

TimeLord - ProDJ Event Script

1.1 Syntax

TimeLord has a small event scripting language available for use with ProDJ Link. Scripts can be assigned to any particular Track in the TimeLord session database, and trigger Actions based on different Events that occur as the DJ plays their set.

The core language is composed of three elements – Events, Actions and Arguments. These are used to form Statements, which have the syntax of:

```
Event Action(Argument[, 'Argument', ...])
```

Events occur when a particular condition has been met for a Track on a supported ProDJ Link player. For example, whenever a track is loaded into a player the onLoad Event is then triggered, and whenever the track is paused the onPause Event will trigger. Whenever these Events exist in any Statements in the tracks, the specified Action will occur.

Arguments for Actions will depend on the specific Action itself, and the expected syntax is described in the Action list later in this document. Arguments are either typed as *Integer* (a number) or *String* (text, words), and separated by a comma if more than one argument is available for the Action. Events and Actions are not case sensitive, eg. onLoad NoTifY('Abc123') is valid.

String arguments must be contained within single quotes, and can only contain double quotes if any. For example:

```
; this is valid - note the double quotes are inside the text,
; and the single quotes surround the text
```

```
onPause MA2('Goto Exec 1."Color" Cue "Blue" Fade 4');
```

```
; this is not valid because the quotes are wrong
onPlay MA2("Goto Exec 1.'Color' Cue 'Red' Fade 4")
```

Blanks lines, or lines beginning with a semi-colon (;) will be ignored, and lines beginning with a semi-colon can be used for comments and documentation.

Some Actions have arguments which are optional, and can be replaced with a # symbol to use the default argument.

```
; Here the optional TimeOut argument of Notify is set to the default value
onLoad Notify('Remember the flame cannons!', #, 255, 255, 0);
```

Only one Statement is triggered per line, though any one Event may have more than one Statement associated with it. In this example, both the "Notify" and "MidiNote" actions will triggered one after another when the "onLoad" Event occurs.

```
; The "onLoad" Event triggers when this Track is loaded into the player
; "Notify" expects at least one argument, the message as a String
onLoad Notify('Remember the flame cannons!');
```

```
; "MidiNote" Action expects three Integers for Channel, Note and Velocity
onLoad MidiNote(1, 64, 127);
```

Conditional statements and math etc. are not yet supported but the language may be extended in the future, with backward compatibility maintained so existing scripts will still trigger.

2.1 Events

Events are triggered as the ProDJ network updates, and mostly focused on player state changes. The currently available events are listed below:

OnLoad - Triggered on loading of this track into the player. This is the first event that will occur on loading this track.

OnUnload - Triggered when the track is unloaded. This is the last event what will occur for this track.

OnStart - Triggered once only after onLoad, this occurs when this track starts playing for the first time since it was loaded.

OnPlay - Triggered every time the track state changes to playing.

OnStop - Triggered when the track has reached the end and stopped automatically.

OnPause - Triggered whenever the track is paused.

OnCued - Triggered whenever the track is cued.

OnReverse - Triggered every time this track's playing direction changes to reverse.

OnForward - Triggered every time the track changes from reverse to forward play.

OnBeat - Triggered on the beat every time the player decides a beat occurs.

OnLoop - Triggered when the track has started looping.

OnSearch - Triggered when the track is being sought by the DJ.

OnCueing - Triggered when the track is being cued by the DJ.

OnTempoChange - Triggered every time the tracks tempo value changes.

OnMaster - Triggered every time when this track becomes the master.

OffMaster - Triggered when the player was master, but is now not.

OnAir - Triggered when the player is LIVE (if a mixer is present).

OffAir - Triggered when the player was LIVE but is now not live (if a mixer is present).

OnFound - Triggered every time this player is found, ie. connected to the network.

OnLost - Triggered every time this player is lost, ie disconnected from the network.

3.1 Actions

Actions are executed when an associated Event occurs. Actions requiring Arguments must have these Arguments contained within parenthesis () and separated by commas. Actions will have different Argument type requirements as described below, the type of the Argument is shown in *Italics*.

Follow(*Integer* Player_Number)

The master output of TimeLord should now follow this player number, and this overrides the auto follow master option. Player number should be between 1 and 4, with 0 being auto.

Example:

```
; whenever this track loads, start following player 3  
onLoad Follow(3)
```

MidiNote(*Integer* Channel, *Integer* Note, *Integer* Velocity)

Sends a MIDI Note out of the default MIDI device, on desired MIDI Channel at specified Velocity. Channel can be between 1 and 16, Note and Velocity can be between 0 and 127.

Example:

```
; every beat send MIDI note 2 on channel 1, with velocity 127  
onBeat MIDINote(1,2,127)
```

MA2(*String* Command)

Send a command to the connected MA2 via MA2-Link as the connected user. You can use double quotes inside the command only, but must be surrounded with single quotes.

Example:

```
; send the connected MA2-Link this command... Goto Exec 1."wash" Cue 2  
onPlay MA2('Goto Exec 1."wash" Cue 2')
```

Notify(*String* Message [, *Integer* TimeOut, *Integer* Red, *Integer* Green, *Integer* Blue, *Integer* Alpha])

Display a notification to the TimeLord user's PC.

Optional:

TimeOut is specified in milliseconds, defaulting to 2000 – 2 seconds.

Red, Green, Blue are 0..255 values that specify the colour of the text, default to 0,255,255.

Alpha is the apparent brightness of the text 0..255, defaulting to 255 as full brightness.

Examples:

```
; Display messages on the TimeLord screen.  
onLoad Notify('Remember Confetti Drop!')  
onPlay Notify('5 Seconds in Red at half brightness", 5000, 255, 0, 0, 127)
```

4.1 Tokens

Tokens are used to include dynamic information from in Action arguments. They are a substituted string, called using the syntax {*\$NAME_OF_TOKEN*} as shown in the example below:

```
; Display the running TimeLord version...  
onLoad Notify('TimeLord Version is {$APP_VER}')
```

Available Tokens include:

P_NUM - The player number that has triggered this script. This can be 0 if the script was not executed by a player, rather by the system.

P_DBID – A number unique to this track and player, same as the TimeLord Session ID for the track as shown in the database.

TIMELORD PRO DJ

P_MA2ID – As configured for the track in the TimeLord Session database. Eg: 123 or 54 or whatever the TimeLord user assigns to the track. If not assigned in the database, will be 0 and will silently fail in MA2 as 0 cannot exist.

P_ARTIST – Track name as a String

P_ALBUM – Album Name, String

P_TITLE – Title of the track loaded, as a String

P_GENRE – Genre of the track loaded, as a String

P_COLOUR – Colour of the track loaded, as a String

"None", "Orange", "Red", "Blue", "Green", "Aqua", "Yellow", "Pink", "Purple"

P_COMMENT – Any comments the track has, as a String

P_BPM – Replaced with the playback BPM, eg: 145.50

P_TBPM – Replaced with the track BPM, eg: 120.00

P_MASTER – Replaced with 1 if this player is master, 0 if this place is not master.

P_ONAIR – Replaced with 1 if this player is in the mix, 0 if not. This is only valid if a DJM mixer is present in the ProDJ Network.

P_EVENT – Special token, and is replaced with the name of the event that has occurred. It can be used to trigger an Action whenever any Event occurs. For example, It can be used like this:

```
; This will call Notify for every Event that ever occurs for this track,  
; and display the name of the Event that occurred.  
{P_EVENT} Notify('Event "{P_EVENT}" has occurred')
```

The following relate the current master player in the ProDJ network, as indicated on the players MASTER button.

PM_MASTER – Number of the current master player

PM_ARTIST – Track name as a String

PM_ALBUM – Album Name, String

PM_TITLE – Title of the track loaded, as a String

PM_GENRE – Genre, String

PM_COLOUR – Colour, String

APP_VER – (*Constant*) Return the version string of the currently running TimeLord software. Of little use.