose of this documentation

This manual contains all information required to program SIMATIC Ident devices via the XML interface.

## Basic knowledge required

Specific knowledge of XML programming, as well as general knowledge of automation technology and identification systems, is required to understand this manual.

## Trademarks

The following and possibly other names not identified by the registered trademark sign ® are registered trademarks of Siemens AG:

SIMATIC ®, SIMATIC RF ®, MOBY ®, RF MANAGER ®, SIMATIC Sensors ®

## Orientation in the documentation

In addition to this manual, you may need the hardware manuals of the particular Ident devices that you want to program. At this time, the SIMATIC RF610R, RF615R, RF650R, RF680R and RF685R readers, as well as the SIMATIC RF185C, RF186C, RF188C, RF186CI and RF188CI communication modules, can be programmed via the XML interface.

You can find the current versions of the relevant manuals on the pages of the Siemens Industry Online Support (<https://support.industry.siemens.com/cs/ww/en/ps/14970/man>).

## Abbreviations and naming conventions

The following terms/abbreviations are used synonymously in this document:

Transponder, tag

Data medium, mobile data storage (MDS)

Communication module (CM)

Interface module (ASM)

## Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit

Link: (<https://www.siemens.com/industrialsecurity>)

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customers' exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under

Link: (<https://support.industry.siemens.com/cs/ww/en/ps/15247/pm>)

# API description

## 2.1 Properties and area of application

### Properties

This protocol represents simple XML communication for a point-to-point connection. The frames (commands/responses) are defined in XML syntax.

### Area of application

Using the following XML commands, you can configure/program and diagnose Ident devices that are not connected to a controller via a user application in an Ethernet-based IT environment.

At this time, the SIMATIC RF610R, RF615R, RF650R, RF680R and RF685R readers, as well as the SIMATIC RF185C, RF186C, RF188C, RF186CI and RF188CI communication modules, can be programmed via the XML interface.

### Integration

The following graphic shows an example hardware configuration: In addition to a PC with the user application, you need a compatible Ident device that is connected to the PC or a third-party controller via Ethernet.

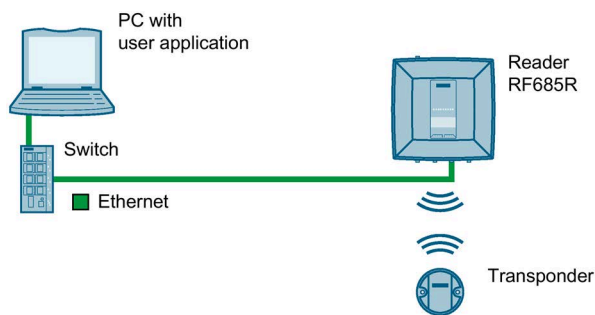


Figure 2-1 Example of a suitable hardware configuration for XML programming

## 2.2 Principle of operation

The XML interface is based on command/reply frames. Each command that you send to the connected Ident device via the XML programming interface is replied to by the Ident device with a reply frame regardless of whether the command was executed successfully or not. If errors occur during communication or if the command could not be executed, the reply frame contains an error description. In addition to these command/reply frames, the Ident device can also send asynchronous reports to the XML programming device.

To ensure unique assignment of the commands and replies, each command must include a unique ID. This ID is repeated in the corresponding reply frame.

Normally, the Ident device replies to a command within 5 seconds. We recommend that you program your application so that if this time is exceeded, the application starts appropriate error handling.

Some commands (e.g. "setConfiguration" or "readTagIds") can take longer than 5 seconds. These commands already include information to this effect.

---

### Note

#### Saving and working through the commands

It is also possible to send several commands without waiting for the replies. The Ident device works through the commands in the precise order in which they were received. Note that the Ident device discards newly arriving commands if there are already approximately 100 commands waiting for execution in the device memory.

---

### Asynchronous notifications (XML reports)

Apart from the synchronous command/reply frames, asynchronous notifications - so-called reports - are also transferred. These reports are generated by the Ident device and may require confirmation of receipt by the user application. Each transfer includes a unique ID (<id>) generated by the Ident device. In contrast to the IDs of the commands, this ID is generated by the Ident device itself. The user application can only acknowledge this notification with the same ID.

The reports are divided into events and alarm messages. An event contains data acquired by the Ident device itself. Alarm messages inform the user application automatically of special or incorrect operating conditions of the Ident device.

The reports can be transferred transacted or not transacted.

- In the non-secure mode, all reports are sent to the user application without waiting for confirmation of receipt. The report is automatically discarded if the connection to the user application does not exist or is interrupted.
- In transacted mode, the receipt of every report must be confirmed by the user application with a reply frame. If no confirmation of receipt is received within approximately 10 seconds, the Ident device sends the report to the user application again.

If there are connection errors or interruptions, the reports are stored on the Ident device until the connection is re-established. Note that saved reports are lost when the Ident device is switched off. If necessary, activate the "Transacted transfer" function in the configuration of the Ident device if the connection between the Ident device and PC is not



stable, for example due to a bad WLAN connection or if data can only be retrieved at longer intervals (batch operation).

You will find further information on the structure of the reply frames in the section "tagEventReport (Page 109)".

Note that the Ident device can save a maximum of 10 000 reports. If this number is exceeded, newly generated reports are discarded internally.

## 2.3 Demo application

A demo application created by Siemens and based on Windows .NET 3.5 is described in this section. This application serves as a model on the basis of which you can program your own user application. All XML commands listed in "Programming (Page 15)" are contained in the demo application. This allows you to test your Ident devices directly using the demo application and to run commands.

You can find the described demo application including source code files as download on the pages of "Siemens Industry Online Support (<https://support.industry.siemens.com/cs/ww/en/view/109768503>)".

---

### Note

#### Disclaimer of liability

Note that Siemens AG accepts no liability for the demo application "RFID Reader XML Demo".

---

### 2.3.1 Startup and structure of the demo application

#### Requirement

.NET 3.5 Runtime is installed on the Windows PC.

#### Start

Follow the steps below to start the demo application on your PC:

1. Download the "SIMATIC\_Ident\_XML-Demoapplikation.exe" file to your PC.

You can find the file on the pages of the Siemens Industry Online Support (<https://support.industry.siemens.com/cs/ww/en/view/109768503>).

2. Run the file.

It is a self-extracting ZIP file.

3. Start the application by double-clicking on the file "RFReader.TestApp.exe".

Result: The demo application starts.

### Structure

The demo application consists of the following main components:

- Demo XML API "RFReader.XmlApi.dll"

The Demo XMP API is implemented using the DLL file. The demo application is based on the XML API. The XML API makes all XML functions available via .NET. To be able to test the XML API in your own application, you need to reference the DLL file in your project.

- User interface of the "RFReader.TestApp.exe" demo application and associated files

Start the demo application by running the "RFReader.TestApp" file. It accesses the Demo XML API (DLL file) to be able to use the XML functions.

This application provides you with a graphical interface with which you can send predefined commands to Ident devices. The application can communicate with multiple devices in parallel. For each connected device, a new instance of the "RFReader.XmlApi" is generated with the "Device > Connect Device" command and managed by the application (with the respective API name).

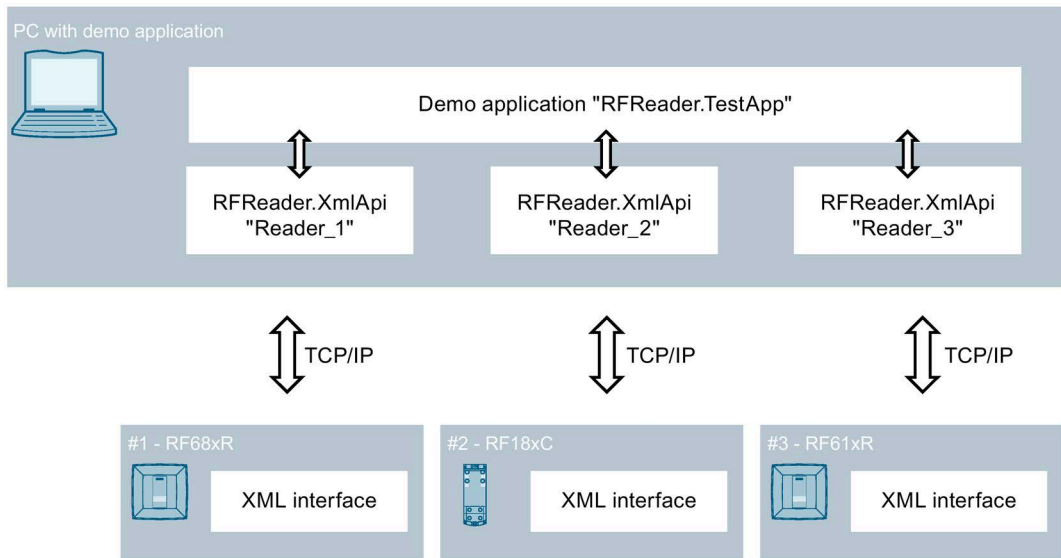
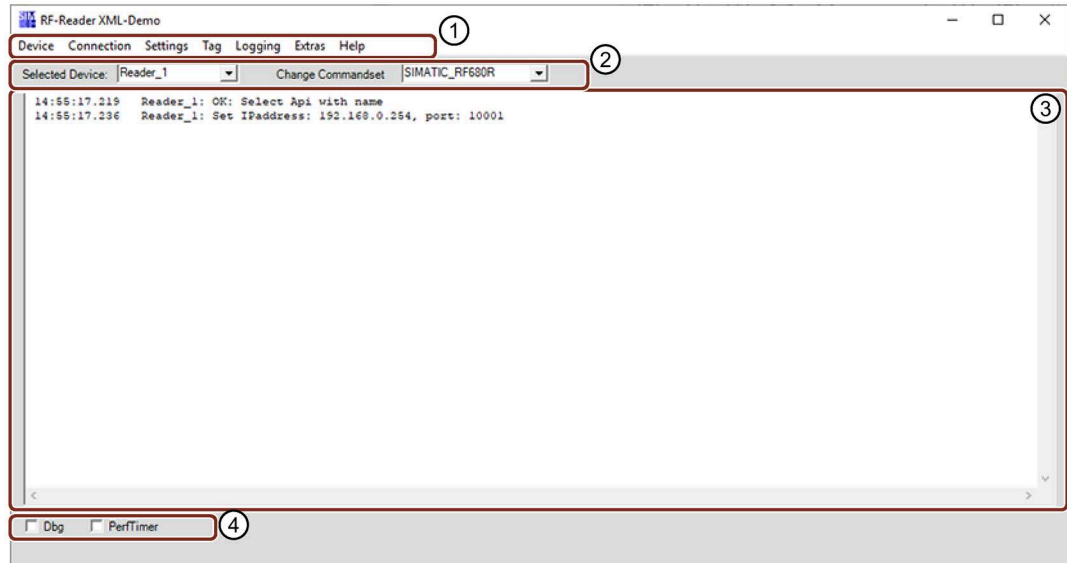


Figure 2-2 Structure/functions of the demo application

## 2.3.2 User interface of the demo application

The start window of the application appears after the demo application has been started. The application is divided into the following areas:



- ① Menu bar
- ② Device display
- ③ Log window
- ④ Display options

Figure 2-3 User interface of the demo application

### Menu bar

The menu bar contains all available commands. These are grouped together in the relevant menus. Each individual command can be selected from the menu tree.

Menu	Description
Device	Functions for connecting to the Ident device, disconnecting from the Ident device and for selecting devices.
XML commands	
Connection	XML commands with which you can control the connection between the user application and Ident device.
Settings	XML commands with which you can control the configuration of the Ident device.
Tag	XML commands with which you can control processing of the transponder data.
Logging	XML commands for reading out and deleting log files.

Menu	Description
Extras	Functions for resetting the log window and for saving transponder data. Using the menu command "SaveTagEventReports", you can save acquired transponder data on your PC as a *.csv file or in an SQL database.
Help	Information on the demo application (e.g. XML API version)

### Device display

Display of the currently selected Ident device with which you are communicating. The selection of the command scope depends on the connected device.

### Log window

Text box in which all executed commands and their return values are displayed. Alarm messages and events sent by the Ident device are also displayed in the log window. The log window can be cleared with the "Extras > Clear log" menu entry.

### Display options

This area contains two check boxes:

- Dbg

With this check box, all transferred data is also displayed in XML format.

- PerfTimer

With this check box, you can display the execution time of each command.

The displayed time relates to the period between sending the command and the arrival of the reply on the PC.

## 2.3.3 Working with the demo application

### Requirements

- The Ident device is connected and has started up.
- A unique IP address has been assigned to the Ident device.
- XML communication has been activated in the WBM of the Ident device.

## Procedure

Follow the steps below to establish a connection to the Ident device:

1. Start the demo application.
2. Select the menu command "Device > Connect device".
3. Enter the IP address of the device in the "Device IP Address" input box.
4. If necessary, select the "Transacted" check box to enable transfer with transaction security in the application.

Please note that transacted transfer only works if it has been activated in the WBM of the Ident device.

5. If necessary, change the API name to have the option of switching over between Ident devices when working with several device.
6. Confirm your entry with "OK".
7. Select the menu command "Connection > HostGreetings".
8. In the "Device type" input box, enter the device type with which the application should connect.

Notation: "SIMATIC\_RF6xxR" (e.g. "SIMATIC\_RF680R")

If this input box is not completed, the application connects to every connected, compatible device.

9. Confirm your entry with "OK".

Remember that after connecting to the Ident device, the "HostGreetings" command must always be executed first.

The connection to the Ident device is established. The API name of the currently active device is displayed in the "Selected device" text box. All commands are sent exclusively to this device.

You can communicate with several Ident devices at the same time. To communicate with another device, repeat the actions described above. Once you have established a connection to multiple devices, you can switch conveniently from one to the other using the "Ident device > Select device" menu command.

After you have established the connection to an Ident device and have executed the "HostGreetings" command, you can communicate with the device. To do this, various commands are available in the menus. These commands are described in the following sections.

## 2.4 Compatibility

The XML API of the demo application was released for the first time with version V2.0. Since then, the XML API has been expanded by some functions and commands and many additional Ident devices and product families have been added that can be programmed using the XML API.

The XML API versions are backward-compatible. The following table provides an overview of the compatible devices and versions.

Table 2- 1 Overview of compatible versions

Device/Product family	Firmware version of the device / product family	Protocol version of the XML API (XML API version)
RF600	V1.0 ... V2.0	V2.0
	V2.1 ... V2.2	V2.0; V2.1
	As of V2.2.1	V2.0; V2.1; V2.2
RF18xC/RF18xCI	V1.3	V3.0

# Programming

## 3.1 Command execution and responses

This section describes all XML commands and responses that you can send to or receive from an Ident device via the XML API.

Each command sent by the XML API application is replied to by the Ident device with a reply frame. If the command was executed successfully, the reply frame has the value "0" in the "ResultCode" parameter. If other values are returned in this parameter, this means that the command was not executed successfully. In this case, the returned value corresponds to the error code.

---

**Note****Parameter sequence in commands**

In all commands (and responses), note that the parameters within a hierarchy level can be in any sequence.

---

**Note****Number of parameters depends on the API/firmware version**

Note that further optional parameters may be added in later development stages. Checking the number of parameters could therefore lead to incompatibilities.

---

## 3.2 Command/response overview

The XML API of the demo application was released for the first time with version V2.0. Since then, the XML API has been expanded by some functions and commands and many additional Ident devices and product families have been added that can be programmed using the XML API.

The following tables provide you with an overview of all XML commands and responses stored in the XML API. There is a separate table for each command/response category. Based on the tables, you can determine the XML API protocol version as of which the respective commands are contained in the XML API and the device firmware versions with which they are compatible. In addition, each XML command/XML response section contains a separate overview of the XML API protocol version (API version) and device firmware versions as of which the respective command or response is valid.

### XML commands

Table 3- 1 Overview of the "Connections" XML commands

XML command	Description	From XML API version	From firmware version	
			RF600	RF1xxC
hostGreetings	Establish connection.	V1.0	V1.0	V1.3
hostGoodbye	Terminate connection.	V1.0	V1.0	V1.3
heartBeat	Monitor connection.	V1.0	V1.0	V1.3
setIPConfig	Specify Ethernet address of device.	V1.0	V2.2.1	V1.3
getIPConfig	Read out Ethernet address of device.	V2.1	V2.2.1	V1.3

Table 3- 2 Overview of the "Reader settings" XML commands

XML command	Description	From XML API version	From firmware version	
			RF600	RF1xxC
setConfiguration	Transfers a configuration to the device and saves it retentively.	V1.0	V1.0	V1.3
getConfiguration	Reads out the saved configuration from the device.	V1.0	V1.0	V1.3
getConfigVersion	Reads out the version ID of the saved configuration from the device.	V1.0	V1.0	V1.3
getActiveConfiguration	Reads out the current values from the device.	V2.0	V1.0	V1.3
getLogfile	Reads out the logbook of the device.	V2.0	V1.0	V1.3
resetLogfile	Deletes the logbook of the device.	V2.0	V1.0	V1.3
setParameter	Sets a value of a device parameter.	V2.0	V1.0	V1.3
getParameter	Reads out a device parameter.	V2.0	V1.0	V1.3
setTime	Sets the internal device clock.	V1.0	V1.0	V1.3
getTime	Reads the time of day of the internal device clock.	V1.0	V1.0	V1.3
setIO	Sets the digital outputs of the device.	V1.0	V1.0	V1.3
getIO	Reads the states of the digital inputs/outputs of the device.	V1.0	V1.0	V1.3
resetReader	Resets the device. Note: Present for compatibility reasons. This command has been replaced by the "resetDevice" command.	V1.0	V1.0	V1.3
resetDevice	Resets the device.	V3.0	--	V1.3
getReaderStatus	Reads out the status information of the device. Note: Present for compatibility reasons. This command has been replaced by the "getDeviceStatus" command.	V1.0	V1.0	V1.3
getDeviceStatus	Reads out the status information of the device.	V3.0	--	V1.3



XML command	Description	From XML API version	From firmware version	
			RF600	RF1xxC
getCMReaderStatus	Reads out the status information of the device.	V3.0	--	V1.3
getAllSources	Reads out the names of all configured read points of the device.	V1.0	V1.0	V1.3

Table 3- 3 Overview of the "Transponder processing" XML commands

XML command	Description	From XML API version	From firmware version	
			RF600	RF1xxC
editBlacklist	Save transponders IDs to or remove them from the "Black list".	V2.0	V1.0	--
getBlacklist	Reads out the transponder IDs from the "Black list".	V2.0	V1.0	--
triggerSource	A read point is displayed for reading. Detected transponders are reported by "tagEventReport".	V1.0	V1.0	V1.3
readTagIDs	The read point performs the desired number of inventories and returns the detected transponders in the reply frame.	V1.0	V1.0	V1.3
writeTagID	Writes a new ID to the transponder.	V1.0	V1.0	--
getObservedTagIDs	Returns all transponders that have the "observed" status.	V2.0	V1.0	V1.3
readTagMemory	Reads out data from a memory area of a transponder.	V1.0	V1.0	V1.3
writeTagMemory	Writes data to a memory area of a transponder.	V1.0	V1.0	V1.3
readTagField	Reads out data from a symbolically addressed memory area of a transponder.	V1.0	V1.0	--
writeTagField	Writes data to a symbolically addressed memory area of a transponder.	V1.0	V1.0	--
killTag	Disables a transponder.	V1.0	V1.0	--
lockTagBank	Locks a memory area of a transponder.	V1.0	V1.0	--
getTagStatus	Reads out the status information of the transponder.	V3.0	--	V1.3
stopCommand	Ends a running/pending command.	V3.0	--	V1.3

## XML reports

Table 3- 4 Overview of XML reports

XML report	Description	From XML API version	From firmware version	
			RF600	RF1xxC
tagEventReport	Sends the transponder data of the respective transponder on every status change (observed, lost).	V1.0	V1.0	V1.3
rssEventReport	Signals every change in the RSSI values when reading the transponders.	V1.0	V1.0	--
ioEventReport	Signals every change of a digital input/output.	V1.0	V1.0	V1.3
logEventReport	Signals new logbook entries.	V3.0	--	V1.3
presenceEventReport	Signals the transponders currently located in the antenna field.	V3.0	--	V1.3
lastAccessEventReport	Sends the transponder data of the respective transponder on a successful transponder command.	V3.0	--	V1.3
alarm	Status, warning or error messages of the device.	V1.0	V1.0	V1.3

## 3.3 XML commands/responses

### Optional command lines

The following sections explain the XML commands and responses in detail. Note that some parameters of the XML commands/responses are optional. These parameters are designated by the code "opt".

### 3.3.1 Connections

This section describes all the commands with which you can control the connection between the user application and the Ident device.

The following diagram shows how a connection is established and terminated.

Table 3- 5 Sequence of connection establishment / termination

Connection establishment / termination	Step	Description
<pre> sequenceDiagram     participant UA as User application     participant ID as Ident device     Note over UA: ①     UA-&gt;&gt;ID: hostGreetings     Note over ID: ②     ID--&gt;&gt;UA: reply     Note over UA: ③     UA-&gt;&gt;ID: heartbeat     Note over ID: ④     ID--&gt;&gt;UA: reply     Note over UA: ⑤     UA-&gt;&gt;ID: hostGoodbye     Note over ID: ⑥     ID--&gt;&gt;UA: reply         </pre>	①	The user application sends the "hostGreetings" command to the Ident device.
	②	The Ident device returns a positive reply frame.
	③	After the connection has been established, the user application communicates with the Ident device. It sends, for example, a heartbeat command at regular intervals.
	④	For each command, the Ident device sends a reply frame.
	⑤	The user application sends the "hostGoodbye" command to terminate the connection.
	⑥	The Ident device sends a positive reply frame. The Ident device then terminates the existing TCP/IP connection.

Note that the Ident device replies with the "ERROR\_INVALID\_READER\_STATUS" error message if the commands are sent without a preceding "hostGreetings" command.

After executing the command, the device blocks connection requests of new clients for 30 seconds. By periodically sending "heartBeat" commands within a period of 30 seconds, you can make sure that no other user applications access the device.

### 3.3.1.1 hostGreetings

Table 3- 6 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

All communication with an Ident device must start with the "hostGreetings" command. Using this command, the device detects the user application connected to the XML interface and its version. In turn, the firmware version of the device is stored in the reply frame of the device. In this way, the XML API can determine which commands are supported by the connected device.

Note that the Ident device replies with the "ERROR\_INVALID\_READER\_STATUS" error message if the commands are sent without a preceding "hostGreetings" command.

Note that up to 20 seconds can elapse until a reply frame is received.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <hostGreetings>
      <readerType> value readerType </readerType> //opt
      <supportedVersions>
        <version> value_version </version>
        <version> value_version </version> //opt
        ...
      </supportedVersions>
    </hostGreetings>
  </cmd>
</frame>
```

//opt  $\triangle$  Optional: The line can be omitted.

## Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <hostGreetings>
      <returnValue>
        <version> value_version </version>
        <configID> value_configID </configID>
      </returnValue>
    </hostGreetings>
  </reply>
</frame>

```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_readerType	Fixed values	SIMATIC_RF610R SIMATIC_RF615R SIMATIC_RF650R SIMATIC_RF680R SIMATIC_RF685R SIMATIC_RF185C SIMATIC_RF186C SIMATIC_RF188C SIMATIC_RF186CI SIMATIC_RF188CI	Device type  If the connected device does not match the value specified, "ERROR_PARAMETER_ILLEGAL_VALUE" is returned.  If this parameter is not specified, the connected device type is not checked.
value_version	Alphanumeric text	Vx.x	Supported XML API protocol version (See section "Compatibility (Page 14)")
value_configID	Alphanumeric text	--	Unique identifier of the transferred configuration. The ID can also be read using the "getConfigVersion" function.

### 3.3.1.2 hostGoodbye

Table 3- 7 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

This command ends communication with the Ident device and terminates the TCP/IP connection.

In the default setting, the device continues working with its current settings. This allows the device to work independently. The data accumulating when the reader is operating independently is stored in the buffer. You will find further information on the buffer or on asynchronous notifications in the section "Principle of operation (Page 8)".

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <hostGoodbye>
  </hostGoodbye>
  </cmd>
</frame>
```

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <hostGoodbye/>
  </reply>
</frame>
```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

#### Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier

### 3.3.1.3 heartBeat

Table 3- 8 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, it is possible to check whether the connection between XML API and the connected Ident device has been interrupted (e.g. wire break) or whether the device is out of operation (e.g. network failure).

After executing the command, the device blocks connection requests of new clients for 30 seconds. By periodically sending "heartBeat" commands within a period of 30 seconds, you can make sure that no other user applications access the device.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <heartBeat/>
  </cmd>
</frame>
```

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <heartBeat/>
  </reply>
</frame>
```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

#### Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier

### 3.3.1.4 setIPConfig

Table 3-9 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V2.2.1	V1.3

With this command, you can change the IP address of the Ident device. If the transferred parameters do not contain any inconsistencies, the device will terminate the connection. This happens as well if the parameters are identical to the previous settings. A reply frame is sent in reply to this command only if there is an error.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <setIPConfig>
      <iPAddress> value_iPAddress </iPAddress> //opt
      <subNetMask> value_subNetMask </subNetMask> //opt
      <gateway> value_gateway </gateway> //opt
      <dHCPEnable> value_dHCPEnable </dHCPEnable> //opt
    </setIPConfig>
  </cmd>
</frame>
```

//opt  $\Delta$  Optional: The line can be omitted.

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <setIPConfig/>
  </reply>
</frame>
```

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".



## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_ipAddress	IP address	1.0.0.1... 255.255.255.254	IP address of the reader Parameter is ignored, if DHCP is activated: value_dHCPEnable = True
value_subNetMask	IP address	1.0.0.1... 255.255.255.254	Subnet mask of the reader Parameter is ignored, if DHCP is activated: value_dHCPEnable = True
value_gateway	IP address	1.0.0.1... 255.255.255.254	Gateway of the reader Parameter is ignored, if DHCP is activated: value_dHCPEnable = True
value_dHCPEnable	Fixed values	True False	Use of DHCP to assign the IP address.

### 3.3.1.5 getIPConfig

Table 3- 10 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V2.2.1	V1.3

With this command, you can read out the current IP address of the Ident device.

## Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <getIPConfig/>
  </cmd>
</frame>
```

## Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getIPConfig>
      <iPAddress> value_iPAddress </iPAddress>
      <subNetMask> value_subNetMask </subNetMask>
      <gateway> value_gateway </gateway>
      <dHCPEnable> value_dHCPEnable </dHCPEnable>
    </getIPConfig/>
  </reply>
</frame>

```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0..4294967295	Unique command identifier
value_iPAddress	IP address	1.0.0.1... 255.255.255.254	IP address of the reader Parameter is ignored, if DHCP is activated: value_dHCPEnable = True
value_subNetMask	IP address	1.0.0.1... 255.255.255.254	Subnet mask of the reader Parameter is ignored, if DHCP is activated: value_dHCPEnable = True
value_gateway	IP address	1.0.0.1... 255.255.255.254	Gateway of the reader Parameter is ignored, if DHCP is activated: value_dHCPEnable = True
value_dHCPEnable	Fixed values	True False	Use of DHCP to assign the IP address.

## 3.3.2 Device settings

This section describes all the commands with which you can control the configuration of the Ident device (reader, communication module).

### 3.3.2.1 setConfiguration

Table 3- 11 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you can transfer a configuration to the Ident device. After the configuration is confirmed, it is activated and stored permanently in the flash memory of the Ident device. As an alternative, you can also load a configuration that was created using the WBM.

Note that up to 20 seconds can elapse until a reply frame is received.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <setConfiguration>
      <configData>
        <![CDATA[value_configData]]>
      </configData>
    </setConfiguration>
  </cmd>
</frame>
```

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <setConfiguration>
      <returnValue>
        <configID> value_configID </configID>
      </returnValue>
    </setConfiguration>
  </reply>
</frame>
```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

**Parameter**

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_ configData	CDATA	--	Configuration data The parameter must be embedded in a CDATA segment.
value_configID	Alphanumeric text	--	Unique identifier of the transferred configuration The ID can also be read out with the "getConfigVersion" command.

**3.3.2.2 getConfiguration**

Table 3- 12 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you can read out the configuration saved in the Ident device. You can also use this configuration, for example, to transfer it to other Ident devices.

**Command**

```
<frame>
  <cmd>
    <id> value_id </id>
    <getConfiguration/>
  </cmd>
</frame>
```

**Reply**

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getConfiguration>
      <returnValue>
        <configID> value_configID </configID>
        <configData>
          <![CDATA[value_configData]]>
        </configData>
      </returnValue>
    </getConfiguration>
  </reply>
</frame>
```

```

    </getConfiguration>
  </reply>
</frame>

```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_ configData	CDATA	--	Configuration data The parameter must be embedded in a CDATA segment.
value_configID	Alphanumeric text	--	Unique identifier of the transferred configuration The ID can also be read out with the "getConfigVersion" command.

### 3.3.2.3 getConfigVersion

Table 3- 13 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you can read out the version ID of the saved configuration.

## Command

```

<frame>
  <cmd>
    <id> value_id </id>
    <getConfigVersion/>
  </cmd>
</frame>

```

## Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getConfigVersion>
      <returnValue>
        <configID> value_configID </configID>
      </returnValue>
    </getConfigVersion>
  </reply>
</frame>
```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique command identifier
value_configID	Alphanumeric text	--	Unique identifier of the transferred configuration

### 3.3.2.4 getActiveConfiguration

Table 3- 14 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V2.0	V1.0	V1.3

With this command, you can request the active parameters with which the Ident device is currently working as a configuration file from the Ident device.

Note that the active parameter values may differ from the stored values.

Note that up to 20 seconds can elapse until a reply frame is received.

## Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <getActiveConfiguration/>
  </cmd>
</frame>
```

## Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getActiveConfiguration>
      <returnValue>
        <configID> value_configID </configID>
        <configData>
          <![CDATA[value_configData]]>
        </configData>
      </returnValue>
    </getActiveConfiguration>
  </reply>
</frame>
```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_ configData	CDATA	--	Configuration data The parameter must be embedded in a CDATA segment.
value_configID	Alphanumeric text	--	Unique identifier of the transferred configuration The ID can also be read out with the "getConfigVersion" command.

### 3.3.2.5 getLogfile

Table 3- 15 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V2.0	V1.0	V1.3

With this command, you can read out the logbook of the Ident device.

Note that up to 20 seconds can elapse until a reply frame is received.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <getLogfile>
      <logType> value_logType </logType>
    </getLogfile>
  </cmd>
</frame>
```

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getLogfile>
      <returnValue>
        <logData>
          <![CDATA[value_configData]]>
        </logData>
      </returnValue>
    </getLogfile>
  </reply>
</frame>
```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".



## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_logType	Fixed values	Diagnosis Logbook Headline ServiceLog	Type of diagnostics file <ul style="list-style-type: none"> <li>Diagnosis (default with RF640R/RF670R): Diagnostics file with logbook entries in CSV format including header. Required with older device versions for reasons of compatibility.</li> <li>Logbook (default): Diagnostics file with logbook entries in CSV format including header. Recommended with newly created applications.</li> <li>Headline: Only returns the header line. Recommended for logbook events.</li> <li>ServiceLog: Service log file in CSV format without header. Siemens technical personnel are responsible for evaluating the log.</li> </ul>
value_logData	CDATA	--	Data logged by the Ident device The parameter must be embedded in a CDATA segment.

## Output format of the "Logbook" diagnostics file

The diagnostics file of the type "Logbook" consists of individual lines of text that are divided into multiple columns by a semicolon ";". A header describing the columns contained is always output as the first line. This formatting enables you to open the file in a spreadsheet program, for example, which improves the readability of the data.

Depending on the Ident device used and its configuration (in the WBM), different diagnostic information is output.

	A	B	C	D	E	F	G	
1	Date_Time	Type	Entry	Command_Type	Readpoint	Call_Param	Return	EPCID_UID
2	2019-06-18T08:11:08.849+00:00	COMMANDS	hostGreetings	CMD		version=V2.0,version	version=V3.0,configl	
3	2019-06-18T08:11:08.849+00:00	COMMANDS	hostGreetings_1	CMD		...=V2.1,version=V2.2,v	...D=5D01041D	
4	2019-06-18T08:11:08.850+00:00	COMMANDS	hostGreetings_2	CMD		...ersion=V3.0		

Figure 3-1 Example of a "Logbook" diagnostics file

A logbook entry can only contain up to 6 kB data. When commands are recorded, multiple logbook entries can therefore be created for a single command. The names of the respective commands and a command-specific consecutive number are inserted in the "Entry" column.

The following table provides an overview of the parameters that can be saved in the logbook entries of the diagnostics file depending on the configuration of your diagnostics.

Table 3- 16 Parameters contained in every logbook entry

Parameter	Type	Values	Description
Date_Time	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2019-06-18T08:11:08.849+00:00
Type	Fixed values	COMMON ERRORS EVENTS COMMANDS FILTER GPIO	Type of entry The configuration of the Ident device (WBM) determines which events/message types are entered in the logbook. They are output accordingly in the diagnostics file.
Entry	Text	--	Text of the logbook entry With a logbook entry of the type "EVENTS", "COMMANDS" and "FILTER", the diagnostics file is supplemented by additional logbook entries (see the following table).

Table 3- 17 Parameters contained in a logbook entry due to the "EVENTS", "COMMANDS" and "FILTER" parameters

Parameter	Type	Values	Description
Command_Type	Fixed values	CMD CMD_CALL CMD_CHN CMD_RPT CMD_CHNRPT CMD_FAIL RPL RPL_CHN RPL_RPT PRL_CHNRPT RPL_FAIL UNKNOWN	Type of the command Is only filled when the "COMMANDS" message type is activated. With all other message types, this column contains no entries.
Readpoint	Text	--	Name of the read point affected by the entry. If no read point can be assigned (e.g. with the XML command "getReader-Status"), this column contains no entries.

Parameter	Type	Values	Description
Call_Param	Text	--	Specification of the command parameters Is only filled when the "COMMANDS" message type is activated. With all other message types, this column contains no entries.
Return	Text	--	Specification of the return values Is only filled when the "COMMANDS" message type is activated.
EPCID_UID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format" Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".

Table 3- 18 Parameters contained in a logbook entry only with RF600

Parameter	Type	Values	Description
TagPC	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters. Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".
RSSI	Decimal value 0...9	0...255	RSSI value
Power	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB
Antenna	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
Polarization	Fixed values	Circular Linear_vertical Linear_horizontal Unknown	Polarization of the antenna With external antennas, the parameter is always unknown.
Channel	Decimal value 0...9	1...50	Channel number on which the transponder was detected.

Parameter	Type	Values	Description
Command_Retry	Decimal value 0...9	0...65535	Number of command repetitions that were needed before the reader received correct data. "0" corresponds to no attempt. The maximum number of attempts is set using the WBM.
Air_Retry	Decimal value 0...9	0...65535	Number of air interface command repetitions that were needed before the reader received correct data. "0" corresponds to no attempt. The maximum number of attempts is specified in the reader firmware and cannot be modified.
FilterData	Fixed values	True False	Indicates whether a filter criterion was received. True: All the data was read or no filter was set. False: Data could not be read.
Inventoried	Decimal value 0...9	0...65535	Indicates how often the transponder was identified via the air interface before it changed to the "Observed" status.
RSSI_Min	Decimal value 0...9	0...255	The lowest RSSI value with which the transponder was detected. This value has no unit and is not directly related to the performance.
RSSI_Max	Decimal value 0...9	0...255	The highest RSSI value with which the transponder was detected. This value has no unit and is not directly related to the performance.

### 3.3.2.6 resetLogfile

Table 3- 19 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V2.0	V1.0	V1.3

With this command, you can delete all logbook entries.

## Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <resetLogfile>
      <logType> value_logType </logType> //opt
    </resetLogfile>
  </cmd>
</frame>
```

//opt  $\hat{=}$  Optional: The line can be omitted.

## Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <resetLogfile/>
  </reply>
</frame>
```

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_logType	Fixed value	Diagnosis Logbook ServiceLog	Type of diagnostics file <ul style="list-style-type: none"> <li>• Diagnosis (default with RF640R/RF670R): Logbook file. Required with older device versions for reasons of compatibility.</li> <li>• Logbook (default): Logbook file. Recommended with newly created applications.</li> <li>• ServiceLog: Service log file. Siemens technical personnel are responsible for evaluating the log.</li> </ul>

### 3.3.2.7 setParameter

Table 3- 20 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V2.0	V1.0	V1.3

With this command, you can set a specific read point/antenna parameter for the Ident device.

Please note that changes using this command are stored in volatile memory in the configuration of the WBM. As a result, the Ident device works with the parameter value specified with "setParameter" but does not display this in the WBM.

Note that up to 20 seconds can elapse until a reply frame is received.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <setParameter>
      <name> value_name </name>
      <value> value_value </value>
      <objType> value_objType </objType> //opt
      <objName> value_objName </objName> //opt
    </setParameter>
  </cmd>
</frame>
```

//opt  $\Delta$  Optional: The line can be omitted.

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <setParameter/>
  </reply>
</frame>
```

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_name	Alphanumeric text	See following table	Name of the supported parameter
value_value	Alphanumeric text	See following table	Parameter value
value_objType	Alphanumeric text	See following table	Specifies the type of the parameter groups to be addressed.
value_objName	Alphanumeric text	See following table	Name of the specific parameter group

## Possible values of the "value\_name" parameter

Table 3- 21 Parameter values for the RF18xC/RF18xCI product family

name	value	objType	objName	Description
Power	2...8	Antenna	Antenna11 Antenna21 Antenna31 Antenna41	Radiated power of the reader antenna in [W] Only relevant when the RF380R reader is connected. 2: 0.5 W 3: 0.75 W 4: 1.0 W 5: 1.25 W 6: 1.5 W 7: 1.75 W 8: 2.0 W

Table 3- 22 Parameter values for the RF600 product family

name	value	objType	objName	Description
Power	0, 5.00...33.00	Antenna	Antenna01 Antenna02 Antenna03 Antenna04	Radiated power of the antenna in [dB] Increment: 0.25 dB
RssiThreshold	0...255	Antenna	Antenna01 Antenna02 Antenna03 Antenna04	RSSI threshold value Transponders with lower RSSI values are not taken into account. This value has no unit and is not directly related to the performance.

name	value	objType	objName	Description
Polarization	Default Circular Linear_vertical Linear_horizontal All	Antenna	Antenna01 Antenna02 Antenna03 Antenna04	Polarization of the antenna  This parameter is only relevant with switchable antennas.  The "Default" value sets the antenna to linear-vertical.
RssiDelta	0...255	Source	See description	Difference for RSSI values  Maximum difference compared with the RSSI value of the transponder with the highest RSSI value that the transponders can have and still be processed.  This value has no unit and is not directly related to the performance.  "objName" must have the name of the corresponding read point.
ModulationScheme	32, 33, 34, 35, 37, 65	General	General	Modulation scheme of the read point  This parameter also specifies which transponder types are identified (ISO 18000-62/-63).

### 3.3.2.8 getParameter

Table 3- 23 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V2.0	V1.0	V1.3

With this command, you can set a specific read point/antenna parameter of the Ident device. The return value contains the value currently being used

### Command

```

<frame>
  <cmd>
    <id> value_id </id>
    <getParameter>
      <name> value_name </name>
      <objType> value_objType </objType>
      <objName> value_objName </objName>
    </getParameter>
  </cmd>
</frame>

```



## Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getParameter>
      <returnValue>
        <value> value_value </value>
      </returnValue>
    </getParameter>
  </reply>
</frame>

```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_name	Alphanumeric text	See following table	Name of the supported parameter
value_value	Alphanumeric text	See following table	Parameter value
value_objType	Alphanumeric text	See following table	Specifies the type of the parameter groups to be addressed.
value_objName	Alphanumeric text	See following table	Name of the specific parameter group

Possible values of the "value\_name" parameter

Table 3- 24 Parameter values for the RF18xC/RF18xCI product family

value_name	value_value	value_objType	value_objName	Description
Power	2...8	Antenna	Antenna11 Antenna21 Antenna31 Antenna41	Transmit power of the reader antenna in [W] Only relevant when the RF380R reader is connected. 2: 0.5 W 3: 0.75 W 4: 1.0 W 5: 1.25 W 6: 1.5 W 7: 1.75 W 8: 2.0 W

Table 3- 25 Parameter values for the RF600 product family

name	value	objType	objName	Description
Power	0, 5.00...33.00	Antenna	Antenna01 Antenna02 Antenna03 Antenna04	Radiated power of the antenna in [dB] Increment: 0.25 dB
RssiThreshold	0...255	Antenna	Antenna01 Antenna02 Antenna03 Antenna04	RSSI threshold value Transponders with lower RSSI values are not taken into account. This value has no unit and is not directly related to the performance.
Polarization	Default Circular Linear_vertical Linear_horizontal All	Antenna	Antenna01 Antenna02 Antenna03 Antenna04	Polarization of the antenna This parameter is only relevant with switchable antennas. The "Default" value sets the antenna to linear-vertical.

name	value	objType	objName	Description
RssiDelta	0...255	Source	See description	Difference for RSSI values Maximum difference compared with the RSSI value of the transponder with the highest RSSI value that the transponders can have and still be processed. This value has no unit and is not directly related to the performance. "objName" must have the name of the corresponding read point.
ModulationScheme	32, 33, 34, 35, 37, 65	General	General	Modulation scheme of the read point This parameter also specifies which transponder types are identified (ISO 18000-62/-63).

### 3.3.2.9 setTime

Table 3- 26 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you can set the internal clock of the Ident device.

### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <setTime>
      <utcTime> value_utcTime </utcTime>
    </setTime>
  </cmd>
</frame>
```

## Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <setTime/>
  </reply>
</frame>
```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique command identifier
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2018-12-24T18:34:56.929+00:00. Note: The reader only accepts time information from 01.01.2000 ... 18.01.2038.

### 3.3.2.10 getTime

Table 3- 27 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you can read out the current time stamp of the internal device clock time.

## Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <getTime/>
  </cmd>
</frame>
```

## Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getTime>
      <returnValue>
        <utcTime> value_utcTime </utcTime>
      </returnValue>
    </getTime>
  </reply>
</frame>

```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffz e.g.: 2018-12-24T18:34:56.929+00:00.

### 3.3.2.11 setIO

Table 3- 28 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you can set the digital outputs of the Ident device.

The general settings for the behavior of the digital outputs such as "Inactivity" or "Reset time" are specified in a configuration using the WBM. You can find detailed information on the configuration of the digital outputs in the WBM sub-section "Digital outputs" of the relevant Ident device manual.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <setIO>
      <outValue> value_outValue </outValue>
    </setIO>
  </cmd>
</frame>
```

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <setIO/>
  </reply>
</frame>
```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_outValue	Characters 0, 1, X	0000.... XXXX... 1111	<p>Each position stands for a digital output of the Ident device:</p> <ul style="list-style-type: none"> <li>• Output0: 1st position (least significant bit right)</li> <li>• Output1: 2nd position</li> <li>• Output2: 3rd position</li> <li>• Output3: 4th position</li> <li>• ...</li> </ul> <p>Depending on the value of the particular position, the corresponding output is set to ON (1) or OFF (0) or remains unchanged (X).</p> <p>Example: A "value_outValue" of "0X11"</p> <ul style="list-style-type: none"> <li>• Sets Output0 to ON</li> <li>• Sets Output1 to ON</li> <li>• Leaves Output2 unchanged</li> <li>• Sets Output3 to OFF</li> </ul> <p>This command outputs a negative reply "ERROR_PARAMETER_OUT_OF_RANGE" if there are more outputs to be set than the Ident device supports. If, for example, the Ident device only supports 2 digital outputs, no output will be set in the example above in which 4 outputs are addressed.</p>

### 3.3.2.12 getIO

Table 3- 29 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you can read out the current states of all digital inputs/outputs of the Ident device.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <getIO/>
  </cmd>
</frame>
```

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getIO>
      <returnValue>
        <inValue> value_inValue </inValue>
        <outValue> value_outValue </outValue>
      </returnValue>
    </getIO>
  </reply>
</frame>
```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".



## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_inValue	Binary characters 0, 1	0000.... 1111	Each position stands for a digital input of the Ident device: <ul style="list-style-type: none"> <li>• Input0: 1st position (least significant bit right)</li> <li>• Input1: 2nd position</li> <li>• Input2: 3rd position</li> <li>• Input3: 4th position</li> <li>• ...</li> </ul> Depending on the value of the particular position, the corresponding input is set to ON (1) or OFF (0). If the Ident device does not support digital inputs/outputs, no value is output.
value_outValue	Binary characters 0, 1	0000.... 1111	Each position stands for an output of the Ident device: <ul style="list-style-type: none"> <li>• Output0: 1st position (least significant bit right)</li> <li>• Output1: 2nd position</li> <li>• Output2: 3rd position</li> <li>• Output3: 4th position</li> <li>• ...</li> </ul> Depending on the value of the particular position, the corresponding output is set to ON (1) or OFF (0). If the Ident device does not support digital inputs/outputs, no value is output.

## 3.3.2.13 resetReader

Table 3- 30 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you can reset the reader to the factory settings or restart the reader.

After a positive reply from the user application, the reader terminates the TCP/IP connection and then performs a reset. Note that, following this, you need to establish the connection again via the user application and restart the reader with the "hostGreetings" command.

From XML API protocol version V3.0, this command is still present for reasons of compatibility. This command has been replaced by the "resetDevice" command.

**Command**

```
<frame>
  <cmd>
    <id> value_id </id>
    <resetReader>
      <resetType> value_resetType </resetType> //opt
    </resetReader>
  </cmd>
</frame>
```

//opt ≙ Optional: The line can be omitted.

**Reply**

Note that the reply frame may not be sent if the reader restarts.

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <resetReader/>
  </reply>
</frame>
```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

**Parameter**

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
resetType	Fixed values	Reset2Factory Reboot Defcfg	Type of reset Reset2Factory: Reset to the factory settings and delete the stored configuration. Reboot (default): Hardware reset of the reader without deleting the stored configuration. After the restart, communication must be re-established. Defcfg: Reset to default configuration. Settings on IP address, user management and log files are retained. The reader is not restarted.

### 3.3.2.14 resetDevice

Table 3- 31 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V3.0	--	V1.3

With this command, you can reset the device to the factory settings.

After a positive reply from the user application, the device terminates the TCP/IP connection and then performs a reset. Note that, following this, you need to establish the connection again via the user application and restart the device with the "hostGreetings" command.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <resetDevice>
      <resetType> value_resetType </resetType> //opt
    </resetDevice>
  </cmd>
</frame>
```

//opt  $\hat{=}$  Optional: The line can be omitted.

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <resetDevice/>
  </reply>
</frame>
```

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

**Parameter**

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
resetType	Fixed values	Reset2Factory Reboot	Type of reset Reset2Factory: Reset to the factory settings and delete the stored configuration. Reboot (default): Hardware reset of the device without deleting the stored configuration. After the restart, communication must be re-established.

**3.3.2.15 getReaderStatus**

Table 3- 32 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you can read out status information from the reader.

From XML API protocol version V3.0, this command is still present for reasons of compatibility. This command has been replaced by the "getDeviceStatus" command.

**Command**

```
<frame>
  <cmd>
    <id> value_id </id>
    <getReaderStatus/>
  </cmd>
</frame>
```

**Reply**

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getReaderStatus>
      <returnValue>
        <readerType> value_readerType </readerType>
        <mLFB> value_mLFB </mLFB>
        <hwVersion> value_hwVersion </hwVersion>
```

```

    <fWVersion> value_fWVersion </fWVersion>
    <serialNumber> value_serialNumber </serialNumber>
    <subVersions>                                     //opt
        <version> value_version </version>
        ...
    </subVersions>
  </returnValue>
</getReaderStatus>
</reply>
</frame>

```

//opt  $\hat{=}$  Optional: The line can be omitted.

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_readerType	Fixed values	SIMATIC_RF610R SIMATIC_RF615R SIMATIC_RF650R SIMATIC_RF680R SIMATIC_RF685R SIMATIC_RF185C SIMATIC_RF186C SIMATIC_RF188C SIMATIC_RF186CI SIMATIC_RF188CI	Device type
value_mLFB	Alphanumeric text	--	Siemens article number of the device Example: 6GT2811-6AA10-0AA0
value_hWVersion	Alphanumeric text	--	Hardware version of the device
value_fWVersion	Alphanumeric text	--	Firmware version of the device
value_serialNumber	Alphanumeric text	--	Serial number of the device
value_version	Alphanumeric text	--	Reader-specific version of components Note that the number of sub-versions can change. In future versions, there may be more or less sub-versions.

### 3.3.2.16 getDeviceStatus

Table 3- 33 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V3.0	--	V1.3

With this command, you can read out status information from the device.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <getReaderStatus/>
  </cmd>
</frame>
```

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getReaderStatus>
      <returnValue>
        <readerType> value_readerType </readerType>
        <mLFB> value_mLFB </mLFB>
        <hWVersion> value_hWVersion </hWVersion>
        <fWVersion> value_fWVersion </fWVersion>
        <serialNumber> value_serialNumber </serialNumber>
        <subVersions> //opt
          <version> value_version </version>
          ...
        </subVersion> //opt
      </returnValue>
    </getReaderStatus>
  </reply>
</frame>
```

//opt ≙ Optional: The line can be omitted.

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_readerType	Fixed values	SIMATIC_RF610R SIMATIC_RF615R SIMATIC_RF650R SIMATIC_RF680R SIMATIC_RF685R SIMATIC_RF185C SIMATIC_RF186C SIMATIC_RF188C SIMATIC_RF186CI SIMATIC_RF188CI	Device type
value_mLFB	Alphanumeric text	--	Siemens article number of the device Example: 6GT2811-6AA10-0AA0
value_hWVersion	Alphanumeric text	--	Hardware version of the device
value_fWVersion	Alphanumeric text	--	Firmware version of the device
value_serialNumber	Alphanumeric text	--	Serial number of the device
value_version	Alphanumeric text	--	Reader-specific version of components Note that the number of sub-versions can change. In future versions, there may be more or less sub-versions.

## 3.3.2.17 getCMReaderStatus

Table 3- 34 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V3.0	--	V1.3

With this command, you can read out status information from the CM/reader. This includes, for example, various statistics counters of the devices currently connected to the communications module. This information can be requested either from a specific device or from all devices.

**Command**

```

<frame>
  <cmd>
    <id> value_id </id>
    <getCMReaderStatus>
      <deviceName> value_deviceName </deviceName> //opt
      <mode> value_mode </mode> //opt
    </getCMReaderStatus>
  </cmd>
</frame>

```

//opt ≙ Optional: The line can be omitted.

**Reply**

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getCMReaderStatus>
      <returnValue>
        <cmReaderStatus>
          <cmReaderStatusData> value_cmReaderStatusData
        </cmReaderStatusData>
        <deviceName> value_deviceName </deviceName>
        <values> //opt
          <value> value_value </value>
          ...
        </values>
      </cmReaderStatus>
      ...
    </returnValue>
  </getCMReaderStatus>
</reply>
</frame>

```

//opt ≙ Optional: The line can be omitted.

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".



## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_ deviceName	Fixed values	Device01 Device02 Device03 Device04	Name of the device that is connected to the respective interface. If this parameter is not set, all interfaces/devices are selected automatically.
value_mode	Fixed values	1 6 235...255	Mode of the status data to be requested. The supported values depend on the reader type assigned to the specified read point. If this parameter is not set, the value "1" is used automatically. The meaning of the values depends on the connected device and the block used. You can find additional information in the relevant block manual.
value_ cmReaderStatus Data	Hexadecimal value 0...9, A...F	--	The raw reply data of the "getCMReader" command as simple byte sequence. It is mode-specific and reader-specific.
value_value	Alphanumeric text	--	Named specific values from the raw reply data. Note that the number of values can change. There may be more or fewer values in future versions.

## 3.3.2.18 getAllSources

Table 3- 35 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you can query the names of all configured read points of the device. You can change the names of the read points in the WBM of the respective device.

### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <getAllSources/>
  </cmd>
</frame>
```

### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getAllSources>
      <returnValue>
        <sourceName> value_sourceName </sourceName>
        ...
        <sourceName> value_sourceName </sourceName> //opt
      </returnValue>
    </getAllSources>
  </reply>
</frame>
```

//opt ≙ Optional: The line can be omitted.

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

### Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").

### 3.3.3 Transponder processing

This section describes all the commands with which you can control the processing of the transponder data. There are two ways of querying transponder data:

- Synchronous transponder command

Commands that return the transponder data in the reply frame.

The reader executes the required action once and then sends back the acquired transponder data.

Algorithms that work with individual transponder commands such as "Read/WritePowerBoost" and "Read/WriteRetry" are active.

- Asynchronous transponder events

"TagEventReports" are sent by the reader to the user application on the reader's initiative.

The transponder data is acquired only by triggers of the read point. The configuration of the reader opens up numerous options for the trigger configuration of a read point.

You specify the message content contained in the reply frames using the "Tag events" in the WBM. You will find more information on the tag events in the WBM sub-section "Communication" of the relevant Ident device manual.

#### 3.3.3.1 editBlackList

Table 3- 36 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V2.0	V1.0	--

With this command, you save EPC IDs to/remove them from the "Black list".

The black list is a filter mechanism with which transponders are filtered out. Transponders whose IDs are stored in the black list are ignored and not processed. Please note that the black list is a circular buffer with a configurable size. If all the entries in the black list are occupied, the next new entry deletes the oldest entry.

You specify the size of the black list buffer in the WBM. You will find more information on the black list in the WBM sub-section "Read points" of the relevant Ident device manual.

#### Command

```
<frame>
<cmd>
  <id> value_id </id>
  <editBlacklist>
    <sourceName> value_sourceName </sourceName>
    <blackListCmd> value_blackListCmd </blackListCmd>
    <tagID> value_tagID </tagID> //opt
    ...
  </editBlacklist>
</cmd>
</frame>
```

```

        <tagID> value_tagID </tagID> //opt
    </editBlacklist>
</cmd>
</frame>

```

//opt  $\triangleq$  Optional: The line can be omitted.

### Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <editBlacklist/>
  </reply>
</frame>

```

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

### Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_blackList-Cmd	Fixed values	Add Add_obs Del Del_all	How "setBlacklist" works: <ul style="list-style-type: none"> <li>• Add: All the following EPC IDs will be saved</li> <li>• Add_obs: All EPC IDs with the "observed" status will be saved in the black list.</li> <li>• Del: All the following EPC IDs will be removed from the black list</li> <li>• Del_all: All entries in the black list will be removed.</li> </ul>
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format" Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".

### 3.3.3.2 getBlackList

Table 3- 37 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V2.0	V1.0	--

With this command, you can read out all EPC IDs currently saved in the "Black list".

The black list is a filter mechanism with which transponders are filtered out. Transponders whose IDs are stored in the black list are ignored and not processed.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <getBlacklist>
      <sourceName> value_sourceName </sourceName>
    </getBlacklist>
  </cmd>
</frame>
```

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getBlacklist>
      <returnValue>
        <tagID> value_tagID </tagID> //opt
        ...
        <tagID> value_tagID </tagID> //opt
      </returnValue>
    </getBlacklist>
  </reply>
</frame>
```

//opt  $\hat{=}$  Optional: The line can be omitted.

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

**Parameter**

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format" Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".

**3.3.3.3 triggerSource**

Table 3- 38 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

Please note that this command was designed for transponder processing using the RF600 readers. This command is also present for RF1xxC for reasons of compatibility, but it is not suitable for use.

With this command, you trigger inventories on the selected read points. In this case, detected transponders are subjected to the smoothing algorithm and can have the different statuses "glimpsed", "observed" and "lost". The data of the identified statuses is sent to the XML user application as "TagEventReport".

The configuration parameters of the read point such as "Smoothing" and the definition of the data of each transponder (tag fields, RSSI value, ...) to be sent are taken from the stored configuration. You set this parameter in the WBM. You will find more information on the parameters in the WBM sub-section "Read points" of the relevant Ident device manual (RF600).

Note that there is no smoothing algorithm with RF1xxC. When a transponder is detected with RF1xxC, an "observed" event is signaled and when a transponder is no longer detected, a "lost" event is signaled.

## Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <triggerSource>
      <sourceName> value_sourceName </sourceName>
      <triggerMode> value_triggerMode </triggerMode> //opt
    </triggerSource>
  </cmd>
</frame>
```

//opt  $\hat{=}$  Optional: The line can be omitted.

## Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <triggerSource/>
  </reply>
</frame>
```

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_ sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_ triggerMode	Fixed values	Single Start Stop	Trigger type With RF600, the duration and quantity can be determined via the WBM. Single (default): Single triggering of the read point. Start: The read point is continuously triggered until a stop command is sent. Stop: Stops triggering of the read point. Note that this command is only effective for previously executed trigger commands. If a continuous trigger was configured with RF600, this command does not affect it.

### 3.3.3.4 readTagIDs

Table 3- 39 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you trigger inventories on the selected read points. A reply frame returns information on which transponders were detected. If no transponder was detected, a positive reply is also returned but it does not contain transponder data.

Note that this command cannot be interrupted with RF600. The number of inventories performed and the time period over which the reader performs inventories depend on the "value\_duration" and "value\_unit" parameters. An acknowledge frame from the reader is only sent after this. You must consider these reply times in the user application. The configuration parameters of the read point (read cycles per trigger, read timeout, ...) are not used. You will find more information on the parameters in the WBM sub-section "Read points" of the relevant Ident device manual.

---

**Note**

**RF600: Filter mechanisms influence the results**

Please note that filter mechanisms defined with RF600 affect the results. With an active filter, only the transponders that have not been filtered out are listed in the reply frame.

You will find more information on the filter mechanisms in the WBM sub-section "Filters" of the relevant Ident device manual.

---

**Note**

**Delay of the reply frames**

Please note that the reply frame of the Ident device may be delayed further due to the configured duration of the command.

---

### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <readTagIDs>
      <sourceName> value_sourceName </sourceName>
      <duration> value_duration </duration> //opt
      <unit> value_unit </unit> //opt
    </readTagIDs>
  </cmd>
</frame>
```

//opt ≙ Optional: The line can be omitted.



## Reply

```

<frame>
  <reply>
    <id>value_id</id>
    <resultCode> 0 </resultCode>
    <readTagIDs>
      <returnValue>
        <tag>
          <tagID> value_tagID </tagID>
          <sourceName> value_sourceName </sourceName> //opt**
          <tagPC> value_tagPC </tagPC> //opt*
          <utcTime> value_utcTime </utcTime> //opt
          <antennaName> value_antennaName </antennaName> //opt*
          <rSSI> value_rSSI </rSSI> //opt*
          <channel> value_channel </channel> //opt*
          <power> value_power </power> //opt*
          <polarization> value_polarization </polarization> //opt*
          <inventoried> value_inventoried </inventoried> //opt*
          <filterDataAvailable> value_filterDataAvailable
          </filterDataAvailable> //opt*
        </tag>
        ...
        <tag> //opt
        ...
        </tag> //opt
      </returnValue>
    </readTagIDs>
  </reply>
</frame>

```

//opt  $\triangleq$  Optional: The line can be omitted. In the reply, the parameter is transferred depending on the configuration settings (Settings - Communication).

\*: Only for the RF600 product families

\*\* : Only for the RF1xxC product families

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName <sup>2)</sup>	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_duration	Decimal value 0...9	0...65535	Duration for which the selected read point will read transponders. If the value "0" is set or the parameter is empty, only one read cycle is executed.
value_unit	Fixed values	Time Count	Specifies the unit for the duration. Time (default): Time in milliseconds Count: Number of inventories
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format" Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".
value_tagPC <sup>1)</sup>	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters. Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2018-12-24T18:34:56.929+00:00.
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
value_rSSI <sup>1)</sup>	Decimal value 0...9	0...255	RSSI value
value_channel <sup>1)</sup>	Decimal value 0...9	1...50	Channel number on which the transponder was detected.
value_power <sup>1)</sup>	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB

Parameter	Type	Values	Description
value_polarization <sup>1)</sup>	Fixed values	Default Circular Linear_vertical Linear_horizontal All	Polarization of the antenna This parameter is only relevant with switchable antennas. The value "Default" $\triangleq$ "Linear_vertical".
value_inventoried <sup>1)</sup>	Decimal value 0...9	0...65535	Indicates how often the transponder was identified via the air interface in this command.
value_filterDataAvailable <sup>1)</sup>	Fixed values	True False	Indicates whether a filter criterion was received. True: All the data was read or no filter was set. False: Data could not be read.

<sup>1)</sup> Only for the RF600 product families

<sup>2)</sup> Only for the RF1xxC product families

### 3.3.3.5 writeTagID

Table 3- 40 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	--

With this command, you write a new ID to a transponder. To ensure clear identification when writing the ID, either there can be only one transponder in the antenna field or the previous ID of the transponder needs to be specified. If there is more than one transponder in the antenna field or if no transponder ID is specified, a negative response is returned.

### Command

```

<frame>
  <cmd>
    <id> value_id </id>
    <writeTagID>
      <sourceName> value_sourceName </sourceName>
      <tagID> value_tagID </tagID> //opt
      <newID> value_newID </newID>
      <idLength> value_idLength </idLength> //opt
      <password> value_password </password> //opt
    </writeTagID>
  </cmd>
</frame>

```

//opt  $\triangleq$  Optional: The line can be omitted.

**Reply**

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <writeTagID>
      <returnValue>
        <tag>
          <tagID> value_tagID </tagID>
          <tagPC> value_tagPC </tagPC> //opt
          <utcTime> value_utcTime </utcTime> //opt
          <antennaName> value_antennaName </antennaName> //opt
          <rSSI> value_rSSI </rSSI> //opt
          <channel> value_channel </channel> //opt
          <power> value_power </power> //opt
          <polarization> value_polarization </polarization> //opt
          <commandRetry> value_commandRetry </commandRetry> //opt
          <airRetry> value_airRetry </airRetry> //opt
          <filterDataAvailable> value_filterDataAvailable
          </filterDataAvailable> //opt
        </tag>
      </returnValue>
    </writeTagIDs>
  </reply>
</frame>

```

//opt ≙ Optional: The line can be omitted. In the reply, the parameter is transferred depending on the configuration settings (Settings - Communication).

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format". This function accesses all transponders with this ID. If this parameter is empty or not even transferred, the function applies to all transponders. Nevertheless only one transponder is then permitted in the antenna field. Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".
value_newID	Hexadecimal value 0...9, A...F	--	New EPC-ID in "RAW Hex Data Format" to be written to the transponder. Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".
value_idLength	Decimal value 0...9	16, 32, 48...496	Length of the new EPC-ID in bits. If this parameter is set, the reader checks the correct length of "value_newID". If the check fails, a negative reply is sent. Without this parameter, the new EPC-ID is only checked if the length is a multiple of 16 bits.
value_password	Hexadecimal value 0...9, A...F	00000000.... FFFFFFFF	Access password for the transponder This parameter does not need to be defined if there is no password protection activated for the transponder.
value_tagPC	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters. Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2018-12-24T18:34:56.929+00:00.

Parameter	Type	Values	Description
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
value_rSSI	Decimal value 0...9	0...255	RSSI value
value_channel	Decimal value 0...9	1...50	Channel number on which the transponder was detected.
value_power	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB
value_polarization	Fixed values	Default Circular Linear_vertical Linear_horizontal All	Polarization of the antenna This parameter is only relevant with switchable antennas. The value "Default" $\hat{=}$ "Linear_vertical".
value_commandRetry	Decimal value 0...9	0...65535	Number of times the reader repeated commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of attempts is set using the WBM.
value_airRetry	Decimal value 0...9	0...65535	Number of times the reader repeated air interface commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of air interface commands is specified in the reader firmware and cannot be modified.
value_filterDataAvailable	Fixed values	True False	Indicates whether a filter criterion was received. True: All the data was read or no filter was set. False: Data could not be read.

### 3.3.3.6 getObservedTagIDs

Table 3- 41 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V2.0	V1.0	V1.3

Please note that this command was designed for transponder processing using the RF600 readers. This command is also present for RF1xxC for reasons of compatibility, but it is not suitable for use.

This command can be used in the following scenarios.

#### Scenario 1: No inventories are currently being performed.

In this case, you trigger inventories on the selected read points with this command. A reply frame returns information on which transponders were detected.

In contrast to the "readTagIDs" command, the smoothing algorithm of the selected read point is also applied with this command. The read point must therefore take enough inventories until a transponder adopts the "observed" status. This can be achieved by selecting the suitable parameter values or by triggering/starting the read point in good time before the command.

If no transponder with the "observed" status was detected, a positive reply is also returned but it does not contain transponder data.

To be reported, a transponder must have had the "observed" status at least once at the start of or during the command execution time. The transponder is reported even if it returned temporarily to the "lost" status.

Note that this command cannot be interrupted with RF600. The number of inventories performed and the time period over which the reader performs inventories depend on the "value\_duration" and "value\_unit" parameters. An acknowledge frame from the reader is only sent after this. You must consider these reply times in the user application. The configuration parameters of the read point (read cycles per trigger, read timeout, ...) are not used. You will find more information on the parameters in the WBM sub-section "Read points" of the relevant Ident device manual.

#### Scenario 2: Inventories are already being performed.

In this case, the reader is already performing inventories. This can take place, for example, by means of a connected light barrier that is configured accordingly. If this command is then executed, all transponders that have the "observed" status are output.

---

#### Note

##### RF600: Filter mechanisms influence the results

Please note that defined filter mechanisms affect the results. With an active filter, only the transponders that have not been filtered out are listed in the reply frame.

You will find more information on the filter mechanisms in the WBM sub-section "Filters" of the relevant Ident device manual.

---

**Note****Delay of the reply frames**

Please note that the reply frame of the Ident device may be delayed further due to the configured duration of the command.

---

**Command**

```
<frame>
  <cmd>
    <id> value_id </id>
    <getObservedTagIDs>
      <sourceName> value_sourceName </sourceName>
      <duration> value_duration </duration> //opt
      <unit> value_unit </unit> //opt
    </getObservedTagIDs>
  </cmd>
</frame>
```

//opt  $\triangleq$  Optional: The line can be omitted.

**Reply**

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getObservedTagIDs>
      <returnValue>
        <tag>
          <tagID> value_tagID </tagID>
          <sourceName> value_sourceName </sourceName> //opt**
          <tagPC> value_tagPC </tagPC> //opt*
          <utcTime> value_utcTime </utcTime> //opt
          <antennaName> value_antennaName </antennaName> //opt*
          <rSSI> value_rSSI </rSSI> //opt*
          <channel> value_channel </channel> //opt*
          <power> value_power </power> //opt*
          <polarization> value_polarization </polarization> //opt*
          <inventoried> value_inventoried </inventoried> //opt*
          <filterDataAvailable> value_filterDataAvailable
          </filterDataAvailable> //opt*
        </tag>
        ...
      </tag> //opt
      ...
    </tag> //opt
```



```

    </returnValue>
  </getObservedTagIDs>
</reply>
</frame>

```

//opt  $\triangle$  Optional: The line can be omitted.

\*: Only for the RF600 product families

\*\*: Only for the RF1xxC product families

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName <sup>2)</sup>	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_duration	Decimal value 0...9	0...65535	Duration for which the selected read point will read transponders. If the value "0" is set or the parameter is empty, only one read cycle is executed.
value_unit	Fixed values	Time Count	Specifies the unit for the duration. Time (default): Time in milliseconds Count: Number of inventories
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format" Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".
value_tagPC <sup>1)</sup>	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters. Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2018-12-24T18:34:56.929+00:00.

Parameter	Type	Values	Description
value_antennaName <sup>1)</sup>	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
value_rSSI <sup>1)</sup>	Decimal value 0...9	0...255	RSSI value
value_channel <sup>1)</sup>	Decimal value 0...9	1...50	Channel number on which the transponder was detected.
value_power <sup>1)</sup>	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB
value_polarization <sup>1)</sup>	Fixed values	Default Circular Linear_vertical Linear_horizontal All	Polarization of the antenna This parameter is only relevant with switchable antennas. "Default" ≙ "Linear_vertical".
value_inventoried <sup>1)</sup>	Decimal value 0...9	0...65535	Indicates how often the transponder was identified via the air interface in this command.
value_filterDataAvailable <sup>1)</sup>	Fixed values	True False	Indicates whether a filter criterion was received. True: All the data was read or no filter was set. False: Data could not be read.

- 1) Only for the RF600 product families
- 2) Only for the RF1xxC product families

### 3.3.3.7 readTagMemory

Table 3- 42 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you can read out data from the selected transponder. If the specified UID or EPC ID is not found, the command is executed for all transponders detected by the read point.

#### Note

##### RF600: Filter mechanisms may influence the results

If no EPC ID is specified, the results are influenced by all defined filter mechanisms. In this case, only the transponders that were not filtered out appear in the reply frame. If an EPC ID is specified, the results are not filtered.

You will find more information on the filter mechanisms in the WBM sub-section "Filters" of the relevant Ident device manual.

The reply frame contains the IDs of all detected transponders with the information as to whether the requested data for the transponder could be read or not. If no transponder was detected, a negative reply without transponder data is returned.

### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <readTagMemory>
      <sourceName> value_sourceName </sourceName>
      <tagID> value_tagID </tagID> //opt
      <password> value_password </password> //opt*
      <tagField>
        <bank> value_bank </bank> //opt*
        <startAddress> value_startAddress </startAddress>
        <dataLength> value_dataLength </dataLength>
      </tagField>
      ...
      <tagField> //opt
        ...
      </tagField> //opt
    </readTagMemory>
  </cmd>
</frame>
```

\*: Only for the RF600 product families

//opt ≙ Optional: The line can be omitted.

## Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <readTagMemory>
      <returnValue>
        <tag>
          <tagID> value_tagID </tagID>
          <sourceName> value_sourceName </sourceName> //opt**
          <tagPC> value_tagPC </tagPC> //opt*
          <success> value_success </success>
          <utcTime> value_utcTime </utcTime> //opt
          <antennaName> value_antennaName </antennaName> //opt*
          <rSSI> value_rSSI </rSSI> //opt*
          <channel> value_channel </channel> //opt*
          <power> value_power </power> //opt*
          <polarization> value_polarization </polarization> //opt*
          <commandRetry> value_commandRetry </commandRetry> //opt*
          <airRetry> value_airRetry </airRetry> //opt*
          <filterDataAvailable> value_filterDataAvailable
          </filterDataAvailable> //opt*
          <tagField> //opt
            <bank> value_bank </bank> //opt*
            <startAddress> value_startAddress </startAddress>
            <dataLength> value_dataLength </dataLength>
            <data> value_data </data>
          </tagField> //opt
          ...
          <tagField> //opt
            ...
          </tagField> //opt
        </tag>
        ...
      <tag> //opt
        ...
      </tag> //opt
    </returnValue>
  </readTagMemory>
</reply>
</frame>

```

//opt ≙ Optional: The line can be omitted. In the reply, the parameter is transferred depending on the configuration settings (engineering/communication).

\*: Only for the RF600 product families

\*\* : Only for the RF1xxC product families

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format". This function accesses all transponders with this ID. If this parameter is empty or not even transferred, the function applies to all transponders. Nevertheless only one transponder is then permitted in the antenna field. Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".
value_tagPC <sup>1)</sup>	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters. Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".
value_bank	--	0...3	Memory bank of the transponder <ul style="list-style-type: none"> <li>• 0: Reserved</li> <li>• 1: EPC</li> <li>• 2: TID</li> <li>• 3: USER MEMORY</li> </ul> If no value is selected, the memory bank 3 "USER MEMORY" is used automatically with RF600. With RF1xxC, this parameter is not relevant.
value_startAddress	Decimal value 0...9	0...4294967295	Start address of the first byte in the memory bank where reading will start.
value_dataLength	Decimal value 0...9	1...1024	Number of bytes to be read.
value_password <sup>1)</sup>	Hexadecimal value 0...9, A...F	00000000... FFFFFFFF	Access password for the transponder This parameter does not need to be defined if there is no password protection activated for the transponder.
value_success	Fixed values	True False	Indicates whether the command was successful for this transponder. True: Command successful False: Command unsuccessful

Parameter	Type	Values	Description
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffz e.g.: 2018-12-24T18:34:56.929+00:00.
value_antennaName <sup>1)</sup>	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
value_rSSI <sup>1)</sup>	Decimal value 0...9	0..255	RSSI value
value_channel <sup>1)</sup>	Decimal value 0...9	1...50	Channel number on which the transponder was detected.
value_power <sup>1)</sup>	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB
value_polarization <sup>1)</sup>	Fixed values	Default Circular Linear_vertical Linear_horizontal All	Polarization of the antenna This parameter is only relevant with switchable antennas. "Default" ≙ "Linear_vertical".
value_commandRetry <sup>1)</sup>	Decimal value 0...9	0...65535	Number of times the reader repeated commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of attempts is set using the WBM.
value_airRetry <sup>1)</sup>	Decimal value 0...9	0...65535	Number of times the reader repeated air interface commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of air interface commands is specified in the reader firmware and cannot be modified.

Parameter	Type	Values	Description
value_ filterDataAvailable <sup>1)</sup>	Fixed values	True False	Indicates whether a filter criterion was received. True: All the data was read or no filter was set. False: Data could not be read.
value_data	Hexadecimal value 0...9, A...F	--	Data that was read. Each byte is represented by two hexadecimal characters. Example: The byte sequence "0x12, 0x34, 0xA3" is represented as the character string "1234A3" in the "value_data" parameter. "value_dataLength" is 3 in this example. If the transponder is detected but the data cannot be read (e.g. if the transponder has no user memory as required), this field remains empty. "value_success" is then also set to "False".

<sup>1)</sup> Only for the RF600 product families

<sup>2)</sup> Only for the RF1xxC product families

### 3.3.3.8 writeTagMemory

Table 3- 43 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

With this command, you write data to the selected transponder. If the specified UID or EPC ID is not found, the command is executed for all transponders detected by the read point. If the specified UID/EPC ID is not found, the command is executed for all transponders detected by the read point.

#### Note

##### RF600: Filter mechanisms may influence the results

If no EPC ID is specified, the results are influenced by all defined filter mechanisms. In this case, only the transponders that were not filtered out appear in the reply frame. If an EPC ID is specified, the results are not filtered.

You will find more information on the filter mechanisms in the WBM sub-section "Filters" of the relevant Ident device manual.

The reply frame contains the IDs of all identified transponders. The "value\_success" parameter indicates for each transponder whether or not the command was successful. If no transponder was identified, a negative reply is returned.

**Command**

```

<frame>
  <cmd>
    <id> value_id </id>
    <writeTagMemory>
      <sourceName> value_sourceName </sourceName>
      <tagID> value_tagID </tagID> //opt
      <password> value_password </password> //opt
      <tagField>
        <bank> value_bank </bank> //opt
        <startAddress> value_startAddress </startAddress>
        <dataLength> value_dataLength </dataLength>
        <data> value_data </data> //opt
      </tagField>
      ...
      <tagField> //opt
      ...
      </tagField> //opt
    </writeTagMemory>
  </cmd>
</frame>

```

//opt ≙ Optional: The line can be omitted.

\*: Only for the RF600 product families

**Reply**

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <writeTagMemory>
      <returnValue>
        <tag>
          <tagID> value_tagID </tagID>
          <sourceName> value_sourceName </sourceName> //opt**
          <tagPC> value_tagPC </tagPC> //opt*
          <success> value_success </success>
          <utcTime> value_utcTime </utcTime> //opt
          <antennaName> value_antennaName </antennaName> //opt
          <rSSI> value_rSSI </rSSI> //opt
          <channel> value_channel </channel> //opt*
          <power> value_power </power> //opt*
          <polarization> value_polarization </polarization> //opt*
          <commandRetry> value_commandRetry </commandRetry> //opt*
          <airRetry> value_airRetry </airRetry> //opt*
          <filterDataAvailable> value_filterDataAvailable
          </filterDataAvailable> //opt*
        </tag>
      </returnValue>
    </writeTagMemory>
  </reply>
</frame>

```



```

        </tag>
        ...
        <tag>           //opt
        ...
        </tag>         //opt
    </returnValue>
</writeTagMemory>
</reply>
</frame>

```

//opt  $\triangleq$  Optional: The line can be omitted.

\*: Only for the RF600 product families

\*\* : Only for the RF1xxC product families

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format". This function accesses all transponders with this ID.  If this parameter is empty or not even transferred, the function applies to all transponders. Nevertheless only one transponder is then permitted in the antenna field.  Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468  For additional information, refer to the "EPC Global Specification".
value_tagPC <sup>1)</sup>	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters.  Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".

Parameter	Type	Values	Description
value_bank	--	0...3	Memory bank of the transponder <ul style="list-style-type: none"> <li>• 0: Reserved</li> <li>• 1: EPC</li> <li>• 2: TID</li> <li>• 3: USER MEMORY</li> </ul> If no value is selected, the memory bank 3 "USER MEMORY" is used automatically with RF600. With RF1xC, this parameter is not relevant.
value_startAddress	Decimal value 0...9	0...4294967295	Start address of the first byte in the memory bank where writing will start.
value_dataLength	Decimal value 0...9	1...1024	Number of bytes to be written. The reader checks the correct length of "value_data". If the check is negative, a negative reply is sent.
value_data	Hexadecimal value 0...9, A...F	--	Data that should be written. Each byte is represented by two hexadecimal characters. Example: The byte sequence "0x12, 0x34, 0xA3" is represented as the character string "1234A3" in the "value_data" parameter. "value_dataLength" is 3 in this example. If the transponder is detected but the data cannot be written (e.g. if the transponder has no user memory as required), this field remains empty. "value_success" is then also set to "False".
value_password	Hexadecimal value 0...9, A...F	00000000... FFFFFFFF	Access password for the transponder This parameter does not need to be defined if there is no password protection activated for the transponder.
value_success	Fixed values	True False	Indicates whether the command was successful for this transponder. True: Command successful False: Command unsuccessful
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2018-12-24T18:34:56.929+00:00.

Parameter	Type	Values	Description
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
value_rSSI	Decimal value 0...9	0...255	RSSI value
value_channel <sup>1)</sup>	Decimal value 0...9	1...50	Channel number on which the transponder was detected.
value_power <sup>1)</sup>	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB
value_polarization <sup>1)</sup>	Fixed values	Default Circular Linear_vertical Linear_horizontal All	Polarization of the antenna This parameter is only relevant with switchable antennas. "Default" $\triangleq$ "Linear_vertical".
value_commandRetry <sup>1)</sup>	Decimal value 0...9	0...65535	Number of times the reader repeated commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of attempts is set using the WBM.
value_airRetry <sup>1)</sup>	Decimal value 0...9	0...65535	Number of times the reader repeated air interface commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of air interface commands is specified in the reader firmware and cannot be modified.
value_filterDataAvailable <sup>1)</sup>	Fixed values	True False	Indicates whether a filter criterion was received. True: All the data was read or no filter was set. False: Data could not be read.

<sup>1)</sup> Only for the RF600 product families

<sup>2)</sup> Only for the RF1xxC product families

### 3.3.3.9 readTagField

Table 3- 44 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	--

With this command, you can read out data from the specific area of the selected transponder. The address of the data area is specified by the name of a tag field. The tag field and the name of the field are specified using the WBM. If the specified EPC ID is not found, the command is executed for all transponders detected by the read point.

**Note**

**RF600: Filter mechanisms may influence the results**

If no EPC ID is specified, the results are influenced by all defined filter mechanisms. In this case, only the transponders that were not filtered out appear in the reply frame. If an EPC ID is specified, the results are not filtered.

You will find more information on the filter mechanisms in the WBM sub-section "Filters" of the relevant Ident device manual.

The reply frame contains the IDs of all detected transponders with the information as to whether the requested data for the transponder could be read or not. If no transponder was detected, a negative reply without transponder data is returned.

### Command

```

<frame>
  <cmd>
    <id> value_id </id>
    <readTagField>
      <sourceName> value_sourceName </sourceName>
      <tagID> value_tagID </tagID> //opt
      <password> value_password </password> //opt
      <tagField>
        <fieldName> value_fieldName </fieldName>
      </tagField>
      ...
      <tagField> //opt
        ...
      </tagField> //opt
    </readTagField>
  </cmd>
</frame>

```

//opt ≙ Optional: The line can be omitted.

## Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <readTagField>
      <returnValue>
        <tag>
          <tagID> value_tagID </tagID>
          <tagPC> value_tagPC </tagPC> //opt
          <success> value_success </success>
          <utcTime> value_utcTime </utcTime> //opt
          <antennaName> value_antennaName </antennaName> //opt
          <rSSI> value_rSSI </rSSI> //opt
          <channel> value_channel </channel> //opt
          <power> value_power </power> //opt
          <polarization> value_polarization </polarization> //opt
          <commandRetry> value_commandRetry </commandRetry> //opt
          <airRetry> value_airRetry </airRetry> //opt
          <filterDataAvailable> value_filterDataAvailable
          </filterDataAvailable> //opt
          <tagField> //opt
            <data> value_data </data>
          </tagField> //opt
          ...
          <tagField> //opt
            ...
          </tagField> //opt
        </tag>
        ...
        <tag> //opt
          ...
        </tag> //opt
      </returnValue>
    </readTagField>
  </reply>
</frame>

```

**//opt**  $\triangleq$  Optional: The line can be omitted. In the reply, the parameter is transferred depending on the configuration settings (engineering/communication).

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format". This function accesses all transponders with this ID. If this parameter is empty or not even transferred, the function applies to all transponders. Nevertheless only one transponder is then permitted in the antenna field. Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".
value_tagPC	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters. Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".
value_fieldName	Hexadecimal value 0...9, A...F	--	Name of tag field Is specified using the WBM.
value_password	Hexadecimal value 0...9, A...F	00000000... FFFFFFFF	Access password for the transponder This parameter does not need to be defined if there is no password protection activated for the transponder.
value_success	Fixed values	True False	Indicates whether the command was successful for this transponder. True: Command successful False: Command unsuccessful
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2018-12-24T18:34:56.929+00:00.
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
value_rSSI	Decimal value 0...9	0..255	RSSI value

Parameter	Type	Values	Description
value_channel	Decimal value 0...9	1...50	Channel number on which the transponder was detected.
value_power	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB
value_polarization	Fixed values	Default Circular Linear_vertical Linear_horizontal All	Polarization of the antenna This parameter is only relevant with switchable antennas. "Default" $\triangleq$ "Linear_vertical".
value_commandRetry	Decimal value 0...9	0...65535	Number of times the reader repeated commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of attempts is set using the WBM.
value_airRetry	Decimal value 0...9	0...65535	Number of times the reader repeated air interface commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of air interface commands is specified in the reader firmware and cannot be modified.
value_filterDataAvailable	Fixed values	True False	Indicates whether a filter criterion was received. True: All the data was read or no filter was set. False: Data could not be read.
value_data	Hexadecimal value 0...9, A...F	--	Data that should be read. Each byte is represented by two hexadecimal characters. Example: The byte sequence "0x12, 0x34, 0xA3" is represented as the character string "1234A3" in the "value_data" parameter. "value_dataLength" is 3 in this example. If the transponder is detected but the data cannot be read (e.g. if the transponder has no user memory as required), this field remains empty. "value_success" is then also set to "False".

### 3.3.3.10 writeTagField

Table 3- 45 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	--

With this command, you write data to the specific area of the transponder. The address of the data area is specified by the name of a tag field. The tag field and the name of the field are specified using the WBM. If the specified EPC ID is not found, the command is executed for all transponders detected by the read point.

**Note**

**RF600: Filter mechanisms may influence the results**

If no EPC ID is specified, the results are influenced by all defined filter mechanisms. In this case, only the transponders that were not filtered out appear in the reply frame. If an EPC ID is specified, the results are not filtered.

You will find more information on the filter mechanisms in the WBM sub-section "Filters" of the relevant Ident device manual.

The reply frame contains the IDs of all identified transponders. The "value\_success" parameter indicates for each transponder whether or not the command was successful. If no transponder was identified, a negative reply is returned.

### Command

```

<frame>
  <cmd>
    <id> value_id </id>
    <writeTagField>
      <sourceName> value_sourceName </sourceName>
      <tagID> value_tagID </tagID> //opt
      <password> value_password </password> //opt
      <tagField>
        <fieldName> value_fieldName </fieldName>
        <data> value_data </data>
      </tagField>
      ...
    </tagField> //opt
    ...
  </tagField> //opt
</writeTagField>
</cmd>
</frame>

```

//opt ≙ Optional: The line can be omitted.



## Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <writeTagField>
      <returnValue>
        <tag>
          <tagID> value_tagID </tagID>
          <tagPC> value_tagPC </tagPC> //opt
          <success> value_success </success>
          <utcTime> value_utcTime </utcTime> //opt
          <antennaName> value_antennaName </antennaName> //opt
          <rSSI> value_rSSI </rSSI> //opt
          <channel> value_channel </channel> //opt
          <power> value_power </power> //opt
          <polarization> value_polarization </polarization> //opt
          <commandRetry> value_commandRetry </commandRetry> //opt
          <airRetry> value_airRetry </airRetry> //opt
          <filterDataAvailable> value_filterDataAvailable
          </filterDataAvailable> //opt
        </tag>
        ...
        <tag> //opt
        ...
        </tag> //opt
      </returnValue>
    </writeTagField>
  </reply>
</frame>

```

**//opt**  $\triangleq$  Optional: The line can be omitted. In the reply, the parameter is transferred depending on the configuration settings (engineering/communication).

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format". This function accesses all transponders with this ID. If this parameter is empty or not even transferred, the function applies to all transponders. Nevertheless only one transponder is then permitted in the antenna field. Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".
value_tagPC	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters. Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".
value_fieldName	Hexadecimal value 0...9, A...F	--	Name of tag field Is specified using the WBM.
value_password	Hexadecimal value 0...9, A...F	00000000... FFFFFFFF	Access password for the transponder This parameter does not need to be defined if there is no password protection activated for the transponder.
value_data	Hexadecimal value 0...9, A...F	--	Data that should be written. Each byte is represented by two hexadecimal characters. Example: The byte sequence "0x12, 0x34, 0xA3" is represented as the character string "1234A3" in the "value_data" parameter. "value_dataLength" is 3 in this example. If the transponder is detected but the data cannot be written (e.g. if the transponder has no user memory as required), this field remains empty. "value_success" is then also set to "False".

Parameter	Type	Values	Description
value_success	Fixed values	True False	Indicates whether the command was successful for this transponder. True: Command successful False: Command unsuccessful
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffz e.g.: 2018-12-24T18:34:56.929+00:00.
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
value_rSSI	Decimal value 0...9	0..255	RSSI value
value_channel	Decimal value 0...9	1...50	Channel number on which the transponder was detected.
value_power	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB
value_polarization	Fixed values	Default Circular Linear_vertical Linear_horizontal All	Polarization of the antenna This parameter is only relevant with switchable antennas. "Default" ≙ "Linear_vertical".
value_commandRetry	Decimal value 0...9	0...65535	Number of times the reader repeated commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of attempts is set using the WBM.
value_airRetry	Decimal value 0...9	0...65535	Number of times the reader repeated air interface commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of air interface commands is specified in the reader firmware and cannot be modified.
value_filterDataAvailable	Fixed values	True False	Indicates whether a filter criterion was received. True: All the data was read or no filter was set. False: Data could not be read.

### 3.3.3.11 killTag

Table 3- 46 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	--

With this command, you disable the selected transponder. If the specified EPC ID is not found, the command is executed for all transponders detected by the read point.

**Note**

**RF600: Filter mechanisms may influence the results**

If no EPC ID is specified, the results are influenced by all defined filter mechanisms. In this case, only the transponders that were not filtered out appear in the reply frame. If an EPC ID is specified, the results are not filtered.

You will find more information on the filter mechanisms in the WBM sub-section "Filters" of the relevant Ident device manual.

The reply frame contains the IDs of all identified transponders. The "value\_success" parameter indicates for each transponder whether or not the command was successful. If no transponder was identified, a negative reply is returned.

### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <killTag>
      <sourceName> value_sourceName </sourceName>
      <tagID> value_tagID </tagID> //opt
      <password> value_password </password>
    </killTag>
  </cmd>
</frame>
```

//opt ≙ Optional: The line can be omitted.

## Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <killTag>
      <returnValue>
        <tag>
          <tagID> value_tagID </tagID>
          <tagPC> value_tagPC </tagPC> //opt
          <success> value_success </success>
          <utcTime> value_utcTime </utcTime> //opt
          <antennaName> value_antennaName </antennaName> //opt
          <rSSI> value_rSSI </rSSI> //opt
          <channel> value_channel </channel> //opt
          <power> value_power </power> //opt
          <polarization> value_polarization </polarization> //opt
          <commandRetry> value_commandRetry </commandRetry> //opt
          <airRetry> value_airRetry </airRetry> //opt
          <filterDataAvailable> value_filterDataAvailable
          </filterDataAvailable> //opt
        </tag>
        ...
        <tag> //opt
        ...
        </tag> //opt
      </returnValue>
    </killTag>
  </reply>
</frame>

```

**//opt**  $\triangleq$  Optional: The line can be omitted. In the reply, the parameter is transferred depending on the configuration settings (engineering/communication).

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format". This function accesses all transponders with this ID. If this parameter is empty or not even transferred, the function applies to all transponders. Nevertheless only one transponder is then permitted in the antenna field. Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".
value_tagPC	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters. Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".
value_password	Hexadecimal value 0...9, A...F	00000000... FFFFFFFF	Disable the kill password of the transponder
value_success	Fixed values	True False	Flag to indicate whether or not the command was successful for this transponder. True: Command successful False: Command unsuccessful
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2018-12-24T18:34:56.929+00:00.
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
value_rSSI	Decimal value 0...9	0..255	RSSI value
value_channel	Decimal value 0...9	1...50	Channel number on which the transponder was detected.

Parameter	Type	Values	Description
value_power	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB
value_polarization	Fixed values	Default Circular Linear_vertical Linear_horizontal All	Polarization of the antenna This parameter is only relevant with switchable antennas. "Default" $\triangleq$ "Linear_vertical".
value_commandRetry	Decimal value 0...9	0...65535	Number of times the reader repeated commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of attempts is set using the WBM.
value_airRetry	Decimal value 0...9	0...65535	Number of times the reader repeated air interface commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of air interface commands is specified in the reader firmware and cannot be modified.
value_filterDataAvailable	Fixed values	True False	Indicates whether a filter criterion was received. True: All the data was read or no filter was set. False: Data could not be read.

### 3.3.3.12 lockTagBank

Table 3- 47 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	--

With this command, you lock the corresponding memory area of the selected transponder. If the specified EPC ID is not found, the command is executed for all transponders detected by the read point.

#### Note

##### RF600: Filter mechanisms may influence the results

If no EPC ID is specified, the results are influenced by all defined filter mechanisms. In this case, only the transponders that were not filtered out appear in the reply frame. If an EPC ID is specified, the results are not filtered.

You will find more information on the filter mechanisms in the WBM sub-section "Filters" of the relevant Ident device manual.

The reply frame contains the IDs of all identified transponders. The "value\_success" parameter indicates for each transponder whether or not the command was successful. If no transponder was identified, a negative reply is returned.

**Structure of the parameters "epcGen2LockAction" and "epcGen2LockMask"**

The first row of the table ('Bit') indicates the bit positions of the action and the masking values. The masking and action values are specified first with the MSB.

Further information on the parameters can be found in the "EPCglobal Specification (<http://www.gs1.org>)".

Table 3- 48 Structure of the parameters "epcGen2LockAction" and "epcGen2LockMask"

Bank	Kill PWD		Access PWD		EPC Memory		TID Memory		User Memory	
Bit	9	8	7	6	5	4	3	2	1	0
Masking	s/w	s/w	s/w	s/w	s/w	s/w	s/w	s/w	s/w	s/w
Action	r/w	p	r/w	p	w	p	w	p	w	p

Explanations on the table:

- The masking value specifies which action value bit is to be set as s/w as shown in the table above ("skip/write" with "skip=0" and "write=1").
- The action value specifies which lock should be defined for each memory bank.
  - "r/w" locks the password for read and write access.
  - "w" locks the password for write access (read access permitted).
  - "p" specifies a permanent lock.

The following tables show possible combinations of the "r/w" and "w" action values with/without lock ("p") and the respective meaning for a memory bank.

The transponder is in the "open" status when it is identified and in the "secured" status when its access password is verified.

Table 3- 49 Possible combinations of the "w" action values

w	p	Description
0	0	Write access to the particular memory bank is possible from both the "open" and the "secured" status.
0	1	Write access to the particular memory bank is permanently possible from both the "open" and the "secured" status; it can never be locked.
1	0	Write access to the particular memory bank is possible from the "secured" status but not from the "open" status.
1	1	No write access to the particular memory bank is possible.



Table 3- 50 Possible combinations of the "r/w" action values

r/w	p	Description
0	0	Read and write access to the particular password is possible from both the "open" and the "secured" status.
0	1	Read and write access to the particular password is permanently possible from both the "open" and the "secured" status; it can never be locked.
1	0	Read and write access to the particular password is possible from the "secured" status but not from the "open" status.
1	1	No read or write access to the particular memory bank is possible.

### Example

Bank	Kill PWD	Access PWD	EPC Memory	TID Memory	User Memory	Hex string value
Masking	(00) 11	11	11	00	00	3F0
Action	(00) 10	10	10	00	00	2A0

In this example, the masking value is "lockMask = 11 1111 0000" (hex: 3F0). As a result, only write access to the memory locations "Kill", "Access" and "EPC" is possible. The action value "lockAction = 10 1010 0000" (hex: 2A0) has the following effects:

- Kill password

Read and write access is possible from the "secured" status, but not from the "open" status. The Access password for the transponder must be known before the Kill password can be read or modified.

- EPC memory bank

Write access is possible from the "secured" status, but not from the "open" status. The Access password must be known before a new ID can be written to the transponder. You will find more detailed information on "epcGen2LockAction" and "epcGen2LockMask" in the "EPC Radio Frequency Identity Protocols Standard Specification".

### Command

```

<frame>
  <cmd>
    <id> value_id </id>
    <lockTagBank>
      <sourceName> value_sourceName </sourceName>
      <tagID> value_tagID </tagID> //opt
      <lockAction> value_lockAction </lockAction>
      <lockMask> value_lockMask </lockMask>
      <password> value_password </password>
    </lockTagBank>
  </cmd>
</frame>

```

//opt  $\triangleq$  Optional: The line can be omitted.

## Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <lockTagBank>
      <returnValue>
        <tag>
          <tagID> value_tagID </tagID>
          <tagPC> value_tagPC </tagPC> //opt
          <success> value_success </success>
          <utcTime> value_utcTime </utcTime> //opt
          <antennaName> value_antennaName </antennaName> //opt
          <rSSI> value_rSSI </rSSI> //opt
          <channel> value_channel </channel> //opt
          <power> value_power </power> //opt
          <polarization> value_polarization </polarization> //opt
          <commandRetry> value_commandRetry </commandRetry> //opt
          <airRetry> value_airRetry </airRetry> //opt
          <filterDataAvailable> value_filterDataAvailable
          </filterDataAvailable> //opt
        </tag>
        ...
        <tag> //opt
        ...
        </tag> //opt
      </returnValue>
    </lockTagBank>
  </reply>
</frame>

```

//opt  $\triangleq$  Optional: The line can be omitted. In the reply, the parameter is transferred depending on the configuration settings (engineering/communication).

The error codes for negative replies (resultCode  $\neq$  0) are described in the section "Error messages (negative XML replies) (Page 104)".

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format". This function accesses all transponders with this ID. If this parameter is empty or not even transferred, the function applies to all transponders. Nevertheless only one transponder is then permitted in the antenna field. Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".
value_tagPC	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters. Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".
value_lockAction	Boolean value 0, 1	000000000... 111111111	"LockAction" is a 10-digit Boolean value. The least significant bits decide the "USER Memory". You will find a full description of the "LockAction" parameter in the "EPC Global Specification".
value_lockMask	Boolean value 0, 1	000000000... 111111111	"LockMask" is a 10-digit Boolean value. The least significant bits decide the "USER Memory". You will find a full description of the "LockMask" parameter in the "EPC Global Specification".
value_password	Hexadecimal value 0...9, A...F	00000000.... FFFFFFFF	Access password for the transponder
value_success	Fixed values	True False	Flag to indicate whether or not the command was successful for this transponder. True: Command successful False: Command unsuccessful
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffz e.g.: 2018-12-24T18:34:56.929+00:00.

Parameter	Type	Values	Description
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
value_rSSI	Decimal value 0...9	0..255	RSSI value
value_channel	Decimal value 0...9	1...50	Channel number on which the transponder was detected.
value_power	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB
value_polarization	Fixed values	Default Circular Linear_vertical Linear_horizontal All	Polarization of the antenna This parameter is only relevant with switchable antennas. "Default" ≙ "Linear_vertical".
value_commandRetry	Decimal value 0...9	0...65535	Number of times the reader repeated commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of attempts is set using the WBM.
value_airRetry	Decimal value 0...9	0...65535	Number of times the reader repeated air interface commands attempting to obtain correct data. "0" corresponds to no attempt. The maximum number of air interface commands is specified in the reader firmware and cannot be modified.
value_filterDataAvailable	Fixed values	True False	Indicates whether a filter criterion was received. True: All the data was read or no filter was set. False: Data could not be read.

### 3.3.3.13 getTagStatus

Table 3- 51 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V3.0	--	V1.3

With this command, you can read out status information of a transponder. To ensure clear identification, either there can be only one transponder in the antenna field or the ID of the transponder needs to be specified. If there is more than one transponder in the antenna field and no transponder ID is specified, a negative response is returned.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <getTagStatus>
      <sourceName> value_sourceName </sourceName>
      <mode> value_mode </mode> //opt
      <tagID> value_tagID </tagID> //opt
    </getTagStatus>
  </cmd>
</frame>
```

//opt  $\triangleq$  Optional: The line can be omitted.

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getTagStatus>
      <returnValue>
        <tagStatusData> value_tagStatusData </tagStatusData>
        <tagID> value_tagID </tagID> //opt
        <values> //opt
          <value> value_value </value>
          ...
        </values>
      </returnValue>
    </getTagStatus>
  </reply>
</frame>
```

//opt  $\triangleq$  Optional: The line can be omitted.

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

**Parameter**

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_mode	Fixed values	1 2 3 239	Mode of the status data to be requested. The supported values depend on the reader type assigned to the specified read point. If this parameter is not set, the value "3" is used automatically.
value_tagID	Hexadecimal value 0...9, A...F	--	UID in "RAW Hex Data Format". This function accesses all transponders with this ID. Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".
value_tagStatusData	Hexadecimal value 0...9, A...F	--	The raw reply data of the "getTagStatus" command as simple byte sequence. It is mode-specific, reader-specific and transponder-specific.
value_value	Alphanumeric text	--	Named specific values from the raw reply data. Note that the number of values can change. There may be more or fewer values in future versions.

### 3.3.3.14 stopCommand

Table 3- 52 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V3.0	--	V1.3

With this command, the currently running command of the reader connected to the read point is canceled.

#### Command

```
<frame>
  <cmd>
    <id> value_id </id>
    <sourceName> value_sourceName </sourceName>
    <stopCommand/>
  </cmd>
</frame>
```

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <stopCommand/>
  </reply>
</frame>
```

The error codes for negative replies (resultCode ≠ 0) are described in the section "Error messages (negative XML replies) (Page 104)".

#### Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_ sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").

### 3.3.4 Error messages (negative XML replies)

Table 3- 53 Protocol version and firmware versions of the command

XML API	RF600	RF1xC
V1.0	V1.0	V1.3

If a command cannot be executed, the user application receives an error message (negative XML message) from the affected device. This reply frame always contains the corresponding result code ("value\_resultCode ≠ 0" or "≠ 0x00"), as well as information on the error that has occurred.

#### Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> value_resultCode </resultCode>
    <error>
      <name> value_name </name>
      <cause> value_cause </cause>
    </error>
  </reply>
</frame>

```

#### Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_resultCode	Decimal value 0...9	0...65535	Result code of the command You will find a list of possible return codes in the following table.
value_name	Text	--	Textual description of the error You will find descriptions of the problems in the following table.
value_cause	Text	--	Brief description of the cause of the negative reply.



## List of possible result codes

The result codes specified below differ in the number space from the possible error numbers in the logbook. The error numbers in the logbook correspond to the error numbers of the Ident instructions. You can refer to the "Ident instruction" column for the assignment of the result code to the Ident instruction error numbers.

Error code Ident instructions	Result code ("value_ resultCode")		Name ("value_name")	Description ("value_cause")
	hex	dec		
--	0x00	0	NO_ERROR	Positive reply. Command processed successfully.
0xE1FE0100	0x11	17	ERROR_MEMORY_ERROR	Memory error of the transponder
0xE1FE0200	0x12	18	ERROR_PRESENCE_ERROR	Presence error The transponder left the antenna field.
0xE1FE0300	0x13	19	ERROR_ADDRESS_ERROR	Address error Access outside the addressed memory area.
0xE1FE0400	0x14	20	ERROR_TAG_DETECTIVE	Initialization error The transponder is unable to execute the initialization command.
0xE1FE0500	0x15	21	ERROR_TAG_MEMORY_OVERFLOW	Access outside the addressed memory area.
0xE1FE0600	0x16	22	ERROR_TAG_UNFORMATTED	The transponder is not formatted.
0xE1FE0700	0x17	23	ERROR_DATA_STRUCTURE	The transponder data is inconsistent.
0xE1FE0800	0x18	24	ERROR_WRONG_EPCID	The transponder does not have the expected EPC ID/UID or has no EPC ID/UID.
0xE1FE0900	0x19	25	ERROR_AIR_COMMAND_NOT_SUPPORTED	The transponder does not support the command of the wireless protocol.
0xE1FE0A00	0x1A	26	ERROR_TAG_LOCKED	The transponder to which data is to be written or which should be deactivated is locked.
0xE1FE8100	0x91	145	ERROR_NO_ANSWER_FROM_TAG	The transponder is not responding.
0xE1FE8200	0x92	146	ERROR_WRONG_PASSWORD	The password you entered is incorrect. Access to the transponder was denied.
0xE1FE8300	0x93	147	ERROR_VERIFY_TAG_FAILED	Verification of the transponder failed.
0xE1FE8400	0x94	148	ERROR_TAG_UNSPECIFIED	General transponder error.
0xE1FE8500	0x95	149	ERROR_TAG_INSUFFICIENT_POWER	There is not enough transponder power.
0xE2FE0100	0x21	33	ERROR_FIELD_DISTURBED	Field error or send error
0xE2FE0200	0x22	34	ERROR_TOO_MANY_TAGS	There are too many transponders in the antenna field.
0xE2FE8100	0xA1	161	ERROR_NO_TAG	There is no transponder with the relevant EPC ID/UID in the antenna field.
0xE2FE8200	0xA2	162	ERROR_NO_DATA	The requested data is not available.
0xE2FE8300	0xA3	163	ERROR_INVALID_CRC	Bad checksum

3.3 XML commands/responses

Error code Ident instructions	Result code ("value_resultCode")		Name ("value_name")	Description ("value_cause")
	hex	dec		
0xE2FE8500	0xA5	165	ERROR_NO_FREQUENCY	No radio channel activated.
0xE2FE8600	0xA6	166	ERROR_NO_CARRIER	No carrier signal activated.
0xE2FE8700	0xA7	167	ERROR_MORE_THAN_ONE_TAG_IN_FIELD	There is more than one transponder in the antenna field.
0xE2FE8800	0xA8	168	ERROR_AIR_PROTOCOL_UNSPECIFIED	General radio protocol error
0xE4FE0100	0x41	65	ERROR_POWER_SUPPLY	Fault in power supply
0xE4FE0300	0x43	67	ERROR_ANTENNA	An antenna error was detected when executing the command.
0xE4FE0500	0x45	69	ERROR_DATA_BUFFER_OVERFLOW	Data buffer overflow
0xE4FE0600	0x46	70	ERROR_INVALID_READER_STATUS	The specified command is not permitted in the current reader status.
0xE4FE0700	0x47	71	ERROR_STARTUP_MESSAGE	Incorrect startup message
0xE4FE8100	0xC1	193	ERROR_TAGFIELD_NOT_FOUND	The specified tag field is unknown.
0xE4FE8A00	0xCA	202	ERROR_MISCELLANEOUS	A general error has occurred.
0xE4FE8B00	0xCB	203	ERROR_CONFIGURATION	A configuration error has occurred.
0xE4FE8C00	0xCC	204	ERROR_PIB_READER	Communication error between Ident profile and reader/CM.
0xE4FE8D00	0xCD	205	ERROR_FIRMWARE_INTERNAL	Internal firmware error
0xE4FE8E00	0xCE	206	ERROR_COMMAND_ABORTED	The command was aborted.
0xE4FE8F00	0xCF	207	ERROR_WRONG_MODULE_CONFIGURATION	Wrong module configuration
0xE4FE9300	0x1F5	501	ERROR_ANTENNA_1	Antenna 1 not connected
0xE4FE9300	0x1F6	502	ERROR_ANTENNA_2	Antenna 2 not connected
0xE4FE9300	0x1F7	503	ERROR_ANTENNA_3	Antenna 3 not connected
0xE4FE9300	0x1F8	504	ERROR_ANTENNA_4	Antenna 4 not connected
0xE4FE9300	0x1F9	505	ERROR_ANTENNA_5	Antenna 5 not connected
0xE4FE9300	0x1FA	506	ERROR_ANTENNA_6	Antenna 6 not connected
0xE4FE9300	0x1FB	507	ERROR_ANTENNA_7	Antenna 7 not connected
0xE4FE9300	0x1FC	508	ERROR_ANTENNA_8	Antenna 8 not connected
0xE4FE9300	0x1FD	509	ERROR_WRONG_TYPE_OR_VERSION_ANTENNA_1	Antenna 1: Antenna type or version is incorrect
0xE4FE9300	0x1FE	510	ERROR_WRONG_TYPE_OR_VERSION_ANTENNA_2	Antenna 2: Antenna type or version is incorrect

Error code Ident instructions	Result code ("value_ resultCode")		Name ("value_name")	Description ("value_cause")
	hex	dec		
0xE4FE9300	0x1FF	511	ERROR_WRONG_TYPE_OR_VERSION_ANTENNA_3	Antenna 3: Antenna type or version is incorrect
0xE4FE9300	0x200	512	ERROR_WRONG_TYPE_OR_VERSION_ANTENNA_4	Antenna 4: Antenna type or version is incorrect
0xE4FE9300	0x201	513	ERROR_WRONG_TYPE_OR_VERSION_ANTENNA_5	Antenna 5: Antenna type or version is incorrect
0xE4FE9300	0x202	514	ERROR_WRONG_TYPE_OR_VERSION_ANTENNA_6	Antenna 6: Antenna type or version is incorrect
0xE4FE9300	0x203	515	ERROR_WRONG_TYPE_OR_VERSION_ANTENNA_7	Antenna 7: Antenna type or version is incorrect
0xE4FE9300	0x204	516	ERROR_WRONG_TYPE_OR_VERSION_ANTENNA_8	Antenna 8: Antenna type or version is incorrect
0xE5FE8100	0xD1	209	ERROR_COMMUNICATION_ACCESS_DENIED	Communication error between CM and reader: Access denied.
0xE5FE8200	0xD2	210	ERROR_COMMUNICATION_RESSOURCE_USED	Communication error between CM and reader: Resource occupied.
0xE5FE8300	0xD3	211	ERROR_COMMUNICATION_SERIAL_INTERFACE	Functional error of the serial interface
0xE5FE8400	0xD4	212	ERROR_COMMUNICATION_COMMON_ERROR	Communication error between CM and reader: General error
0xE6FE0100	0x61	97	ERROR_COMMAND_NOT_SUPPORTED	The command is not supported by the reader or is unknown.
0xE6FE0200	0x62	98	ERROR_INVALID_COMMAND_INDEX	Invalid command index (CI)
0xE6FE0300	0x63	99	ERROR_TAG_CROSSED_FIELD_WITHOUT_CMD_PROCESSED	A transponder has passed through the antenna field without being processed.
0xE6FE0400	0x64	100	ERROR_PARAMETER_INVALID_VALUE	A parameter has an invalid value.
0xE6FE0500	0xE1	225	ERROR_PARAMETER_MISSING	A necessary parameter has not been specified.
0xE6FE8100	0xE2	226	ERROR_PARAMETER_INVALID_FORMAT	A parameter has an incorrect format
0xE6FE8200	0xE3	227	ERROR_PARAMETER_INVALID_TYPE	A parameter has the wrong format or the wrong data type.
0xE6FE8300	0xE4	228	ERROR_PARAMETER_NOT_SUPPORTED	A parameter is not supported by this reader.
0xE6FE8400	0xE5	229	ERROR_WRONG_MESSAGE_FORMAT	The XML format is incorrect. An error was detected when parsing the command.
0xE6FE8600	0xE6	230	ERROR_INVENTORY_COMMAND_FAILED	The "Inventory" command failed.
0xE6FE8700	0xE7	231	ERROR_READ_COMMAND_FAILED	The read command failed.

Error code Ident instructions	Result code ("value_ resultCode")		Name ("value_name")	Description ("value_cause")
	hex	dec		
0xE6FE8800	0xE8	232	ERROR_WRITE_ COMMAND_FAILED	The write command failed.
0xE6FE8900	0xE9	233	ERROR_WRITETAGID_ COMMAND_FAILED	Writing the EPC-ID failed.
0xE6FE8A00	0xEA	234	ERROR_LOCK_ COMMAND_FAILED	The "Lock" command failed.
0xE6FE8B00	0xEB	235	ERROR_KILL_ COMMAND_FAILED	The "Kill" command failed.
0xE6FE8C00	0xEC	236	ERROR_STOP_ COMMAND_FAILED	The "Stop" command failed.

## 3.4 XML reports

This section describes all frames that can be sent automatically by the connected device to the user application (XML reports). These reports can be transferred transacted or not transacted and are differentiated based on their type - event or alarm. You will find more information on XML reports in the section "Principle of operation (Page 8)".

### 3.4.1 Events

This section describes all events. Events are asynchronous frames sent automatically by the device to the user application and containing data acquired by the reader itself.

#### 3.4.1.1 tagEventReport

Table 3- 54 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

On a status change (observed, lost) of a transponder, a transponder event report records the transponder data and sends it to the user application.

With RF600, the configuration of the trigger (continuous, I/O level, ...) and the definition of the transponder data (tag fields, RSSI value, ...) to be sent are taken from the stored configuration. These parameters cannot only be changed using the WBM.

The transponder data in each event report is grouped according to read points (<source>).

A reply frame from the user application is necessary only when transacted transfer is set. A reply frame can, however, also be sent in non-secure mode and this has no negative influence.

### Report

```
<frame>
  <report>
    <id> value_id </id>
    <ter>
      <source>
        <sourceName> value_sourceName </sourceName>
        <tag>
          <tagID> value_tagID </tagID>
          <sourceName> value_sourceName </sourceName> //opt**
          <tagPC> value_tagPC </tagPC> //opt*
          <event> value_event </event> //opt
          <utcTime> value_utcTime </utcTime> //opt
          <antennaName> value_antennaName </antennaName> //opt*
          <rSSI> value_rSSI </rSSI> //opt*
```

```

        <rSSIMin> value_rSSIMin </rSSIMin> //opt*
        <rSSIMax> value_rSSIMax </rSSIMax> //opt*
        <channel> value_channel </channel> //opt*
        <power> value_power </power> //opt*
        <polarization> value_polarization </polarization> //opt*
        <inventoried> value_inventoried </inventoried> //opt*
        <filterDataAvailable> value_filterDataAvailable
        </filterDataAvailable> //opt*
        <tagField> //opt
            <fieldName> value_fieldName </fieldName>
            <bank> value_bank </bank>
            <startAddress> value_startAddress </startAddress>
            <dataLength> value_dataLength </dataLength>
            <data> value_data </data>
        </tagField> //opt
        <tagField> //opt
            ...
        </tagField> //opt
    </tag>
    ...
    <tag> //opt
        ...
    </tag> //opt
</source>
<source> //opt
    ...
</source> // opt
</ter>
</report>
</frame>

```

//opt ≙ Optional: The number of parameters varies depending on device configuration and/or device type. Multiple Tag events can be reported in a frame.

\*: Only for the RF600 product families

\*\* : Only for the RF1xxC product families

## Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <ter/>
  </reply>
</frame>
```

A reply frame from the user application is optional. It is only necessary when transacted mode is set.

No negative replies are defined for reports.

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName <sup>2)</sup>	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format" Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".
value_tagPC <sup>1)</sup>	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters. Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".
value_event	Fixed values	New <sup>1)</sup> Glimpsd <sup>1)</sup> Observed Lost	The transponder events are also generated by a smoothing algorithm in the read point. The algorithms are specified in the WBM. A report can also contain more than one event for the same transponder. Each individual event has its own transponder structure. If a report contains two events for the same transponder, two transponder structures exist with the same "value_tagID" but different values for "value_event".

Parameter	Type	Values	Description
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2018-12-24T18:34:56.929+00:00.
value_antennaName <sup>1)</sup>	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
value_rSSI <sup>1)</sup>	Decimal value 0...9	0..255	RSSI value This value has no unit and is not directly related to the performance.
rSSIMin <sup>1)</sup>	Decimal value 0...9	0...255	The lowest RSSI value with which the transponder was detected. This value has no unit and is not directly related to the performance.
rSSIMax <sup>1)</sup>	Decimal value 0...9	0...255	The highest RSSI value with which the transponder was detected. This value has no unit and is not directly related to the performance.
value_channel <sup>1)</sup>	Decimal value 0...9	1...50	Channel number on which the transponder was detected.
value_power <sup>1)</sup>	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB
value_polarization <sup>1)</sup>	Fixed values	Circular Linear_vertical Linear_horizontal Unknown	Polarization of the antenna With external antennas always unknown.
value_inventoried <sup>1)</sup>	Decimal value 0...9	0...65535	Indicates how often the transponder was identified via the air interface before it changed to the "observed" status.
value_filterDataAvailable <sup>1)</sup>	Fixed values	True False	Indicates whether a filter criterion was received. True: All the data was read or no filter was set. False: Data could not be read.
value_fieldName <sup>1)</sup>	Text	--	Name of tag field Is specified in the WBM.
value_bank	--	0...3	Memory bank of the transponder <ul style="list-style-type: none"> <li>• 0: Reserved</li> <li>• 1: EPC</li> <li>• 2: TID</li> <li>• 3: USER MEMORY</li> </ul>
value_startAddress	Decimal value 0...9	0...65535	Start address of the first byte in the memory bank where reading will start.



Parameter	Type	Values	Description
value_dataLength	Decimal value 0...9	1...510	Number of bytes to be read.
value_data	Hexadecimal value 0...9, A...F	--	Data that should be read. Each byte is represented by two hexadecimal characters. Example: The byte sequence "0x12, 0x34, 0xA3" is represented as the character string "1234A3" in the "value_data" parameter. "value_dataLength" is 3 in this example. If the transponder is detected but the data cannot be read (e.g. if the transponder has no user memory as required), this field remains empty. "value_success" is then also set to "False".

- 1) Only for the RF600 product families
- 2) Only for the RF1xxC product families

### 3.4.1.2 rssiEventReport

Table 3- 55 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	--

An RSSI event report signals every change in the RSSI values when reading the transponders. RSSI events are generated separately for each antenna.

The data type and amount are specified by the settings in the configuration in the WBM.

## Report

```
<frame>
  <report>
    <id> value_id </id>
    <rssier>
      <tag>
        <tagID> value_tagID </tagID>
        <sourceName> value_sourceName </sourceName> //opt**
        <tagPC> value_tagPC </tagPC> //opt*
        <utcTime> value_utcTime </utcTime> //opt
        <antennaName> value_antennaName </antennaName> //opt*
        <rSSI> value_rSSI </rSSI> //opt*
        <channel> value_channel </channel> //opt*
        <power> value_power </power> //opt*
        <polarization> value_polarization </polarization> //opt*
      </tag>
      ...
      <tag> //opt
      ...
      </tag> //opt
    </rssier>
  </report>
</frame>
```

//opt  $\triangleq$  Optional: The number of parameters varies depending on device configuration and/or device type. Multiple RSSI events can be reported in a frame.

\*: Only for the RF600 product families

\*\* : Currently not supported by the RF1xxC product families

## Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <rssier/>
  </reply>
</frame>
```

A reply frame from the user application is optional. It is only necessary when transacted mode is set.

No negative replies are defined for reports.

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format" Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468 For additional information, refer to the "EPC Global Specification".
value_sourceName <sup>2)</sup>	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_tagPC <sup>1)</sup>	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters. Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffz e.g.: 2018-12-24T18:34:56.929+00:00.
value_antennaName <sup>1)</sup>	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
value_rSSI <sup>1)</sup>	Decimal value 0...9	0..255	RSSI value This value has no unit and is not directly related to the performance.
value_channel <sup>1)</sup>	Decimal value 0...9	1...50	Channel number on which the transponder was detected.
value_power <sup>1)</sup>	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB
value_polarization <sup>1)</sup>	Fixed values	Circular Linear_vertical Linear_horizontal Unknown	Polarization of the antenna With external antennas always unknown.

<sup>1)</sup> Only for the RF600 product families

<sup>2)</sup> Only for the RF1xxC product families

### 3.4.1.3 ioEventReport

Table 3- 56 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V1.0	V1.0	V1.3

An IO event report signals changes at a digital input/output.

The sending of IO events can be configured in the WBM. You will find more information on the events in the WBM sub-section "Communication" of the relevant Ident device manual.

### Report

```

<frame>
  <report>
    <id> value_id </id>
    <ioer>
      <io>
        <ioName> value_ioName </ioName>
        <ioEvent> value_ioEvent </ioEvent>
        <utcTime> value_utcTime </utcTime>
      </io>
      <io> //opt
      ...
      </io> //opt
    </ioer>
  </report>
</frame>

```

//opt  $\Delta$  Optional: Multiple "IO events" can be reported in a frame.

### Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <ioer/>
  </reply>
</frame>

```

A reply frame from the user application is optional. It is only necessary when transacted mode is set.

No negative replies are defined for reports.

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_ioName	Fixed values	Inport00 Inport01 ... Inport08 Output00 Output01 ... Output08	Name of the digital input/output
value_ioEvent	Fixed values	High Low	Indicates the new status of the digital input/output.
value_utcTime	Time	--	UTC time stamp (Universal Time Co-ordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2018-12-24T18:34:56.929+00:00.

### 3.4.1.4 logEventReport

Table 3- 57 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V3.0	--	V1.3

A log event report signals new logbook entries.

## Report

```

<frame>
  <report>
    <id> value_id </id>
    <ler>
      <log>
        <logEntry> value_logEntry </logEntry>
        <utcTime> value_utcTime </utcTime>
      </log>
      <log> //opt
      ...
      </log> //opt
    </ler>
  </report>
</frame>

```

//opt ≙ Optional: Multiple logbook entries can be reported in a frame.

## Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <ler/>
  </reply>
</frame>

```

A reply frame from the user application is optional. It is only necessary when transacted mode is set.

No negative replies are defined for reports.

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique command identifier
value_logEntry	Text	--	Logbook entry
value_utcTime	Time	--	UTC time stamp (Universal Time Co-ordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffz e.g.: 2018-12-24T18:34:56.929+00:00.

### 3.4.1.5 presenceEventReport

Table 3- 58 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V3.0	--	V1.3

A presence event report signals the number of transponders currently present in the antenna field whenever the number of transponders in the antenna field changes.

#### Report

```
<frame>
  <report>
    <id> value_id </id>
    <per>
      <presence>
        <sourceName> value_sourceName </sourceName>
        <noOfTags> value_noOfTags </noOfTags>
        <utcTime> value_utcTime </utcTime>
      </presence>
      ...
      <presence> //opt
        ...
      </presence> //opt
    </per>
  </report>
</frame>
```

//opt ≙ Optional: The presence can be signaled for multiple read points in a frame.

#### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <ler/>
  </reply>
</frame>
```

A reply frame from the user application is optional. It is only necessary when transacted mode is set.

No negative replies are defined for reports.

**Parameter**

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_noOfTags	Decimal value	0...x	Number of transponders currently located in the antenna field. 0: No transponder in the antenna field 1: At least one transponder in the antenna field x: If supported by the reader, the exact number of transponders is specified if there is more than one transponder in the antenna field.
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2018-12-24T18:34:56.929+00:00.

**3.4.1.6 lastAccessEventReport**

Table 3- 59 Protocol version and firmware versions of the command

XML API	RF600	RF1xxC
V3.0	--	V1.3

A last access event report records the transponder data of the transponder that was read or written to with the last successful command. Using this report, all successfully executed transponder commands can be recorded - even those that are issued in parallel by a controller.

**Report**

```

<frame>
  <report>
    <id> value_id </id>
    <laer>
      <client>
        <clientName> value_clientName </clientName>
        <command> value_command </command>
      <tag>
        <tagID> value_tagID </tagID>
        <sourceName> value_sourceName </sourceName> //opt
        <tagPC> value_tagPC </tagPC> //opt
      </tag>
    </laer>
  </report>
</frame>

```



```

        <utcTime> value_utcTime </utcTime> //opt
        <antennaName> value_antennaName </antennaName> //opt
        <rSSI> value_rSSI </rSSI> //opt
        <channel> value_channel </channel> //opt
        <power> value_power </power> //opt
        <polarization> value_polarization </polarization> //opt
        <tagField> //opt
            <bank> value_bank </bank>
            <startAddress> value_startAddress </startAddress>
            <dataLength> value_dataLength </dataLength>
            <data> value_data </data>
        </tagField> //opt
        <tagField> //opt
        ...
        </tagField> //opt
    </tag>
    ...
    <tag> //opt
    ...
    </tag> //opt
</client>
<client> //opt
    ...
</client> // opt
</laer>
</report>
</frame>

```

**//opt**  $\triangleq$  Optional: The parameters may be present, depending on the device parameter assignment or the device type.

## Reply

```

<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <tder/>
  </reply>
</frame>

```

A reply frame from the user application is optional. It is only necessary when transacted mode is set.

No negative replies are defined for reports.

Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_clientName	Fixed values	XML1 XML2 XML3 XML4 WBM OPCUA EIP PNIO PB READPOINT	Name of the user application that executed the last successful transponder command.  READPOINT: A device-internal application of the read point initiated the command.
value_command	Fixed values	See description	Name of the last successfully executed transponder command. The possible values depend on the user application used: <ul style="list-style-type: none"> <li>• PNIO, PB, EIP (Ident profile): INVENTORY, WRITE_ID, LOCK_TAG_BANK, PHYSICAL_READ, PHYSICAL_WRITE, FORMAT, TAG_STATUS, GET, PUT, NEXT</li> <li>• PNIO, PB (FB 45): WRITE, READ, INIT, MDS-STATUS</li> <li>• OPCUA: Scan, ReadTag, WriteTag, KillTag, LockTag, SetTagPasword, Read, Write</li> <li>• WBM, XML: readTagIDs, writeTagID, readTagMemory, writeTagMemory, killTag, lockTagBank, getTagStatus</li> <li>• READPOINT: Observed</li> </ul>
value_tagID	Hexadecimal value 0...9, A...F	--	UID or EPC ID in "RAW Hex Data Format"  Example of a 96-bit EPC ID of an RF600 transponder: 3005FB63AC1F3681EC880468  For additional information, refer to the "EPC Global Specification".

Parameter	Type	Values	Description
value_rwData	Hexadecimal value 0...9, A...F	--	Read/written data of the last command Is only output if the last command was a transponder-specific read/write command.
value_sourceName	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_tagPC	Hexadecimal value 0...9, A...F	--	Tag PC (Protocol Control) 16-bit value represented by four hexadecimal characters. Example: The value "1234" corresponds to the binary value "0001.0010.0011.0100".
value_utcTime	Time	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2018-12-24T18:34:56.929+00:00.
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of the antenna
value_rSSI	Decimal value 0...9	0..255	RSSI value This value has no unit and is not directly related to the performance.
value_channel	Decimal value 0...9	1...50	Channel number on which the transponder was detected.
value_power	Decimal value with two decimal places 0...9	0, 5.00...33.00	Used radiated power of the antenna in [dB] Increment: 0.25 dB
value_polarization	Fixed values	Circular Vertical Horizontal Unknown	Polarization of the antenna With external antennas always unknown.
value_bank	--	0...3	Memory bank of the read/written data of the last successfully executed transponder command. Multiple memory banks are also possible with the "XML" user applications, depending on the command. RF600: <ul style="list-style-type: none"> <li>• 0: Reserved</li> <li>• 1: EPC</li> <li>• 2: TID</li> <li>• 3: USER MEMORY</li> </ul> RF300: 0

Parameter	Type	Values	Description
value_startAddress	Decimal value 0...9	0...65535	Start address of the first byte of the read/written data of the last successfully executed transponder command for reading/writing.  Multiple memory banks are also possible with the "XML" user applications, depending on the command.
value_dataLength	Decimal value 0...9	1...1024	Number of bytes of the read/written data of the last successfully executed transponder command for reading/writing.  Multiple memory banks are also possible with the "XML" user applications, depending on the command.
value_data	Hexadecimal value 0...9, A...F	--	Read/written data of the last successfully executed transponder command for reading/writing.  Multiple memory banks are also possible with the "XML" user applications, depending on the command.  Each byte is represented by two hexadecimal characters.  Example: The byte sequence "0x12, 0x34, 0xA3" is represented as the character string "1234A3" in the "value_data" parameter. "value_dataLength" is 3 in this example.  If the transponder is detected but the data cannot be read (e.g. if the transponder has no user memory as required), this field remains empty. "value_success" is then set to "False".

## 3.4.2 alarm

This section describes all alarms. Alarms are asynchronous frames and are used as status, warning or error messages of the reader.

The meaning of an alarm is indicated by the error number ("value\_errorNumber") and other optional parameters.

### Alarm report

```
<frame>
  <alarm>
    <id> value_id </id>
    <error>
      <utcTime> value_utcTime </utcTime>
      <errorNumber> value_errorNumber </errorNumber>
      <errorText> value_errorText </errorText>
      <sourceName> value_sourceName </sourceName> //opt**
      <commandName> value_commandName </commandName> //opt**
      <clientName> value_clientName </clientName>
      <eventType> value_eventType </eventType> //opt
      <paramXY> value_param_xy </paramXY> //opt
      ...
    </error>
  </alarm>
</frame>
```

// opt  $\triangleq$  Optional: The parameters may be present, depending on the alarm message.

\*\* : Only for the RF1xxC product families

### Reply

```
<frame>
  <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <error/>
  </reply>
</frame>
```

A reply frame from the user application is optional. It is only necessary when secured mode is set.

No negative replies are defined for reports.

## Parameter

Parameter	Type	Values	Description
value_id	Decimal value 0...9	0...4294967295	Unique command identifier
value_errorNumber	Decimal value 0...9	0...65535	Error number You will find a list of error numbers in the following table.
value_utcTime	Text	--	UTC time stamp (Universal Time Coordinated) in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2018-12-24T18:34:56.929+00:00.
value_errorText	Text	--	Message text of the alarm
value_sourceName <sup>1)</sup>	Text	--	Read point name The name is specified in the WBM of the device (WBM menu "Read points").
value_commandName <sup>1)</sup>	Text	--	Name of the executed command
value_clientName	Fixed values	XML1 XML2 XML3 XML4 WBM OPCUA EIP PNIO PB READPOINT	Name of the user application that executed the command. READPOINT: A device-internal application of the read point initiated the command.
value_eventType	Fixed values	Coming Going	Status display Some alarms have a simple status indication. This indicates whether the error occurred and is still present or whether it has been eliminated.
value_paramXY	Text	--	Additional parameter The name of the parameter depends on the error. Depending on the error number, an alarm contains a different number of parameters.

<sup>1)</sup> Only for the RF18xC/RF18xCI product families

### List of possible alarm messages

Error number ("value_ errorNumber")		Description
hex	dec	
0x1511	5393	ERROR_NO_ANSWER_FROM_TAG The transponder is not responding.
0x1512	5394	ERROR_WRONG_PASSWORD The password you entered is incorrect. Access to the transponder was denied.
0x1513	5395	ERROR_VERIFY_TAG_FAILED Verification of the transponder failed.
0x1514	5396	ERROR_TAG_UNSPECIFIED General transponder error
0x1515	5397	ERROR_TAG_INSUFFICIENT_POWER There is not enough transponder power.
0x1521	5409	ERROR_NO_TAG There is no transponder in the antenna field.
0x1522	5410	ERROR_NO_DATA The requested data is not available.
0x1523	5411	ERROR_INVALID_CRC Bad checksum
0x1525	5413	ERROR_NO_FREQUENCY No radio channel activated.
0x1526	5414	ERROR_NO_CARRIER No carrier signal activated.
0x1527	5415	ERROR_MORE_THAN_ONE_TAG_IN_FIELD There is more than one transponder in the antenna field.
0x1528	5416	ERROR_AIR_PROTOCOL_UNSPECIFIED General radio protocol error
0x1599	5529	ERROR_WRONG_TYPE_OR_VERSION_ANTENNA_1 Wrong antenna type or wrong antenna version at connector ANT 1.
0x159A	5530	ERROR_WRONG_TYPE_OR_VERSION_ANTENNA_2 Wrong antenna type or wrong antenna version at connector ANT 2.
0x159B	5531	ERROR_WRONG_TYPE_OR_VERSION_ANTENNA_3 Wrong antenna type or wrong antenna version at connector ANT 3.
0x159C	5532	ERROR_WRONG_TYPE_OR_VERSION_ANTENNA_4 Wrong antenna type or wrong antenna version at connector ANT 4.
0x154A	5450	General error
0x154D	5453	Internal firmware error
0x1567	5479	ERROR_READ_COMMAND_FAILED The read command failed.
0x1591	5521	Antenna error at connector ANT 1
0x1592	5522	Antenna error at connector ANT 2

Error number ("value_ errorNumber")		Description
hex	dec	
0x1593	5523	Antenna error at connector ANT 3
0x1594	5524	Antenna error at connector ANT 4
0x7B71	31601	Overflow - "alarm": Send buffer for "alarm" is full. Alarm messages can be discarded until the next alarm is received.
0x7B73	31603	Overflow - "tagEventReport": Send buffer for "tagEventReport" is full. Data can be lost until the next "tagEventReport" is received.
0x9BFD	39933	ERROR_PARAMETER_INVALID_VALUE A parameter has an invalid value.
0x9CC5	40133	ERROR_ANTENNA An antenna error was detected when executing the command.
0x9CC7	40135	Power supply error The power supply is very close to the low limit.
0x9D8E	40334	ERROR_TOO_MANY_TAGS There are too many transponders in the antenna field.
0x9DF1	40433	ERROR_MEMORY_OVERRUN Access outside the addressed memory area.
0x9DEA	40426	ERROR_TAG_LOCKED The transponder to which data is meant to be written or which should be deactivated is locked.
0x7A152	500050	Connection attempt failed.
0x7A153	500051	Configuration successfully loaded.
0x7A154	500052	Connection established.
0x7A155	500053	Connection interrupted.



# Service & Support

## Industry Online Support

In addition to the product documentation, you are supported by the comprehensive online information platform of Siemens Industry Online Support at the following Internet address:

Link: (<https://support.industry.siemens.com/cs/de/en/>)

Apart from news, you will also find the following there:

- Project information: Manuals, FAQs, downloads, application examples etc.
- Contacts, Technical Forum
- The option to submit a support request:  
Link: (<https://support.industry.siemens.com/My/ww/en/requests>)
- Our service offer:

Right across our products and systems, we provide numerous services that support you in every phase of the life of your machine or system - from planning and implementation to commissioning, through to maintenance and modernization.

You will find contact data on the Internet at the following address:

Link: ([https://www.automation.siemens.com/aspa\\_app/?ci=yes&lang=en](https://www.automation.siemens.com/aspa_app/?ci=yes&lang=en))

## RFID homepage

For general information about our identification systems, visit RFID home page ([www.siemens.com/ident](http://www.siemens.com/ident)).

## Online catalog and ordering system

The online catalog and the online ordering system can also be found on the Industry Mall home page (<https://mall.industry.siemens.com>).

## SITRAIN - Training for Industry

The training offer includes more than 300 courses on basic topics, extended knowledge and special knowledge as well as advanced training for individual sectors - available at more than 130 locations. Courses can also be organized individually and held locally at your location.

You will find detailed information on the training curriculum and how to contact our customer consultants at the following Internet address:

Link: (<https://new.siemens.com/global/en/products/services/industry/sitrain.html>)

