

# **RTDS**

## **Resource Tracking Directory Server**

**Draft Version** : 1.1.0  
**Date** : 02 July 2020  
**Author** : Thinkal VB  
**Copyright ©** : [thinkalvb@gmail.com](mailto:thinkalvb@gmail.com)

## **1.0 ABBREVIATIONS**

RTDS	Resource Tracking Directory Server
ASCII	American Standard Code for Information Interchange
IP	Internet Protocol
SAP	Source Address Pair (IP address and port number)
BG	Broadcast Group
BGID	Broadcast Group Identifier
BGT	Broadcast Group Tag

## **2.0 INTRODUCTION**

RTDS is a resource tracking server for computers, services, or other resources connected to the Internet. RTDS helps in the communication between resources and services within a mutual area of interest. RTDS is built to ensure the persistence and integrity of distributed services without compromising the simplicity of the process. All RTDS commands and responses are strictly confined to ASCII (8 bit extended ASCII) character encoding. The RTDS server follows a command-response model in strictly sequential order. All RTDS command and response must end with a new line character '\n' (ASCII 10)

## **3.0 COMPONENTS**

### **3.1 ARCHITECTURE**

RTDS server follows server-client architecture in command-line mode. An IP address and port number together are known as the source address pair (SAP). Client's can join a broadcast group with some tag which will describe some property of the participating client.

### **3.2 COMMANDS**

An RTDS server is governed using commands. RTDS command may be preceded by optional or mandatory arguments which are separated by horizontal tab (ASCII 9). A new command may only be issued after resolving the previous command. All commands must be in small letters and are case sensitive.

### **3.3 BROADCAST GROUP** (default max 128 characters)

Every broadcast group is distinguished using a unique string termed as Broadcast Group Identifier. BGID can contain any ASCII printable character except a white space (ASCII 32). The BGID of a Broadcast Group can be anything including a URL, email ID, a phone number or even a torrent URI.

### **3.4 IP VERSION**

IP version can have possibly two values as v4 representing IP version 4 and v6 representing IP version 6. IPv4 address is represented in dotted-decimal notation and IPv6 address is represented as eight groups of four hexadecimal digits, each group representing 16 bits separated by colons (:). Suitable abbreviations for the IP address shall be used in conformance with RFC 5156.

### 3.5 BG TAG (default max 32 characters)

Broadcast Group Tag is used to specify the subgroup within every individual Broadcast Group. BGT can contain any ASCII printable character except a white space (ASCII 32). BGT and BGID must be more than one character in length. BGT can be used to form sub-groups like server, client, moderator etc....

### 3.6 RESPONSE

RTDS will respond to every request in a certain response format. Every response from RTDS is prefixed using a message specifier which will determine the type of response. The message specifier is enclosed in two square brackets “[ ]”. The actual response will entail after the message prefix followed by a horizontal tab ‘\t’ (ASCII 9).

Message Specifiers:

[M]	A broadcast message.
[W]	A broadcast message from some UDP peer (whisper).
[R]	Response to an issued command.
[C]	A new peer joined the Broadcast Group.
[D]	A peer disconnected from Broadcast Group.

Response [R]:

success	Command is successful.
bad_command	Bad command.
bad_param	Bad parameter provided with the command.
wait_retry	Wait and try again later.
is_in_bg	Client is already listening to a Broadcast Group.
not_in_bg	Client is not listening to any Broadcast Group.
not_allowed	Command is not allowed.

## 4.0 COMMANDS

### 4.1 PING

Ping command will return the public IP version, address and the port number of the command issuing system.

command format: **ping**

response format: **IP\_ver IP\_addr Port**

response example:

[R] v4 12.45.87.3 143

[R] v6 2001:0db8:85a3:0000:0000:8a2e:0370:7334 80

### 4.2 LEAVE

The leave command is used to exit from the Broadcast Group. If the client is not participating in any BG then a `not_in_bg` response will be returned by the RTDS.

command format: **leave**

response example:

[R] success

[R] not\_in\_bg

### 4.3 EXIT

The client system will be disconnected from the RTDS server. The client will be automatically removed from the directory when the TCP connection with the server is broken.

command format: **exit**

### 4.4 LISTEN

This command will add the client to a Broadcast Group under a certain tag. Parameters BGT and BGID are mandatory. A TCP connection can only participate in a single Broadcast Group. Leave command has to be used before switching to another broadcast group or tag. If the client is participating in any BG then an `is_in_bg` response will be returned by the RTDS. All participants in the group will be notified about the new peer in the group (Note: BGT won't be published with the notification).

command format: **listen** bgid bgt

command example:

*listen* bitcoin@rtds server

response example:

```
[R]    success
[R]    not_in_bg
[R]    wait_retry
```

#### 4.5 HEAR

This command is the same as the “listen” command except when you hear a BG under a certain tag the other group members won’t be notified of your presence.

command format: **hear**      bgid    bgt

command example:

```
hear  bitcoin@rtds server
```

#### 4.6 BROADCAST (default max 256 characters)

The broadcast command is used to send a message across all the clients in the participating Broadcast Group. The BGT can be used to specify for which sub-group the message will be delivered. If the client is not participating in any BG then a not\_in\_bg response will be returned by the RTDS. The message can contain any ASCII printable character including white spaces (ASCII 32). For binary messages, any binary-to-text encoding can be used to broadcast the same. For broadcasting to the entire listening group use ‘\*’ asterisk (ASCII 42).

command format: **broadcast**      message      bgt

command example:

```
broadcast  Hello world  server
broadcast  Hello_world *
```

response example:

```
[R]    success
[R]    not_in_bg
[R]    wait_retry
```