

Sigfox Extension User Guide

Version 1.0

**Copyright © 2016 PTC Inc. and/or Its Subsidiary Companies. All Rights Reserved.**

User and training guides and related documentation from PTC Inc. and its subsidiary companies (collectively “PTC”) are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed software user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the PTC copyright notice and any other proprietary notice provided by PTC. Training materials may not be copied without the express written consent of PTC. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes.

Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC.

UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION. PTC regards software piracy as the crime it is, and we view offenders accordingly. We do not tolerate the piracy of PTC software products, and we pursue (both civilly and criminally) those who do so using all legal means available, including public and private surveillance resources. As part of these efforts, PTC uses data monitoring and scouring technologies to obtain and transmit data on users of illegal copies of our software. This data collection is not performed on users of legally licensed software from PTC and its authorized distributors. If you are using an illegal copy of our software and do not consent to the collection and transmission of such data (including to the United States), cease using the illegal version, and contact PTC to obtain a legally licensed copy.

**Important Copyright, Trademark, Patent, and Licensing Information:** See the About Box, or copyright notice, of your PTC software.

**United States Governments Rights**

PTC software products and software documentation are “commercial items” as that term is defined at 48 C.F.R. 2.101. Pursuant to Federal Acquisition Regulation (FAR) 12.212 (a)-(b) (Computer Software) (MAY 2014) for civilian agencies or the Defense Federal Acquisition Regulation Supplement (DFARS) at 227.7202-1 (a) (Policy) and 227.7202-3 (a) (Rights in commercial computer software or commercial computer software documentation) (FEB 2014) for the Department of Defense, PTC software products and software documentation are provided to the U.S. Government under the PTC commercial license agreement. Use, duplication or disclosure by the U.S. Government is subject solely to the terms and conditions set forth in the applicable PTC software license agreement.

**PTC Inc., 140 Kendrick Street, Needham, MA 02494 USA**



[Software Change Log 2](#_Toc442699506)

[Introduction and Installation 2](#_Toc442699507)

[About the Sigfox Extension 2](#_Toc442699508)

[Installing the Sigfox Extension 2](#_Toc442699509)

[Configuration and Usage 4](#_Toc442699510)

[Sigfox.GenericDeviceThingShape 4](#_Toc442699511)

[Sigfox.DevicesConnectedScheduler 4](#_Toc442699512)

[Sigfox.RestInterface 4](#_Toc442699513)

[Sigfox.CallbackInterface 5](#_Toc442699514)

[FrameUtilitiesThing 5](#_Toc442699515)

[Usage of the FrameUtilitiesThing 6](#_Toc442699516)

[Known Limitations and Compatibility 8](#_Toc442699517)

[Limitations 8](#_Toc442699518)

[Compatibility 8](#_Toc442699519)

[Document Revision History 8](#_Toc442699520)

# Software Change Log

|  |  |  |
| --- | --- | --- |
| **Version** | **Release Date** | **Changes** |
| 1.0 | 01/18/2016 | Initial Release |

# Introduction and Installation

Extensibility is a core aspect of the architecture and design of ThingWorx. Partners, third parties, and ThingWorx users can easily add new functionality into the system in a seamless manner. Extensions can be Service (function/method) libraries, Connector Templates, Widgets, and more.

This document provides installation and usage instructions for the Sigfox Extension.

# About the Sigfox Extension

The Sigfox Extension for the Thingworx Platform simplifies the work of delivering Sigfox enabled applications by providing an application template and some helper utilities. So, this extension provides a clean starting architecture to build on and extend. Included you can also find the Dataframe Utilities Extension which simplifies the work of encoding and decoding complex frames of data in order to communicate with low level devices. So, you can skip the complex part of doing bit parsing and in order to obtain all the fields data inside a frame, and instead, you just need to create an infortable that describes the frame format.

## Installing the Sigfox Extension

|  |  |
| --- | --- |
| 1. From a web browser, launch ThingWorx Composer. 2. Log into Composer as an administrator. |  |
| 1. Go to **Import/Export > Import**. |  |
| 1. Click **Choose File** and select SigfoxExtension.zip 2. Click **Import**.  Note: If an **Import Successful** message does not appear, contact your ThingWorx System Administrator. 3. Configure the PlatformSubsystem: Enable the “Allow Request Method Switch” and disable “Filter content type”. This changes are required for Sigfox Communication 4. StartSSL CA certificates imported into the global Java Keystore (required for communication to Sigfox). A short script on how this can be done can be found [here](https://gist.github.com/mohamed-el-habib/00deef599e8ba1cdbece). |  |
|  |  |
|  |  |
|  |  |

# 

# Configuration and Usage

## Sigfox.GenericDeviceThingShape

Figure The properties on a Sigfox.GenericDeviceThingShape

A generic thingshape with properties that you can apply on all your sigfox enabled things. This thing shape encapsulates properties like the timestamp of the last messages, the location of the device, state, sequence numbers, and much more. Some of the properties are logged, so be careful because you'll need to provide your own Value Stream. The thingshape also wraps some alerts: for example, messages received out of order and for lack of communication.

## Sigfox.DevicesConnectedScheduler

A Scheduler used to raise the alert that a **Sigfox.GenericDeviceThingShape** has not received messages.

## Sigfox.RestInterface

Maps a couple the most used REST API endpoints: will help with device type and device listing, callback creation, enrollment of new devices. Consult the services list for more info. To configure, just use you API credentials in the properties. Consult the service documentation for more details.

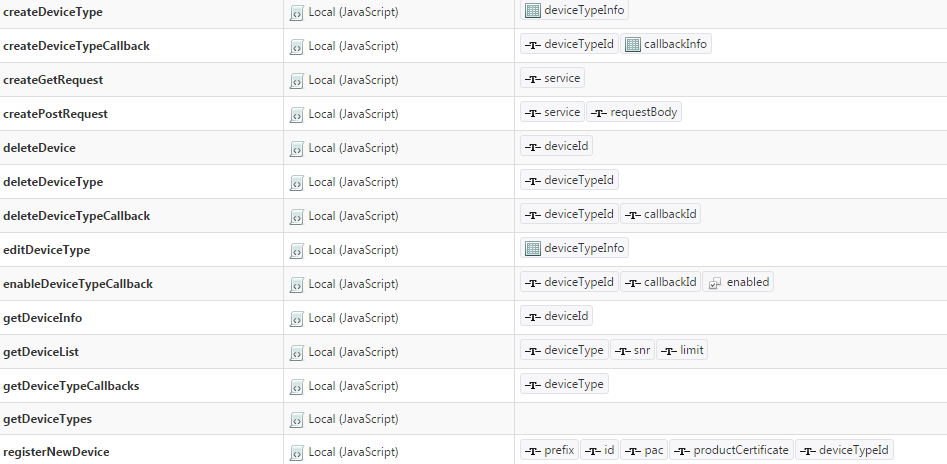


Figure The services implemented in the Sigfox.RestInterface

## Sigfox.CallbackInterface

This thing contains the service that is called by SIGFOX as a callback. On the sigfox side you will have to add a batch callback like this:

|  |  |
| --- | --- |
| **URL pattern** | [https://THINGWORX\_IP/Thingworx/Things/SigfoxInterface/Services/CallbackDataBatch?appKey=APP\_KEY&batch={batch}](https://thingworx_ip/Thingworx/Things/SigfoxInterface/Services/CallbackDataBatch?appKey=APP_KEY&batch=%7bbatch%7d) |
| **Line Pattern** | {time},{device},{snr},{rssi},{avgSignal},{station},{data},{lat},{lng},{seqNumber}; |

The service **CallbackDataBatch**will match the device in the callback to a Thing with the ​**Sigfox.GenericDeviceThingShape**, update its properties, and then call the **UpdateData**service, with the parameters **data**(the raw data from the device) and **receiveTimestamp**(when Sigfox says it received the data). The implementation of this service is application specific, so it is not included in this package. Its job is to parse the raw data and set device specific properties. Also, you can configure this service to also create Things dynamically if needed.

## FrameUtilitiesThing

In order to you use the **FrameUtilitiesThing**, you must examine the firstly create an Infotable using the **FrameFormatDataShape.** Inside this datashape you can define the properties for each of the fields of the frame.

The **FrameUtilitiesThing** thing provides the following services:

* Decode Frame: uses a frame format Infotable and a data string and returns a JSON with the decoded data. The data string represents a string with data in hexadecimal format, represented using big-endian. The result JSON contains the value for each of the fields.
* Encode Frame: uses a frame format Infotable, with all the properties set, and returns a string in hexadecimal format, using big-endian, for the data.

### Usage of the FrameUtilitiesThing

At the base of this extension is the **FrameFormatDataShape** datashape. Using infotables with this datashape you can map the fields of the frame.



Figure FrameFormatDataShape fields

For example, if you have a frame from devices described as the following table:

Table Sample frame definition

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **BITS** | **0-7** | **8** | **9-15** | **16-23** | **24-39** | **64-79** |
| **FIELD** | CODE | Conf Status | Frame counter | Transmit Period | Sensor Value (LSB) | Sensor Offset  (LSB) |

The infotable that describes the preceding table is presented in the following figure. You can see that a row is used for each of the fields in the frame.

The extension supports fields of data that span across multiple bytes, with a portion of the bits in one byte, and another portion in another byte.

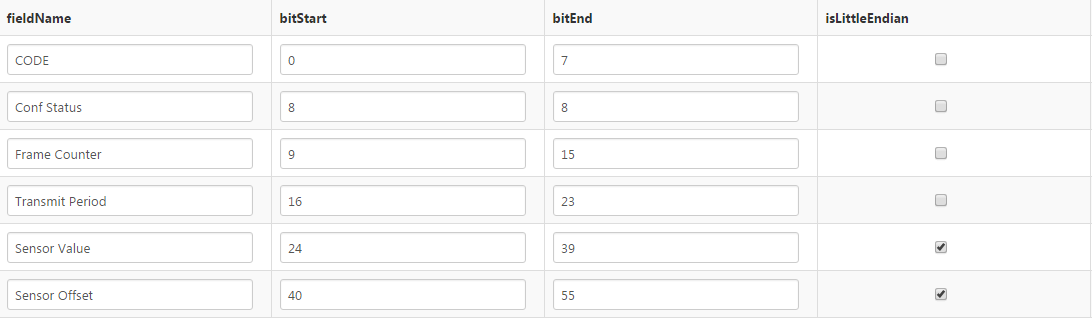


Figure Sample FrameFormatDatashape

The **FrameFormatDataShape** has the following fields:

* **fieldName:** The name of the field. This name is also used to construct the result JSON for frame decoding.
* **bitStart:** The bit where this frame starts. Please note that **bitStart** is included in the field.
* **bitEnd:** The bit where this frame ends. Please note that **bitEnd** is included in the field.
* **isLittleEndian:** For fields spanning across multiple bytes, specifies if the field is little endian(LSB first).
* **value:** The value for this field. Used for frame encoding.

The **MessageDecoderThing** has the following services:

* **decodeFrame**: Decodes a data frame, using a given specification
  + **Inputs**:
    - **frameFormat**: An infotable with the **FrameFormatDataShape** describing the frame
    - **data**: The data to decode, represented as a string with hexadecimal data, where first byte (byte 0) represent the first two characters.
  + **Result**: A string containg JSON data. For each of the fields in the input frame format, a corresponding property exists in the JSON, with the value equal to the computed value from the data
* **encodeFrame**:
  + **Inputs**:
    - **frameFormat**: An infotable with the **FrameFormatDataShape** describing the frame. The value field in this infotable must be filled.
  + **Result:** The encoded data, represented as a string with hexadecimal data, where first byte (byte 0) represent the first two characters.

# Known Limitations and Compatibility

## Limitations

## Compatibility

This extension has been tested for compatibility with the following ThingWorx Platform releases and Java JRE versions:

|  |  |
| --- | --- |
| ThingWorx Platform Version | Java JRE Version |
| ThingWorx 6.0.0 | Java version 8, update 51 |
| ThingWorx 6.5.0 | Java version 8, update 65 |
| ThingWorx 6.6.2 | Java version 8, update 65 |

# Document Revision History

|  |  |  |
| --- | --- | --- |
| **Revision Date** | **Version** | **Description of Change** |
| January 18, 2016 | 1.0 | Initial Release |