

**CONCEPT: FOLLOW FEATURE (MASTER/SLAVE TRACK)**  
**(Buzap, 9Oct2009)**

One track becomes **master track** for the **follow feature**. Other tracks can be activated to follow the master track and become **slave track(s)**. Whenever a loop is switched in the master track, the current loop for a slave track is remembered. When switching back to the previous loop in the master track, the previous loops of the slave tracks will also be played automatically.

Here is an example how to use the follow feature:

Track1=Master	Track2=Slave	
Loop1 (Verse)	Rhythm Loop1	
Loop2 (Chorus)	Rhythm Loop1	<i>switch to loop2 in master track</i>
Loop2 (Chorus)	Rhythm Loop2	<i>new loop2 in slave track</i>
Loop1 (Verse)	Rhythm Loop1	<i>switch back to loop1 in master track, slave track follows &amp; remembers loop1</i>
Loop1 (Verse)	Rhythm Loop3	<i>new loop3 in slave track</i>
Loop2 (Chorus)	Rhythm Loop2	<i>switch back to loop2 in master track, slave track follows &amp; remembers loop2</i>
Loop1 (Verse)	Rhythm Loop3	<i>switch back to loop1 in master track, slave track follows &amp; remembers loop3</i>

You can use this i.e. with a song structure in the master track. The loops in the slave track(s) will follow your master track automatically.

For a slave track, there are four main **follow modes**, defining behaviour when switching a loop in the master track:

- **off** :  
Regular behaviour, slave track keeps going on.
- **follow** :  
Current loop in slave track will be remembered. Corresponding loop in slave track is played. If there is no corresponding loop in slave track, slave track keeps going on.
- **exist** :  
Similar to *follow*. When there is no corresponding loop in slave track, slave track will be muted..
- **mute** :  
This is a forced mute. Slave track will be muted with every switch in master track.

Here is a comparison how the slave track would behave with different *follow mode* settings:

Track 1	Track 2 (slave track)				
Master	off	follow	exist	mute	(different follow modes)
Loop1	Loop1	Loop1	Loop1	Loop1	
Loop2	Loop1	Loop1	-	-	<i>switch to loop2 in master track</i>
Loop2	Loop2	Loop2	Loop2	Loop2	<i>play loop2 in slave track</i>
Loop1	Loop2	Loop1	Loop1	-	<i>switch back to loop1 in master track</i>

Here are detailed specifications of the *follow* feature:

## VARIABLES (GLOBAL)

**followMaster**      0 | <track number>

Specifies which track is master track. With 0, follow feature is completely turned off. Possibly, this could be activated with a nice "M"-button in the GUI (track settings).

**followMasterMute** off | on

Determines if all slave tracks will be muted when master track is muted.

## VARIABLES (FOR EACH TRACK)

**followMode**      off | follow | exist | mute | remember

- **off** - Regular behaviour. Nothing will happen when switching loop in master track.
- **follow** - Current track is a *slave track*. When a loop is switched in master track, the current loop in the slave track is remembered. When switching back to the previous loop in the master track, the corresponding loop will be played in the slave track.  
When there is no corresponding loop for the slave track (i.e. new loop in master track), nothing happens. I.e., the current loop in the slave track could keep on playing.  
This is the default mode for using the follow feature. You can use it i.e. for a track with percussive loops that accompany the master track.
- **exist** - Current track is a *slave track*. This works similar to *follow* mode. However, when there is no corresponding loop for the slave track, the slave track will be muted. There could be no corresponding loop in the slave track due to a new master track loop or if previously the corresponding loop was reset/muted in slave track.  
You can use this mode i.e. to add a short loop with sound textures to enhance your master track loop. It will be turned off when you switch to another loop in the master track, allowing you a fresh start.
- **mute** - This is a forced mute. Slave track will be muted with every switch in master track. Keep in mind that current track is still a *slave track*. So, if you switch back to *follow* mode, the corresponding loops will still be remembered. This mode is useful for temporarily adding sounds to your master track loop that you would like turned off later.
- **remember** - Current track behaves just like a normal track with *follow mode off*. However, corresponding loops will still be remembered when switching the master track loop. If *followMasterMute* is turned on, this track will not be muted.  
This is useful if you don't want the current track to be a slave track now. But you want to turn on the follow mode later on.

The *followMode* is obsolete when the current track is master track or follow feature is inactive (followMaster=0).

### **followOutput**      *off | follow | reduce*

In addition to following loops, you can also keep track and follow the *output audio levels* when switching master track loops. The *followOutput* is only active when the follow feature is used.

- **off** - When loops are switched, the current track output levels remain.
- **follow** - When switching a loop in master track, also the current track output level will be remembered and automatically switched to previous output level. This is useful for having different mixes for the output levels.
- **silent** - Behaves like *follow*. In addition, instead of muting the slave track (i.e. with *followMode = exist* or *mute*), output level will be reduced to 0.

Be aware that this parameter also affects master track output level!

### **followMuteDetect** *off | on*

Normally, no corresponding loop in the slave track will be remembered when it has been empty or muted. In addition, with *followMuteDetect*, a slave track can be treated just like a muted track when the current output level is turned down (lower than the value for *followMuteSensitivity*).

This is useful in a situation where you had the slave track volume turned down. Otherwise, a slave track loop you didn't hear before could jump back in while switching.

### **followMuteSensitivity**      *<output level>*

Defines the sensitivity for *followMuteDetect*. Obviously, only a low value (i.e. 12) would make sense. A value of 0 would require the current track output volume to be completely turned off. The sensitivity defines what is considered "almost turned off".

## **FUNCTIONS**

### **FollowNextMute**

Forces the next master track loop switch to behave like *followMode = mute* for the current (slave) track. After that it returns to previous behaviour.

You can use this feature i.e. if you work with *followmode = follow*. However, with the next loop in your master track, you want to go to a different direction and don't want a slave track to follow.

### **FollowNextOff**

Forces the next master track loop switch to behave like *followMode = off* for the current (slave) track. After that it returns to previous behaviour.

### **FollowFlushTrackMemory**

All remembered corresponding loops & output levels for the current (slave) track will be reset.

### **FollowFlushAllMemory**

All remembered corresponding loops & output levels will be reset for all tracks.

### **FollowFlushOutputMemory**

All remembered corresponding output levels only for all tracks will be reset.

**Reset, General Reset, Global Reset**

These standard functions also need to reset the remembered corresponding loops for a slave track (or all tracks).

- Buzap.