SIEMENS

SIMATIC Ident

RFID Systems
SIMATIC RF XML programming

Function Manual

08/2012

A5E03125519-04

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.

⚠DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
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:

⚠WARNING
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.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Order number: A5E03125519
Ⓟ 09/2012 Technical data subject to change

Table of contents

1 Introduction

This document describes the RFID Reader XML Interface for SIMATIC RF readers.

RFID Reader XML Interface Overview 2

2.1 Preparing the reader

The RFID Reader XML Interface enables you to work with a simple XML interface to exploit the SIMATIC RF reader features. While the interface offers a way to access the reader’s capabilities to process RFID tags, the initial setup of the reader has to be done beforehand by means of a base configuration.

The SIMATIC RF-MANAGER Basic creates such configurations which can be transferred to a SIMATIC RF reader where they are stored on the onboard flash memory to be available even after a power cycle.

Moreover, base configurations created with the SIMATIC RF-MANAGER Basic can be exported to a file to be loaded to the SIMATIC RF reader via the XML interface.

Apart from this means of initializing a reader, the XML interface is not intended to configure a reader but to operate it. Settings such as ‘how many antennas will be used’ or ‘setting out port 1 to LOW when a tag is detected’ or ‘the initial state of outport 3 is High’ must be configured with the SIMATIC RF-MANAGER Basic.

Nevertheless, the typically parameters which often need to be changed during normal operating mode or for commissioning issues such as the antenna power, can be adjusted via the XML interface.

Every SIMATIC RF reader supporting the RFID Reader XML Interface is ready to operate out-of-the-box by providing a permanent default base configuration. So even without bothering to use the SIMATIC RF-MANAGER Basic you will be able to do an initial commissioning of a SIMATIC RF reader.

Additionally, it is possible to resort to the default configuration by means of the XML interface at any time.

The settings the default configuration uses are identical to those of a newly created project with the SIMATIC RF-MANAGER Basic. There is a single data source for all four antennas (although not all of the antennas have to be attached).

If you need to go into details, just open the SIMATIC RF-MANAGER Basic and check the initial settings.

The physical connection to a SIMATIC RF reader is established via an Ethernet interface. The necessary communication protocol is TCP/IP.

The factory defaults of the Ethernet parameters for a SIMATIC RF reader are:

IP Address:	192.168.0.254
Subnet Mask:	255.255.255.0
Port:	10001
Gateway:	192.168.0.1
DHCP:	off

All Ethernet parameters can be easily changed with the SIMATIC RF-MANAGER Basic. In case the IP address of a device is no longer known, the SIMATIC RF-MANAGER Basic provides a reader discovery feature that allows detecting and reconfiguring such readers.

For more information about preparing a SIMATIC RF reader have a look at the corresponding SIMATIC RFxxx system manual.

2.2 Configuration settings via

The following table gives you a reference to the settings of the SIMATIC RF-MANAGER Basic which could affect the behavior of the XML interface.

Topic	Where to find	Description
Configuration file	Menu > Reader > Export User API Configuration	Generate a configuration file, which could be loaded onto the reader via the command setConfiguration.
Ethernet Settings	Menu > Reader > Configure Reader Address	Configures the Ethernet settings of the reader.
Startup behavior	Tab RFID Device > System	This page defines the startup behaviour of the reader.
Buffering mode	Tab RFID Device > Connection > Buffering while connection lost	Defines if reports and alarms work in buffering or non buffering mode.
Out Ports	Tab I/O Ports	The general settings of the out port behavior such as "Inactivity Level" or "Automatic Reset Time"
IoEventReport	Tab I/O Ports > General > Send notifications	If the property "Send notification" is checked, the reader will send IoEventReports for this port.
Source	Tab Sources	On this tab you can configure the behavior of the source.
Name of the source	Tab Sources > General	Name of the source.
Tag Selector	Tab Source > Tag Selector	Assignment which Tag Selector is active for the source. A tag selector defines beside filter criteria's also tag fields. These tag fields could be addressed via XML interface by their names. Tag fields connected to a source via Tag Selector will also be transmitted in TagEventReports.
Name of the tag field	Tab Tag Selector > General > Tag field name	Defines the name of a tag field. Be aware, the field is only valid for such sources, which include the corresponding tag selector. See Tag Selector
Data Selector	Tab Notification channels > General	Assignment which Data Selector is active for the notification channel. The selected Data Selector defines which additional tag data (`<rSSI>`, `<utcTime>`, `<antennaName>`) the reader sends.

Topic	Where to find	Description
RSSI Value	Tab Data Selector > General > Field Names > RSSI value	If this field name is checked, the optional tag parameter `<rSSI>` will be included in the corresponding XML messages. But only if the corresponding Data Selector is connected to the notification channel. See Data Selector.
UTC time	Tab Data Selector > General > Field Names > Event time (UTC)	If this field name is checked, the optional tag parameter `<utcTime>` will be included in the corresponding XML messages. But only if the corresponding Data Selector is connected to the notification channel. See Data Selector.
Antenna name	Tab Data Selector > General > Field Names > Sending Antenna	If this field name is checked, the optional tag parameter `<antennaName>` will be included in the corresponding XML messages. But only if the corresponding Data Selector is connected to the notification channel. See Data Selector.

2.3 Connecting the reader

Only one logical connection is necessary for a client application to communicate with the reader. After power-on a SIMATIC RF reader listens on TCP/IP port 10001 for a connection request from a client application.

Whenever a client application has connected via TCP/IP, the reader will reject request from other clients as long as it considers the connection valid.

A connection is considered valid if it shows any activity within a timeout of 30 seconds. After a period of inactivity longer than 30 seconds, new client connection requests are accepted again.

A client application should use a cyclic heartbeat telegram to keep its connection valid and block other clients.

After a client has established the connection to the reader, the RFID Reader XML interface is ready for use.

2.4 The XML Interface

The programming interface is an XML based messaging interface which uses a command-response messaging protocol for communication.

That is, each command that is sent by a client application will be answered by the reader with a response message no matter if the command could be operated successfully or not. In case of errors the response contains the reason for failure.

Each command must use a unique ID which is included by the reader in the corresponding response. This ensures unambiguous association of messages to responses in case the interface is not operated synchronously.

It is possible to send multiple commands subsequently without waiting for each command to respond before sending the next command. The reader ensures that every command will be answered with a response message and that command execution happens in order.

Clients have to associate responses with messages via the unique command ID in such cases.

Normally, the reader will respond to a command within 5 seconds. Whenever this timeout is exceeded a client application may take appropriate action e.g. warning the user of a potential mal-function in the reader. Keep in mind that the reader will not send a timeout response.

Only some commands such as setConfiguration or readTagIds may take more than 5 seconds to complete. Such commands will have an explicit note.

In addition to the normal command/response type messages, this specification defines asynchronous notifications. These will always be sent by the reader and may require an acknowledgment from the client application.

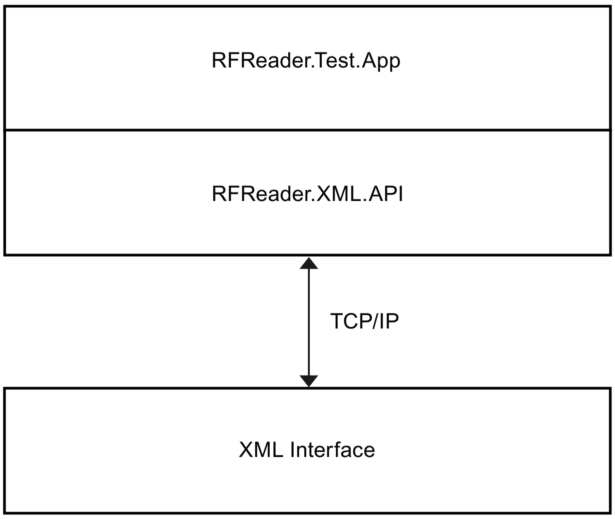
The chapter "RFID Reader XML Interface Reference (Page 19)" describe each command of the interface in detail including its parameters and return values.

2.5 Testing the XML Interface

Within the SIMATIC RF-MANAGER Basic installation you get an example for a Windows .NET 2.0 Client application including the complete source code.

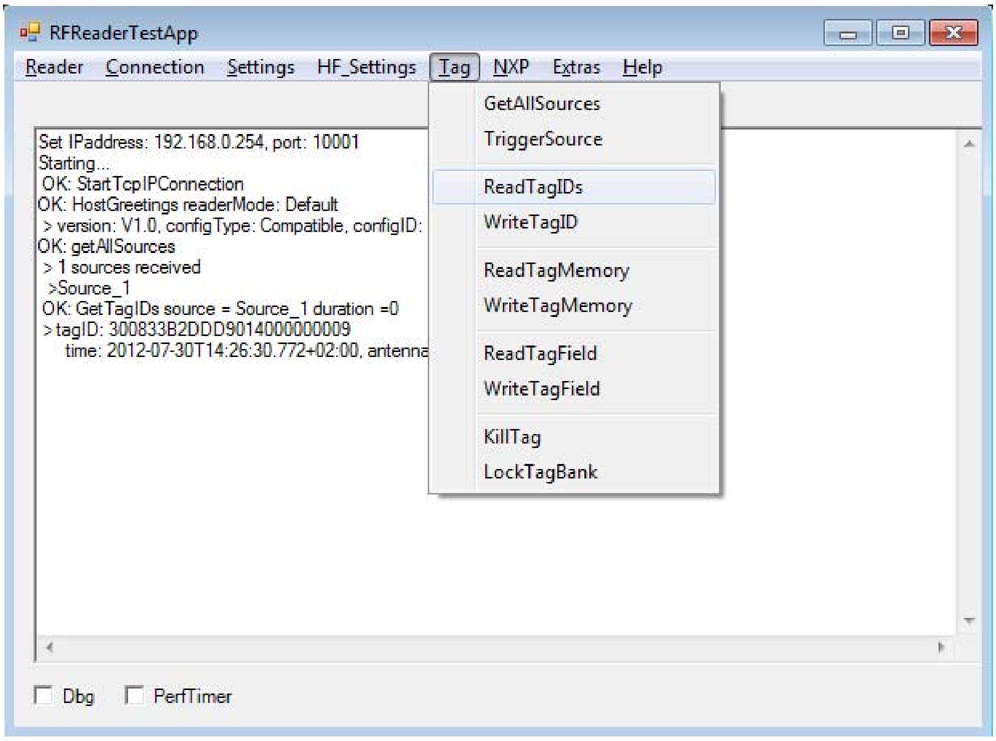
You could easily start the sample application via the start menu "RF Reader XML Demo". If you copy the application to other PCs you only have to make sure that the runtime components of .NET 2.0 are also installed.

If you want to change some code, you require a derivate of the Microsoft Visual Studio 2005 as starting point. The Express versions will also be sufficient.



The example is divided into two parts.

- The RFReader.XmlApi handle the XML interface on the PC side and provide the complete functionality via .NET.
- The RFReader.TestApp is a simple Windows application which allows users to send single commands to the reader.



```
Set IPaddress: 192.168.0.254, port: 10001
Starting...
 OK: StartTcpIPConnection
OK: HostGreetings readerMode: Default
 > version: V1.0, configType: Compatible, configID:
OK: getAllSources
 > 1 sources received
  >Source_1
 OK: GetTagIDs source = Source_1 duration =0
 >tagID: 300833B2DDD9014000000009
     time: 2012-07-30T14:26:30.772+02:00, antenna
```

The RFReader.TestApp application consists of three sections. Beginning from top to down you find:

- The menu bar.

  All available commands are grouped into corresponding menu entries. Each single command can be selected via the menu hierarchy.

- The log box.

  This is a text box displaying all information about executed commands and their return values.

  Alarms and Events which were sent autonomous from the reader were also reported in the log box. You can clear the log box via the menu entry "Extras > Clear log".

- Check boxes.

  Check the Dbg checkbox if the transmitted XML data stream should also be displayed.

  With a checked PerfTimer checkbox the execution time for each command will be reported.

For getting tag data out of your reader you simply have to:

1. Connecting the hardware.

   On the hardware side, you need a SIMATIC RF reader supporting the RFID Reader XML Interface, at least with one connected antenna.

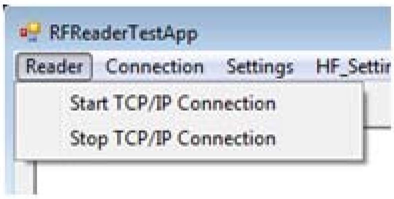
   The reader must have an Ethernet connection to your PC running the RFReader.TestApp application.

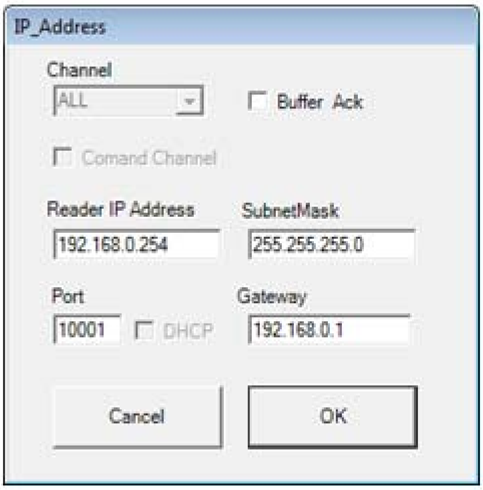
   The factory setup of the reader includes a default configuration. So you may only have to change the default IP address of the SIMATIC RF reader.

   For more information see the system manual RFxxx.

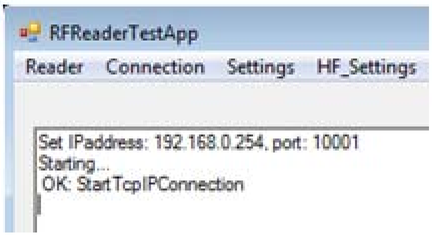
2. Configure the TCP/IP Connection.

   Start the TCP/IP connection via the RFReader.TestApp application:





Channel: ALL
□ Buffer Ack
□ Comand Channel
Reader IP Address: 192.168.0.254
SubnetMask: 255.255.255.0
Port: 10001
□ DHCP
Gateway: 192.168.0.1



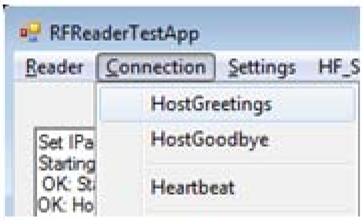
```
Set IPaddress: 192.168.0.254, port: 10001
Starting...
OK: StartTcpIPConnection
```

   – Select "Reader > Start TCP/IP Connection".

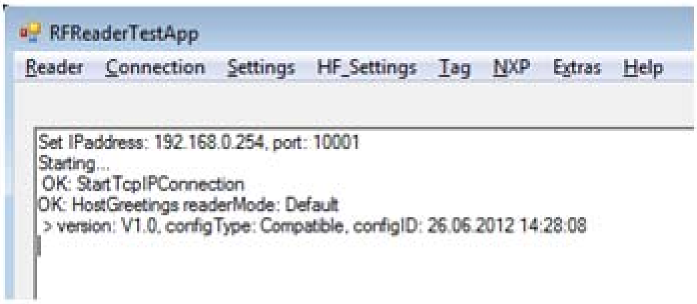
   – Check the correctness of the Reader IP Address and than press the "OK" button. If the reader is reachable, you will get the positive response in the text box.

3. Start communication with the reader.

   The first command to the reader is always the hostGreetings command:



```
Set IPa
Starting
 OK: St
OK: Ho
```



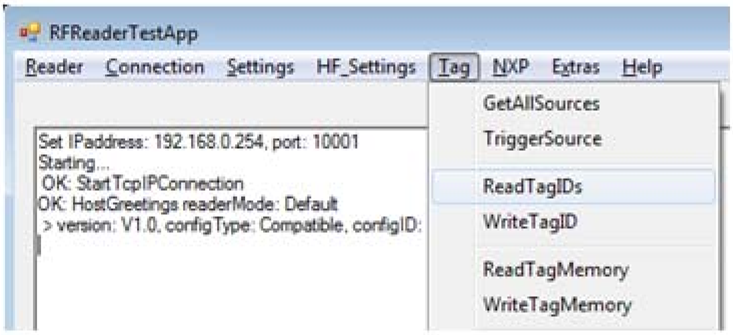
```
Set IPaddress: 192.168.0.254, port: 10001
Starting...
 OK: StartTcpIPConnection
OK: HostGreetings readerMode: Default
 > version: V1.0, configType: Compatible, configID: 26.06.2012 14:28:08
```

   - Select "Connection > HostGreetings" and quit the following pop up window with "OK".
   - In the text box you will get the answer for the HostGreetings command from the reader.

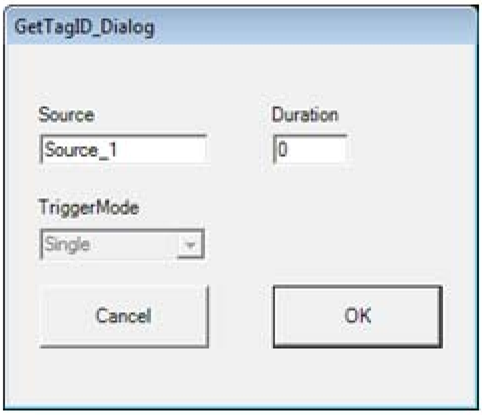
4. Read Tag Data.

   Now we are ready to get data from RFID tags. Don’t forget to put a Tag in reach of the antenna. You find all related functions for getting tag data inside the menu "Tag".

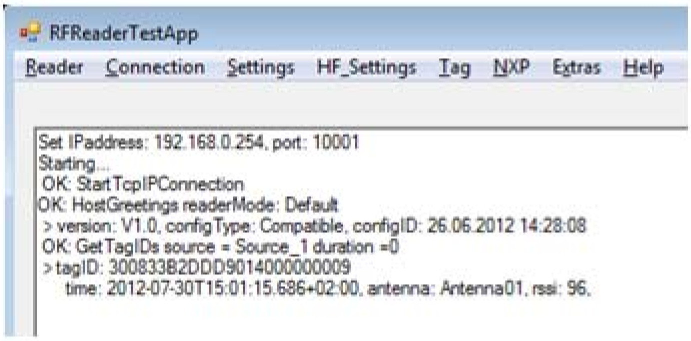
   For example we read the Tag IDs:



```
Set IPaddress: 192.168.0.254, port: 10001
Starting...
 OK: StartTcpIPConnection
OK: HostGreetings readerMode: Default
 > version: V1.0, configType: Compatible, configID:
```



Source: Source_1
Duration: 0
TriggerMode: Single



```
Set IPaddress: 192.168.0.254, port: 10001
Starting...
 OK: StartTcpIPConnection
OK: HostGreetings readerMode: Default
 > version: V1.0, configType: Compatible, configID: 26.06.2012 14:28:08
 OK: GetTagIDs source = Source_1 duration =0
 >tagID: 300833B2DDD9014000000009
     time: 2012-07-30T15:01:15.686+02:00, antenna: Antenna01, rssi: 96,
```

– Select "Tag > ReadTagIDs"
– Check the correct name of the source. Only necessary if the reader has not the default configuration (e.g. after factory setup) loaded.
– After pressing the "OK" button, you will get in the text box all Tag IDs which are read by the reader.

3 RFID Reader XML Interface Reference

The RFID XML Interface provides a way to access the RFID features that the RFID Reader device provides. End users can utilize these features by implementing custom client applications.

This chapter provides a detailed description of all available functions and their parameters.

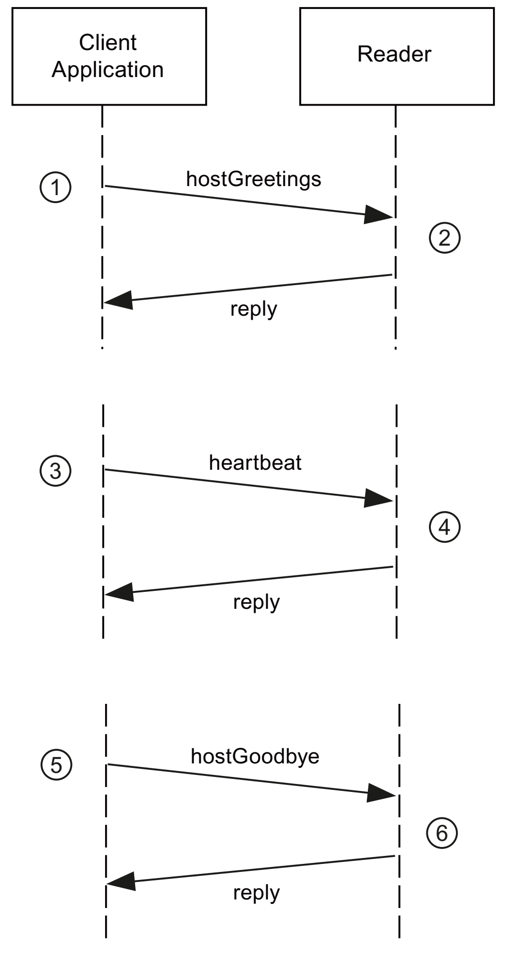
3.1 Commands

This chapter describes all possible commands that a client application can send to a SIMATIC RF reader.

3.1.1 Connection

This section describes all commands involved with the connection between client application and reader.

The following sequence diagram illustrates a successful connection and disconnection procedure.

Connecting procedure	Step	Description
 Client Application Reader 1 hostGreetings 2 reply 3 heartbeat 4 reply 5 hostGoodbye 6 reply	1	Client application sends first a hostGreetings command to the reader
	2	The reader will respond with a positive response.
	3	After the connection has been established, the client application communicates with the reader. For example sends heartbeat telegrams at regular intervals.
	4	The reader will respond each command with a response.
	5	The client application will sends a hostGoodbye command to inform the reader that the connection should be disconnect.
	6	The reader will respond with a positive response. Afterwards the reader disconnect the existing TCP/IP connection.

3.1.1.1 hostGreetings

Every communication with a reader has to start with a hostGreetings command to make the XML interface client known to the reader.

Whenever commands are sent without a preceding hostGreetings message, the reader will answer with "invalid mode". The only exception to this rule is the reset command which is always accepted to allow reinitializing of a reader.

With the hostGreetings command, the XML interface client is not only made known to the reader but compatibility between the current reader configuration and the client is checked. Only if the contained configuration is compatible with the XML interface will hostGreetings succeed. Such configurations can be created and transferred with the SIMATIC RF-MANAGER Basic configuration tool.

If the reader is operated using SIMATIC RF-MANAGER Professional, more features will be available but will only work together with a SIMATIC RF-MANAGER Professional runtime and no longer with the XML interface. Whenever such configurations are loaded to the reader, hostGreetings' compatibility check will give a negative answer.

Use SIMATIC RF-MANAGER Basic to transfer a valid configuration. Alternatively, transfer existing configurations using the XML interface's setConfiguration command or revert to the default configuration by using the reset command.

The currently stored base configuration defines the startup behavior, i.e. whether a reader is stopped or started after power on (see chapter "Configuration settings via (Page 9)").

Via hostGreetings the current state of the reader can be switched from stop mode to running or vice versa.

The XML interface may support multiple versions. A client is allowed to specify within hostGreetings which version of the XML interface it wants to work with. The reader response will contain the version the XML interface will use.

Command

```
<frame>
 <cmd>
    <id> value_id </id>
    <hostGreetings>
        <readerType> value_readerType </readerType>    // opt
        <readerMode> value_ReaderMode </readerMode>    // opt
        <supportedVersions>
            <version> value_version </version>
            <version> value_version </version>         // opt
            …
        </supportedVersions>
    </hostGreetings>
 </cmd>
</frame>
```

`// opt` → Optional: line can be omitted.

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_readerType	Fixed values	SIMATIC_RF670R SIMATIC_RF640R	Optional Type of reader.
value_ReaderMode	Fixed values	Default Stop Run	Optional Requested behavior of the reader. If this parameter is not specified the default will be Default.
			Default Reader continuouslworking without change in behavior.
			Stop Reader stops working. Neither reports nor alarms will be sent to client application. Antennas will turn off their HF field
			Run If Reader was in Stop mode and a compatible configuration is available it will be started.
value_version	Alphanumeric text	V1.0	Supported protocol version

Parameter	Type	Values	Description
value_configID	Alphanumeric text	-	A unique identification for the transferred configuration. ID can also be read with function getConfigVersion.
value_configType	Fixed values	Compatible PcRuntime	Return value. Signals type of currently loaded configuration.
			Compatible Configuration is full compatible with XML
			PcRuntime The reader runs with an RF_MANAGER generated configuration. The reader accepts only the following XML commands: • setConfiguration • getReaderStatus • setIPConfig • hostGoodbye • resetReader • heartbeat • startReader • stopReader

3.1.1.2 hostGoodbye

This function ends communication with the reader and the Ethernet connection will be closed.

In default mode, the reader will work on with its current settings. This allows an autonomous operation of the reader. After the next hostGreetings command all collected tag data will be delivered to the client application via tagEventReport. Pre-condition is that buffering mode of reports is activated.

Optionally, the reader mode may be switched to stop.

Command

```
<frame>
 <cmd>
    <id> value_id </id>
    <hostGoodbye>
        <readerMode> value_ReaderMode </readerMode>    // opt
    </hostGoodbye>
 </cmd>
</frame>
```

`// opt` → Optional: line can be omitted.

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_ReaderMode	Fixed values	Default Stop	Optional Requested behavior of the reader. If this parameter is not specified the default will be Default.
			Default Reader will not change its behavior.
			Stop Reader switches into stop mode.

3.1.1.3 startReader

This command starts the reader, but only if the reader was in stop mode before.

The stored base configuration defines the startup behavior (see chapter "Configuration settings via (Page 9)"). Either the reader is stopped or started after power on.

The command startReader the reader can be switched from stop mode into running mode and the command stopReader switch the reader into stop mode.

Command

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

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

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

3.1.1.4 stopReader

This command stops the reader, but only if the reader was in running mode before.

The stored base configuration defines the startup behavior (see chapter "Configuration settings via (Page 9)"). Either the reader is stopped or started after power on.

The command startReader the reader can be switched from stop mode into running mode and the command stopReader switch the reader into stop mode.

Command

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

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

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

3.1.1.5 heartBeat

On the one hand this function can be used to check if the connection is broken (e.g. broken cable), or if the reader is down (e.g. no power).

On the other hand the reader blocks requests for a connection by new clients for a maximum of 30 seconds after receiving the last command via the existing connection. Periodically sending heartBeat commands within a time frame of 30 seconds guarantees that no other client application accidentally gets access to the reader.

Command

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

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

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

3.1.1.6 setIPConfig

This command sets the IP parameters of the reader.

The reader will first send a positive response to this command. Afterwards the reader will disconnect the existing connection.

Command

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

`// opt` → Optional: line can be omitted.

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_iPAddress	IP Address	1.0.0.1… 255.255.255.254	IP Address The network infrastructure is to be considered for valid values
value_subNetMask	IP Address	1.0.0.0… 255.255.255.254	Subnet mask The network infrastructure is to be considered for valid values
value_dHCPEnable	Fixed values	True False	Use of DHCP for getting IP Address
			True DHCP active
			False DHCP off
value_gateway	IP Address	1.0.0.1… 255.255.255.254	IP Address of gateway. The network infrastructure is to be considered. for valid values

3.1.2 Reader Settings

This section describes all commands referring to the configuration of the reader.

3.1.2.1 setConfiguration

This command transfers a base configuration to the reader. After validating the given configuration it will be activated and permanently stored in the flash memory of the reader.

With each power cycle, the reader will use the stored base configuration.

A base configuration can be created using the SIMATIC RF-MANAGER Basic. Once the file is exported it can be used as is for a transfer to a SIMATIC RF reader.

Command

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

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_configData	CDATA	-	Configuration data. Unchanged exported configuration file from SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). Must be included in a CDATA section!
value_configID	Alphanumeric text	-	A unqique identification for the transferred configuration. ID can also be read with function getConfigVersion.

3.1.2.2 getConfiguration

This command retrieves a stored configuration from the reader.

Command

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

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_configData	CDATA	-	Configuration data. Unchanged configuration file content as exported from (see chapter "Configuration settings via (Page 9)"). Is included in a CDATA section!
value_configID	Alphanumeric text	-	The unique identification of the transferred configuration. ID can also be read with function getConfigVersion.

3.1.2.3 getConfigVersion

This command retrieves the version of the stored configuration.

Command

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

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_configType	Fixed values	Compatible PcRuntime	Return value. Signals type of currently loaded configuration.
			Compatible Configuration is full compatible with XML Interface commands.
			PcRuntime The reader runs with a RF_MANAGER generated configuration. The reader accepts only the following XML commands: • setConfiguration • getReaderStatus • setIPConfig • hostGoodbye • resetReader • heartbeat • startReader • stopReader
value_configID	Alphanumeric text	-	A unique identification of the transferred configuration.

3.1.2.4 setTime

This command set the reader’s internal clock.

Command

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

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00.

3.1.2.5 getTime

This command retrieves the current timestamp of the reader’s internal clock.

Command

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

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00.

3.1.2.6 setIO

This command sets the reader’s digital out ports.

The general settings of the out port behavior such as "Inactivity Level" or "Automatic Reset Time" are defined via the SIMATIC RF-MANAGER Basic (see also chapter "Configuration settings via (Page 9)").

Command

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

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_outValue	Characters 0, 1, X	0000…. XXXX… 1111	Each digit represents an out port of the reader: • Outport00: 1. digit (right least significant bit) • Outport01: 2. digit • Outport02: 3. digit • Outport03: 4. digit • … According to the value of each digit the corresponding port will be set to High level (1), reset to Low level (0) or left unchanged (X) Example: A value_outValue of "0X11" will • set Outport00 to High • set Outport01 to High • leave Outport02 unchanged • set Outport03 to Low The function will return the negative result ERROR_PARAMETER_OUT_OF_RANGE if more ports should be set as the reader supports. E.g. if the reader supports only 2 out ports, the above example for addressing 4 ports doesn't set any port at all.

3.1.2.7 getIO

This command retrieves the current levels of all in ports and out ports of the reader.

Command

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

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_inValue	Binary characters 0, 1	0000…. 1111	Each digit represents an in port of the reader: • Inport00: 1. digit (right least significant bit) • Inport01: 2. digit • Inport02: 3. digit • Inport03: 4. digit • … A digit value of 0 signals a Low level of the corresponding port. A digit value of 1 signals a High level of the corresponding port. If the reader doesn’t support any port the value will be empty.
value_outValue	Binary characters 0, 1	0000…. 1111	Each digit represents an out port of the reader: • Outport00: 1. digit (right least significant bit) • Outport01: 2. digit • Outport02: 3. digit • Outport03: 4. digit • … A digit value of 0 signals a Low level of the corresponding port. A digit value of 1 signals a High level of the corresponding port. If the reader doesn’t support any port the value will be empty.

3.1.2.8 resetReader

This command resets the reader. The following options are available.

- Delete the stored configuration.

  After a positive reply to the client application, the reader will terminate the connection and perform afterwards a hardware reset.

  The reader will restart with the default configuration.

  The client application must do a reconnect and start with command hostGreetings again.

- Hardware reset of the reader without deleting the stored configuration.

  After a positive reply to the client application, the reader will terminate the connection and afterwards perform a hardware reset.

  The client application must do a reconnect and start with command hostGreetings again.

- Delete the stored configuration and load the default configuration.

  But without resetting the reader or terminating the connection to the client application.

  A positive reply signals that the default configuration is loaded.

Command

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

`// opt` → Optional: line can be omitted.

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_resetType	Fixed values	Default Reboot Defcfg	Optional Type of reset. If this parameter is not specified the default will be Default.
			Default Delete the stored configuration. After a positive reply to the client application, the reader will terminate the connection and perform afterwards a hardware reset. The reader will restart with the default configuration. The client application must do a reconnect and start with command hostGreetings again.
			Reboot Hardware reset of the reader without deleting the stored configuration. After a positive reply to the client application, the reader will terminate the connection and perform afterwards a hardware reset. The client application must do a reconnect and start with command hostGreetings again.
			Defcfg Delete the stored configuration and load the default configuration. But without resetting the reader or terminating the connection to the client application. A positive reply signals that the default configuration is loaded.

3.1.2.9 getReaderStatus

This command retrieves information about the reader.

Command

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

Response

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

`// opt` → Optional: line can be omitted.

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_readerType	Fixed values	SIMATIC_RF670R SIMATIC_RF640R	Type of reader.
value_ReaderMode	Fixed values	Stop Run	Current behavior of the reader.
			Stop Reader is in stop. Neither reports nor alarms will be sent to client applications. Antennas have their HF field turned off.
			Run Reader is in run.
value_mLFB	Alphanumeric text	-	SIEMENS order number of reader. e.g. 6GT2811-0AB00-0AA0
value_hWVersion	Alphanumeric text	-	Hardware version of the reader. e.g. V1.0.0.0_1.1.0.34
value_fWVersion	Alphanumeric text	-	Firmware version of the reader. e.g. V1.0.0.0_1.1.0.34
value_version	Alphanumeric text	-	Reader specific version of sub parts of the reader. Be aware that the number of sub versions is subject to change. For future releases there may be more or less amount of sub versions.

3.1.3 Protocol configuration

This section describes all commands referring to the HF-settings of the reader.

3.1.3.1 setProtocolConfig

This command sets air protocol specific parameters.

Command

```
<frame>
 <cmd>
    <id> value_id </id>
    <setProtocolConfig>
        <initialQ> value_initalQ </initialQ>    // opt
        <profile> value_profile </profile>      // opt
        <channels> value_channels </channels>   // opt
        <retry> value_retry </retry>            // opt
        <idLength> value_idLength </idLength>   // opt
        <writeBoost> value_writeBoost </writeBoost>  // opt
    </setProtocolConfig>
 </cmd>
</frame>
```

`// opt` → Optional: line can be omitted.

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_initialQ	Decimal value 0..9	0..15	Optional. Air protocol parameter Initial Q for ISO 18000 6C tags. Refer to EPC global specification.
value_profile	Decimal value 0..9	0, 2, 4, 5, 7, 9, 10, 11	Optional. Profile ID of modulation schema. 0 = Tx:40kbps/Rx:80kbps/FM0 2 = Tx:40kbps/Rx:160kbps/FM0 4 = Tx:40kbps/Rx:40kbps/Miller4 5 = Tx:80kbps/Rx:160kbps/FM0 7 = Tx:80kbps/Rx:40kbps/Miller4 9 = Tx:80kbps/Rx:160kbps/Miller2 10 = Tx:80kbps/Rx:80kbps/Miller4 11 = Tx:80kbps/Rx:40kbps/Miller8
value_channelList	Decimal value with separator 0..9	103, 106, 109, 112	Optional. List of used ETSI channels. Channels are separated by comma. For FCC reader, this parameter has no effect. Example: 103, 106, 109
value_retry	Decimal value 0..9	0..255	Optional. Maximum number of retries if an air protocol command fails.
value_idLength	Decimal value 0..9	0, 16, 32, 48…496	Optional. Length of tag ID in bit. If only tags with the identical tag ID length are used, the reader can do a special reading algorithm, which speeds up reading. Valid length of tag IDs are multiples of 16. A value of 0 indicates standard handling of tags. In this case tags with different tag ID length can be read.
value_writeBoost	Decimal value with 2 decimal places 0..9 decimal point	0.00…12.00	Optional. Power increasing for write commands on air protocol in dB. Reader accepts values only in increments of 0.25 dB. Decimal point has to be a point

3.1.3.2 getProtocolConfig

This command gets air protocol specific parameters.

Command

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

Response

```
<frame>
 <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getProtocolConfig>
        <returnValue>
            <initialQ> value_initalQ </initialQ>
            <profile> value_profile </profile>
            <channels> value_channels </channels>
            <retry> value_retry </retry>
            <idLength> value_idLength </idLength>
            <writeBoost> value_writeBoost </writeBoost>
        </returnValue>
    </getProtocolConfig>
 </reply>
</frame>
```

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_initialQ	Decimal value 0..9	0..15	Air protocol parameter Initial Q for ISO 18000 6C tags. Refer to EPC global specification.
value_profile	Decimal value 0..9	0, 2, 4, 5, 7, 9, 10, 11	Profile ID of modulation schema. 0 = Tx:40kbps/Rx:80kbps/FM0 2 = Tx:40kbps/Rx:160kbps/FM0 4 = Tx:40kbps/Rx:40kbps/Miller4 5 = Tx:80kbps/Rx:160kbps/FM0 7 = Tx:80kbps/Rx:40kbps/Miller4 9 = Tx:80kbps/Rx:160kbps/Miller2 10 = Tx:80kbps/Rx:80kbps/Miller4 11 = Tx:80kbps/Rx:40kbps/Miller8
value_channelList	Decimal value with separator 0..9	103, 106, 109, 112	Optional. List of used ETSI channels. Channels are separated by comma. For FCC reader, this parameter has no effect. Example: 103, 106, 109
value_retry	Decimal value 0..9	0..255	Maximum number of retries if an air protocol command fails.
value_idLength	Decimal value 0..9	0, 16, 32, 48…496	Length of tag ID in bit. If only tags with identical tag ID lengths are used, the reader can use a special reading algorithm, which speeds up reading. Valid length of a tag IDs are multiples of 16. A value of 0 indicates standard handling of tags. In this case tags with different tag ID length can be read.
value_writeBoost	Decimal value with 2 decimal places 0..9 decimal point	0.00…48.00	Power increasing for write commands on air protocol in dB. Decimal point has to be a point

3.1.3.3 setAntennaConfig

This command sets antenna specific parameters.

Command

```
<frame>
 <cmd>
    <id> value_id </id>
    <setAntennaConfig>
        <antenna>
            <antennaName> value_antennaName </antennaName>
            <power> value_power </power>               // opt
            <cableLoss> value_cableLoss </cableLoss>   // opt
            <gain> value_gain </gain> opt
            <rSSIThreshold> value_rSSIThreshold </rSSIThreshold> // opt
        </antenna>
        …
        <antenna>     // opt
            …
        </antenna>    // opt
    </setAntennaConfig>
 </cmd>
</frame>
```

`// opt` → Optional: line can be omitted.

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna
value_power	Decimal value 0..9	0..2000	Optional. Antenna power in mW. Reader accepts values only in increments of : 10 mW for range 50...100 mW and 100 mW for range 100...2000 mW
value_cableLoss	Decimal value with 2 decimal places 0..9 decimal point	0.00…63.75	Optional. Cable loss in dB. Reader accepts values only in increments of 0.25 dB. Decimal point has to be a point.
value_gain	Decimal value with 2 decimal places 0..9 decimal point	-32.00…31.75	Optional. Antenna gain in dB. Reader accepts values only in increments of 0.25 dB. Decimal point has to be a point.
value_rSSIThreshold	Decimal value 0..9	0..255	Optional. Threshold for RSSI values. Tags with lower RSSI values will be discarded. This is a value without any unit and without any direct relationship to the power level.

3.1.3.4 getAntennaConfig

This command gets antenna specific parameters for all Antennas.

Command

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

Response

```
<frame>
 <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <getAntennaConfig>
        <returnValue>
            <antenna>
                <antennaName> value_antennaName </antennaName>
                <power> value_power </power>
                <cableLoss> value_cableLoss </cableLoss>
                <gain> value_gain </gain>
                <rSSIThreshold> value_rSSIThreshold </rSSIThreshold>
            </antenna>
            …
            <antenna>
                …
            </antenna>
        </returnValue>
    </getAntennaConfig>
 </reply>
</frame>
```

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna
value_power	Decimal value 0..9	0..2000	Antenna power in mW. Reader accepts values only in increments of : 10 mW for range 50...100 mW and 100 mW for range 100...2000 mW
value_cableLoss	Decimal value with 2 decimal places 0..9 decimal point	0.00…63.75	Cable loss in dB. Reader accepts values only in increments of 0.25 dB. Decimal point has to be a point.

Parameter	Type	Values	Description
value_gain	Decimal value with 2 decimal places 0..9 decimal point	-32.00…31.75	Antenna gain in dBi. Reader accepts values only in increments of 0.25 dB. Decimal point has to be a point.
value_rSSIThreshold	Decimal value 0..9	0..255	Optional. Threshold for RSSI values. Tags with lower RSSI values will be discarded. This is a value without any unit and without any direct relationship to the power level.

3.1.4 Tag processing

This section describes all commands managing tag data.

There are principle two mechanisms for getting tag data:

- Commands which return the tag data in their response telegram.

  The reader performs only once the requested action and returns afterwards the collected tag data.

  There will be no repetitions of the commands. The configuration parameter of the source like "Read cycles per trigger" or "Read timeout" will have no effect.

  The collected tag data won’t be included in any smoothing algorithm of the source nor be transferred via TagEventReports.

- TagEventReports are sent autonomous from the reader to the client application.

  The tag data will only be collected via triggering of the source. The SIMATIC RF-MANAGER Basic enable a wide range of possibilities to configure the trigger of a source.

  Via the XML interface the function "triggerSource" will perform a single trigger of the source.

3.1.4.1 getAllSources

Get the names of all configured sources of the reader.

Command

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

Response

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

`// opt` → Optional: line can be omitted.

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). The default configuration has only one source named: Source_1

3.1.4.2 triggerSource

Trigger the source to perform a read action. Data of detected tags will be sent to the client application as a TagEventReport.

The configuration parameter of the source ("Read cycles per trigger", "Read timeout", "smoothing"...) and the definition which data of a tag (tagfields, RSSI Value...) is sent, are used from the stored configuration. Changing of these parameters is only possible via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)").

Command

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

Response

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

For negative response (non-zero resultCode) see chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). The default configuration has only one source named: Source_1
value_triggerMode	Fixed values	Single Start Stop	Mode of trigger.
			Single Performs one single trigger of the source.

Parameter	Type	Values	Description
			Start The source will be continuously triggered until a stop command is performed.
			Stop Stops triggering the source. Be aware that this command only acts on a previous trigger source commands. It doesn’t stops any continuous trigger defined in the configuration.

3.1.4.3 readTagIDs

Perform a read cycle at the requested source and return all detected tags in the response telegram.

If no tag was detected the function will return with a positive response but without any tag data.

Be aware that all filters (configured via tag selector at the source see chapter "Configuration settings via (Page 9)") will be active. Only tags that survive the filtering will be in the response telegram.

Command

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

Response

```
<frame>
 <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <readTagIDs>
        <returnValue>
            <tag>
                <tagID> value_tagID </tagID>
                <utcTime> value_utcTime </utcTime>   // opt
                <antennaName> value_antennaName </antennaName>  // opt
                <rSSI> value_rSSI </rSSI>            // opt
            </tag>
            …
            <tag>          // opt
                …
            </tag>         // opt
        </returnValue>
    </readTagIDs>
 </reply>
</frame>
```

`// opt` → Optional: line can be omitted.

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). The default configuration has only one sources named: Source_1
value_duration	Decimal value 0..9	0..65535	Optional Duration time in ms. Determines how long the requested source should read tags. A value of 0 or no parameter at all, performs only a single read cycle. The command will be active for the complete duration time. This has to be considered for the implementation of the client application, where typically a timeout monitoring is implemented. The configuration parameter of the source (Read cycles per trigger, read timeout... chapter "Configuration settings via (Page 9)") won't be used.
value_tagID	Hexadecimal value 0..9, A…F	-	Tag ID in "RAW Hex Data Format" Example for a 96 Bit tag ID: 3005FB63AC1F3681EC880468 For more information see EPC global specification.
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00. Whether this parameter is transmitted depends on the loaded configuration. The configuration must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). A configuration is created with the SIMATIC RF-MANAGER Basic

Parameter	Type	Values	Description
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). A configuration is created with the SIMATIC RF-MANAGER Basic.
value_rSSI	Decimal value 0..9	0..255	RSSI value. This is value without any unit and without any direct relationship to the power level. The configurations must contain the corresponding "Field Name" in its "Data Selector" of its "RFID Device" (see chapter "Configuration settings via (Page 9)"). A configuration is created with the SIMATIC RF-MANAGER Basic.

3.1.4.4 writeTagID

Write a new ID onto a tag. There is only a single valid tag allowed in the antenna field, otherwise the function returns with a negative response.

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` → Optional: line can be omitted.

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). The default configuration has only one source named: Source_1
value_tagID	Hexadecimal value 0..9, A…F	-	Optional. Tag ID in "RAW Hex Data Format". Either this Function will work on tags with this ID. Or if this parameter is empty or not transmitted at all, the function works on any tag. But in this case only a single tag is allowed within the antenna field. Example for a 96 Bit tag ID: 3005FB63AC1F3681EC880468 For more information see EPC global specification.
value_newID	Hexadecimal value 0..9, A…F	-	New Tag ID in "RAW Hex Data Format" which should be written onto the tag. Example for a 96 Bit tag ID: 3005FB63AC1F3681EC880468 For more information see EPC global specification.

Parameter	Type	Values	Description
value_idLength	Decimal value 0..9	16, 32, 48…496	Optional. Length of the new tag ID in bit. If this parameter exists, the reader checks the correct length of the "value_newID". If the check fails, the function returns with a negative response. Without this parameter, the new tag ID is only be checked if the length is a multiple of 16 Bits.
value_password	Hexadecimal value 0..9, A…F	00000000…. FFFFFFFF	Optional. Access password of the tag. Parameter can be omitted, if the tag has no activated password protection.

3.1.4.5 readTagMemory

Read data from the requested tag.

If no tag ID is supplied, the function will work on all tags detected by the given source. Be aware that all filters (configured via tag selector at the source see chapter "Configuration settings via (Page 9)") will be active. Only tags that survive the filtering will be in the response telegram if no tag ID is given.

The response telegram will include each detected tag together with information whether the requested data could be read for the tag or not.

If no tag was detected the function will return with a positive response but without any tag data.

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>
            <startAddress> value_startAddress </startAddress>
            <dataLength> value_dataLength </dataLength>
        </tagField>
        …
        <tagField>         // opt
            …
        </tagField>        // opt
    </readTagMemory>
 </cmd>
</frame>
```

`// opt` → Optional: line can be omitted.

Response

```
<frame>
 <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <readTagMemory>
        <returnValue>
            <tag>
                <tagID> value_tagID </tagID>
                <success> value_success </success>
                <utcTime> value_utcTime </utcTime>  // opt
                <antennaName> value_antennaName </antennaName>  // opt
                <rSSI> value_rSSI </rSSI>           // 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
        </returnValue>
    </readTagMemory>
 </reply>
</frame>
```

`// opt` → Optional: line can be omitted.

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). The default configuration has only one source named: Source_1
value_tagID	Hexadecimal value 0..9, A…F	-	Optional Tag ID in "RAW Hex Data Format". Either this Function will work on tags with this ID. Or if this parameter is empty, or not transmitted at all, the source tries to read from all recognized tags. Example for a 96 Bit tag ID: 3005FB63AC1F3681EC880468 For more information see EPC global specification.
value_bank	-	0..3	Memory bank of tag. 0: Reserved 1: EPC 2: TID 3: USER MEMORY
value_startAddress	Decimal value 0..9	0…65535	Number of the first byte in the given memory bank which should be read.
value_dataLength	Decimal value 0..9	1…510	Number of bytes which should be read.
value_password	Hexadecimal value 0..9, A…F	00000000…. FFFFFFFF	Optional. Access password of the tag. Parameter can be omitted, if the tag has no activated password protection.
value_success	Fixed values	True False	Flag indicating if function was successful for this tag. True: Function successful False: Function failed

Parameter	Type	Values	Description
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00. Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.

Parameter	Type	Values	Description
value_rSSI	Decimal value 0..9	0..255	RSSI value. This is a value without any unit and without any direct relationship to the power level. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_data	Hexadecimal value 0..9, A…F	-	Data which should be read. Each Byte is represented by two hexadecimal characters. Example: The Byte order 0x12, 0x34, 0xA3 is represented in the parameter value_data as charactersting: 1234A3. The value_dataLength is in this example is 3. If the function detects the tag, but could not read the data data (e.g. if the tag has no requested user memory at all) , this field will be empty. In this case the value_success will also be set to False

3.1.4.6 writeTagMemory

Write data onto the requested tag.

If no tag ID is supported the function will work on all the source detected tags. Be aware that all filters (configured via tag selector at the source see chapter "Configuration settings via (Page 9)") will be active in this case. Only tags that survive the filtering will be in the response telegram.

The response telegram will include each detected tag. Each tag is flagged if the function was successful for this tag or not.

If no tag was detected the function will return with a negative response.

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>
            <startAddress> value_startAddress </startAddress>
            <dataLength> value_dataLength </dataLength>
            <data> value_data </data>         // opt
        </tagField>
        …
        <tagField>        // opt
            …
        </tagField>       // opt
    </writeTagMemory>
 </cmd>
</frame>
```

`// opt` → Optional: line can be omitted.

Response

```
<frame>
 <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <writeTagMemory>
        <returnValue>
            <tag>
                <tagID> value_tagID </tagID>
                <success> value_success </success>
                <utcTime> value_utcTime </utcTime>      // opt
                <antennaName> value_antennaName </antennaName>  // opt
                <rSSI> value_rSSI </rSSI>               // opt
            </tag>
            …
            <tag>           // opt
                …
            </tag>          // opt
        </returnValue>
    </writeTagMemory>
 </reply>
</frame>
```

`// opt` → Optional: line can be omitted.

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). The default configuration has only one source named: Source_1
value_tagID	Hexadecimal value 0..9, A…F	-	Optional Tag ID in "RAW Hex Data Format". Either this Function will work on tags with this ID. Or if this parameter is empty or not transmitted at all, the source tries to manage all recognized tags. Example for a 96 Bit tag ID: 3005FB63AC1F3681EC880468 For more information see EPC global specification.
value_bank	-	0..3	Memory bank of tag. 0: Reserved 1: EPC 2: TID 3: USER MEMORY
value_startAddress	Decimal value 0..9	0...65535	Number of the first byte in the given memory bank which should be written.
value_dataLength	Decimal value 0..9	1...510	Number of bytes which should be written. The reader checks the correct length of the "value_data"... If the check fails, the function returns with a negative response.
value_password	Hexadecimal value 0..9, A…F	00000000…. FFFFFFFF	Optional. Access password of the tag. Parameter can be omitted, if the tag has no activated password protection.

Parameter	Type	Values	Description
value_data	Hexadecimal value 0..9, A…F	-	Data which should be written. Each Byte is represented by two hexadecimal characters. Example: The Byte order 0x12, 0x34, 0xA3 is represented in the parameter value_data as character string: 1234A3. The value_dataLength in this example is 3.
value_success	Fixed Values	True False	Flag indicating if function was successful for this tag. True: Function successful False: Function failed
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00. Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_rSSI	Decimal value 0..9	0..255	RSSI value. This is a value without any unit and without any direct relationship to the power level. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.

3.1.4.7 readTagField

Read data from the requested tag. The address of the data region is determined by the name of a TagField. The TagField and its name are configured via the SIMATIC RF-MANAGER Basic (by means of a TagSelector).

The following predefined fieldnames could be used in any case:

- tagId
- accessPassword
- killPassword

If no tag ID is supported the function will work on all from the source detected tags. Be aware that all filters (configured via tag selector at the source see chapter "Configuration settings via (Page 9)") will be active in this case. Only tags that survive the filtering will be in the response telegram.

The response telegram will include each detected tag. Each tag is flagged if the function was successful for this tag or not.

If no tag was detected the function will return with a positive response but without any tag data.

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: line can be omitted.

Response

```
<frame>
 <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <readTagField>
        <returnValue>
            <tag>
                <tagID> value_tagID </tagID>
                <success> value_success </success>
                <utcTime> value_utcTime </utcTime>              // opt
                <antennaName> value_antennaName </antennaName>  // opt
                <rSSI> value_rSSI </rSSI>       // 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
        </returnValue>
    </readTagField>
 </reply>
</frame>
```

`// opt` → Optional: line can be omitted.

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Negative Reply (Page 96)"). The default configuration has only one source named: Source_1
value_tagID	Hexadecimal value 0..9, A…F	-	Optional Tag ID in "RAW Hex Data Format". Either this Function will work on tags with this ID. Or if this parameter is empty or not transmitted at all, the source tries to manage all recognized tags. Example for a 96 Bit tag ID: 3005FB63AC1F3681EC880468 For more information see EPC global specification.
value_fieldName	Text	-	Name of the tagField. Is defined within the SIMATIC RF-MANAGER Basic.
value_password	Hexadecimal value 0..9, A…F	00000000…. FFFFFFFF	Optional. Access password of the tag. Parameter can be omitted, if the tag has no activated password protection.
value_success	Fixed values	True False	Flag indicating if function was successful for this tag. True: Function successful False: Function failed
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00. Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.

Parameter	Type	Values	Description
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_rSSI	Decimal value 0..9	0..255	RSSI value. This is a value without any unit and without any direct relationship to the power level. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_bank	-	0..3	Memory bank of tag. 0: Reserved 1: EPC 2: TID 3: USER MEMORY
value_startAddress	Decimal value 0..9	0...65535	Number of the first byte in the given memory bank which should be read.
value_dataLength	Decimal value 0..9	1...510	Number of bytes which should be read.
value_data	Hexadecimal value 0..9, A…F	-	Data which is read. Each Byte is represented by two hexadecimal characters. Example: The Byte order 0x12, 0x34, 0xA3 is represented in the parameter value_data as charactersting: 1234A3. The value_dataLength is in this example is 3. If the function detects the tag, but could not read the data, this field will be empty. In this case the value_success will also be set to False.

3.1.4.8 writeTagField

Write data to the requested tag. The address of the data region is determined by the name of a TagField. The TagField and its name are configured via the SIMATIC RF-MANAGER Basic (by means of a TagSelector).

The following predefined fieldnames could be used in any case:

- tagId
- accessPassword
- killPassword

If no tag ID is supported the function will work on all from the source detected tags. Be aware that all filters (configured via tag selector at the source see chapter "Configuration settings via (Page 9)") will be active in this case. Only tags that survive the filtering will be in the response telegram.

The response telegram will include each detected tag. Each tag is flagged if the function was successful for this tag or not.

If no tag was detected the function will return with a negative response.

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: line can be omitted.

Response

```
<frame>
 <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <writeTagField>
        <returnValue>
            <tag>
                <tagID> value_tagID </tagID>
                <success> value_success </success>
                <utcTime> value_utcTime </utcTime>              // opt
                <antennaName> value_antennaName </antennaName>  // opt
                <rSSI> value_rSSI </rSSI>              // opt
            </tag>
            …
            <tag>      // opt
                …
            </tag>     // opt
        </returnValue>
    </writeTagField>
 </reply>
</frame>
```

`// opt` → Optional: line can be omitted.

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). The default configuration has only one source named: Source_1
value_tagID	Hexadecimal value 0..9, A…F	-	Optional Tag ID in "RAW Hex Data Format". Either this Function will work on tags with this ID. Or if this parameter is empty or not transmitted at all, the source tries to manage all recognized tags. Example for a 96 Bit tag ID: 3005FB63AC1F3681EC880468 For more information see EPC global specification.
value_fieldName	Text	-	Name of the tagField. Is defined within the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)").
value_password	Hexadecimal value 0..9, A…F	00000000…. FFFFFFFF	Optional. Access password of the tag. Parameter can be omitted, if the tag has no activated password protection.
value_data	Hexadecimal value 0..9, A…F	-	Data which should be written. Each Byte is represented by two hexadecimal characters. Example: The Byte order 0x12, 0x34, 0xA3 is represented in the parameter value_data as charactersting: 1234A3. The value_dataLength is in this example is 3.
value_success	Fixed values	True False	Flag indicating if function was successful for this tag. True: Function successful False: Function failed

Parameter	Type	Values	Description
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00. Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_rSSI	Decimal value 0..9	0..255	RSSI value. This is a value without any unit and without any direct relationship to the power level. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.

3.1.4.9 killTag

Kill the requested tag.

If no tag ID is supported the function will work on all from the source detected tags. Be aware that all filters (configured via tag selector at the source see chapter "Configuration settings via (Page 9)") will be active in this case. Only tags that survive the filtering will be in the response telegram.

The response telegram will include each detected tag. Each tag is flagged if the function was successful for this tag or not.

If no tag was detected the function will return with a negative response.

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: line can be omitted.

Response

```
<frame>
 <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <killTag>
        <returnValue>
            <tag>
                <tagID> value_tagID </tagID>
                <success> value_success </success>
                <utcTime> value_utcTime </utcTime>              // opt
                <antennaName> value_antennaName </antennaName>  // opt
                <rSSI> value_rSSI </rSSI>            // opt
            </tag>
            …
            <tag>       // opt
                …
            </tag>      // opt
        </returnValue>
    </killTag>
 </reply>
</frame>
```

`// opt` → Optional: line can be omitted.

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). The default configuration has only one source named: Source_1
value_tagID	Hexadecimal value 0..9, A…F	-	Optional Tag ID in "RAW Hex Data Format". Either this Function will work on tags with this ID. Or if this parameter is empty or not transmitted at all, the source tries to manage all recognized tags. Example for a 96 Bit tag ID: 3005FB63AC1F3681EC880468 For more information see EPC global specification.
value_password	Hexadecimal value 0..9, A…F	00000000…. FFFFFFFF	Kill password of the tag.
value_success	Fixed values	True False	Flag indicating if function was successful for this tag. True: Function successful False: Function failed
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00. Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.

Parameter	Type	Values	Description
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_rSSI	Decimal value 0..9	0..255	RSSI value. This is a value without any unit and without any direct relationship to the power level. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.

3.1.4.10 lockTagBank

Lock a complete bank of the requested tag.

If no tag ID is supported the function will work on all from the source detected tags. Be aware that all filters (configured via tag selector at the source see chapter "Configuration settings via (Page 9)") will be active in this case. Only tags that survive the filtering will be in the response telegram.

The response telegram will include each detected tag. Each tag is flagged if the function was successful for this tag or not.

If no tag was detected the function will return with a negative response.

Below is a short description of the parameters epcGen2LockAction and epcGen2LockMask. The first line of the table ('Bit') defines the bit positions of the action and the mask values. The mask and action values are given with the MSB first.

Bank	Kill PWD		Access PWD		EPC Memory		TID Memory		User Memory	
Bit	9	8	7	6	5	4	3	2	1	0
Mask	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

The mask value determines which of the action value bits should be applied as shown in the table above as s/w (skip/write with skip=0 and write=1).

The action value determines how the bank should be locked for each memory bank.

The flag r/w defines a password lock for read/write.

The flag w defines a password lock for write (read access allowed).

The p flag defines a permanent lock.

The following tables list possible combinations of r/w and w flags with p flags and their combined meanings for one memory bank.

The tag is in the open state if it is identified and in the secured state if its access password is verified.

w	p	Description
0	0	Associated memory bank can be write-accessed from either open or secured state.
0	1	Associated memory bank can be permanently write-accessed from either open or secured state and can never be locked.
1	0	Associated memory bank can be write-accessed from the secured state but not from the open state.
1	1	Associated memory bank cannot be write-accessed from any state.

r/w	p	Description
0	0	Associated password location can be read and write-accessed from either open or secured state.
0	1	Associated password location can be permanently read and write-accessed from either open or secured state and can never be locked.
1	0	Associated password location can be read and write-accessed from the secured state but not from the open state.
1	1	Associated memory bank cannot be read or write-accessed from any state.

Example

Bank	Kill PWD	Access PWD	EPC Memory	TID Memory	User Memory	Hex String Value
Mask	(00) 1 1	1 1	1 1	0 0	0 0	3F0
Action	(00) 1 0	1 0	1 0	0 0	0 0	2A0

In the above example, the lockMask is 11 1111 0000 (hexadecimal 3F0). This means that you can only write action bits to the Kill, Access and EPC memory locations. The lockAction fields are 10 1010 0000 (hexadecimal 2A0) which results in the following:

- Kill Password:

  Can be read and write-accessed from the secured state but not from the open state. The Access Password of the tag must be known before the kill password can be read or changed.

- EPC Memory Bank:

  Can be write-accessed from the secured state but not from the open state. The Access Password must be known before a new ID can be written to the tag. For detailed information about the epcGen2LockAction and epcGen2LockMask, see 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>
    </lockTag>
 </cmd>
</frame>
```

`// opt` → Optional: line can be omitted.

Response

```
<frame>
 <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <lockTag>
        <returnValue>
            <tag>
                <tagID> value_tagID </tagID>
                <success> value_success </success>
                <utcTime> value_utcTime </utcTime>              // opt
                <antennaName> value_antennaName </antennaName>  // opt
                <rSSI> value_rSSI </rSSI>             // opt
            </tag>
            …
            <tag>       // opt
                …
            </tag>      // opt
        </returnValue>
    </lockTag>
 </reply>
</frame>
```

`// opt` → Optional: line can be omitted.

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). The default configuration has only one source named: Source_1
value_tagID	Hexadecimal value 0..9, A…F	-	Optional Tag ID in "RAW Hex Data Format". Either this Function will work on tags with this ID. Or if this parameter is empty or not transmitted at all, the source tries to manage all recognized tags. Example for a 96 Bit tag ID: 3005FB63AC1F3681EC880468 For more information see EPC global specification.
value_lockAction	Bool value 0, 1	0000000000… 1111111111	The Lock Action is a Boolean value with 10 digits. The Least significant bits define the USER Memory. For complete definition of the lock action see EPC global specification.
value_lockMask	Bool value 0, 1	0000000000… 1111111111	The Lock Mask is a Boolean value with 10 digits. The Least significant bits define the USER Memory. For a complete definition of the lock mask see EPC global specification.
value_password	Hexadecimal value 0..9, A…F	00000000…. FFFFFFFF	Access password of the tag.
value_success	Fixed values	True False	Flag indicating if function was successful for this tag. True: Function successful False: Function failed

Parameter	Type	Values	Description
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00. Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_rSSI	Decimal value 0..9	0..255	RSSI value. This is a value without any unit and without any direct relationship to the power level. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.

3.1.4.11 nXP_SetReadProtect

Set read protection on specific NXP-manufactured tags.

A read-protected tag returns only zeros as a tag ID. The TID and the User memory bank are no longer accessible.

The tag must have a valid access password (non zero). Otherwise the function returns with negative response.

If no tag ID is supported the function will work on all from the source detected tags. Be aware that all filters (configured via tag selector at the source see chapter "Configuration settings via (Page 9)") will be active in this case. Only tags that survive the filtering will be in the response telegram.

The response telegram will include each detected tag. Each tag is flagged if the function was successful for this tag or not.

If no tag was detected the function will return with a negative response.

Command

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

`// opt` → Optional: line can be omitted.

Response

```
<frame>
 <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <nXP_SetReadProtect>
        <returnValue>
            <tag>
                <tagID> value_tagID </tagID>
                <success> value_success </success>
                <utcTime> value_utcTime </utcTime>              // opt
                <antennaName> value_antennaName </antennaName>  // opt
                <rSSI> value_rSSI </rSSI>           // opt
            </tag>
            …
            <tag>           // opt
                …
            </tag>          // opt
        </returnValue>
    </nXP_SetReadProtect>
 </reply>
</frame>
```

`// opt` → Optional: line can be omitted.

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). The default configuration has only one source named: Source_1
value_tagID	Hexadecimal value 0..9, A…F	-	Optional Tag ID in "RAW Hex Data Format". Either this Function will work on tags with this ID. Or if this parameter is empty or not transmitted at all, the source tries to manage all recognized tags. Example for a 96 Bit tag ID: 3005FB63AC1F3681EC880468 For more information see EPC global specification.
value_password	Hexadecimal value 0..9, A…F	00000000…. FFFFFFFF	Access password of the tag.
value_success	Fixed values	True False	Flag indicating if function was successful for this tag. True: Function successful False: Function failed
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00. Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.

Parameter	Type	Values	Description
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_rSSI	Decimal value 0..9	0..255	RSSI value. This is a value without any unit and without any direct relationship to the power level. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.

3.1.4.12 nXP_ResetReadProtect

Reset read protection on specific NXP-manufactured tags.

A read-protected tag returns only zeros as a tag ID. The TID and the User memory bank are no longer accessible.

The tag must have a valid access password (non zero). Otherwise the function returns with negative response.

Because protected tags don’t return its ID the function will work on all from the source detected tags. Be aware that all filters (configured via tag selector at the source see chapter "Configuration settings via (Page 9)") will be active in this case. Only tags that survive the filtering will be in the response telegram.

The response telegram will include each detected tag. Each tag is flagged if the function was successful for this tag or not.

If no tag was detected the function will return with a negative response.

Command

```
<frame>
 <cmd>
    <id> value_id </id>
    <nXP_ResetReadProtect>
        <sourceName> value_sourceName </sourceName>
        <password> value_password </password>
    </nXP_ResetReadProtect>
 </cmd>
</frame>
```

Response

```
<frame>
 <reply>
    <id> value_id </id>
    <resultCode> 0 </resultCode>
    <nXP_ResetReadProtect>
        <returnValue>
            <tag>
                <tagID> value_tagID </tagID>
                <success> value_success </success>
                <utcTime> value_utcTime </utcTime>              // opt
                <antennaName> value_antennaName </antennaName>  // opt
                <rSSI> value_rSSI </rSSI>             // opt
            </tag>
            …
            <tag>         // opt
                …
            </tag>        // opt
        </returnValue>
    </nXP_ResetReadProtect>
 </reply>
</frame>
```

`// opt` → Optional: line can be omitted.

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). The default configuration has only one source named: Source_1
value_password	Hexadecimal value 0..9, A…F	00000000…. FFFFFFFF	Access password of the tag.
value_success	Fixed values	True False	Flag indicating if function was successful for this tag. True: Function successful False: Function failed
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00. Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.

Parameter	Type	Values	Description
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna Whether this parameter is transmitted depends on the loaded configuration. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.
value_rSSI	Decimal value 0..9	0..255	RSSI value. This is a value without any unit and without any direct relationship to the power level. The configurations must contain the corresponding "Field Name" in its "Data Selector" (see chapter "Configuration settings via (Page 9)"). Create such a configuration with the SIMATIC RF-MANAGER Basic.

3.2 Reports

This chapter describes all possible reports which may be sent from the reader to the client application.

Reports are able to work in a buffering and a non-buffering mode. This behavior is configured with the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)").

In non-buffering mode all reports are sent to the client application in a ‘fire and forget’ manner, i.e. without waiting for an acknowledging telegram. As a consequence the report will be thrown away without any notice if the connection to the client application doesn’t exist or if it is broken.

In buffering mode each report must be acknowledged by the client application with a response telegram. if no acknowledge telegram is received within 10 seconds, the reader will sent the report again to the client application. In case of connection errors, or no connection at all, the reports will be stored at the reader until the connection is established again.

A maximum of 1000 reports can be buffered at the reader. Whenever this maximum is reached new reports will be thrown away.

Reports are divided into events and alarms. An event contains autonomous collected data from the reader. Alarms informs about abnormal conditions of the reader.

3.2.1 Events

Any data that is sent asynchronously from the reader is reported as event.

3.2.1.1 tagEventReport

A source which is triggered collects tag data which is sent via a tag event report to the client application.

The configuration of the trigger (Continous, Application Request, IO Level...) and the definition which data of a tag (tagfields, RSSI Value...) is send, are used from the stored configuration. Changing of these parameters is only possible via the SIMATIC RF-MANAGER Basic.

The tag data inside each event report is grouped by the corresponding sources.

A response telegram from the client application is only necessary if buffering mode is configured. Nevertheless, sending a response telegram in non buffering mode is possible without any negative effect.

Report

```
<frame>
 <report>
    <id> value_id </id>
    <ter>
        <source>
            <sourceName> value_sourceName </sourceName>
            <tag>
                <tagID> value_tagID </tagID>
                <event> value_event </event>                    // opt
                <utcTime> value_utcTime </utcTime>              // opt
                <antennaName> value_antennaName </antennaName>  // opt
                <rSSI> value_rSSI </rSSI>           // 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: line can be omitted.

Response from Client

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_sourceName	Text	-	Name of the source. The name is defined via the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)"). The default configuration has only one source named: Source_1
value_tagID	Hexadecimal value 0..9 ,A…F	-	Optional Tag ID in "RAW Hex Data Format". Example for a 96 Bit tag ID: 3005FB63AC1F3681EC880468
value_event	Fixed Values	New Glimpsed Observed Lost	Tag events are generated through smoothing algorithm in the source. Refer to the SIMATIC RF-MANAGER Basic for more information. There could be more than one event for the same tag in a single report. Each individual event has its own tag structure. Whenever a report includes two events for the same tag, there will be two tag structures with the same value_tagID but different value_event. In SIMATIC RF-MANAGER Basic you configure with the "Event filters" of the corresponding "Data Selector" which events will be sent from a specific source (see chapter "Configuration settings via (Page 9)").
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00.

Parameter	Type	Values	Description
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna Whether this parameter is transmitted depends on the loaded configuration. In SIMATIC RF-MANAGER Basic the entry "Sending Antenna" inside the "Field names" of the "Data Selector" of the corresponding source must be selected (see chapter "Configuration settings via (Page 9)").
value_rSSI	Decimal value 0..9	0..255	RSSI value. This is a value without any unit and without any direct relationship to the power level. In SIMATIC RF-MANAGER Basic the entry "RSSI values" inside the "Field names" of the "Data Selector" of the corresponding source must be selected (see chapter "Configuration settings via (Page 9)").
value_fieldName	Text	-	Name of the tagField. Is defined within the SIMATIC RF-MANAGER Basic (see chapter "Configuration settings via (Page 9)").
value_bank	-	0..3	Memory bank of tagField 0: Reserved 1: EPC 2: TID 3: USER MEMORY
value_startAddress	Decimal value 0..9	0…65535	Number of the first byte in the given memory bank which have been read.
value_dataLength	Decimal value 0..9	1...510	Number of bytes which have been read.
value_data	Hexadecimal value 0..9 ,A…F	-	Data which is read. Each Byte is represented by two hexadecimal characters. Example: The Byte order 0x12, 0x34, 0xA3 is represented in the parameter value_data as charactersting: 1234A3. The value_dataLength in this example is 3. If the source detects the tag, but could not read the data (e.g. if the tag has no requested user memory at all) , this field will be empty.

3.2.1.2 rssiEventReport

An RSSI event report signals a change of RSSI Values when reading tags.

The type and amount of data is determined by the settings of the base configuration created with the SIMATIC RF-MANAGER Basic.

Report

```
<frame>
 <report>
    <id> value_id </id>
    <rssier>
        <tag>
            <tagID> value_tagID </tagID>
            <utcTime> value_utcTime </utcTime>
            <antennaName> value_antennaName </antennaName>
            <rSSI> value_rSSI </rSSI>
        </tag>
        …
        <tag>      // opt
            …
        </tag>     // opt
    </rssier>
 </report>
</frame>
```

`// opt` → Optional: line can be omitted.

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_tagID	Hexadecimal value 0..9 ,A…F	-	Optional Tag ID in "RAW Hex Data Format". Example for a 96 Bit tag ID: 3005FB63AC1F3681EC880468
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00.
value_antennaName	Fixed values	Antenna01 Antenna02 Antenna03 Antenna04	Name of Antenna
value_rSSI	Decimal value 0..9	0..255	RSSI value. This is a value without any unit and without any direct relationship to the power level.

3.2.1.3 ioEventReport

An IO event report signals a change of an in port or out port level.

Inside SIMATIC RF-MANAGER Basic each port can be configured to send io events individually (see chapter "Configuration settings via (Page 9)").

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` → Optional: line can be omitted.

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_ioName	Fixed values	Inport00 Inport01 Inport02 Inport03 Outport00 Outport01 Outport02 Outport03	Name of the port.
value_ioEvent	Fixed values	High Low	Indicates the new state of the port.
value_utcTime	Time	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00.

3.2.2 Alarm

Asynchronous messages report a status, warning or malfunction of the reader.

The meaning of an alarm is identified by its error number and optionally some additional parameters. See "Listing of possible alarms".

Report

```
<frame>
 <alarm>
    <id> value_id </id>
    <error>
        <utcTime>value_utcTime </utcTime>
        <errorNumber>value_errorNnumber </errorNumber>
        <errorText>value_errorText </errorText>
        <???>value_param_xy </???>              // opt
        …
    </error>
 </alarm>
</frame>
```

`// opt` → Optional: line can be omitted.

Response

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

Error codes for negative responses (non-zero resultCode) are listed in chapter "Negative Reply (Page 96)".

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_errorNumber	Decimal value 0..9	0…65535	Error number. Description see below
value_utcTime	Text	-	Universal Time Coordinated timestamp in ISO 8601 format: yyyy-MM-ddTHH:mm:ss.fffzzz e.g.: 2009-12-24T18:34:56.929+00:00.
value_errorText	Text	-	Message text of the alarm.
value_param_xy	Text	-	Additional parameter. The name of the parameter is the xml tag. Depending on the error number an alarm has different quantity of parameters.

Listing of possible Alarms

errorNumber (decimal)	Text
31001	No data available. <commandName>
31002	CRC-Error. <commandName>
31003	Required License not available <commandName>
31004	Invalid value <value> <commandName>
31005	Invalid Tag-Standard. - <CommandName>
31006	Currently no antenna activated for measurement. - <CommandName>
31007	Currently no frequency activated for measurement. - <CommandName>
31008	Currently no carrier activated for measurement. - <CommandName>
31009	Antenna-Error. - <CommandName>
31010	No tag in field. - <CommandName>
31011	More than one tag in field. - <CommandName>
31012	Wrong Licence key. - <CommandName>
31013	Activation of new loaded firmware failed. - <CommandName>
31014	Command not allowed in current mode. - <CommandName>
31015	Tag with referenced Handle not available. - <CommandName>
31016	Currently no radio profile activated for measurement. - <CommandName>
31128	Unspecific return value. - <CommandName>
31201	No tag response. ID: <TagID> Antenna: < ReadPointName> - <CommandName>
31202	Access to tag denied. ID: <TagID> Antenna: <ReadPointName> - <CommandName>
31203	Verification of tag failed. ID: <TagID> Antenna: <ReadPointName> - <CommandName>
31204	Tag responded with errorcode UNSPECIFIC. ID: <TagID> Antenna: <ReadPointName> - <CommandName>
31328	Tag responded with errorcode OTHER. ID: <TagID> Antenna: <ReadPointName> - <CommandName>
31331	Tag memory not available or invalid protocol control word. ID: <TagID> Antenna: <ReadPointName> - <CommandName>
31332	Tag memory is locked. ID: <TagID> Antenna: <ReadPointName> - <CommandName>
31339	Tag has insufficient power for writing. ID: <TagID> Antenna: <ReadPointName> - <CommandName>
31343	Unspecific tag error. ID: <TagID> Antenna: <ReadPointName> - <CommandName>
31400	No free channel for sending available.
31401	Antenna error at port ANT 1.
31402	Antenna error at port ANT 2.
31403	Antenna error at port ANT 3.
31404	Antenna error at port ANT 4.
31409	Invalid power setting for ETSI at port ANT 1.
31410	Invalid power setting for ETSI at port ANT 2.
31411	Invalid power setting for ETSI at port ANT 3.

errorNumber (decimal)	Text
31412	Invalid power setting for ETSI at port ANT 4.
31601	Overflow - Alarms will be discarded
31602	Overflow of Alarms ended
31603	Overflow - Tagevents will be discarded
31604	Overflow of Tagevents ended
500050	Connection request failed
500051	Configuration successful
500052	Connection established
500053	Connection broken

3.3 Negative Reply

Malfunctions during command execution will be signaled with a non zero resultCode inside the reply message. Additional information about the malfunction will be also transmitted.

Response

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

Parameters

Parameter	Type	Values	Description
value_id	Decimal value 0..9	0..4294967295	Unique identification of command.
value_resultCode	Decimal value 0..9	0…65535	Identification number for the malfunction. Listing of all possible return codes see table below
value_name	Text	-	Textual name of the malfunction List of all possible names see table below.
value_cause	Text	-	Short information about the real cause for sending this negative response.

Listing of possible result codes

Result Code (decimal)	Name	Description
0	NO_ERROR	Positive response.
1	ERROR_UNKNOWN	Unspecified error.
2	ERROR_COMMAND_NOT_SUPPORTED	The command is not supported or unknown to the reader.
3	ERROR_PARAMETER_INVALID_FORMAT	A parameter has the wrong format.
4	ERROR_PARAMETER_MISSING	A required parameter is not specified.
5	ERROR_PARAMETER_INVALID_DATATYP	Not supported. A parameter has the wrong format.
6	ERROR_PARAMETER_ILLEGAL_VALUE	Using a parameter value that is illegal.
7	ERROR_PARAMETER_OUT_OF_RANGE	A parameter value is out of range.
8	ERROR_PARAMETER_NOT_SUPPORTED	A parameter is not supported by this reader.
9	ERROR_PARAMETER_LENGTH_EXCEEDED	The length of the given parameter was too long.
10	ERROR_TOO_MANY_TAGSELECTORS	Reader Internal Error Raised when trying to add more than the maximum number of tag selectors.
11	ERROR_TOO_MANY_TRIGGERS	Reader internal Error Raised when trying to add more than the maximum number of triggers.
12	ERROR_TAGSELECTOR_NOT_FOUND	Reader internal Error Raised when the given tag selector parameter is not known.
13	ERROR_TRIGGER_NOT_FOUND	Reader internal Error Raised when the given trigger parameter is not known.
14	ERROR_READPOINT_NOT_FOUND	Reader internal Error Raised when the given read point parameter is not known.
15	ERROR_CHANNEL_NOT_FOUND	Reader internal Error Raised when the given channel parameter is not known.
16	ERROR_SOURCE_NOT_FOUND	Raised when the given source parameter is not known.
17	ERROR_SOURCE_READ_ONLY	Reader internal Error Raised when trying to remove asource that cannot be removed, e.g. if it is one of the preconfigured sources.
18	ERROR_DATASELECTOR_NOT_FOUND	Reader internal Error Raised when the given DataSelector parameter is not known.

Result Code (decimal)	Name	Description
19	ERROR_TAGS_DETECTED_NOTREAD	Tags were detected by the reader, but cannot be read successfully.
20	ERROR_NO_TAG	No tag was in the field of the reader when trying to write or kill.
21	ERROR_TAG_LOCKED	The tag to be written or killed is locked.
22	ERROR_WRONG_PASSWORD	The given access password or the kill code is wrong.
23	ERROR_MULTIPLE_TAGS	Multiple tags were detected in the range of the reader when trying to kill.
24	ERROR_READONLY_TAG	The tag was a read-only tag that cannot be written.
25	ERROR_OBJECT_EXISTS	Reader internal Error Raised when trying to create an object and an object of the same type with the same name exists already.
26	ERROR_READPOINT_IN_USE	Reader internal Error The reader does not allow ReadPoints shared between Sources and the ReadPoint is already assigned to another Source.
27	ERROR_TAGFIELD_NOT_FOUND	Reader internal Error Raised when the given tag field parameter is not known
28	ERROR_PARSING_MESSAGE	Raised when an error occurs parsing a message (e.g. malformed message).
256	ERROR_READER_STOPPED	Reader is in stop. Command not allowed in stop mode.
257	ERROR_CONFIGURATION_ INCOMPATIBLE	Active configuration not usable via RFID reader XML interface.
4096	ERROR_IOPORT_NOT_FOUND	Referenced IO port not known to the reader.
4098	ERROR_INVALID_ADMIN_STATUS	Reader doesn't accept commands. Either connection is not established -> call hostGreetings() first, or reader is still loading configuration. Try later on.

3.4 References

- SIMATIC RF-MANAGER Basic
- System Manual RF600