

MPACT

Integration With Receivers



ZEBRA

Copyright

© 2020 ZIH Corp. and/or its affiliates. All rights reserved. ZEBRA and the stylized Zebra head are trademarks of ZIH Corp., registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners.

COPYRIGHTS & TRADEMARKS: For complete copyright and trademark information, go to www.zebra.com/copyright.

WARRANTY: For complete warranty information, go to www.zebra.com/warranty.

END USER LICENSE AGREEMENT: For complete EULA information, go to www.zebra.com/eula.

Terms of Use

Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries ("Zebra Technologies"). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Table of Contents

Copyright	-2
Terms of Use	-2
Proprietary Statement	-2
Product Improvements	-2
Liability Disclaimer	-2
Limitation of Liability	-2
Table of Contents	-3
Receiver Bootstrapping	-4
Sample Badge_Config.JSON	-4
Bootstrap Process	-5
Receiver Filter Configuration	-5
Sample Simple Receiver Filter Configuration	-5
Sample Response From Server For Time Request By Receiver	-8
Sample HeartBeat Event Sent By Receiver	-8
Sample Asynchronous Event Sent By Receiver	-8
Sample Beacon Data Sent by Receiver	-9
Sample Receiver Filter Configuration With Support For SuperBeacons	-10
Sample Beacon Data Sent By Receiver With SuperBeacons	-12

Receiver Bootstrapping

If a receiver has not been configured before then it tries to initialize itself.

Receiver acts as a WIFI client. It uses the following WIFI connection details to join the network.

SSID: mpact_init
Security Type: wpa2
Security Key: mpact123

Upon joining the network the receiver continues with its initialization process.

The network settings and the gateway or endpoint-server way settings are needed for the receiver to complete its initialization process. The receiver expects it to be available from a machine with following details:

IP: 192.168.1.100
Port: 8005

The receiver expects a JSON configuration file called *badge_config.json* to be served from this location. The schema for the bootstrap JSON file is available in the schema document [*JSON Schema for Receivers.doc*].

Sample Badge_Config.JSON

```
{
  "wifiProfiles": [{
    "ssid": "wifinetwork123",
    "securityKey": "abcd1234",
    "securityType": "wpa2",
    "wpaEnterpriseUser": "",
    "wpaEnterprisePassword": "",
    "wpaEnterpriseOuterIdentity": "",
    "eapType": "",
    "enable": true
  }],
  "gatewayConfigs": [{
    "user": "superuser",
    "password": "mpact123",
    "receiverConfigURL": "http://192.168.1.100:8005/gateway-client/config/v1/receiver",
    "configPullFrequencyInMins": 5,
    "enable": true
  }]
}
```

Bootstrap Process

badge_config.json has *wifiProfiles* section that tells the receiver how to authenticate itself and connect to the WIFI network. The end point server connection details are provided in the *gatewayConfigs* section.

The receiver would validate the badge_config.json store the details and reboot itself and join the WIFI network provided in the *wifiProfiles* section. After successfully joining the WIFI network the receiver would try to pull a beacon filter configuration JSON file by making a GET request to the URL specified in *receiverConfigURL* JSON attribute. If the server requires basic 64 authentication, the username and password provided in the *user* and *password* sections would be used by the receiver. Receiver would periodically pull the beacon filter configuration from the server with the frequency specified in the *configPullFrequencyInMins* attribute and use changes to beacon filter configuration if any.

Receiver Filter Configuration

The beacon filtering details as well as the event, health endpoint details are to be provided to the receiver in the JSON file. The schema for the beacon filter JSON file is available in the schema document [*JSON Schema for Receivers.doc*]. The server should implement the endpoints for the following:

1. Provide beacon filter configuration requested by receiver using HTTP GET [Mandatory]
2. Provide time data
3. requested by receiver using HTTP GET [Mandatory]
4. Accept health events sent by receiver using HTTP POST [Mandatory]
5. Accept beacon data events sent by receiver using HTTP POST [Mandatory]
6. Provide firmware requested by receiver using HTTP GET [Optional]

Sample Simple Receiver Filter Configuration

```
{
  "scanIntervalInMilliseconds": 500,
  "wifiTransmissionInterval": 8,
  "heartbeatIntervalInSeconds": 30,
  "timePullIntervalInMinutes": 5,
  "bootTimeWaitInSeconds": 4,

  "beaconWhiteList": [{
    "uuid": "FE913213-B311-4A42-8C16-47FAEAC938AB",
    "type": "mpact"
  },
  {
    "uuid": "FE913213-B311-4A42-8C16-47FAEAC938EF",
    "type": "mpact"
  }
  ],

  "profiles": [{
    "name": "Location Beacons",
    "profileLow": 1,
```

```

    "profileHigh": 1000,
    "beaconType1": "Fixed",
    "proximityRanges": [{
      "name": "Near",
      "rssiLow": null,
      "rssiHigh": null,
      "beaconHitCount": 1,
      "sendClosestOnly": false,
      "suppressRepeats": false
    }]
  },
  {
    "name": "Asset Beacons",
    "profileLow": 2000,
    "profileHigh": 3000,
    "beaconType1": "Mobile",
    "proximityRanges": [{
      "name": "Near",
      "rssiLow": null,
      "rssiHigh": null,
      "sendClosestOnly": false,
      "suppressRepeats": true
    }]
  }
],
"serviceUrls": {
  "timeService": "http://10.21.2.52:80050/gettime",
  "eventService": "http://10.21.2.52:80050/event/",
  "healthService": "http://10.21.2.52:80050/health/",
  "firmwareService": "http://10.21.2.52:80050/firmware/"
},
"firmware": {
  "downloadInSecond": "true",
  "firmwareDownloadDebug": "true",
  "gaugeFirmware": "2.7.6.0-057R_GAUGE-BT000311-01_9689CECC_14DDC8BE.srec",
  "bleFirmware": "2.7.6.0-057R_BLE-FIRMWARE_C6D54CAF_F789E6B6.bin",
  "wifiFirmware": "2.7.6.0-057R_WIFI-FW_B561BEFF_9A23DE30.bin",
  "bootloader": "2.7.6.0-057R_BOOTLOADER-LOW_0613CBA3_52F02E35.bin",
  "mainFirmware": "2.7.6.0-057R_MPACT-MB5000-01-WR_12B56998_E58DD0CA.elf"
}
}

```

As mentioned earlier the *receiverConfigURL* specified in the bootstrap file would serve a beacon filter configuration JSON file like the one above. It would be different for different deployments based on their requirements. The meaning of each of these attributes is available as part of schema file. Here are some important attributes and their purpose:

Attribute Name	Description
scanIntervalInMilliseconds	The frequency of the receiver to scan for BLE beacons
wifiTransmissionInterval	The frequency of the receiver to post the filtered beacon data to the server. This attribute is a multiple of <i>scanIntervalInMilliseconds</i> . So if <i>wifiTransmissionInterval</i> is 8 and <i>scanIntervalInMilliseconds</i> is 500 then the receiver would post beacon data every (500*8)ms.
heartbeatIntervalInSeconds	The frequency at which the receiver should send its health message.
timePullIntervalInMinutes	The receiver does not have a real-time clock and synchronizes its time with the server time. The frequency at which the receiver should updates its time.
beaconWhiteList	A list of beacon UUIDs that should be processed by the receiver.
serviceURLs	<p>These are the URL endpoints where the receiver would post the information.</p> <p><i>timeService</i>: This is the URL from where the receiver would pull the time information by using a HTTP GET method. The data format expected by the receiver is part of the JSON schema file.</p> <p><i>healthService</i>: This is the URL where the receiver would send asynchronous events as well as health information by using HTTP POST method. The POST data format is available as part of the JSON schema file.</p> <p><i>eventService</i>: This is the URL where the receiver would send periodic beacon data events by using HTTP POST method. The POST data format is available as part of the JSON schema file.</p> <p><i>firmwareService</i>: This is the URL from where the receiver would pull and update its firmware by using a HTTP GET method.</p>

Sample Response From Server For Time Request By Receiver

```
{
  "time": 1580853295594
}
```

Sample HeartBeat Event Sent By Receiver

```
{
  "timestamp": 1567799222930,
  "firmware": "2.7.6.0-057R_MPACT-MB5000-01-WR.bin",
  "type": "HeartBeat",
  "receiver": {
    "type": "FR",
    "identity": {
      "mac": "40:83:DE:D9:61:EB"
    }
  }
}
```

Sample Asynchronous Event Sent By Receiver

```
{
  "timestamp": 1567799770734,
  "firmware": "2.7.6.0-057R_MPACT-MB5000-01-WR.bin",
  "type": "Reboot.Unforced",
  "details": {
    "bootCode": 12
  },
  "receiver": {
    "type": "FR",
    "identity": {
      "mac": "40:83:DE:D9:61:EB"
    }
  }
}
```


Sample Beacon Data Sent by Receiver

```
{
  "beacons": [{
    "major": 100,
    "uuid": "fe913213-b311-4a42-8c16-47faeac938ef",
    "bl": 90,
    "ts": 1553191045018,
    "mac": "a0:e6:f8:79:d4:13",
    "rssi": -52,
    "minor": 0
  }, {
    "major": 100,
    "uuid": "fe913213-b311-4a42-8c16-47faeac938ef",
    "bl": 90,
    "ts": 1553191045080,
    "mac": "a0:e6:f8:79:d8:6f",
    "rssi": -56,
    "minor": 0
  }],
  "receiver": {
    "type": "FR",
    "identity": {
      "mac": "40:83:DE:D9:6A:69"
    }
  }
}
```

Sample Receiver Filter Configuration With Support For SuperBeacons

```
{
  "scanIntervalInMilliseconds": 500,
  "wifiTransmissionInterval": 8,
  "heartbeatIntervalInSeconds": 30,
  "timePullIntervalInMinutes": 5,
  "bootTimeWaitInSeconds": 4,

  "superBeaconFilters": {
    "rssiLow": -80,
    "rssiHigh": -30
  },
  "superBeaconProfiles": [{
    "name": "Location Beacons1",
    "profileLow": 700,
    "profileHigh": 730,
    "proximityRanges": [{
      "name": "Near",
      "rssiLow": null,
      "rssiHigh": -30
    }]
  }], {
    "name": "Location Beacons2",
    "profileLow": 2000,
    "profileHigh": 2500,
    "proximityRanges": [{
      "name": "Near",
      "rssiLow": null,
      "rssiHigh": -30
    }]
  }],
  "superBeaconWhiteList": [{
    "uuid": "FE913213-B311-4A42-8C16-47FAEAC938DB",
    "type": "ibeacon"
  },
  {
    "uuid": "FE913213-B311-4A42-8C16-47FAEAC938EF",
    "type": "mpact"
  }
  ],
  "beaconWhiteList": [{
    "uuid": "FE913213-B311-4A42-8C16-47FAEAC938EF",
    "type": "mpact"
  }],

  "profiles": [{
```

```

    "name": "Location Beacons",
    "profileLow": 1,
    "profileHigh": 100,
    "beaconType1": "Mobile",
    "proximityRanges": [{
      "name": "Near",
      "rssiLow": null,
      "rssiHigh": null,
      "sendClosestOnly": false,
      "suppressRepeats": false
    }]
  }, {
    "name": "Asset Beacons",
    "profileLow": 2500,
    "profileHigh": 3000,
    "beaconType1": "Fixed",
    "proximityRanges": [{
      "name": "Near",
      "rssiLow": null,
      "rssiHigh": null,
      "sendClosestOnly": false,
      "suppressRepeats": true
    }]
  }],
  "serviceUrlsHostName": {
    "timeService": "http://rcvr-test.mweng.lan:8005/gettime",
    "eventService": "http://rcvr-test.mweng.lan:8005/event/",
    "healthService": "http://rcvr-test.mweng.lan:8005/health/",
    "firmwareService": "http://rcvr-test.mweng.lan:8005/firmware/"
  },
  "serviceUrls": {
    "timeService": "http://10.21.2.52:80050/gettime",
    "eventService": "http://10.21.2.52:80050/event/",
    "healthService": "http://10.21.2.52:80050/health/",
    "firmwareService1": "http://10.21.2.52:80050/firmware/"
  },
  "firmware": {
    "downloadInSecond": "true",
    "firmwareDownloadDebug": "true",
    "gaugeFirmware": "2.7.6.0-057R_GAUGE-BT000311-01_9689CECC_14DDC8BE.srec",
    "bleFirmware": "2.7.6.0-057R_BLE-FIRMWARE_C6D54CAF_F789E6B6.bin",
    "wifiFirmware": "2.7.6.0-057R_WIFI-FW_B561BEFF_9A23DE30.bin",
    "bootloader": "2.7.6.0-057R_BOOTLOADER-LOW_0613CBA3_52F02E35.bin",
    "mainFirmware": "2.7.6.0-057R_MPACT-MB5000-01-WR_12B56998_E58DD0CA.elf"
  }
}

```

Sample Beacon Data Sent By Receiver With SuperBeacons

```
{
  "beacons": [{
    "major": 2002,
    "uuid": "FE913213-B311-4A42-8C16-47FAEAC938DB",
    "bl": -1,
    "ts": 1570812881878,
    "mac": "a0:e6:f8:6c:70:62",
    "sbMAC": "a4:34:f1:9f:ef:49",
    "rssi": -86,
    "rssiAt1m": -63,
    "minor": 61
  }, {
    "major": 2501,
    "uuid": "fe913213-b311-4a42-8c16-47faeac938ef",
    "bl": -1,
    "ts": 1570812882018,
    "mac": "a0:e6:f8:6c:71:17",
    "rssi": -73,
    "rssiAt1m": -53,
    "minor": 32
  }, {
    "major": 2500,
    "uuid": "fe913213-b311-4a42-8c16-47faeac938ef",
    "bl": -1,
    "ts": 1570812882534,
    "mac": "a0:e6:f8:6c:6f:5d",
    "rssi": -73,
    "rssiAt1m": -53,
    "minor": 90
  }],
  "superBeacons": [{
    "sbVer": 3,
    "sbBL": 83,
    "sbTS": 1570812881878,
    "sbChnl": 39,
    "sbRSSI": -63,
    "sbMAC": "a4:34:f1:9f:ef:49",
    "sbRSSIAt1m": -50
  }, {
    "sbVer": 2,
    "sbBL": 85,
    "sbTS": 1570812881879,
    "sbChnl": 39,
    "sbRSSI": -81,
    "sbMAC": "a4:34:f1:9f:ee:37",
```

```
    "sbRSSIAt1m": -50
  }],
  "receiver": {
    "type": "FR",
    "identity": {
      "mac": "40:83:DE:D9:AE:BB"
    }
  }
}
```

