Universal Tuning Editor

Hπ INSTRUMENTS
Aaron Andrew Hunt
Changes from Previous Documentation

Current Version, v6 — 27. April 2019

Previous Versions

Introduction

Features List
Version Roadmap
User Interface Basics
Maximising the Detail View
Maximising the Tuning List
Toolbar
Bug Reporting & Feedback
Feature Requests

1. MIDI Settings

Input Sources & Output Destinations
Rescanning MIDI
Synthesizer Settings
MIDI Instrument Input
Working with Devices

2. Instruments

uInstruments
Instrument Definition Files (.uinst)
The Structure of a uInstrument
uVersions & uOptions
MIDI Notes & Channels
Altering Default uInstruments
Creating a New uInstrument

3. Tones

Reference Tone
Tonic
4. Scales

Tunings vs. Scales .................................................................18
Supported Tuning Files ..........................................................18
Importing a Scale .................................................................20
Caveats About Importing Scala .scl Files ..................................21
Mapping a Scale .................................................................21
Mapping from All Keys ..........................................................21
Mapping from Selected Keys ..................................................21
Mapping from Octave & Key ....................................................22
Tuning Equal Divisions ..........................................................22
Exporting Scales .................................................................23
Scale Library ........................................................................23

5. Tuning List

Input, Tuning, & Output Columns ........................................24
Units .....................................................................................24
About Period Numbers ..........................................................24
Tuning Entries .................................................................25
Comments ............................................................................25
Column Width Adjustments ....................................................25

6. Editing Tunings by Selection

Transpose .............................................................................26
Repeat ....................................................................................26
Reverse Order .......................................................................26
Tune Scale ............................................................................26
Tune Equal Division .............................................................27
Tune Equal Steps .................................................................27
Convert to Ratio .....................................................................27
Convert to Decimal ...............................................................27
Convert to Units .....................................................................28
Reduce ET Notation ...............................................................28
Reduce Ratios.................................................................28
Reduce Entry ...............................................................28
Set Period.................................................................28
Period Up / Down ........................................................28

7. Display Options ........................................................29
Tuning List ....................................................................29
Text on keys ..................................................................29
Animate Instrument ......................................................29

8. Devices — TBX2 .......................................................30
MIDI Interface Requirements .......................................30
Editing Data Stored on Your TBX2 .................................31
Programming a Tuning ..................................................31
Batch Programming Tunings ..........................................31
Programming a Preset ....................................................33
Setting Global Parameters ............................................34
Setting USR Data ........................................................34
USR Data Example 1 .....................................................36
USR Data Example 2 .....................................................36
Backing up & Restoring Data (.udevicebackup) .................37
Updating Firmware (.ubootsyx) .......................................37
Troubleshooting ........................................................38

APPENDIX: uInstrument (.uinst) Short Guide .................39
Names & Descriptions ..................................................39
Shapes ........................................................................39
Colors ........................................................................39
Positions: Left & Top ...................................................39
Structure .....................................................................40
Display Names ............................................................40
MIDI Notes & Channels ...............................................40
Versions .....................................................................40
Options .............................................................................................................41
Navigation .......................................................................................................41
Default Tuning ..............................................................................................41

Credits 42
Changes from Previous Documentation

Here are lists of changes for each version of this documentation.

Please report typos or problems with this text via email to hpiinstruments@zentral.zone

Current Version, v6 — 27. April 2019
• Documentation updated with release of UTE v1.4.0
• Chapter 4: Exporting Scales added description of the new .scl from selected notes options
• Chapter 6: added description of the new Repeat function.
• Changed how entries in this section are handled in the table of contents.

Previous Versions

v5 — 8. April 2019
• Documentation updated with release of UTE v1.3.7
• Chapter 6: fixed typo under Tune Equal Steps.

v4 — 27. March 2019
• Documentation updated with release of UTE v1.3.5
• Introduction: updated Features List and Version Roadmap.
• Chapter 2: added new sections Altering Default uInstruments and Creating New uInstruments.
• Chapter 5: added sections Comments and Column Width Adjustments.
• Chapter 6: added sections Tune Equal Steps, Reduce ET Notation, Reduce Ratios, and Reduce Entries.
• Chapter 7: updated Display Window image.

v3 — 2. March 2019
• Chapter 3: updated Tones Window images and definition of Tonic to reflect changes.
• Chapter 4: updated Scales Window images and text about how scales are mapped to keys.
  Added new formats to Export chart
• Chapter 5: removed text about the ø character in Tuning Entries because use of this character is not supported in UTE due to the paradigm change from octave-based entries to period-based entries. Added section on Units column.
• Chapter 6: added Convert to Ratio, Convert to Decimal and Convert to Units new Selection menu item function descriptions.
• Chapter 7: updated Display Window images and description.
• Appendix: edited some .uinst tag descriptions for clarity.

v2 — 15. February 2019
• Introduction: Updated text and added new features to features list, and added a section outlining the toolbar buttons.
• Chapter 1: Added section on rescanning MIDI (Windows-only necessity), and a section explaining the new MIDI Instrument Input functionality.
• Chapter 4: rewrote text on the topic of Mapping to reflect new feature set of the Scales window. Added sections on Tuning Equal Divisions, and Exporting Scales.
• Chapter 6: Added chapter for Editing Tunings by Selection to explain the options in the newly added Selection menu.
• Chapter 7: Added short chapter Display Options.
• Chapter 8: Devices - this chapter was previously chapter 6.

v1.3 — 28. January 2019
• Chapter 7: Added information about MIDI Interface requirements, added a diagram and LCD image for TBX2 Firmware Update Startup, and a Troubleshooting section for MIDI problems.
• Moved this section out of the introduction and gave it chapter heading status.
• Added publication date to page header.

v1.2 — 3. October 2018
• Chapter 7: Added information on updating TBX2 firmware.

v1 — January 2018
• Initial release
Introduction

Universal Tuning Editor (UTE, for Mac OSX and Windows) is the result of over a decade of experience developing tuning hardware and software through H-Pi Instruments. UTE combines the strongest features of existing H-Pi software in a newly conceived codebase, improving the shortcomings of existing software and expanding possibilities for the future. A revolutionary idea behind UTE is to approach tunings according to any arbitrary geometry of any MIDI keyboard instrument imaginable. This is done by implementing XML instrument definition files which users can write themselves.

A future aim of the software is to provide a translator for the many different file formats now used for tuning. CSE has been doing this job for some time, and with UTE a new concept is introduced: the .utuning file, a universal container for any existing type of tuning file.

UTE is developed along with TBX2, a hardware device capable of storing over 8000 tunings. TBX2 is designed to receive input from a MIDI controller and send microtonal output to a MIDI synthesizer. UTE allows TBX2 owners to upload tunings, update firmware, manage presets and parameters, and so on. Support for other devices will be added in future releases of UTE.

Features List

- user-definable instrument geometry
- internal synthesizer microtonal output
  - Mac version supports soundfonts, Windows version depends on system
- import tunings in any existing tuning format
- export tunings in a variety of tuning formats
- map tunings automatically to keyboard keys
- user-definable reference tone and tonic
- tune any key to any pitch using flexible tuning entries
- play MIDI in from any instrument to hear the pitches you define
- animated graphic instrument responds to incoming MIDI
- create equal division tunings (octave or non-octave)
- create equal steps ascending or descending of any size
- transpose tones up or down by any tuning entry
- set or alter period placement of any tone
- convert between ratios, decimals, and logarithmic units (like cents)
- Copy / Paste and Paste Into Selection options
- native filetypes:
  - .ute - for projects
  - .uinst - for instrument definitions
  - .utuning - a container for tuning files in any format
  - .ubootsyx, .uboothex - for firmware files
- .udevicebackup - for device data backup files
- features for TBX2 users:
  - upload tunings to memory
  - send firmware updates
  - program any of 40 presets
  - program global parameters
  - program any of 10 USR datasets
  - backup and restore memory
- automatic and menu-option bug reporting

**Version Roadmap**
Current UTE development is focused on editing and export features for working with scale files, instruments, and a tuning device (TBX2). In each new release, more advanced features are added. It is planned that UTE will eventually incorporate a searchable Scale Database similar to ScalaVista, but more flexible.
**User Interface Basics**

UTE is a multi-document (multi-project) application featuring a straightforward, single-window interface with a simple toolbar and menu options. The area across the top of the window shows an overhead view of the currently loaded instrument. This area is called the **Overview**. In the Overview is a yellow box called the **Viewport**. The contents of the Viewport are shown below the Overview in expanded form in an area called the **Detail View**. At the far right edge of the Overview is a vertical navigation bar listing the number of periods in the instrument. Clicking on a number in the navigation bar moves the Viewport to that Period of the instrument. The majority of the right area of the window contains a list of all the tones which are assigned to the keys of the instrument, called the **Tuning List**.
**Maximising the Detail View**

The views can be made larger or smaller, and can also be moved in order to make best use of space for a given instrument. If an instrument takes up little horizontal space, the Overview can be pushed to the right by double-clicking the shutter icon on the left side, so that the Detail view can fill the entire left side of the window.
Maximising the Tuning List
Alternatively, the Overview can be pushed to the left by double-clicking the shutter icon on the right side, to give more space to the Tuning List.

Toolbar
The order of topics in this text roughly follows the order of toolbar buttons from left to right.

- **MIDI IN** — set the MIDI Input Port (Chapter 1)
- **MIDI OUT** — set the MIDI Output Port and Synthesizer Options (Chapter 1)
- **Instruments** — load a .uinst file and select options for an instrument (Chapter 2)
- **Tones** — define Reference tone and Tonic frequencies (Chapter 3)
- **Scales** — import, create, and manage scales (Chapter 4)
**Device** — select a device or perform a function for the selected device (*Chapter 8*)

**Display** — decide how aspects of the interface should look and respond (*Chapter 7*)

**Preferences** — select global options to be applied to new projects

### Bug Reporting & Feedback

Please report any problems you may experience with UTE directly by using the menu item *Report a Bug*. Before doing so, please also check the [UTE reports webpage](#), which lists all known issues and feature requests. Feedback which is not about bugs may be sent by email directly or using the menu item *Send an Email*.

```
Please report bugs as described, and support will proceed via email to resolve the issue.
```

### Feature Requests

If the software does not do something you would like it to do, and you are willing to pay for the feature you want, use the menu item *Request a Feature* to describe the feature and make an initial offer to pay for it. A professional wage for programming is not expected. 20 € is an acceptable starting point, considering that adding any feature requires several hours of work. If your idea makes sense and your offer is reasonable, then a payment schedule is agreed upon and a testing stage begins. Once testing is done and the feature is verified as working, a new version of the software is released including the new feature(s).

```
Not all features are possible or will be considered relevant for the majority of users. A minimum offer of 20 € is standard for all Feature Requests.
```
1. MIDI Settings

UTE communicates with MIDI devices, so an important first step when using the software is to properly set up MIDI connections.

**Input Sources & Output Destinations**

All available MIDI inputs and outputs are shown in the MIDI window, which can be opened by selecting the menu item *MIDI > Input / Output* (Command+M) or by clicking the MIDI icon in the toolbar and selecting *MIDI Input / Output* from the popup menu. On Mac, UTE creates one virtual input port and one output port which can be used to route MIDI to and from other software. It is however recommended to use Apple’s native IAC Bus for this because the connections will persist from one session to the next without having to reselect the port upon each use. To set up the IAC Bus ports, open *Utilities > Audio MIDI Setup*, go to *Window > Show MIDI Setup*, and follow the instructions provided by Apple using the *Help* menu.

Windows users who want to use virtual MIDI ports must install a driver for them such as LoopBe1 or MIDIYoke.

**Rescanning MIDI**

Mac users do not need to rescan MIDI ports when connecting and disconnecting MIDI gear, as the device list gets updated in real time. Windows users need to open the *MIDI* window and click the *Rescan* button (also available in many *Device* windows) whenever a device is connected or disconnected.

**Synthesizer Settings**

Once an Instrument is loaded, access to the internal synthesizer appears in the MIDI popup menu. Selecting this option opens the Synthesizer Options window. On Mac, options to select a Soundfont, and set *Pitch Cents Offset*, *Reverb*, and *Volume* are included. Note that the *Pitch Cents Offset* value should be changed only when a given Soundfont is pitched at some other level than A = 440.0 Hz, otherwise output will not be correct according to the values set to your *Reference Tone* and *Tonic* (see *Chapter 3, Tones for more information*). On Windows, only the Patch, Volume and Panning can be selected.

For internal microtonal output on Windows, it is best to route MIDI Out to a third-party synthesizer due to the unacceptable latency of the built-in Windows MIDI synthesizer.
**MIDI Instrument Input**
Since UTE is an editor for defining tunings for MIDI instruments, it is assumed that you are working with the instrument that you are editing, and that you have connected that instrument as the MIDI Input Source in UTE. When you have done this, incoming MIDI notes will sound the pitches you have defined for those keys of the instrument. The instrument keys onscreen can also be animated, and the entries in the Tuning List selected, according to MIDI Input. This option is controlled in the Display window for the current project, and under the Display panel of the Preferences Window for new projects.

![Animate instrument and list with incoming MIDI Notes]

Note that animating the keyboard can result in MIDI response latency. For best MIDI response, uncheck this option.

**Working with Devices**
When working with a tuning device such as TBX2 to upload tunings, manage presets and so on, the device should be connected both to MIDI Input and Output, so that data will not only be sent to the unit, but also received from it (see Chapter 8, Devices for more information). Please be aware that some MIDI interfaces are not usable in this configuration because they will produce feedback loops, and some MIDI interfaces are not usable because they do not support sysex messages properly (see Chapter 8, MIDI Interface Requirements).
2. Instruments

A UTE (.ute) Project begins by selecting an instrument. Click the Instrument icon to open a window showing a list of available instruments, and add an instrument to your project.

**Instruments**

Each available instrument in UTE is a virtual model of a MIDI controller. The virtual model is referred to as a *instrument* (pronounced “U-Instrument”). UTE includes a number of *instruments* modelling MIDI controllers which have been used for microtonal music, such as standard piano keyboards, or microtonal keyboards like the Tonal Plexus TPX and U-Plex. The default set of instruments all actually exist, but models can be made for instruments that do not yet exist. New instrument models are added using instrument definition files.

**Instrument Definition Files (.uinst)**

Any MIDI keyboard instrument can be added to UTE as long as a definition file can be made. This is done using XML. The default .uinst files can be studied by those who wish to add a new instrument to UTE, to see how the XML is structured. A quick reference is included in the Appendix of this documentation. The general user does not need to study the XML, but should become familiar with how UTE interprets the structure of a loaded instrument.

**The Structure of a *instrument***

All *instruments* are understood as collections of keys called *keys*, where each key corresponds to a given MIDI Note on a given MIDI channel. Each *key* can be assigned a specific tuning, which is the main purpose of UTE.

The keys of any keyboard are normally arranged in some periodically repeating geometry. UTE defines this geometry at four levels. The lowest level is the key, called a *key* by UTE. *Keys* are arranged into *groups*, which are combined into *collections*, which are finally organised into a *period*. The *period* is the largest repeating structure of a keyboard, which normally is called an octave. The name “period” is used as a more general term since the
tuning of the keys is not known and can be anything. In this text the period may nevertheless be referred to as an “octave”. To review, the structure of a \textit{uInstrument} is as follows:

\begin{align*}
\text{[ uKeys } \rightarrow \text{ uGroups } \rightarrow \text{ uCollections } \rightarrow \text{ uPeriod }] = \text{ uInstruments}
\end{align*}

\textbf{uVersions & uOptions}

Each \texttt{.uinst} definition contains a list of available \textit{uVersions} of the \textit{uInstruments} which normally vary in the number of octaves, and possibly starting and ending keys within the \textit{uPeriod}. \textit{uOptions} allow different key-colour options to be available.

\textbf{MIDI Notes & Channels}

The MIDI standard allows 16 channels having 128 notes each, so that MIDI instruments are practically limited to having 2048 keys. This limitation is maintained by UTE, because \textit{uInstruments} are MIDI devices by definition. Each \textit{uInstruments} definition determines the way channels and notes are assigned to keys, which is supposed reflect the real structure of the actual MIDI controller, so that the MIDI data output by the actual instrument matches the virtual instrument model.

\textbf{Altering Default uInstruments}

If you find a problem with any of the default instruments, please do not attempt to alter the \texttt{.uinst} file manually. Instead, report the problem using the menu item \textit{Report a Bug}. You may duplicate default files and alter the duplicate as long as you rename both the file itself and the instrument name within the file.

\textbf{Creating a New uInstruments}

You can define your own instruments by writing your own \texttt{.uinst} files. See the \textit{Appendix} for more information.
3. Tones

UTE uses two primary tones to define all tunings: a Reference Tone, and a Tonic. In the simplest case, these two tones are the same, but they need not be the same, because they do have different purposes.

**Reference Tone**

The modern standard reference tone is $\text{A} = 440.0 \text{ Hz}$ (or $440.0 \text{ cps} = \text{cycles per second}$). UTE prompts you to choose a common note name and assign to it a reference frequency in Hz. This determines not only how all tones will be assigned to frequencies in your tunings, but also how notes and intervals are named. Normally a reference frequency is assigned somewhere in the middle octave (the octave from middle C up to but not including the C above it) but that is not a requirement.

![Reference Tone](image)

**Tonic**

A Tonic represents the root frequency (or value of $1/1$) of any scale you work with in UTE. It is defined as an interval above the Reference Tone. This is easily demonstrated by a ratio in the form $R:T$ where $R$ is the Reference Tone and $T$ is Tonic. An entry of $1:1$ means that the Reference Tone and Tonic are the same (simplest case). An entry of $4:5$ would mean that the Reference Tone is 4 and the Tonic is 5, making an interval of $(4:5)$ which can also be expressed $(5/4)$ both of which mean that the Tonic is a Perfect Small Major Third above the Reference Tone. Whenever Tonic is not $1:1$, it will sound either above or below the Reference Tone according to your choice of placement. You can define the Tonic as any valid tuning entry. In the example above, $(4,12)$ means 4 equal tempered halfsteps above the Reference Tone A 440 Hz, which would be C in twelve tone equal temperament. See Chapter 4 for more information on tuning entries. If a Hz entry is used for the Tonic, it is not treated as an absolute frequency but rather as an interval ratio from the Reference Tone. If the frequency you enter is more than an octave ($2/1$) away from the Reference Tone, it will automatically be transposed by successively doubling or halving the value until it is within one octave of the Reference Tone, either above or below according to the option you have selected.
4. Scales

A scale is an arrangement of any number of tones in ascending order. The word “tuning” is sometimes used interchangeably with “scale”, which sometimes makes sense, for example when discussing tuning files, some of which contain scale information, and some of which do not, but in each case the imported data can be treated as a scale. One of the main functions of UTE is the importing of tuning files and assigning tones of scales to instrument keys — a process called “mapping”.

Tunings vs. Scales

The word “tuning” is a general term meaning any set of tones used to make music. A scale is normally a smaller set of organised tones from which a larger set of tones called a tuning may be derived. A tuning does not necessarily include a scale and it doesn’t have to be organised in any conventional way; it can be a collection of tones in any order with no discernible subsets. This distinction between scale and tuning is useful when getting to know different tuning file types.

A tuning and a scale are not considered to be the same thing in UTE.

Supported Tuning Files

UTE supports seven types of tuning files. The file extension must be included with the file name in order for the file to be visible to UTE.

<table>
<thead>
<tr>
<th>Extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.scl</td>
<td>text list of tones as scales in ratio or decimal (cents) form</td>
</tr>
<tr>
<td>.csv</td>
<td>text list of 128, 512, or 2048 MIDI Note + 14-bit Pitch Bend values</td>
</tr>
<tr>
<td>.hz</td>
<td>text list of 128, 512 or 2048 Hz values</td>
</tr>
<tr>
<td>.tun</td>
<td>text list of 128 cents values</td>
</tr>
<tr>
<td>.mtx</td>
<td>text list of frequency values with other special information</td>
</tr>
<tr>
<td>.gly</td>
<td>binary file exported from Absynth (Native Instruments)</td>
</tr>
<tr>
<td>.tonex</td>
<td>text list of scale tones in a variety of formats using xml tags</td>
</tr>
<tr>
<td>.utuning</td>
<td>text file container for any of the tuning file types listed above</td>
</tr>
</tbody>
</table>
The Scala format is by far the most widely used and widely available tuning files. It is a list of any number of tones expressed either as ratios or cents values. For more details on this format and how to use it, see [http://www.huygens-fokker.org/scala/scl_format.html](http://www.huygens-fokker.org/scala/scl_format.html).

This is a file format used by H-Pi Instruments Custom Scale Editor (CSE), Tonal Plexus Editor (TPXE), and H-Pi Lo-Fi Microstudio (HPLF) software. These files consist of comma-separated lists of MIDI bytes as MIDI NOTE, PITCH BEND MSB, PITCH BEND LSB. In a .csv file, there is no information concerning a scale as such. The files may list 128, 512, or 2048 values. A list of 128 values is a tuning for one MIDI channel. A file listing 512 values covers 4 MIDI channels of 128 notes each. The 512 tone files are used by CSE for Tuning Box TBX1 hardware, which supports tunings in four layers, one layer per MIDI channel. 2048 values covers all 16 MIDI channels, used by TPXE for Tonal Plexus keyboards, where each physical octave of the keyboard is mapped to two MIDI channels.

This file format was introduced by H-Pi Instruments H-Pi Lo-Fi Microstudio (HPLF), also supported by CSE and TPXE as an export format. They are text files consisting of a simple list of frequency values in Hz. The numbers of tones listed in the file correspond to the same options listed above for .csv files; that is, 128, 512, or 2048.

These files are also known as VAZ-Anamark, and can exist in a few different varieties, but the majority consist of a list 128 cents values. The cents values are given in reference to the lowest MIDI note 0, and this form of the file contains no scale information as such. More information on this format can be found at [http://www.mark-henning.de](http://www.mark-henning.de).

These files are created by Max Magic Microtuner software. Details on this format can be found at [http://digilandere.libero.it/microtuner/MicrotunerFormat.pdf](http://digilandere.libero.it/microtuner/MicrotunerFormat.pdf).

These files are created by Native Instruments Absynth software, and the format is also exported by Scala (by Manuel Op de Coul), and CSE. Because .gly is a binary file, there are no public details on the file type, and it is impossible to make the file as text or to view the file contents as text, except in a hex editor. The file data itself is an encrypted list of 128 Hz values, one value per MIDI note.
.tonex
This is a filetype supported by other applications such as CSE, microsynth, and others, which uses xml tags to specify different aspects of a tuning. UTE supports simple .tonex files which list tones in scale order using tuning entries.

.utuning
This is an experimental filetype introduced by UTE, which uses xml tags to contain data from any other type of tuning file, in order to be able to manage any type of tuning file in a single format.

**Importing a Scale**
Click the Scale icon to open the Scale Window, and click the Import button to import a tuning file. If the file selected is a valid compatible tuning file, it will appear in the list of imported scales. Imported scales are stored with each project file, and optionally added to a global scale library.

The tones of the scale can be viewed in a popup list by clicking the grey triangle next to the number of tones.
Caveats About Importing Scala .scl Files

The Scala .scl file format allows many kinds of tonal structures to be represented, not only scales. For this reason, some Scala files will be rejected, because they do not represent proper scales. Here are some rules for preparing Scala files as scale files to import into UTE.

1. Tones should be in low to high order with the last tone (period) being the largest value.
2. Files containing tones out of order will likely produce incorrect results.
3. Files containing a period (final) value of less than 1 will likely produce incorrect results.
4. Do not use negative cents values. Files containing negative cents must include a positive period (final) value, or the negative cents cannot be properly converted.

Mapping a Scale

The Scale Window allows you to map a scale after importing it. Double-click a scale, or select the scale and click the Map Scale button to map the scale to the loaded Instrument. Mapping is done according to the Reference Tone and Tonic you have chosen in the Tones Window. Select a key of the instrument on which to place the Tonic (1/1) of the scale. The tones of the scale are then assigned to the remaining keys of the instrument according to the order of MIDI Channels and Notes as defined in the instrument’s definition file. Mapping the Tonic can be done in several ways, each of which is explained below.

Mapping from All Keys

If you prefer working from an overview of the instrument, you can select a key from a list of all the available keys on the instrument to assign to the Tonic frequency. The key number is listed, followed by the assigned MIDI channel and MIDI Note number for the key.
Mapping from Octave & Key

If you prefer to think in terms of the structure of the instrument rather than counting keys or working from selected keys, the tonic can be mapped by selecting an instrument Period and then a key of the instrument within that period.

The name for the Period is taken from the \texttt{ulnstrument} definition file, so the label \textit{“Octave”} used in this example means that the \texttt{uPeriod} for this instrument has been given the name \textit{“Octave”} in its definition file. This will be the most common case, but the period can be called anything, and whatever name is defined will appear here.

Tuning Equal Divisions

Any octave-based or non-octave equal division can be quickly tuned from the Scales Window by selecting \textit{Equal Division} from the \texttt{Import} button menu. This option is also available under Selection > Tune Equal Division in the main menubar. The Equal Division window appears to let you define the scale with any whole number of divisions. The Period may be an octave (2/1) or any other valid tuning entry (see Chapter 5: Tuning Entries).

If you check the \texttt{Import} option, the scale will be added to the \texttt{Scale List} in the Scales window. If left unchecked, the scale will not be not added to the \texttt{Scale List}. After you click \texttt{OK}, the Scales window will open with the Map interface, so that you can map the scale to the instrument (see Mapping options above).
Exporting Scales
The tunings you create in UTE can be exported in a number of standard tuning file types, using the menu item File > Export Tuning… Currently the following seven formats are supported. Except for the .scl selected-notes-only options, each exported file contains 128 values, so that each MIDI channel is stored in a single file. Additional formats and file length options can be added in future versions of UTE. If something is missing here that you need, you can always use the menu option Request a Feature.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.scl</td>
<td>Scala format, selected tones as cents values, sorted or non-sorted</td>
</tr>
<tr>
<td>.scl + .kbm</td>
<td>Scala format, 128 tones as cents values, plus a keyboard map file</td>
</tr>
<tr>
<td>.csv</td>
<td>128 MIDI Note + 14-bit Pitch Bend values</td>
</tr>
<tr>
<td>.hz</td>
<td>128 Hz values</td>
</tr>
<tr>
<td>.tun</td>
<td>128 cents values</td>
</tr>
<tr>
<td>.syx</td>
<td>MIDI Tuning Standard Bulk Tuning Dump binary file, 128 values</td>
</tr>
<tr>
<td>.mtx</td>
<td>Max Magic Microtuner file, 128 Hz values</td>
</tr>
<tr>
<td>.Temperament_ Hauptwerk_xml</td>
<td>Temperament xml file for Hauptwerk, 128 Hz values (132 nominal)</td>
</tr>
<tr>
<td>.txt</td>
<td>Script for Kontakt, 128 14-bit Pitch Bend values</td>
</tr>
</tbody>
</table>

Scale Library
This functionality will be added to a future release of UTE. The idea is to create a searchable database for all imported scales, regardless of the format of the original scale file, with scales added to the global library by default when you import them. This will save you from having to import the same scale more than once. You will be able to control the behaviour by unchecking Add to Library.
5. Tuning List

The Tuning List shows each key with all of its associated information and allows you to navigate a *uninstrument* somewhat in the manner of a spreadsheet. From left to right, there are roughly three “stages” of columns in the List. On the left side are *Input Columns*, in the middle are *Tuning Columns*, and on the right side are *Output Columns*.

Input, Tuning, & Output Columns

Input Columns include the ordinal number and period number of the key, and its MIDI input channel and note as defined by the *uninstrument*. Tuning Columns include the tuning entry, the interval size in Units (normally cents, see below), the period number, and pitch and interval names. The Output Columns list the resulting MIDI note and its offset in cents from standard 12-tone equal temperament.

Units ¢

The values appearing in the Units column depend on which Units you have selected in the Preferences. Units are defined by two values: *Ruler* and *Divisions*, and each Unit has both a *Name* and a *Token*. A Ruler can be of any size, and can have any number of divisions. The Ruler is defined from a Tuning Entry (see below), normally a ratio. The Divisions value is usually a whole number, but it can also be a decimal (whole number with a fractional part). The default Unit is the *Cent* (1/1200 of an octave), where the Ruler = 2/1, the Divisions = 1200, and the Token is ¢. In the Preferences window you can select your preferred units such as the *Savart* or *Jot*, and you can define any number of your own units. Note that because the Ruler and Divisions values define a Unit themselves, it follows logically that Units entries cannot be used when defining the Ruler with a Tuning Entry. The heading of the Units column displays the Units Token to the right of the word “Units”.

About Period Numbers

The numbers in the Period column count the number of iterations of the scale (the period count). If you have used other H-Pi software, you may notice that this numbering is not based on C octaves, which is a different concept than has been implemented in other H-Pi software. The change has been made to avoid octave transposition problems, and for the sake of simplicity and more natural support for non-octave tunings.
Tuning Entries

A tuning entry in UTE is an expression which represents a tone in different ways using math. Examples of valid tuning entries are shown below. Parentheses must be used when there are 2 terms and some operator, as shown in the last three examples.

- 1.2365256341287 ... decimal values
- 2:3 ... interval ratios a:b, where a < b
- 3/2 ... tone ratios a/b, where a > b
- 2,12 ... a degree of an equal division of an octave, first degree = 1 (or 0 by Preference)
- 3*7 ... exponents, such as 3 to the 7th power
- +35.2 ... an interval up, in cents (or other units, by Preference)
- -4.3 ... an interval down, in cents (or other units, by Preference)
- f=261.6255653 ... Hz values, within an available range of 7.9430 to 12911.4169 Hz
- (1.232)*(2:3) ... is 1.232 transposed up by 2:3
- (4/3)/(1.112) ... is 4/3 transposed down by 1.112
- (5/4)+3.1 ... is 5/4 transposed up by 3.1 cents

Any number of operations can be strung together in an entry using parentheses. Units (cents) transposition however must always be at the end in an entry using parentheses. An example semi-complex tuning entry using parentheses is given below.

\[(4,13)/(11/8)*(9/8)-23.1\]

This is 4th degree of 13ET transposed down by 11/8, then transposed up by 9/8, lastly transposed down by 23.1 cents.

Comments

You may want to add notes for individual keys of a tuning. In the project window or in the Preferences Window under the Display toolbar item, check the Comments option to enable this column in the Tuning List. The Note that comments are not currently exported with any of the tuning export formats.

Column Width Adjustments

To resize the widths of various columns, simply drag the dividers in the list header. Bear in mind that the Tuning Entry column is given the most space by default, and the list adjusts its width proportionally according to the size of the Detail View. Therefore, it is best to adjust the size of the views before adjusting the column widths.
6. Editing Tunings by Selection

When you have selected keys of an Instrument, rows in the Tuning List are highlighted and the Selection menu becomes available. With the items in this menu you can edit the tuning entries assigned to the selected keys at once.

**Transpose**

Tuning entries can be transposed by other tuning entries. To do this for a range of selected keys, with keys selected, open the Transpose Window by choosing the menu item Selection > Transpose or use the keyboard shortcut Command + T.

Transposed entries appear as a chain of values. For example, the entry \(9/8\) transposed up by \(3/2\) will appear as \((9/8)^*(3/2)\). The same entry transposed down by the same interval will appear as \((9/8)/(3/2)\). Note that transpositions by the Period value (2/1 for normal octave-based tunings) can be done using the Transpose window, but it makes much more sense to transpose by the Period using the Period Up and Period Down functions (see below). Note also that any tuning entry which defines an absolute frequency, for example the entry \(f=440.0\) will not allow transposition, so any selected keys having frequency entries will not be altered by a Transpose operation.

**Repeat**

A reasonable way to design a uInstrument tuning is to make one structure and then repeat it across the instrument. This function lets you do that, in either or both directions, tuning all keys without gaps, or tuning keys only in a selected pattern. Period values of the newly tuned keys can be updated or left unchanged as needed.

**Reverse Order**

Added at customer request, with this function you can quickly create “left handed” versions of any tuning. The order of the tuning entries assigned to the instrument keys is reversed.

**Tune Scale**

This is a convenience for opening the Scale Window with the option to tune selected keys only preselected in the Map Scale panel.
Tune Equal Division
Any octave-based or non-octave equal division can be quickly tuned using the menu option Selection > Tune Equal Division, or by typing the keyboard shortcut Command + E. Because this option involves the Scales Window, it is explained above in Chapter 4: Scales, Tuning Equal Divisions.

Tune Equal Steps
Any succession of equally sized steps can be quickly tuned using the menu option Selection > Tune Equal Steps, or by typing the keyboard shortcut Shift + Command + E. This can be used to create tunings that have uniformity but do not necessary have a larger repeating period. Equal steps can be created in both ascending and descending forms.

Convert to Ratio
Any number can be approximated by a ratio of integers within some degree of error. To do such calculations, values are needed to specify the largest prime number which can be used in a ratio, called the Prime Limit, and how accurate an approximation is desired, or the Largest Error. By default, UTE uses a Prime Limit of 499 and a Largest Error of 0.5 cents, but you can select other values in the Preferences window. When a ratio within the Prime Limit cannot be found within the desired range of error, a cents adjustment is appended to the ratio.

The prime numbers listed in the Prime Limit popup list are those under 10,000. For the vast majority of users this range is more than adequate, but it is of course possible to add higher prime numbers to the list. If you would like to work with larger numbers for ratio conversion, please send a message to discuss your needs.

Convert to Decimal
When this option is selected, tuning entries will be converted to a decimal values between 1.0 and 2.0. This may be useful if certain entries become unnecessarily long.
**Convert to Units**

Use this option to change entries into units format, according to the Units you have selected in the Preferences. The selected entries will then appear in the format +Units.

**Reduce ET Notation**

Use this option to combine multiple Equal Temperament expressions into a single expression. The appearance of the expressions depends on your selection of 1-based entries (the default setting, where scale degrees are counted beginning with 1) or 0-based entries (where 0 implies an interval of no distance, as in set theory). In 0-based notation, \((2,12)^*(11,12)\) reduces to \((1,12)\). The same in 1-based notation would appear as \((3,12)^*(12,12)\) reducing to \((2,12)\). Expressions containing different-sized steps are calculated and reduced using common factors, for example in 0-based notation \((3,24)/(1,36)\) reduces to \((7,72)\). In 1-based notation this would appear as \((4,24)/(2,36)\) reducing to \((8,72)\). Keep in mind that combining expressions having no common factors will result in expressions having large numbers.

**Reduce Ratios**

Use this to combine multiple ratios into a single ratio reduced by common factors. For example the entry \((9/8)*(9/8)\) becomes \((81/64)\). The entry \((3/2)*(17/16)\) reduces to \((51/32)\). Keep in mind that combining ratios may result in numbers which are quite large. On the other hand, UTE makes an effort to reduce by common divisors, so that ratios are expresses in simplest terms.

**Reduce Entry**

This option combines Reduce ET Notation with Reduce Ratios, resulting in an entry that is as compact as possible without converting the form of the original expression.

**Set Period**

The menu item Selection > Set Period, keyboard shortcut Command + P, allows setting the tone height of a selected entries according to the Period value which has been assigned according to a given tuning. The Period of a tuning is defined in a Scale (see Chapter 4: Scales) and is normally an octave \((2/1)\) but may be any value (non-octave tuning). If no period has been defined, a default value of \(2/1\) is assumed.

**Period Up / Down**

To quickly transpose a pitch up or down in pitch height according to the given Period value, use the menu item Selection > Period Up, and Selection > Period Down. Keyboard shortcuts use the up and down arrow keys Command + ↑ and Command + ↓.
7. Display Options

The Display toolbar icon opens a window with options for items to be shown in the Tuning List, options for displaying data on instrument keys, and for animating instrument keys with MIDI input.

Tuning List

Eleven items are shown which may be displayed in the Tuning List. Presently two of these items remain unavailable as placeholders for possible future options.

Text on keys

Text from the Tuning List can be displayed on instrument keys by selecting Text on keys and choosing a desired option. Presently the last three options will produce no text.

Animate Instrument

When playing keys on a MIDI keyboard connected as the input source to UTE, the keys of the loaded instrument can be animated, and rows of the Tuning List can be highlighted, for each key played. This is useful for seeing your tunings in action as you try them out on the physical instrument. This animation is made optional because in some cases animating the keys may slow down MIDI response.
8. Devices — TBX2

UTE has been developed for use with the H-Pi Instruments Tuning Box TBX2, but is also designed to support other tuning devices, such as Tonal Plexus keyboards. This chapter outlines features for TBX2. For detailed information on any of the features mentioned here, see the device documentation.

**MIDI Interface Requirements**

TBX2 is programmed over MIDI, using the MIDI DIN connectors. This requires a MIDI Interface, which normally has a USB connection on one end and MIDI DIN jacks or plugs on the other. It is important to be aware that not all MIDI interfaces are made the same. TBX2 programming is done using System Exclusive MIDI messages (known as “sysex”), and some MIDI interfaces do not handle these messages properly. Below is a list of interfaces which we know do NOT work properly with sysex. Note that these interfaces may work fine with some very short sysex messages, but they do not have large enough buffers for large sysex messages. This is a hardware problem and cannot be solved with driver updates, so these interfaces simply should not be used.

<table>
<thead>
<tr>
<th>DO NOT USE THESE MIDI INTERFACES</th>
</tr>
</thead>
</table>

* Roland UM-ONE  
* E-MU 1x1 Tab  
* All cheapo / no-name interfaces from China

Because this is an important requirement for our hardware devices, we will be offering our own MIDI interface in the future. Until our MIDI interface is available, we recommend the following third-party product which is known to work correctly for all sysex:

<table>
<thead>
<tr>
<th>WE RECOMMEND THIS MIDI INTERFACE WHICH TRANSMITS SYSEX CORRECTLY</th>
</tr>
</thead>
</table>

* iConnectivity mio

For problems sending MIDI, see the troubleshooting section at the end of this chapter.
Editing Data Stored on Your TBX2

If you want to edit data already programmed in your TBX2, UTE needs to be able to ask TBX2 for information (referred to as a “query”), and TBX2 needs to be able to send data to UTE (called a “response”). To do this, you need to connect both MIDI IN and MIDI OUT from your TBX2 to your MIDI Interface, and select that interface using the popup buttons at the top of the window from which TBX2 is being programmed. This will be referred to below as having “TBX2 I/O Connected”.

Programming a Tuning

To program TBX2 with your tuning, click Devices toolbar icon and select Send Tuning to TBX2. Be sure to have TBX2 I/O Connected.

TBX2 assigns one tuning table per MIDI channel, with 128 notes per tuning table, so if your tuning contains more than 128 notes, it will be broken into multiple tables in TBX2 memory. For example, Tonal Plexus tunings contain 2048 notes (16 x 128 = 2048), so the tuning is spread across 16 tuning tables in TBX2 memory. UTE automatically figures out how many TBX2 tuning tables are needed to store your tuning, and provides a popup menu for selecting the table numbers to program, with three arrow controls for navigating the table numbers more easily in increments of 10, 100, or 1000.

When the tuning spans multiple TBX2 tuning tables, the names of the tables are numbered. In the example shown above, the table names will be Tonal Plexus 1, Tonal Plexus 2, Tonal Plexus 3, and so on, through Tonal Plexus 16. The checkbox Names Only also allows you to update only the table names without sending the tuning itself.

Batch Programming Tunings

To upload many tunings TBX2 at once in sequence, prepare a folder containing the tuning files, click the Devices toolbar icon, and select Send Batch Tunings to TBX2. Click the Choose Folder button and navigate to the folder containing the files. The tunings may be in any format which can be imported by UTE (see Chapter 4). Once imported, you may change
the order of the files by dragging them up or down in the list. Select the table numbers to be assigned to the files using the popup menu at the upper right.

TBX2 limits tuning names to 16 characters. Because TBX2 stores thousands of scales, it can be helpful to include the number of tones in the scale as a prefix to the tuning name in order to aid navigation. Options for converting the tuning names with prefixes and a shortened number of characters are provided as a convenience. If names are not shortened, any name longer than 16 characters will simply be cut off at the 16th character when sent to TBX2. If you have chosen to number the files, you can use the button marked 123-abc Sort to order the files by both number and name.

Check the option Keep converted files ... to archive the unique order of the imported files and save time should you want to upload the files again later. The converted files have the extension .tbxtuning and are read faster by the batch window than the other file types. The names of the converted files are numbered in the order you have placed them in the listbox. Be sure to have TBX2 I/O Connected and select the correct MIDI Destination before clicking Send to Device. Uploading one file takes less than one second, but if you upload thousands of files, be aware that this can take some time. For example, to upload the entire Scala archive of about 4700 scales takes about 90 minutes.

Be aware that tuning files containing more than 128 notes will be incomplete when using batch-upload, because TBX2 stores tuning tables in the MIDI standard of 128 notes. Although it is possible to import these larger scales and map them manually, this cannot be done using batch-import. Keep in mind the limitations concerning Scala files listed in Chapter 4.
Programming a Preset

TBX2 has 40 presets (numbered 0-39) which can be programmed at the device itself, but sometimes it will be easier to set up your presets using UTE, for example particularly when you are working with multi-table presets such as those needed for Tonal Plexus keyboards. Click the Devices toolbar icon and select Program a Preset to open the window.

To edit a preset already programmed on your TBX2, be sure to have TBX2 I/O Connected, then after selecting a preset number, click the Query TBX2 button at the bottom left corner of the window, and the data for the selected preset will appear in the window for editing.

Clicking the Select Tuning Tables button opens a window allowing you to easily set 16 tables in a row to the 16 MIDI input channels, since this will be a common need for those working with Tonal Plexus keyboards (and possibly other microtonal controllers).
Setting Global Parameters

TBX2 parameters can be edited directly at the device without using UTE, but sometimes using UTE will make more sense. Click the Devices toolbar icon and select Set Global Parameters.

![Setting Global Parameters](image)

Setting USR Data

TBX2 provides 10 USR presets (numbered 0-9). USR Presets allow sending different kinds of sysex messages to retune keyboards according to manufacturers’ specifications.

![Setting USR Data](image)
Unlike other features of TBX2, USR Data functions are rather involved, so they cannot be programmed at the device, and must be programmed using UTE. Each USR Preset Data consists of two parts: header bytes, and options. The header bytes may vary in length anywhere from 0 to 32 bytes. The options are as follows:

<table>
<thead>
<tr>
<th>Options</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Include program number?</td>
<td>Yes / No</td>
</tr>
<tr>
<td>Include tuning table name?</td>
<td>Yes / No</td>
</tr>
<tr>
<td>Include checksum byte?</td>
<td>Yes / No</td>
</tr>
<tr>
<td>Notes formula (4 options)</td>
<td></td>
</tr>
<tr>
<td>1 byte ± cents offset per note</td>
<td>1 byte ± cents</td>
</tr>
<tr>
<td>2 bytes pitch bend MSB, LSB</td>
<td>2 bytes pitch bend</td>
</tr>
<tr>
<td>3 bytes MTS data per note</td>
<td>3 bytes MTS data</td>
</tr>
<tr>
<td>3 bytes pitch bend data per note</td>
<td>3 bytes pitch bend</td>
</tr>
<tr>
<td>do not send any note data</td>
<td>do not send any</td>
</tr>
<tr>
<td>Tuning table start key</td>
<td>(0..127) [default =</td>
</tr>
<tr>
<td>Tuning table end key</td>
<td>Note 60, middle C]</td>
</tr>
</tbody>
</table>

When the USR Preset is selected at TBX2, a sysex message is sent having the following form:

<table>
<thead>
<tr>
<th>Description</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysex start</td>
<td>1</td>
</tr>
<tr>
<td>[ header bytes ]</td>
<td>none or 1 to 32</td>
</tr>
<tr>
<td>[ program number ]</td>
<td>none or 1</td>
</tr>
<tr>
<td>[ tuning name ]</td>
<td>16</td>
</tr>
<tr>
<td>[ notes data ]</td>
<td>none or 1 to 128</td>
</tr>
<tr>
<td>[ checksum ]</td>
<td>none or 1</td>
</tr>
<tr>
<td>sysex end</td>
<td>1</td>
</tr>
</tbody>
</table>

Refer to the documentation of your synthesizer to determine the correct header bytes and options to assign to a USR Preset. The TBX2 USR Data window has a popup menu for selecting any of the MIDI Tuning Standard (MTS) formats to simplify the process of inputting all the parameters if your keyboard requires one of the MTS message formats.
**USR Data Example 1**

The Roland FP-30 keyboard supports the MTS Scale / Octave Tuning 1-Byte Format. USR Data for this message can be prepared simply by selecting that option from the popup menu. The USR Data options will be automatically populated as follows.

<table>
<thead>
<tr>
<th>USR Data Options for setting up Roland FP-30 MTS Sysex Retuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Bytes</td>
</tr>
<tr>
<td>7E, 7F, 08, 08, 03, 7F</td>
</tr>
<tr>
<td>Include program number?</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Include tuning table name?</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Include checksum byte?</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Notes formula (4 options)</td>
</tr>
<tr>
<td>1 byte ± cents offset per note (MTS 1-byte cents form)</td>
</tr>
<tr>
<td>Tuning table start key</td>
</tr>
<tr>
<td>Note 60, middle C</td>
</tr>
<tr>
<td>Tuning table end key</td>
</tr>
<tr>
<td>Note 71, B above middle C</td>
</tr>
</tbody>
</table>

**USR Data Example 2**

The Roland FP-30 can also be retuned across its entire range to any scale (not just 12 tones) by TBX2 in POLY mode, but only when the keyboard is first put into GM1 or GM2 mode. You can push some buttons on the Roland to set that up, but there is also a sysex message which forces the keyboard into the needed mode. So you can use a TBX2 USR preset to send that message, and then press a button on TBX2 to put the synth into the correct mode, after which you can press other preset buttons programmed to POLY. In the user manual we find that the sysex message required by the Roland for “GM1 System On” is: F0, 7E, 7F, 09, 01, F7. We prepare USR Data for this message as follows.

<table>
<thead>
<tr>
<th>USR Data Options for setting up Roland FP-30 GM System ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Bytes</td>
</tr>
<tr>
<td>7E, 7F, 09, 01</td>
</tr>
<tr>
<td>Include program number?</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Include tuning table name?</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Include checksum byte?</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Notes formula (4 options)</td>
</tr>
<tr>
<td>Do Not Send</td>
</tr>
<tr>
<td>Tuning table start key</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>Tuning table end key</td>
</tr>
<tr>
<td>-</td>
</tr>
</tbody>
</table>

Once this USR Data is assigned to a USR number (0-9) then you can program the USR preset at TBX2 by selecting that USR number as the mode for the preset.
Backing up & Restoring Data (.udevicebackup)

Because TBX2 is a device having a large memory for presets, USR Data, Global Parameters, and Tunings, it is useful to make backup copies of the contents of the device for example, if it is used for a series of concerts where many presets are involved. Before updating TBX2 firmware, it is also a good idea to backup data, as under some rare conditions memory corruption may occur during a firmware update. A full backup takes about half an hour to complete, and a full restore takes about an hour. Normally you will not need to do a full backup / restore, as it is unlikely that you will be using all 8192 tunings very often, so UTE gives you the option to select a range of tunings to back up. A normal partial backup / restore process can be done in a few minutes.

Updating Firmware (.ubootsyx)

TBX2 firmware can be updated at any time, to fix reported problems or add new features. When you have TBX2 I/O Connected to UTE, the software can query the firmware version installed on your TBX2 and alert you to any available updates. Firmware updates are issued as files having a .ubootsyx extension, which is transferred to the device over MIDI. Click the Devices toolbar icon and select Update Firmware to select a firmware file and upload it to TBX2. The process normally takes about two minutes to complete. Be sure all of the following takes place.

- Always connect TBX2 to external power when updating firmware!
- Use TBX2 Firmware Update Startup:
TBX2 Firmware Update Startup

Press the rightmost white button and hold it down while you turn on the unit with the blue power switch on the left. This tells the unit “we are ready to update the firmware”. The display will read as shown below.

- Make sure TBX2 is properly I/O connected (see start of this Chapter).
- While firmware is sent, UTE shows a progress bar, and TBX2 displays a percentage complete. If not, see Troubleshooting below.
- At the end, UTE shows an alert, and TBX2 shows a brief message, then restarts itself.

**Troubleshooting**

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBX2 seems “frozen” when it is I/O connected and I try to send MIDI.</td>
<td>Your system is producing a MIDI feedback loop. This can happen if you have some other software running which accesses MIDI devices and routes MIDI input and output. Quit the other application(s) or disconnect the cable from TBX2 MIDI OUT and try again.</td>
</tr>
<tr>
<td>Trying to send the firmware, TBX2 just shows “TBX2 B/L v.1.4 Please run UTE!”.</td>
<td>MIDI is not properly connected, or you have selected the wrong MIDI destination in UTE.</td>
</tr>
<tr>
<td>Sending the firmware, TBX2 does not display an increasing percentage, stays at 0%, or reaches some other number and stops there.</td>
<td>This means your MIDI Interface does not transmit sysex correctly. You must use a different MIDI Interface.</td>
</tr>
</tbody>
</table>
APPENDIX: uInstrument (.uinst) Short Guide

Many .uinst files are included with UTE, which you can study in order to learn how to make your own definition files. This short guide is intended to help make that process a bit easier.

**Names & Descriptions**

All objects have **Name** and **Description** tags. The **Name** is mandatory and must be unique within a group of objects. Apart from the name of the instrument itself, object names are only used internally within the XML to refer to the objects, and are not shown to users. **Descriptions** are optional.

**Shapes**

The following shape primitives are provided for each level of the structure: **Polygon**, **Oval**, **Circle**, **Rectangle**, **Square**, and **Figure**. The last of these is used to define complex forms such as the natural keys of a piano keyboard. The properties which must be defined for each shape are as follows:

<table>
<thead>
<tr>
<th>Polygon</th>
<th>Oval</th>
<th>Circle</th>
<th>Rectangle</th>
<th>Square</th>
<th>Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotation, Diameter, &amp; Sides</td>
<td>Height &amp; Width</td>
<td>Diameter</td>
<td>Height &amp; Width</td>
<td>Diameter</td>
<td>Points</td>
</tr>
</tbody>
</table>

The points of a **Figure** define the ends of straight lines which are connected in series. The resulting shape is closed, where the last point defines the end of a line which connects to the first point. Shapes are assigned to objects using `<myShape>` tags.

**Colors**

All shapes have two colour properties: **FillColor** and **BorderColor**. A colour used in many places should be defined at the start of a file as a **uColor** which can then be referred to by name using brackets within the **FillColor** and / or **BorderColor** tags of a shape.

**Positions: Left & Top**

The horizontal and vertical positions of objects are specified using **Left** and **Top** values respectively, where a **Left** value of 0 is the start of left and positive values move right, and a **Top** value of 0 is the top and increasing values move towards the bottom. All values must be integers, so for complex geometries a scale must be chosen which allows the objects to be in approximately correct relation to each other.
Structure

uKeys are arranged into uGroups, combined into uCollections, finally organised into a uPeriod, the largest repeating structure of a keyboard. uVersions refer to uPeriods to define different versions of an instrument.

Each level of the structure refers to one level lower, using a tag in the form <u“Object”ByName>, where “Object” is “Key”, “Group”, “Collection”, or “Period”.

Display Names

Object Names are used to refer to the objects within the .uinst XML. Display names can be assigned to uKeys, uGroups, uCollections and uPeriods using the tag <DisplayName>, to be shown to users of the instrument in UTE. If no display name is defined, for example for a uGroup, then the uGroup will simply be displayed to the user in UTE as “Group”.

MIDI Notes & Channels

Notes and channels are assigned to uKeys in series, in the order the keys are defined in groups, collections, and finally in the period structure. Notes and channels start counting from zero, and after each 128 notes, the channel is incremented. To force a channel to increment at the onset of a structure, the tag <NewMIDIChannel> is used. The tags <StartAtMIDIChannel> and <StartAtMIDINote> can be used to begin assigning channels or notes at something other than zero. For example, these tags can be used within the <uPeriodByName> tags of a <uVersion> so that a given period starts with a desired channel and note.

Versions

Instrument versions are defined within <uVersion> tags, normally stating the number of periods, tuning octaves and MIDI channels for a given version of an instrument. For example, a piano keyboard may have 61, 76, 88, or 128 keys, such that the starting MIDI octave and starting MIDI note values must vary. The tags <StartAtMIDIChannel>, <StartAtMIDINote>, and <DefaultTuningStartAtOctave> are useful in this regard. A period which is repeated some number of times is defined by <Repeat> tags.
Options
Options common to all versions of a uInstrument are placed in <CommonOptions> tags under <uVersions> before any uVersions are defined. An option can replace colours of objects using the form <AddTag><ReplaceColor>[oldColor],[newColor]</ReplaceColor></AddTag>.

Navigation
The <Navigation> tag contents defines how arrow keys behave when pressed by the user in the Detail View. The values assigned to Right, Left, Up, and Down refer to numbers of keys traversed when the user presses the respective arrow key direction. The numbers can be positive or negative.

Default Tuning
The default tuning of an instrument is defined within <DefaultTuning> tags in <tonex> tags. If no default tuning is included, the tuning is assumed to be the modern default 12-tones-per-octave equal temperament.
Credits

All versions of UTE are designed and programmed by Aaron Andrew Hunt, using Xojo and MBS Plugins on a Mac.

Special thanks to everyone who pre-ordered TBX2, the primary tuning device with which this software is intended to operate. Without your support UTE would not have reached its initial release.

This documentation is written by Aaron Andrew Hunt, using Apple Pages.

Thank you for supporting H-Pi Instruments and UTE.

©2019 H-Pi Instruments · FOR THE FUTURE OF MUSIC