



OPPO BDP-9X/10X Blu-ray Disc Player Network Remote Control Protocol (Version 1.0, September 25, 2012)

Note: This document describes a preliminary version of the network remote control protocol for OPPO BDP-93/95/103/105 players. This feature allows a network-connected device such as a PC or a smartphone to control the OPPO Blu-ray player. Please note the network remote control feature and protocol are still under development, future revisions may not be compatible with this preliminary version, and you will need to reprogram any controllers that use this version of protocol.

The commands and responses listed in this document require firmware version of:

- BDP-93/95: Main firmware of BDP9x-50-0608 or newer, MCU firmware of MCU93-09-0218 (for BDP-93), MCU95-08-0218 (for BDP-95) or newer.
- BDP-103/105: Main firmware of BDP10x-32-0921 or newer, MCU firmware of MCU103-05-0916 (for BDP-103), MCU105-04-0916 (for BDP-105) or newer.

In the remaining of this document, we use “*Player*” for OPPO BDP-93/95/103/105 player and “*Controller*” for the network-connected controlling device.

The complete network remote control procedure can be described in four sections: *DISCOVER*, *SIGN IN*, *SIGN OUT* and *SEND COMMAND*.

1. DISCOVER

Once the Controller is launched, it shall search for any available Player on the same local network. This search is done by broadcasting a message of “NOTIFY OREMOTE LOGIN” to UDP port 7624 (see sample code at the end of this section). Any Player available on the same network will respond upon receiving the broadcasted message. The data in the response message will contain text like “BDP-93 REPORT ADDRESS TO OREMOTE: 172.16.133.170:19999”. Here “BDP-93” is the Player model number, “172.16.133.170:19999” is the IP address and UDP port of the Player.

In subsequent communications, the Controller shall use this UDP address to send messages to and to receive responses from the Player. The reason for choosing UDP over TCP is that UDP does not require any connection setup. This is useful to mobile devices that can easily enter sleeping mode and lose all previous network connections.

Sample code for the broadcasting message (written in C):

```
int broadcast = 1;
int sockfd = -1;
struct sockaddr_in sin_addr, sin_broadcast;
const char* OREMOTE_BROADCAST_MESSAGE = "NOTIFY OREMOTE LOGIN";
short BROADCAST_PORT = 7624;
sockfd = socket(AF_INET, SOCK_DGRAM, 0);
```

```
setsockopt(sockfd, SOL_SOCKET, SO_BROADCAST, (const char*) & broadcast,
sizeof(int));
sin_broadcast.sin_family = AF_INET;
sin_broadcast.sin_addr.s_addr = inet_addr("255.255.255.255");
sin_broadcast.sin_port = htons (BROADCAST_PORT);
int message_length = (int)strlen(OREMOTE_BROADCAST_MESSAGE);
int send_length = bt_socket_udp_sendto(sockfd, OREMOTE_BROADCAST_MESSAGE,
message_length, (struct sockaddr *) &sin_broadcast, 0);
```

2. SIGN IN

The sign-in procedure is done in the following sequence:

- 1) The Controller sends a REQUEST (request) message to the UDP address of the Player to request permission to control. The message is written in XML format and a sample is shown at the end of this section.
- 2) The Player responds with a REQUEST ACKNOWLEDGE (request-ack) message. Since the Player can only be controlled by one Controller at a time, it only respond to the first received SIGN IN request and ignore any subsequent ones.
- 3) After sending the REQUEST message, the Controller shall wait for the REQUEST ACKNOWLEDGE message for up to 1 second.

If no REQUEST ACKNOWLEDGE message is received within this period, the Controller may re-send the REQUEST message again. If there is still no REQUEST ACKNOWLEDGE message after 3 attempts, the Controller shall exit the SIGN IN mode and return to the previous DISCOVER mode.

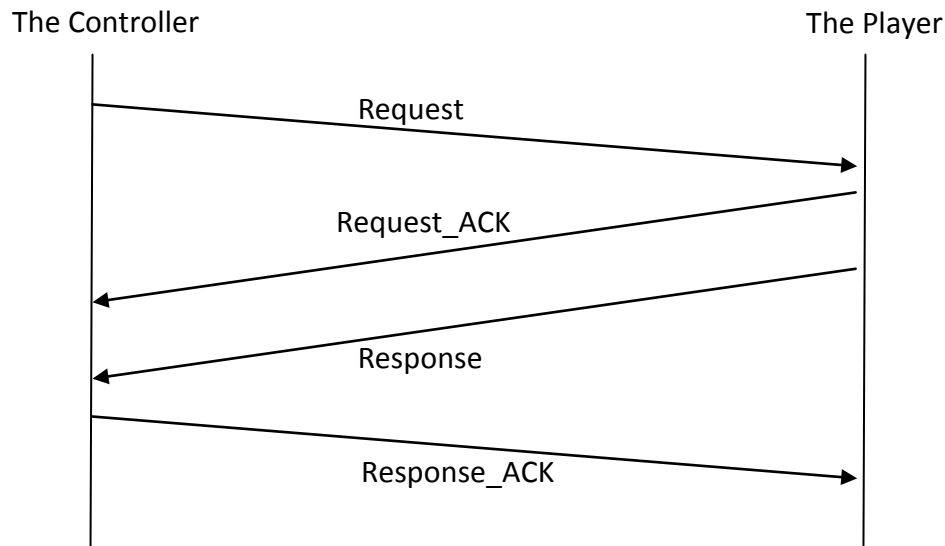
If the Controller successfully receives the REQUEST ACKNOWLEDGE message, it shall stay in the current mode and wait for a RESPONSE (response) message from the Player.

- 4) After receiving the RESPONSE message from the Player, the Controller shall acknowledge with a RESPONSE ACKNOWLEDGE (response_ack) message. The Controller and Player can then proceed to the next section (SIGN OUT or SEND COMMAND).

If no RESPONSE ACKNOWLEDGE message is received, the Player will re-send the same RESPONSE message for up to 3 times.

Please note that the Controller must not start a new session until the previous session is completed. Here “completed” means that the Controller has received the RESPONSE and sent the RESPONSE ACKNOWLEDGE to the Player.

A general flow diagram for this sign-in procedure is displayed as below:



A sample of the sign-in REQUEST message:

```

<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/oppo/remote/">
  <body>
    <request>
      <name>signin</name>
      <uid>fffbefbfa7060200180373d48892</uid>
      <requesttag>OSeWFcV78yJe8f9D</requesttag>
      <password>oppo80031722</password>
      <signintype>1</signintype>
    </request>
  </body>
</s:Envelope>
  
```

- `xmlns:s=http://schemas.xmlsoap.org/soap/oppo/remote/`: this is the namespace. All the XML messages listed in this protocol should use the same namespace.
- `<request>` defines the current message as a "Request". In the following sections, we will define more message types like Response, Acknowledge, and Push.
- `<name>` defines the current message as a "signin". In the following sections, we will define more message types such as Sign Out and Command Request.
- `<uid>` is the identification value for the Controller. This value shall be unique and the same value shall be used in all sections of the remote control procedure.
- `<requesttag>` is the identification value for a session. This value should be unique and the same value should be used in all sessions of this Sign-In section.

- <password>: this tag is reserved for future use. At this time please use the password in the above example as-is.
- <signintype>: this tag is reserved for future use. At this time please keep the value as 1.

A sample of the REQUEST ACKNOWLEDGE (request_ack) message:

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/oppo/oremote/">
  <body>
    <ack>
      <name>request_ack</name>
      <uid>fffbabbfa7060200180373d48892</uid>
      <requesttag>OSeWFcV78yJe8f9D</requesttag>
    </ack>
  </body>
</s:Envelope>
```

- Note that the <uid> and <requesttag> must be the same values as in the previous REQUEST (request) message, otherwise the Controller shall ignore this REQUEST ACKNOWLEDGE message.

A sample of the sign-in RESPONSE message:

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/oppo/oremote/">
  <body>
    <response>
      <name>signin</name>
      <errorcode>0</errorcode>
      <uid>fffbabbfa7060200180373d48892</uid>
      <requesttag>OSeWFcV78yJe8f9D</requesttag>
    </response>
  </body>
</s:Envelope>
```

- Note that the <name>, <uid> and <requesttag> must be the same values as in the previous REQUEST (request) message, otherwise the Controller shall ignore this RESPONSE message.
- <errorcode> is the result for this sign in request: 0 means success any other values mean failure. Please refer to Appendix A for detailed information.

A sample of the RESPONSE ACKNOWLEDGE (response_ack) message:

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/oppo/oremote/">
  <body>
```

```
<ack>
  <name>response_ack</name>
  <uid>fffbabbfa7060200180373d48892</uid>
  <requesttag>OSeWFcV78yJe8f9D</requesttag>
</ack>
</body>
</s:Envelope>
```

- Note that the <uid> and <requesttag> must be the same values as in the previous REQUEST (request) message, otherwise the Player will ignore this RESPONSE ACKNOWLEDGE message.

3. SIGN OUT

If a Controller decides to stop controlling the Player, it needs to “sign out” from the Player so that another Controller can request control from the Player. This procedure is quite similar to the SIGN IN procedure, and the message samples are presented below:

A sample of the sign-out REQUEST message:

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/oppo/oremote/">
  <body>
    <request>
      <name>signout</name>
      <uid>fffbabbfa7060200180373d48892</uid>
      <requesttag>roMzlzh3pm7nj1KN</requesttag>
      <signintype>0</signintype>
    </request>
  </body>
</s:Envelope>
```

A sample of the REQUEST ACKNOWLEDGE (request_ack) message:

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/oppo/oremote/">
  <body>
    <ack>
      <name>request_ack</name>
      <uid>fffbabbfa7060200180373d48892</uid>
      <requesttag>roMzlzh3pm7nj1KN</requesttag>
    </ack>
  </body>
</s:Envelope>
```

A sample of the sign-out RESPONSE message:

```
<?xml version="1.0"?>
```

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/oppo/oremote/">
  <body>
    <response>
      <name>signout</name>
      <errorcode>0</errorcode>
      <uid>fffbabbfa7060200180373d48892</uid>
      <requesttag>roMzlzh3pm7nj1KN</requesttag>
    </response>
  </body>
</s:Envelope>
```

A sample of the RESPONSE ACKNOWLEDGE (response_ack) message:

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/oppo/oremote/">
  <body>
    <ack>
      <name>response_ack</name>
      <uid>fffbabbfa7060200180373d48892</uid>
      <requesttag>roMzlzh3pm7nj1KN</requesttag>
    </ack>
  </body>
</s:Envelope>
```

4. SEND COMMAND

Once the session is established between the Controller and Player, the Controller can send commands to remotely control the Player. Each command contains a content of 3 characters (details in the following Command List) and commands are executed in the order they are received. The Controller shall wait for enough time for the command to be executed before sending the next command.

A sample of the SEND COMMAND (command) message:

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/oppo/oremote/">
  <body>
    <request>
      <name>command</name>
      <uid>fffbabbfa7060200180373d48892</uid>
      <requesttag>Y9ta7teRQC88bW3H</requesttag>
      <content>SRC</content>
    </request>
  </body>
</s:Envelope>
```

- The SEND COMMAND message mimics the infrared remote control behavior so there will be no response or acknowledgement from the Player. At this time this is one-way command only..

Command List:

The supported commands maps directly to the infrared remote (IR) control keys. No parameters are needed for these commands. The Player handles the commands as if it receives the same IR remote commands.

For actions that require a multiple-key sequence, such as go to a certain chapter (GOT command followed by multiple numeric key commands and the SEL command), the Player receives each command individually until the last command is received, at that time the action is carried out by the Player.

Command Code	Remote Key	Function
SRC	SOURCE (BDP-93/95) INPUT (BDP-103/105)	Go to Internet Menu to select Internet apps (BDP-93/95) Select the Input Source (BDP-103/105)
EJT	OPEN	Open/close the disc tray
POW	POWER	Toggle power (In network remote control, this command cannot turn on the Player. The Player must be turned on by other means before it can accept network remote commands.)
PON	ON (93/95 only)	Discrete on (Not in use. The Player must be turned on by other means before it can accept network remote control.)
POF	OFF (93/95 only)	Discrete off
SYS	P/N (93/95 only)	Switch output TV system: NSTC, PAL or MULTI(AUTO)
DIM	DIMMER	Dim the front panel display
PUR	PURE AUDIO	Pure audio mode (no video)
VUP	VOL +	Increase volume
VDN	VOL -	Decrease volume
MUT	MUTE	Mute audio
NU1	1	Numeric key 1
NU2	2	Numeric key 2
NU3	3	Numeric key 3
NU4	4	Numeric key 4
NU5	5	Numeric key 5
NU6	6	Numeric key 6
NU7	7	Numeric key 7
NU8	8	Numeric key 8
NU9	9	Numeric key 9
NU0	0	Numeric key 0
CLR	CLEAR	Clear numeric input
GOT	GOTO	Play from a specified location
HOM	HOME	Go to Home Menu to select media source or start Internet apps
PUP	PAGE UP	Show the previous page
PDN	PAGE DOWN	Show the next page
OSD	DISPLAY (93/95) INFO (103/105)	Show/hide the on-screen display

TTL	TOP MENU	Show BD top menu or DVD title menu
MNU	POP-UP MENU	Show BD pop-up menu or DVD menu
NUP	Up Arrow	Navigation
NLT	Left Arrow	Navigation
NRT	Right Arrow	Navigation
NDN	Down Arrow	Navigation
SEL	ENTER	Navigation
SET	SETUP	Enter the player Setup menu
RET	RETURN	Return to the previous menu or mode
RED	RED	Function varies by content
GRN	GREEN	Function varies by content
BLU	BLUE	Function varies by content
YLW	YELLOW	Function varies by content
STP	STOP	Stop playback
PLA	PLAY	Start playback
PAU	PAUSE	Pause playback
PRE	PREV	Skip to previous
REV	REV	Fast reverse play
FWD	FWD	Fast forward play
NXT	NEXT	Skip to next
AUD	AUDIO	Change audio language or channel
SUB	SUBTITLE	Change subtitle language
ANG	ANGLE (93/95 only)	Change camera angle
ZOM	ZOOM	Zoom in/out and adjust aspect ratio
SAP	SAP (93/95 only)	Turn on/off Secondary Audio Program
ATB	AB REPLAY	Repeat play the selected section
RPT	REPEAT	Repeat play
PIP	PIP (93/95 only)	Show/hide Picture-in-Picture
HDM	RESOLUTION	Switch output resolution
SUH	SUBTITLE (hold)	Press and hold the SUBTITLE key. This activates the subtitle shift feature
NFX	NETFLIX (103/105 only)	Netflix access button.
VDU	VUDU (103/105 only)	VUDU access button.
OPT	OPTION (103/105 only)	Brings up the context-related menu.
M3D	3D (103/105 only)	Enable 2D/3D conversion or bring the 3D adjustment menu.

Appendix A

This table describes the <errorcode> values and the corresponding results, as used in the sign-in and sign-out RESPONSE messages.

<errorcode>	Result Description
0	Success.
-1000	The sign-in request failed because the Player is already controlled by another Controller.
-1001	The sign-in request failed because the <password> value is incorrect.
-1002	The Player failed to generate the xml document for RESPONSE message.
-1003	The Player failed to parse the xml document of REQUEST message.
-1004	The <uid> is invalid.
-1005	The UDP address is invalid.
-1008	The Player failed to parse the parameters from COMMAND message.
-1019	The Player failed to parse the parameters from sign-in REQUEST message.
-1021	The network cannot be connected.