



# USING PALLADIUM WITH REPLAY APPLICATIONS

issue 04, 19<sup>th</sup> December 2007

## CONTENTS

1. Introduction .....	2
2. MIDI Communications .....	2
2.1 Different Computers .....	2
2.2 Same Computer .....	2
3. Who is the Master? .....	2
4. Controlling the Replay Application .....	3
4.1 Follow Mode .....	3
4.2 Linear Mode .....	3
4.3 Non-Linear Mode .....	4
4.3.1 Buffer Considerations .....	4
5. Oops, I didn't mean to press go .....	5
6. MIDI Protocol .....	5
6.1 Stop .....	5
6.2 Pause .....	6
6.3 Cue .....	6
6.4 Play .....	6

## 1. Introduction

This document provides a brief introduction to using Palladium in conjunction with replay applications. Since there are a number of such applications in existence today, and the likelihood is that more will become available in the future, this document is primarily written in general terms without too much reference to any one specific application. Having said that, there are a couple of specific examples at the end of the document.

## 2. MIDI Communications

When using Palladium with a replay application, there are two distinct scenarios:

- Palladium and the replay application are running on different computers
- Palladium and the replay application are running on the same computer

### 2.1 Different Computers

In this scenario, the replay application needn't be running on a PC at all; it could be just as easily be running on a MAC or Linux box. However, in all cases the MIDI communications will need to be in the form of a physical cable between the two computers. Whilst this cable is most likely to be MIDI, it is also possible to send MIDI data over other channels such as Ethernet, if the appropriate drivers are used.

### 2.2 Same Computer

In this case, while a cable could still be used between two of the PC's ports, this isn't a very "neat" solution. It also has the disadvantage of using two ports which are probably a scarce resource, and better used elsewhere in the sound system.

An alternative approach is therefore to use a special type of driver to pipe the data from one application to the other. One such driver which has been tested by CH Sound Design is LoopBe. This provides a virtual MIDI connection between a number of applications running on the same PC, and is available in a number of versions ranging from single in / single out to 8 ins and 8 outs. LoopBe can be found at [www.nerds.de/en/download.html](http://www.nerds.de/en/download.html).

## 3. Who is the Master?

During performances, there should ideally be one application which is being controlled directly by the sound operator. This application will then generate MIDI data which is sent to the second application via the above mentioned MIDI communications channel. In the following discussions, the former application will be referred to as the "master", with the latter

application referred to as the “slave”.

Since Palladium is capable of generating or receiving almost any type of MIDI data, as are most replay applications, the question then arises as to which application should be the master? At CH Sound Design, we believe Palladium should be the master for the following reasons:

- Palladium’s primary focus is actors and their entrances and exits, and thus its cue list is likely to have significantly more cues than the replay application’s cue list. Taking a recent production of “Les Miserables” as an example, Palladium had 182 cues in its cue list, of which only 6 were replay related. Now, if the replay application were the master, it would need to have a large number of “empty” cues added just so it could trigger Palladium for each actor movement. This is a lot of extra work for no real benefit.
- If an extra actor movement is added to the show, having the replay application as the master means that two cue lists need to be updated rather than just one.
- If Palladium were the slave, it is likely that some of its key features such as cross fades and emergencies would become either unavailable, or at least difficult to access.

## **4. Controlling the Replay Application**

Now that we have decided that Palladium is the master and the replay application is the slave, we can turn our attention to exactly how we want to control the replay application. There are a number of ways in which this can be done:

### **4.1 Follow Mode**

In this mode, Palladium simply broadcasts its current cue Id via MSC (MIDI Show Control) whenever this changes, and it is up to the replay application to take the appropriate action. Whilst this is conceptually very simple, it can suffer from two drawbacks:

- If a replay related cue Id in Palladium changes, it will be necessary to update the corresponding cue Id in the replay application.
- Since Palladium is broadcasting Ids for all cues, irrespective of whether or not they are replay related, the replay application will likely be receiving a large number of Ids for which it does not have any corresponding cues. Depending upon the robustness of the application, this may or may not cause problems.

### **4.2 Linear Mode**

In this mode, Palladium treats the replay application as a playback device, and simply sends it a generic “go” command whenever it wants the next track to be played, much as it does now with a CD player with an auto-pause feature. This approach requires that the replay

application's cue list is in sync with Palladium's, and can lead to highly undesirable results if the two end up out of sync for some reason. This is particularly likely during technical rehearsals when one is tending to jump around the script a bit (a lot!), or if the sound operator accidentally presses Palladium's "next" button (refer "Oops, I didn't mean to press go", below). Therefore, this is the least desirable mode of all.

This mode can however be improved upon if the replay application supports "go to cue" commands, as this at least allows Palladium to send such a command just before it sends the go command.

### 4.3 Non-Linear Mode

In this mode, Palladium again treats the replay application as a playback device, but now sends the replay application a "Go" command which includes the required cue Id. This approach, while taking slightly longer to program, means that the two applications will never end up out of sync, and is thus a much safer way of doing things.

Also, now that Palladium is telling the replay application which cue to trigger (and thus which sound to play), there is no longer a need for the replay application to maintain a complete cue list.

Taking a simple example, if a show contained 3 identical telephone rings plus 3 identical door bells, all the replay application really needs is one of each. Palladium will then simply tell the replay application to either play the telephone ring, or play the doorbell.

This approach is particularly useful where the number of times each effect is played changes, as now it is only Palladium's cue list that needs to be updated; the replay application's cue list is unaffected by these changes.

#### 4.3.1 Buffer Considerations

There is however one important issue which needs to be considered when using non-linear mode. Some replay applications maintain only the next effect in a RAM buffer ready to go, with all other effects residing on the hard drive. Thus if Palladium were to ask for an effect other than the next one, there might be a short delay (latency) while the replay application loaded the required effect from the hard drive into RAM.

Thus it is important to check whether or not your replay application behaves in this regard, and if it has any options which can be set to overcome this.

Another possible solution might be to assign a "hotkey" to each effect; this should cause the replay application to always maintain all "hotkey" effects in RAM ready to go.

## 5. Oops, I didn't mean to press go

In addition to the “Next” button, Palladium also provides a “Previous” button. This button has two functions:

- It provides a simple mechanism for stepping backwards a couple of cues during rehearsals.
- It provides an “undo” for those situations where the sound operator presses the “Next” button too soon.

However, consider the following scenario: Palladium is in cue 42, and cue 43 contains a trigger for the replay application. Now, if the sound operator accidentally presses “Next”, and then realising their mistake presses “Previous”, the replay application will still have received its trigger and be playing the sound, even though this is no longer required.

This can be easily overcome by adding a “cue to the start of sound” trigger to cue 42. That way, when the sound operator goes back to cue 42, Palladium will send a command to the replay application which will cause it to stop playing the sound, and re-cue to the start ready for when we really do want to go to cue 43.

Similarly, for those cues where a sound is meant to be stopped, if the replay application supports pause and resume, then all that is needed is to add a resume command to the cue before the pause command.

## 6. MIDI Protocol

Since Palladium is capable of generating almost any MIDI command, and most replay applications are capable of accepting almost any MIDI command, you could in fact use almost any commands you like for the communications between the two applications.

However, such an approach will likely result in a large number of incompatible files, which will make it difficult for files created by one user to be easily shared with others. CH Sound Design would therefore like to recommend the following command set, which corresponds to each of Palladium's playback device control buttons. However, we realise that not all functions will be applicable to all playback applications.

### 6.1 Stop

This is intended to stop all sounds instantly. We recommend the MIDI “all notes off” command for this purpose (Bn.7B.00), where n is the MIDI channel number.

## 6.2 Pause

This is intended to either pause a particular track, or resume a previously paused track. We recommend the MIDI “note off” command for this purpose (8n.TT.00), where n is the MIDI channel number, and TT is the effect or track number.

## 6.3 Cue

This is intended to cause the replay application to jump to a particular track in its cue list. We recommend the MIDI “program change” command for this purpose (Cn.TT), where n is the MIDI channel number, and TT is the effect or track number.

## 6.4 Play

This is intended to start the playing of a particular effect from its beginning. We recommend the MIDI “note on” command for this purpose (9n.TT.00), where n is the MIDI channel number, and TT is the effect or track number. The track number 00 is reserved for the next track in the replay application’s cue list.