



Parco RTLS Messaging

Confidential & Proprietary
Overview

Overview

- Parco passes xml messages between system level programs via the TCP/IP protocol. SDK clients can receive data from a published Parco system resource.
 - A resource is defined as a program running at a particular TCP/IP address and port which supplies real time Parco geo-location data. Resources may be found by querying the RTLS database for the type of resource that you need.
- From the SDK client's point of view, resources are available in two flavors: full stream and subscription.
 - Full stream resources relay all received data back out to all connected SDK clients.
 - Subscription resources relay only the data that is subscribed to by the SDK client.
- The Parco messaging system is built into the RTLS SDK.
 - The developer only needs to create business objects and respond to events raised by the SDK.
 - Accessing a resource without using the SDK is possible providing that you know what the messaging requirements are in order to communicate with a given resource.

Message Definitions

Request Message

- The messaging system relies on a request/response paradigm.
- To establish a connection, you must connect to the SDK server via a TCP/IP socket.
 - XML messages are case sensitive and are required to be UTF-8 encoded.

Request Message

```
<?xml version="1.0" encoding="utf-8" ?>
  : <parco version="1.0">
      <type>request</type>
      <request>BgnStrm/EndStrm/AddDvc/RemDvc</request>
      <reqid>yourID</reqid>
  :   <params>
          <deviceid data="true">XXXX</deviceid>
          <deviceid data="false">12345</deviceid>
      </params>
  </parco>
```

- Each request message is processed and a corresponding response message is returned to the requester. If you do not receive a response message you must assume that the request was not received. The response message's <reqid> tag contains the senders request id and <request> tag contains original request tag value. If the <msg> tag is empty, your request was valid for the resource and the available data will be sent. If the resource could not process your request, the <msg> tag will include the reason why your request was not granted.

Message Definitions

Response Message

Response Message

```
<?xml version="1.0" encoding="utf-8" ?>
<parco version="1.0">
  <type>response</type>

  <request>BgnStrm/EndStrm/AddDvc/RemDvc</request>
  <reqid>yourID</reqid>
  <msg>Any Error is here</msg>
</parco>
```

Message Definitions

Data Message

- Parco Tag data is delivered by Data message. The <type> tag indicates what type of data is present and may be anything. The current values are listed below and may change in the future. The <gis> tag contains device and geo-location data.
 - Tag descriptions:
 - <id> is the Device id
 - <ts> is the timestamp of the data in UTC.
 - <gwid> is for future use in tracking the originating gateway of the data
 - <cnf> is for future use in determining the confidence factor of the x/y/z data.
 - <x>, <y>, and <z> are the positional coordinates
 - <bat> is the battery indicator, value 0 through 15 where a value of 15 indicates fully charged.
 - <data> is for future use and will contain data transmitted by a device.
 - <type> is the data type an may be any value. The current values are:
 - R: 3-D calculation valid for X,Y, and Z
 - T: 2-D calculation valid for X, and Y. Z will be the Z location for the hub
 - O: 2-D estimated calculation for X and Y, Z will be the Z location for the hub.
 - P: Presence indicator with X, Y, and Z as all zeros. In the near future, the presence data will contain the X, Y, and Z position of the receiver that detected the device.

Data Message

```
<?xml version="1.0" encoding="utf-8" ?>
:   <parco version="1.0">
:     <type>Ptaganything</type>
:     <gis>
:       <id>1234aeb</id>
:       <ts>07/07/2003 10:35:27.45 PM</ts>
:       <gwid>25x</gwid>
:       <cnf>234</cnf>
:       <x>23.2</x>
:       <y>34.2</y>
:       <z>12.1</z>
:       <bat>15</bat>
:     </gis>
:     <data>any data here</data>
:   </parco>
```

Connecting to Resources

Full Stream

- Full stream resources supply all incoming data to the SDK client.
 - A resource receiving 20,000 tags per second will relay 20,000 tags per second to your application.
 - If the resource is not receiving data, your application will not receive data.
 - Full stream resources should be created to allow applications to connect without overloading the client.
- To connect to a Full Stream resource:
 - 1) Create a request message with these tag values:

```
<type>request</type>
<request>BgnStrm</request>
<reqid>AUniqueValue</reqid>
```
 - 2) The <params> tag should be empty as full stream resources do not look for device subscriptions.
 - 3) Connect to the SDK server on the published address and port.
 - 4) Send the request message via the TCP socket.
 - 5) Wait for the response message. If you do not receive a response message you should assume that the request was not received.
 - 6) Check the response message for an error message. A message will be present if the resource can not process your request. If there is not an error message, data will be arriving shortly.
 - 7) When you are done receiving data, create and send a request message with these values to the resource.

```
<type>request</type>
<request>EndStrm</request>
<request>yourId</request>
```
 - 8) The resource will send a response message and then disconnect your socket.

Connecting to Resources

Subscription

- 1) To connect to a Subscription resource:
- 2) Create a request message with these tag values:
`<type>request</type>`
`<request>BgnStrm</request>`
`<reqid> yourId </reqid>`
- 3) Add at least 1 device tag to the params tag. The device id you wish to subscribe to goes in the device tag. If you wish to also receive transmitted device data, set the data attribute to true. Device data is not currently available, but is included for future use.
`<deviceid data="true">256</deviceid>`
- 4) Connect to the SDK server on the published address and port.
- 5) Send the request message via the TCP socket.
- 6) Wait for the response message. If you do not receive a response message you should assume that the request was not received.
- 7) To add a device to your subscription, send an AddDvc request message with one or more devices in the params tag.
- 8) Wait for the response message. If your request was processed, the msg tag will be empty.
- 9) To remove a device from your subscription, send a RmvDvc request message with one or more devices to remove.
- 10) Wait for the response message. If your request was processed, the msg tag will be empty.
- 11) If your device subscription count drops to 0 at any time, the resource will send you a response message and disconnect your socket.
- 12) When you are done receiving data, send an EndStrm message to the resource.
- 13) The resource will send a response message and then disconnect your socket.