

ads-tec GmbH

IRF2000 IWL3000 SRC1000

**Application Note - Apps with OSGi -
Condition Monitoring with WWH push**



Document History

1.0	01/02/2014	Initial Version
1.1	28/04/2014	Updated
2.0	28/10/2014	New adress

Table of Contents

Introduction.....	3
Getting started	4
1.1 Create a table.....	4
1.2 Use WWH push inside an OSGi bundle	7
1.3 Deploy the bundle.....	9
1.4 Read from the database	11

Introduction

Condition monitoring and distributed data collection are one of the main aspects of today's buzz words like Industrie 4.0, The Internet of Things or Big-Data.

This guide will lead you through an example of how to use the Big-LinX Distributed Condition Monitoring with WWH Data Push together with a custom OSGi Java Application on the IRF2000.

Since the storage on all embedded devices is always limited and since data transfers especially over mobile networks is expensive this is a great method to filter and preselect the data before transmitting them to central databases.

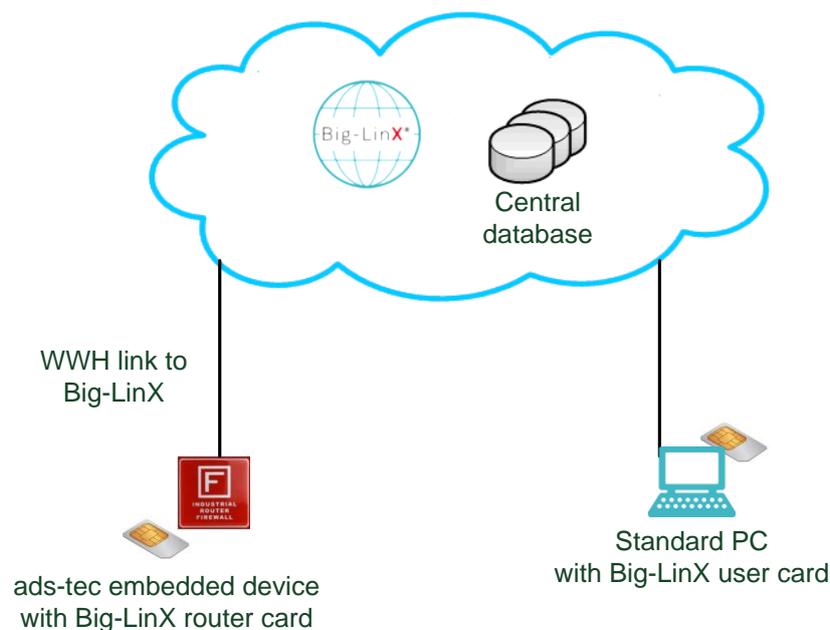
With the **WorldWideHeartbeat** push functionality any OSGi app can send data to the database inside the Big-LinX cloud easily. Therefore a Big-LinX routercard with a PIN is needed to provide a highly secure and authenticated connection.

In this Application note the WWH push from an OSGi app will be discussed. Furthermore an outlook on reading the data will be given.

Getting started

It is recommended to read the OSGi getting started Application note. Also an active Big-LinX connection for the ads-tec embedded device and the user PC is required. Instead of a second Big-LinX user card there is also the possibility to connect the user PC via an ads-tec Site2Site Router (contact sales for further information about Site2Site routing).

The following pictures show the topology of all involved devices.



1.1 Create a table

First step is to create a table in which the pushed data will be stored. This step requires an existing Big-LinX connection between the PC and the Big-LinX cloud. Now with the help of a browser a connection to the database can be made.

Open a browser and go to:

<https://GATEWAYIP/phpmyadmin>

Replace the GATEWAYIP with the correct IP address which can be found in the information from ads-tec containing also the user credentials for the database.

Note: the GATEWAYIP can also be determined by examining the routing table of the PC. Take the VPN IP from the Big-LinX client window e.g. 10.x.y.z



And look for a matching line in the output of the command "route print"

```

=====
Aktive Routen:
Netzwerkziel   Netzwerkmaste Gateway   Schnittstelle Anzahl
0.0.0.0       0.0.0.0   10.20.128.1 10.20.128.135 10
10.1.0.0      255.255.0.0 10.x.y.z    10.x.y.z      30
10.1.1.137   255.255.255.255 127.0.0.1 127.0.0.1    30
    
```

The GATEWAYIP is the first IP in the network. In this example 10.1.0.1.

Now login with username and password



Click on "create table" in the left section of the window.



In the dialog describe your table. For example enter columns as shown in the picture below.

Table name:

Structure

Column	<input type="text" value="ID"/>	<input type="text" value="time"/>	<input type="text" value="value"/>
Type	<input type="text" value="BIGINT"/>	<input type="text" value="TIMESTAMP"/>	<input type="text" value="INT"/>
Length/Values¹	<input type="text" value="20"/>	<input type="text"/>	<input type="text"/>
Default²	<input type="text" value="None"/>	<input type="text" value="None"/>	<input type="text" value="None"/>
Collation	<input type="text"/>	<input type="text"/>	<input type="text"/>
Attributes	<input type="text"/>	<input type="text"/>	<input type="text"/>
Null	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index	<input type="text" value="PRIMARY"/>	<input type="text" value="---"/>	<input type="text" value="---"/>
AUTO_INCREMENT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Comments	<input type="text"/>	<input type="text"/>	<input type="text"/>
MIME type	<input type="text"/>	<input type="text"/>	<input type="text"/>
Browser transformation	<input type="text"/>	<input type="text"/>	<input type="text"/>
Transformation options³	<input type="text"/>	<input type="text"/>	<input type="text"/>

Table comments:

Storage Engine:

Collation:

PARTITION definition:

As you can see we create a quite simple table named TEST with three rows. ID: will hold a unique identifier for the data source

- time: will hold a timestamp of the record
- value: a single simple numeric value

Of course you will most likely add many more rows or many different tables in a real world setup.

Application Note - Apps with OSGi - Condition Monitoring with WWH push 2.0

- It is also possible to use a mySQL client and connect directly to the server with the GATEWAYIP and the port 3306. To add a table the SQL statement could look like this:

```
CREATE TABLE `TEST` (  
  `ID` bigint(20) NOT NULL AUTO_INCREMENT,  
  `time` timestamp NULL DEFAULT NULL,  
  `Value` int(11) NOT NULL,  
  PRIMARY KEY (`ID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

1.2 Use WWH push inside an OSGi bundle

Pushing data out of an OSGi bundle is a simple task. Every WWH push consists of one table name (e.g. "TEST") and some data. Each row of data will be inserted into one row of the table. The first element of the data row will be inserted into column number 1. So for our "TEST" table the JSON Object could look like this:

```
{  
  "table": "TEST",  
  "data": [  
    [  
      null,  
      "1394529351",  
      1234  
    ],  
    [  
      ...  
    ]  
  ]  
}
```

Note that the data object could have more than one row if needed. Regarding bandwidth and protocol overhead it is much better to transmit an array of 10 records every 10 minutes instead of one record every minute!

The following Java code shows how to create this JSON Object and how to send it via WWH push to the central Big-LinX database.

```

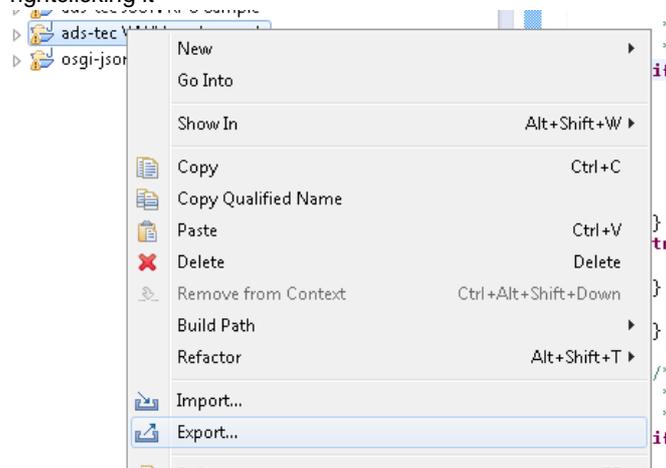
/*
 * build the following json request { "table":"TEST", "data":[ [
 * null, "1394529351", 1234 ], ]}
 */
Date now = new java.util.Date();
java.text.SimpleDateFormat sdf =
    new java.text.SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
String timestamp = sdf.format(now);
List<Object> values = new ArrayList<Object>();
List<Object> value = new ArrayList<Object>();
Map<String, Object> vars_push = new HashMap<String, Object>();
value.add("null"); // Auto increment
value.add(timestamp); // timestamp
value.add(counter); //sample data: counter
values.add(value);
vars_push.put("table", "TEST"); //tablename
vars_push.put("data", values);
JSONRPC2Request request = new JSONRPC2Request("call", 1);
List<Object> params = new ArrayList<Object>();
params.add(Activator.sid);
params.add("wwh"); // JsonRpc method
params.add("push"); // JsonRpc action
params.add(vars_push);
request.setPositionalParams(params);
try {
    JSONRPC2Response response = Activator.RPCSession
        .send(request);
    if (response.indicatesSuccess()) {
        Activator.eventlog.log(LogService.LOG_ERROR,
            "Successfully pushed data");
        counter++;
    } else {
        Activator.eventlog.log(LogService.LOG_WARNING,
            "Error while pushing data: "
            +
response.getError().getMessage());
    }
} catch (JSONRPC2SessionException je) {
    Activator.eventlog.log(LogService.LOG_ERROR,
        "Could not send JSON RPC", je);
}

```

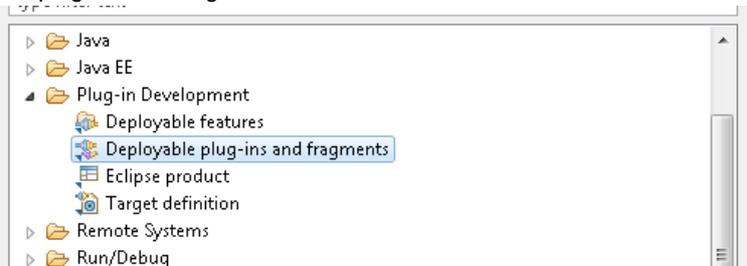
1.3 Deploy the bundle

Use eclipse to write bundles. This is explained in another application note which should be considered before reading this chapter.

First export the bundle by rightclicking it



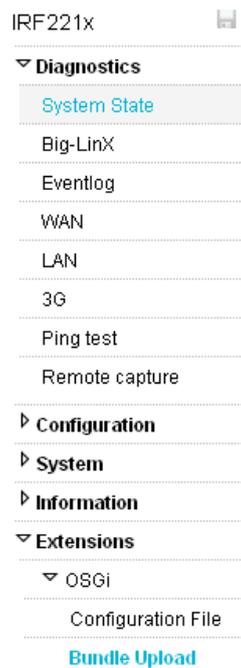
Then select “Deployable plug-ins and fragments”



Press „Next“ and then the „Finish“ Button.

Second step is to upload the bundle to the target platform. To achieve this point your browser to the webinterface of the ads-tec embedded device (e.g. <http://192.168.0.254>)

Then navigate to the bundle upload page

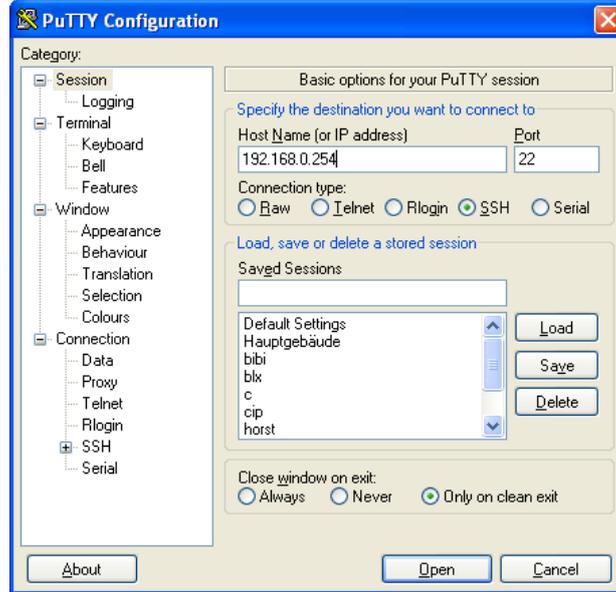


Application Note - Apps with OSGi - Condition Monitoring with WWH push 2.0

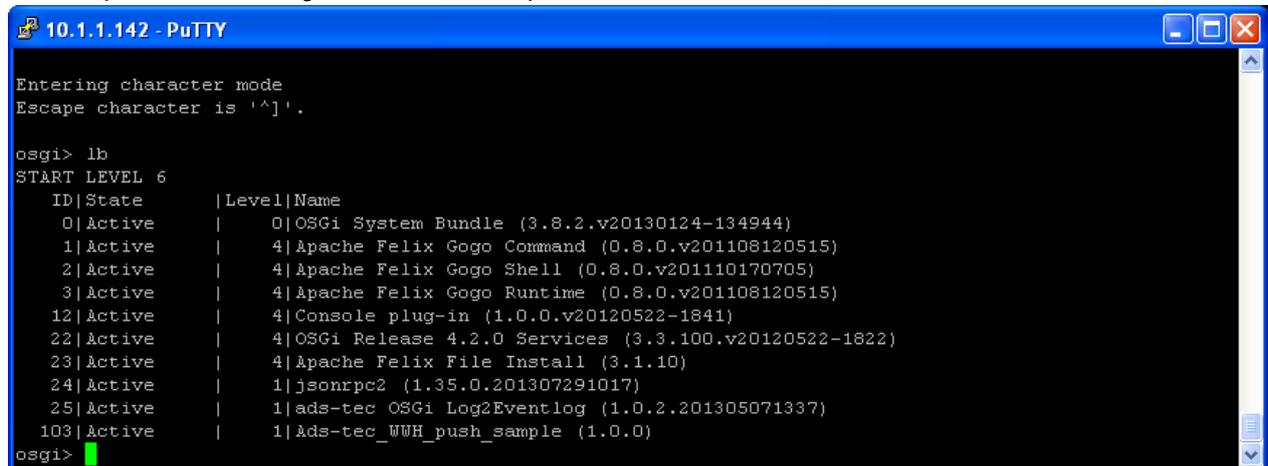
And select the bundle on your computer. With a click on “Upload OSGi bundle” the upload begins



To check if the bundle is running start a SSH client program for example putty. Then enter the IP address of the ads-tec embedded device e.g. 192.168.0.254. Make sure that the connection type is SSH



Press “Open” and enter “osgi” as username and password.



With the „lb“ command all installed bundles can be listed. In this example the WWH push bundle has the number 103 and is running. In the next chapter it is shown how to read the pushed data back from the database.

1.4 Read from the database

To read the data back from the database every mySQL client can be used. The following SQL statement reads all rows from the table "TEST".

```
SELECT * FROM `TEST`;
```

For example the values can be shown using phpmyadmin and executing the SQL statement. The result is shown in the picture below.

	ID	time	Value
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	1	2014-03-12 14:28:30	84
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	2	2014-03-12 14:28:31	85
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	3	2014-03-12 14:28:32	86
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	4	2014-03-12 14:28:34	87
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	5	2014-03-12 14:28:35	88
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	6	2014-03-12 14:28:36	89
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	7	2014-03-12 14:28:37	90
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	8	2014-03-12 14:28:38	91
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	9	2014-03-12 14:28:39	92
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	10	2014-03-12 14:28:40	93
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	11	2014-03-12 15:42:50	0
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	12	2014-03-12 15:42:53	1
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	13	2014-03-12 15:42:54	2
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	14	2014-03-12 15:42:55	3
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	15	2014-03-12 15:42:56	4
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	16	2014-03-12 15:42:58	5
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	17	2014-03-12 15:42:59	6
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	18	2014-03-12 15:43:00	7
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	19	2014-03-12 15:43:02	8
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	20	2014-03-12 15:43:03	9
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	21	2014-03-12 15:43:04	10
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	22	2014-03-12 15:43:05	11
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	23	2014-03-12 15:43:07	12
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	24	2014-03-13 11:41:40	0
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	25	2014-03-13 11:41:43	1
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	26	2014-03-13 11:41:45	2
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	27	2014-03-13 11:41:46	3
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	28	2014-03-13 11:41:47	4
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	29	2014-03-13 11:41:48	5
<input type="checkbox"/> Bearbeiten <input type="checkbox"/> Direkt bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen	30	2014-03-13 11:41:50	6

Please note that in a real world setup you will now not only have to read the data but also delete it after you have copied them to your own companies' ERP system or data analysis software.