**xBRC Guest to Vehicle Association**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Author** | **Description** |
| 1.0 | 8/17/2012 | Arkady Glabek | Release Version |
| 1.1 | 8/21/2012 | Michael Semick | Revision |
| 1.2 | 1/4/2013 | Arkady Glabek | New configuration parameters |
| 1.3 | 1/16/2013 | Arkady Glabek | Added videvent, vidhello, vaalgorithm. Deprecated term AVMS in favor of VID. |

**Document Approvers & Sign-Off**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Approver** | **Role** | **Document Accept/Reject** |
| 4/10/13 | Mike Broz | Project Manager | Accept |

Table of Contents

[1 Introduction 4](#_Toc353202652)

[1.1 Purpose 4](#_Toc353202653)

[1.2 Reference 4](#_Toc353202654)

[2 Vehicle Association Considerations 4](#_Toc353202655)

[2.1 Attraction Types 5](#_Toc353202656)

[2.2 Vehicle Types and Behavior 6](#_Toc353202657)

[3 Installation Requirements 6](#_Toc353202658)

[3.1 Long Range (xBR) Readers 6](#_Toc353202659)

[3.2 Vehicle Reader 8](#_Toc353202660)

[4 Vehicle Association using the VEHICLE event 8](#_Toc353202661)

[4.1 Vehicle reader registration 8](#_Toc353202662)

[4.2 Collection of VEHICLE and xBand events 9](#_Toc353202663)

[4.3 Vehicle association algorithms 10](#_Toc353202664)

[4.4 The choice of algorithm 13](#_Toc353202665)

[4.5 Vehicle association algorithm using GPIO events 13](#_Toc353202666)

[5 Message Format 14](#_Toc353202667)

[5.1 VEHICLE message 14](#_Toc353202668)

[5.2 VEHICLE hello message 15](#_Toc353202669)

[5.3 INVEHICLE message 15](#_Toc353202670)

[6 Configuration 17](#_Toc353202671)

[6.1 xBR reader configuration 17](#_Toc353202672)

[6.2 Assign the Vehicle reader to a location 18](#_Toc353202673)

[6.3 Assign xTPRA reader to a location 18](#_Toc353202674)

[6.4 Configuration Settings 18](#_Toc353202675)

[6.5 Sample Configuration settings 21](#_Toc353202676)

[7 System Interaction 24](#_Toc353202677)

[8 Monitoring 26](#_Toc353202678)

[8.1 Facility View (Subway Map) 26](#_Toc353202679)

[8.2 xBRC Controller Log 27](#_Toc353202680)

[8.3 Eventdump.txt 27](#_Toc353202681)

[8.4 Messages REST call 27](#_Toc353202682)

[9 Testing Using a Simulator 27](#_Toc353202683)

[9.1 Simulation Setup 28](#_Toc353202684)

# Introduction

## Purpose

For those attractions equipped with vehicles it is desirable to associate guests to the vehicle in which they are traveling. This document describes the vehicle association process. First, the considerations affecting the guest to vehicle association are presented. Next, the reader installation requirements are listed. Lastly, the vehicle association algorithm is explained as it is currently implemented in the xBRC.

The information contained in this document should aid a system designer in figuring out what type of hardware is needed, where it should be installed, and what is the expected performance for a given attraction. Also, it should serve as a guide in configuring the xBRC installation, specifically the vehicle association configuration.

## Reference

|  |  |
| --- | --- |
| Document Number | Title and Revision |
| 900-0058 | xBRC Interface Control Document |
|  |  |
|  |  |

# Vehicle Association Considerations

Guest to vehicle association is the process of associating guests wearing xBands to a particular vehicle in which they are riding through the attraction. The outcome of vehicle association is a message sent out to any recipient systems. This message includes a list of guest IDs and a single vehicle ID. In attractions equipped with beam break readers in addition to the VEHICLE reader, the message also includes the car number within the train. There are two general methods for performing guest to vehicle association.

The first method uses some form of an electronic reader mounted externally of the vehicle. As the vehicle passes near this reader, the vehicle ID is read from the vehicle and a message containing the vehicle ID is sent to the xBRC. At the same physical location as the external reader there is an array of long range readers to read xBand events from guest bands passing under the array. The xBRC uses an algorithm to associate the xBand events to the vehicle message by using the signal strength of the xBand events and the time of the event generation for both the xBand events and the vehicle message.

For attractions equipped with beam break readers (xTPRA), the association process is similar to the above, with the exception that as the car passes under the vehicle association array, a beam break is detected and translated into a GPIO event sent to the xBRC. The xBRC uses the GPIO event time rather than the VEHICLE message time to associate guests to the car. The outcome of the association is the combination of car number and vehicle ID.

The second method of associating guests to vehicles uses an xBand that is physically attached to the vehicle itself. Somewhere along the ride an array of long range readers receives the xBand events from the guests as well as those from the vehicle in which they are riding. The xBRC uses an algorithm to associate the xBand events sent by the guests with the events sent by the xBand attached to the vehicle.

The remainder of this document concentrates on the first method of associating guests to vehicles.

## Attraction Types

There are at least three main archetypes for attractions.

**Omnimover**: Attractions where individual cars are mounted on a track, spaced evenly, and are in continuous movement (e.g. Haunted Mansion, Buzz Lightyear Space Ranger Spin)

**Water Ride**: Water channel based attractions where individual cars flow in a channel of water (e.g. Pirates of the Caribbean, It’s a Small World)

**Roller Coaster**: Generally high speed track based, with cars linked together to form a train.

There are other archetypes such as Road based attractions where vehicles free travel on a road (e.g. Kilimanjaro Safaris Expedition at Animal Kingdom), and some that blend features of two of the main archetypes (e.g. Splash Mountain is both a roller coaster, and a water ride.

The above archetypes and the sub categories of variations may require specific hardware solutions necessary to produce timely vehicle messages with vehicle IDs. What works in one attraction may not work in another. For example, a close proximity reader that can be installed for an Omnimover may not work for a water ride where the vehicles do not follow the tight path necessary for the reader to come into proximity of the vehicle.

## Vehicle Types and Behavior

Vehicles vary in size, passenger capacity, proximity to each other, speed, travel direction, periods of non-movement. Some of these factors may affect the vehicle association algorithm.

**Proximity of Vehicles to Each Other**

The closer the vehicles are to each other, the more difficult it is to associate the guests to the correct vehicle. Synapse will provide detailed guidance for each attraction where vehicle association is required, to characterize the level of resolution possible with the current technology.

**Vehicle Speed**

Vehicles or trains of cars travel at different speeds depending on the archetype of the attraction. Additionally some attractions may run at variable speeds, such as the Omnimover archetype. The xBand was designed with variable chirp frequency from one to ten chirps per second to address the ability to read the band in variable or high speed situations.

**Periods of Non-Movement**

The vehicle association algorithm must be able to tolerate the vehicles coming to a stop for some period of time. The guests frequently require additional help loading or unloading the vehicles requiring them to stop.

# Installation Requirements

## Long Range (xBR) Readers

The long range readers should be installed somewhere along the attraction in a location best satisfying the following requirements.

**Constant Vehicle Speed**

Ideally the vehicle association should happen at a location where the vehicles are moving at a constant speed. This is especially necessary if the vehicle reader is physically located not exactly right under the long range readers. In this case it may be necessary to adjust the vehicle message receive time by a configuration parameter, but this will only work if the vehicle speed is constant. If the vehicle speed cannot be guaranteed to be constant at all times then the vehicle reader must be directly under the long range readers.

**Moderate Vehicle Speed**

Because the vehicle association is time based, it is important to detect the guest when he is directly under the long range readers. The guest bands chirp at some interval between 1 and 10 chirps per second. Since the guest is moving at some speed under the readers, the time delay between the chirps can introduce an error. The faster the vehicle is moving the faster should be the guest xBand chirp speed.

The following table shows the distance traveled in feet between each chirp of an xBand given different vehicle speeds and band chirp speeds.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Vehicle | Speed | Distance Traveled in Feet at given chirp speed | | | | |  |
| MPH | FPS | 10 chirp/s | 8 chirp/s | 6 chirp/s | 4 chirp/s | 2 chirp/s | 1 chirp/s |
| 5 | 7.33335 | 0.733335 | 0.916669 | 1.222225 | 1.833338 | 3.666675 | 7.33335 |
| 10 | 14.6667 | 1.46667 | 1.833338 | 2.44445 | 3.666675 | 7.33335 | 14.6667 |
| 20 | 29.3334 | 2.93334 | 3.666675 | 4.8889 | 7.33335 | 14.6667 | 29.3334 |
| 30 | 44.0001 | 4.40001 | 5.500013 | 7.33335 | 11.00003 | 22.00005 | 44.0001 |
| 40 | 58.6668 | 5.86668 | 7.33335 | 9.7778 | 14.6667 | 29.3334 | 58.6668 |
| 50 | 73.3335 | 7.33335 | 9.166688 | 12.22225 | 18.33338 | 36.66675 | 73.3335 |
| 60 | 88.0002 | 8.80002 | 11.00003 | 14.6667 | 22.00005 | 44.0001 | 88.0002 |
| 70 | 102.6669 | 10.26669 | 12.83336 | 17.11115 | 25.66673 | 51.33345 | 102.6669 |

From the table above one can see that even at 10 chirps per second, if the vehicle is traveling at 50 miles per hour, the distance traveled between each chirp is just over 7 feet. If the cars were within 7 feet of each other, this could result in the vehicle association placing the guest in the incorrect car.

Additionally, the array of antennas typically used for vehicle association have a focused read zone. In some cases the vehicle speed could cause the chirps to occur outside the read zone of the antenna, falling before and / or after the vehicle transits under the array.

**Proximity of Vehicles to Each Other**

In those attractions where the vehicles are not connected to each other, it is advantageous to install the long range readers where the vehicles are furthest apart from each other.

## Vehicle Reader

The vehicle reader must be installed in such a location as to produce a vehicle message when the middle of the vehicle passes under the long range readers. The timing of the vehicle message is critical for producing a correct vehicle association.

# Vehicle Association using the VEHICLE event

The described below vehicle association method is based on a VEHICLE message sent by an external system to the xBRC in response to a vehicle passing a fixed waypoint equipped with the vehicle reader. At the same physical waypoint as the vehicle reader, a series of long range xBR long range readers collect events sent by the guests. The xBRC examines the collected VEHICLE events and the guest xBand events and associates guests with vehicles. An INVEHICLE message is sent out by the xBRC every time a guest is associated to a vehicle. The VEHICLE and INVEHICLE messages are defined later in this document.

## Vehicle reader registration

In order for the xBRC to be able to receive the VEHICLE events from the external system, a new “Vehicle ID” (VID) type reader must be added to the xBRC configuration and it must be associated with a location containing xBR readers. The reader of type “Vehicle ID” is automatically created by the xBRC the first time a VEHICLE message is received. A unique name and mac address are generated for this reader based on the scene id and location id contained in the VEHICLE message. Next, the administrator must associate the new reader with a location. The location type should be either LOAD or WAYPOINT. Once this is done, the xBRC will start processing the VEHICLE events. The following figure illustrates the vehicle reader registration.



Figure 1: Vehicle Reader Registration

## Collection of VEHICLE and xBand events

The xBRC decides which vehicle to associate a guest with based on some number of aggregated xBand events received at the vehicle reader location while the guest approaches, travels under, and leaves the location. These aggregated xBand events are compared against one or more VEHICLE events received while the guest was singulated to the location. The singulation algorithm has not been modified to accomplish this task. Just as at other locations, the guest is singulated to the vehicle reader location based on the signal strength threshold configured in the xBRC configuration and the strength of each xband event. The xband events are collected for each guest until the guest stops singulating to the vehicle reader location for some configurable number of seconds. The figure below illustrates how the VEHICLE and aggregated xband events are collected for the vehicle association.

 Figure 2: VEHICLE and xBand event collection

## Vehicle association algorithms

Once the guest stops singulating to the vehicle reader location for some number of seconds (default 2, onridetimeout Config parameter), the xBRC examines the collected aggregated xBand events.

A configurable minimum number of xBand events are required to even attempt vehicle association (minreadstoassociate Config parameter). This is to ensure some level of accuracy for the calculated score as well as to reject pre-mature singulations that produced a delay of 2 seconds (configurable) before the next singulation. The default minimum number of xBand events is 2. This minimum applies to all the xBand events collected for a single guest as the guest traveled under the vehicle association xBR reader array. This means that the xBand transmit speed in conjunction with the singulation signal strength threshold must produce enough xBand events while the guest travels through the vehicle reader location.

The xBand events are examined with respect to the collected GPIO or VEHICLE events using one or both of the following algorithms: “nearevents” or “closestpeak”.

### The nearevents algorithm

Using this algorithm, the xBRC calculates a mathematical score for each vehicle that produced a GPIO or VEHICLE event while the guest was singulated to the vehicle reader location. The score is simply the average maximum signal strength of some maximum number of aggregated xBand events (default 10) immediately surrounding the VEHICLE event in time. The same number of xBand events is examined before and after the VEHICLE event. If the VEHICLE event happens at the beginning or the end of the xBand event sequence then less than the maximum number of events are examined. The average mathematical function is used to account for the fact that not the same number of events may be examined for each VEHICLE event.

(NOTE: When xTPRA readers are used the GPIO event is used instead of the VEHICLE event. See section 4.4 below.)

The following illustration shows a sample vehicle association score calculation.



### The closestpeak algorithm

This algorithm attempts to find the time when the guest was directly under the VA array. It does this by locating two strongest band pings near each other and then finds a time between them adjusted proportionally toward the stronger of the two signal strengths. The calculated time when the guest was under the VA array is then compared to all GPIO or VEHICLE events and the closest GPIO or VEHICLE event is found.

The advantage of the closestpeak algorithm is the ability to associate a guest to a vehicle given very few band events. Whereas the “closestpeak” algorithm requires that the GPIO (or VEHICLE) event is surrounded by at least 2 band events, the “closestpeak” algorithm does not impose that requirement.

## The choice of algorithm

There is a new xBRC configuration parameter called “vaalgorithm”. It controls which algorithm the xBRC will use for VA.

It may be set to one of three values: nearevents, closestpeak, closestpeakfallback.

When set to “closestpeakfallback” the xBRC uses the “nearevents” algorithm and if no vehicle association was produced it then uses the “closestpeak” algorithm.

## Vehicle association algorithm using GPIO events

In attractions equipped with the laser break readers (xTPRA) the vehicle score calculation is performed based on the time of the GPIO event rather than the time of the VEHICLE event. The VEHICLE event is only used to identify the train or vehicle, but not for timing purposes. The xBRC assigns guests to cars numbered starting at car number 1. Once the VEHICLE message is received, the combination of the car number (from GPIO event) and train id (from VEHICLE event) becomes the vehicle ID.

The xBRC uses one of two different algorithms to associate the GPIO events to a VEHICLE event: 1) a timeout based algorithm or 2) counter based algorithm.

### Using timeout to associate GPIO and VEHICLE events

A train contains multiple cars. Further, there are multiple trains following one another. A VEHICLE event identifies a train, but how does the xBRC know that say the first 10 cars belong to the first train and the following 10 cars belong to the next train. One way to accomplish this is use the fact that trains are separated by some distance and it takes some minimum time for the next train to arrive at the vehicle association location. The xBRC starts a timer when it receives a VEHICLE event. All subsequent and any pending GPIO events are associated with the VEHICLE for some configurable amount of time (traintimeoutsec Config parameter) After that the xBRC throws away the VEHICLE event in anticipation of the next VEHICLE event.

The timeout-based algorihm above will only work if the train does not stop or significantly slow down after the xBRC received the VEHICLE event and before all GPIO events are received. If the train was to stop, the VEHICLE event would time out and the GPIO events would not be associated with the correct train.

### Using counter to associate GPIO and VEHICLE events

In attractions where the trains or vehicles may stop or slow down a timeout cannot be used to pair up the VEHICLE and GPIO events. A counter must be used instead. For this method to work the xBRC must know how many GPIO events to expect for every VEHICLE event. The xBRC must also be told how may GPIO events will occur prior to the VEHICLE event and how many after the VEHICLE event. The “laserbreaksbeforevehicle” and “laserbreaksafterVEHICLE” Config parameters are used for this purpose. Knowing how many before and after GPIO events to expect allows the xBRC to correctly recover from lost GPIO or VEHICLE events. Otherwise the counts would get out of synch and all vehicle associations after a lost GPIO or VEHICLE event would be incorrect.

# Message Format

## VEHICLE message

The vehicle message coming from the vehicle reader must follow this format.

<message type="VEHICLE" time="2012-02-02T22:07:52.5000000-05:00">

<vehicleid>123456</vehicleid>

<attractionid>WMKHAMA</attractionid>

<sceneid>13B</sceneid>

<locationid>01</locationid>

<confidence>99</confidence>

</message>

All the fields in the message are echoed back in the INVEHICLE event published when a guest is associated to a vehicle.

The <attractionid> <sceneid> and <locationid> fields are used by the xBRC to generate a unique reader name. These fields should stay constant for all messages produced by the same reader.

The VEHICLE message must be sent in a restful call to the xBRC using the HTTP PUT method to the following url: http://<xbrc-ip>:<xbrc-port>/videvent (http://<xbrc-ip>:<xbrc-port>/avmsevent is deprecated in xBRC version 1.6). Refer to the “900-0058 Rev 1 0 xBRC Interface Control Document” for more information on the REST interface.

## VEHICLE hello message

The Hello vehicle message is similar in format to the VEHICLE message described above. The Hello vehicle message must be sent by the Vehicle ID system once every minute for system health reporting. The Hello vehicle message coming from the vehicle reader must follow this format. Any additional elements are ignored.

<message type="VEHICLE" time="2012-02-02T22:07:52.5000000-05:00">

<attractionid>WMKHAMA</attractionid>

<sceneid>13B</sceneid>

<locationid>01</locationid>

</message>

The <attractionid> <sceneid> and <locationid> fields are used by the xBRC to generate a unique reader name. These fields should stay constant for all messages produced by the same reader.

The Hello VEHICLE message must be sent in a restful call to the xBRC using the HTTP PUT method to the following url: http://<xbrc-ip>:<xbrc-port>/vidhello (http://<xbrc-ip>:<xbrc-port>/avmshello is deprecated in xBRC version 1.6). Refer to the “900-0058 Rev 1 0 xBRC Interface Control Document” for more information on the REST interface.

## INVEHICLE message

Once each guest is associated to a vehicle the INVEHICLE message is sent either on JMS bus or as an HTTP REST call. The message format is as follows.

<message type="INVEHICLE" time=”timestamp”>

<guestid>guest id</guestid>

<publicid>public id of card or band</publicid>

<linkid>*link id value*</linkid>

<linkidtype>[*xbms-link-id | other*]</linkidtype>

<bandtype>[*Guest | Attraction Card | Cast Member*]</bandtype>

<xpass>true/false</xpass>

<readersection> *reader section* </readersection>

<readerlocation>reader location</readerlocation>

<vehicleid>vehicle id</vehicleid>

<vehicle>vehicle tag id</vehicle>

<car>car number of train</car>

<row>row number of car or vehicle</row>

<seat>seat number or identifier</seat>

<attractionid>attraction id</attractionid>

<sceneid>scene id</sceneid>

<locationid>location id</locationid>

<confidence>confidence</confidence>

<sequence>sequence</sequence>

</message>

|  |  |
| --- | --- |
| Element | Purpose |
| guestid | xConnect system identifier of the guest that triggered the in-vehicle message. |
| publicid | Identifies the card or band used at the location of the touch or long range read event for this location. In long range reads, the band’s provisioned LRID is the public ID (pid). This value is in decimal. |
| linkid | Global identifier the guest. Typically the xBMS link identifier GUID of a guest. |
| linkidtype | Indicates the type of link identifier used in the linkid. Typically “xbms-link-id” |
| bandtype | Value used to indicate the role of the card or band which was used to identify the owner. Typically “Guest”, other forms may include “Attraction Card”, “TEST”, “Cast Member” |
| xpass | “true” if guest entered the venue through an xPass queue. “false” if through standby queue |
| readersection | Identifies the “section” (a configured string) associated with the reader location. |
| readerlocation | Identifies the reader(s) reporting the event. Typically, there will be multiple readers “ganged” in each reader location. The *readerlocation* element identifies the abstract location of the readers (e.g. “entry”, “load”, etc.). |
| vehicleid | Identifies the hybrid vehicle associated with the guestid. This id combines the <vehicle>-<car> tags when the INVEHICLE event represents a train, otherwise it would be just the <vehicle> tag value with no car id indicator. Value is a string concatenation of fields separated with a hyphen. |
| vehicle | Identifies the “vehicle” associated with the guestid. This id is received in the AGC VEHICLE event. Value is a string. |
| car | Identifies the “car” associated with the guestid. This id is based on the count of cars that make up a train for a rollercoaster style attraction. This is a sequence number that counts starting with 1. Value is in decimal. This is optional in the message format. |
| row | Identifies the “row” associated with the guestid. This id is based on the count of rows that make up a car (in train scenarios) or vehicle (boat or single car attraction). This is a sequence number that counts starting with 1. Value is in decimal. This is optional in the message format. |
| seat | Seat number of theater or car/vehicle associated with the guestid. Value is a string. 13A, 2, 4-L, etc. This is optional in the message format. |
| attractionid | Attraction id received in the AGC VEHICLE event. |
| sceneid | Scene id received in the AGC VEHICLE event. |
| locationid | Location Id received in the AGC VEHICLE event. |
| confidence | Confidence received in the AGC VEHICLE event. |
| sequence | Sequence received in the AGC VEHICLE event. |

Refer to the “xBRC Interface Control Document” for more information on how to receive the INVEHICLE message from the xBRC.

# Configuration

## xBR reader configuration

### Assign readers to Load or Waypoint type location

Once the xBR (Long Range) readers are physically mounted and configured to send Hello messages to the xBRC, they need to be associated with a reader location of type Load or Waypoint. This location is later referred to as the vehicle association location. The xBRC UI may be used to accomplish this task.

### Set the reader signal strength threshold

As described in section 4.2 the xBRC analyzes multiple xBand events from a guest to associate a guest to a vehicle. The “Signal Strength Threshold” setting for each xBR reader at the vehicle association location determines how many events will be collected by the xBRC for the analysis. Generally, the signal strength threshold should be set fairly low (-70) thus allowing for guest xBand events to singulate to the vehicle association location few seconds before the guest arrives at the location. The current algorithm requires that the guest has been singulated to the vehicle association location before the VEHICLE event for the vehicle in which the guest is traveling is received by the xBRC. Also, the xBRC will not perform vehicle association if fewer than 10 (configurable as described later in this document) guest xBand events have been singulated.

## Assign the Vehicle reader to a location

When the first VEHICLE event is received by the xBRC using the PUT /videvent REST endpoint, a new reader is created. This reader will be of type “Vehicle ID”. The new reader must now be associated with the vehicle association location. This can be accomplished using the xBRC UI Location Editor page.

## Assign xTPRA reader to a location

When using the xTPRA readers, the VEHICLE and xTPRA readers must be assigned to the same location.

## Configuration Settings

The configuration settings described below control the behavior of the vehicle association algorithm. These settings are stored in the Config table in the Mayhem MySQL database and may be modified using the “XBRC Configuration Edit” web page of the XBRMS website.

|  |  |  |
| --- | --- | --- |
| Setting | Default Value | Description |
| onridetimeout | 2 (seconds) | As the xBRC collects xBand events from a guest at the vehicle association location, it waits until all events for that guest have been received before analyzing them. The xBRC knows that all events have been received when there is a period of silence (no xBand events) for some number of seconds. This setting controls how long the period of silence is. |
| minreadstoassociate | 2 | The xBRC requires a minimum number of guest xBand events to attempt vehicle association. Whenever less than these minimum number of events have been received an informational message is written to the xBRC log. The signal strength threshold reader setting described earlier affects how many xBand events are singulated to the vehicle association location. |
| usevehicleeventtime | false | Whether to use the timestamp from the VEHICLE event or the time when the VEHICLE event is received by the xBRC. Since the vehicle association is time-critical, even a sub-second error in the VEHICLE event time can generate faulty results. If this flag is set to true then the system clocks must be perfectly synchronized between the xBRC system and the system producing the VEHICLE event. |
| vehicletimeoffsetms | 0 | If the VEHICLE event is received by the xBRC after a constant delay then this parameter may be used to automatically adjust for that delay. Generally, this value should not be used to compensate for the fact that the physical reader generating the VEHICLE event is located significantly away from the vehicle association location. In this case the delay of the VEHICLE event may not always be constant and will not produce reliable results. However, a valid use case for this parameter would be a known constant system processing time. |
| maxanalyzeguest eventspervehicle | 10 | As described in section 4.3 above, the xBRC analyzes some maximum number of events immediately surrounding the VEHICLE event. This setting controls how many. At minimum this setting should be set to 2 allowing for the two most immediate events to be analyzed. |
| requirevehiclelaserevent | false | If set to true then the xBRC will expect to receive GPIO events from the xTPRA beam break readers. No singulation will be performed if no GPIO events are received. If set to false then only VEHICLE events are needed. |
| traintimeoutsec | 60 | The number of seconds to cache the last train id received from the VEHICLE event, before clearing it in expectation of the next train id. This time should be slightly smaller than the minimum time between the trains. This parameter is ignored if either laserbreaksbeforevehicle or laserbreaksaftervehicle is non zero. |
| laserbreaksbeforevehicle | 0 | The number of expected GPIO events (produced by the xTPRA reader) before the VEHICLE event. Leave this set to 0 if using timeout based algorighm to associate GPIO and VEHICLE events. |
| laserbreaksaftervehicle | 0 | The number of expected GPIO events (produced by the xTPRA reader) after the VEHICLE event. Leave this set to 0 if using timeout based algorighm to associate GPIO and VEHICLE events. |
| vaalgorithm | closestpeakfallback | The choice of vehicle association algorithm. The choices for this parameter are: nearevents, closestpeak, closestpeakfallback. |

## Sample Configuration settings

Vehicle association is controlled by the configuration settings described above. It can be difficult to figure out which settings to use so that the xBRC performs the desired algorithm for the given attraction. This section attempts to help with this process.

### VEHICLE events only

The simplest vehicle association setup is to only use the Vehicle ID reader which sends a single VEHICLE message for each car that passes under the vehicle association long range reader array.

Physical Setup

1. Long range reader array at the association location
2. Vehicle ID reader at the association location

Configuration

|  |  |  |
| --- | --- | --- |
| Setting | Value | Description |
| onridetimeout | 2 (seconds) |  |
| minreadstoassociate | 20 | Reduce for fast moving attractions or where no bleed-over band reads occur. Increase for slow moving attractions where bleed-over band reads occur. |
| usevehicleeventtime | false | No clock synchronization between Vehicle ID system and xBRC so set this to false. |
| vehicletimeoffsetms | 0 to +-2000 | Adjust this paremeter to account for the physical placement of the Vehicle ID reader with respect to the long range reader array as well as Vehicle ID processing time. |
| maxanalyzeguest eventspervehicle | 2 | Only look at 2 band chirps around each VEHICLE event. |
| requirevehiclelaserevent | false | Not using xTPRA readers so set this to false. |
| traintimeoutsec | 60 | N/A |
| laserbreaksbeforevehicle | 0 | N/A |
| laserbreaksaftervehicle | 0 | N/A |

### VEHICE events, GPIO events, train timeout

Use this configuration in attractions equipped with trains consisting of multiple cars. The Vehicle ID reader sends a VEHICLE event identifying the train as the train passes through the vehicle association location. The xTPRA sends a GPIO event for every car that passes under the long range reader array. A timeout occurs before the next train passes through the vehicle association location. Once the VEHICLE event is sent by the Vehicle ID system, all of the cars pass under the long range reader array without stopping.

Physical Setup

1. Long range reader array at the association location
2. Vehicle ID reader at the association location
3. xTPRA reader under the long range reader array

Configuration

|  |  |  |
| --- | --- | --- |
| Setting | Value | Description |
| onridetimeout | 2 (seconds) |  |
| minreadstoassociate | 20 | Reduce for fast moving attractions or where no bleed-over band reads occur. Increase for slow moving attractions where bleed-over band reads occur. |
| usevehicleeventtime | false | Set this to false, unless it has been observed that the GPIO events are not received in a timely fasion due to network or processing delays. If changed to true then the xBRC must synchronize the xTPRA and xBRC system clocks very precisely in some cases within milliseconds. |
| vehicletimeoffsetms | 0 to +-2000 | Adjust this paremeter to account for the physical placement of the xTPRA reader with respect to the long range reader array as well as xTPRA processing time. |
| maxanalyzeguest eventspervehicle | 2 | Only look at 2 band chirps around each VEHICLE event. |
| requirevehiclelaserevent | true | Using the xTPRA reader so set this to true. |
| traintimeoutsec | 5 to 300 sec | The entire train must pass under the vehicle association array within this time. The next train cannot arrive before this time passes. |
| laserbreaksbeforevehicle | 0 | Must be set to 0 |
| laserbreaksaftervehicle | 0 | Must be set to 0 |

### VEHICE events, GPIO events, counting GPIO events

Use this configuration in attractions where the xTPRA reader is used and the trains or vehicles may stop or significantly slow down. The xBRC will count the GPIO events corresponding to each VEHICLE events. No timeout is used due to the attraction stopping or slowing down.

Physical Setup

1. Long range reader array at the association location
2. Vehicle ID reader at the association location
3. xTPRA reader under the long range reader array

Configuration

|  |  |  |
| --- | --- | --- |
| Setting | Value | Description |
| onridetimeout | 2 (seconds) |  |
| minreadstoassociate | 20 | Reduce for fast moving attractions or where no bleed-over band reads occur. Increase for slow moving attractions where bleed-over band reads occur. |
| usevehicleeventtime | false | Set this to false, unless it has been observed that the GPIO events are not received in a timely fasion due to network or processing delays. If changed to true then the xBRC must synchronize the xTPRA and xBRC system clocks very precisely in some cases within milliseconds. |
| vehicletimeoffsetms | 0 to +-2000 | Adjust this paremeter to account for the physical placement of the xTPRA reader with respect to the long range reader array as well as xTPRA processing time. |
| maxanalyzeguest eventspervehicle | 2 | Only look at 2 band chirps around each VEHICLE event. |
| requirevehiclelaserevent | true | Using the xTPRA reader so set this to true. |
| traintimeoutsec | N/A | Timeout is not used |
| laserbreaksbeforevehicle | 0 to 100 | If the xTPRA reader is physically installed in front of the Vehicle ID reader (ie. the vehicle passes the xTPRA reader first) then set this parameter to the number of GPIO events that are received by the xBRC before the VEHICLE event. |
| laserbreaksaftervehicle | 0 to 100 | Set this parameter to the number of GPIO events received after the VEHICLE event is received. |

# System Interaction

The following figure illustrates the flow of messages necessary to perform vehicle association.



Message Flow

The following sequence diagram shows a typical sequence of events leading to a vehicle association.



Vehicle Association Sequence Diagram

# Monitoring

The output of guest to vehicle association is the INVEHICLE event described earlier in this document. This event is generated upon successful vehicle association. The sources described below may be examined for further details on the vehicle association process.

## Facility View (Subway Map)

The Facility View web page of the xBRC UI shows guests moving through the attraction. Clicking on the guest icons shows a detailed list of all guests at a particular location. The detailed list of guests includes the Car ID column which is populated with the vehicle ID after successful vehicle association.

## xBRC Controller Log

The xBRC records any errors as well as trace and debug information to the /var/log/xbrc/xbrcController.log log file. To see the most information set the logging level to TRACE in the /usr/share/xbrc/log4j.xml log file and restart the xBRC. At the TRACE level the xBRC will log detailed vehicle association information.

## Eventdump.txt

The xBRC records all VEHICLE and INVEHICLE messages in the /var/log/xbrc/eventdump.txt log file. The starting point in diagnosing vehicle association problems is checking to see if VEHICLE messages are received by the xBRC.

## Messages REST call

The http exposes a REST call http://<xbrc-ip>:<xbrc-port>/messages (e.g. <http://localhost:8080/messages>) that may be used to examine the messages that were generated by the xBRC including the INVEHICLE message.

# Testing Using a Simulator

Guest to vehicle association may be tested using the “Reader” simulator. The Reader simulator is a C program compiled to run on a linux system. The Reader simulator is capable of simulating pre-defined set of readers. These include the Long Range readers, the VEHICLE reader, as well as the xTPRA beam break reader. The Reader simulator is capable of sending a pre-configured sequence of read events for each configured reader.

The simulation involves running an xBRC and the Reader simulator. The xBRC can run with or without the IDMS system depending if band-to-guest resolution is required. The xBRC can run with or without a configured JMS broker depending if JMS traffic is being monitored. As an alternative to monitoring the JMS traffic, the “messages” REST call may be used to view the generated JMS messages.

The exact nature of the simulation will depend on the specific reader messages that are configured to be sent by the Reader simulator. Generally speaking, the reader messages will simulate one or more guest traveling under the vehicle association array. As the guest travels under the vehicle association array, simulated VEHICLE messages are generated for one or more vehicles. These may refer to a single vehicle or a train. Lastly, beam break GPIO events may be simulated using the xTPRA simulated reader.

The outcome of the test is a guest to vehicle association which can be monitored as described in the Monitoring section above.

## Simulation Setup

Setting up the simulation is a multi-step process that involves installting the xBRC, configuring the Mayhem MySQL server database, creating the xml configuration files for the Reader simulator.

This document does not include the complete scripts necessary to configure the Mayhem database nor the xml configuration files for the Reader simulator. Instead, path names are provided to the GIT repository where these scripts are stored.

### Install the xBRC

First, install the attraction model xBRC. The xBRC version must be 1.4 or later. During the installation the Mayhem database will be automatically created, but it will not contain the necessary locations or readers.

### Create Locations and Readers in the Mayhem database

Next, the simulated locations and readers need to be configured in the Mayhem database. At a minimum, there must be at last a single Location configured. The easiest way is to create one Entry type location and an Exit type location. Under the Entry location a set of readers must be configured. The readers must include one or more Long Range readers, a single VEHICLE reader and a single xTPRA reader. The Exit location should contain one or more long range reader.

The following MySQL script may be used to load a set of Locations and Readers into the Mayhem databse suitable for the simulation.

disney.xBandController/src/linux/Reader/test/vehicle/create\_readers.sql

### Create xml configuration files for the Reader simulator

The Reader simulator requires multiple xml configuration files. The main configuration file contains information about the readers and the xBRC. Below is a sample main configuration file.

<?xml version="1.0"?>

<configuration>

<controller url="http://localhost:8080" />

<readers start="0" end="13661600">

<reader name="exit-1" webport="8012" mac="00:00:00:00:00:02" type="lrr" fileformat="json" file="test/vehicle/1user/LRR\_EXIT\_1user.txt"/>

<reader name="invehicle-1" webport="8015" mac="00:00:00:00:00:05" type="lrr" fileformat="json" file="test/vehicle/1user/LRR\_INVEHICLE\_1user.txt"/>

<reader name="car-1" webport="8016" mac="00:00:00:00:00:06" type="car" fileformat="json" file="test/vehicle/1user/LRR\_WDI\_Car\_Events\_1user.txt"/>

<reader name="car-laser" webport="8017" mac="00:00:00:00:00:07" type="xtpra" fileformat="json" file="test/vehicle/1user/XTPRA\_Laser\_Events\_1user.txt"/>

</readers>

</configuration>

In this sample Reader configuration file four simulated readers are defined, exit-1, invehicle-1, car-1, and car-laser. They are of type “lrr”, “lrr”, “car”, and “xtpra” respectively. The “car” type produces VEHICLE events while the “xtpra” type produces GPIO events. The “file” attribute for each reader specifies the file with a list of events sent by that reader. For example the LRR\_Exit\_1user.txt file starts with these events.

{ "events" :

[

{ "lrid":"FF9DB70C00", "pno":"001", "ss":"-80.00", "chan":"0", "freq":"2401", "dt":"60800" },

{ "lrid":"FF9DB70C00", "pno":"002", "ss":"-75.71", "chan":"0", "freq":"2424", "dt":"61120" },

The exact format of the event files is beyond the scope of this document at this time.

The Reader configuration files can be found at the following GIT location:

disney.xBandController/src/linux/Reader/test/vehicle/1user

### Running the Reader simulator

Once all the configuration files are in place and the Mayhem database is configured with the desired Locations and Readers the test can be started.

First start the xBRC. Also, bring up the UI at <http://localhost:8090/UI> and click on the “Facility View” link. The Facility View page will show the guest as he enters the attraction and is associated with a vehicle.

To start the simulator run the Reader program providing it the path to the main xml configuration file as the only argument.

To run the simulator using the sample configuration file included in disney.xBandController/src/linux/Reader/test/vehicle/1user.

First swith to the disney.xBandController/src/linux/Reader directory. Next run the simulator:

Debug/Reader test/vehicle/1user/1user.xml

### GPIO events

The xBRC can operate in two modes. Mode one, where guests are only associated with vehicle ID provided in the VEHICLE event. Mode two, where guests are associated with the vehicle ID provided in the VEHICLE event as well as the car number provided in the GPIO beam break event. Mode two may be enabled by setting the Config table parameter “requirevehiclelaserevent” to “true”.

### Understanding the Simulator Results

Upon a successful run of the Reader simulator, a guest should appear on the Facility View page of the xBRC UI. The guest will initially start in the HASENTERED state and after a successful vehicle association will transition to the RIDING state. The guest will then transition to EXITED state upon reciving reader events at the Exit location. While in the RIDING or EXITED state, the guest may be clicked on to see the vehicle with which the guest was associated.