

Songlib: bend

written by: John C. Lusth

Revision Date: February 13, 2012

The bend family of note playing functions is similar to the play family, except the notes start out a little flat or a little sharp before resolving to the correct pitch. The bend family follows the n,r,d convention.

```
void bend(double beats,int instrument,int octave,int pitch,  
          double bendFactor,double startLength, double resolvedBeats);
```

```
void nbend(double beats,int instrument,int numberedNote,  
           double bendFactor,double startLength,double resolvedBeats);
```

```
void rbend(double beats,RRA *r,  
           double bendFactor,double startLength,double resolvedBeats);
```

```
void dbend(double beats,int *data,int length,  
           double bendFactor,double startLength,double resolvedBeats);
```

A *bendFactor* less than one means the note starts out flat. For example, a bend factor of `1 / SEMITONE` means the note starts out one semitone low. A *bendFactor* greater than one means the note starts out sharp. For example, a bend factor of `SEMITONE` means the note starts out one semitone higher.

The *startLength* argument specifies how long the note should remain at its initial low or high pitch. The *resolvedBeats* argument specifies how long it takes the note to shift from the low or high start to the final resolved pitch.

Here is a typical use. The note plays for a whole note and starts out one semitone flat:

```
bend(W,harmonica,3,C,1 / SEMITONE,Q,I);
```

The note starts out at B2 and remains there for one quarter note. At that point, the note will start to travel upwards to reach C3. It will take one eighth note to resolve.