Capella Scripts

This Guide contains translations of pages assembled by the German users of capella. Our thanks go to the following script authors:

<table>
<thead>
<tr>
<th>Paul Villiger</th>
<th>Stefan Thierfeldt</th>
<th>Rado Kvon Saiis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hartmut Ring</td>
<td>Hartmut Lemmel</td>
<td>Bernd Jungmann</td>
</tr>
<tr>
<td>Andreas Herzog</td>
<td>Lutz Haase</td>
<td>Urs Ganz</td>
</tr>
<tr>
<td>Peter Becker</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note.

1. The dialogue box bitmaps in this guide are taken from the German guide and in German. Most of the scripts, however, are translated.
2. Script names are put in <> brackets to make them easier to locate.
An introduction to the capella scripts

**STAR SCRIPTS – THE ONES WE WERE DELIGHTED TO SEE!**

<Triplet fix> by Peter Becker

<Triplet bracket fix>

<Beam over marked rests> by Bernd Youngman

Extending beaming over marked rests

<Beam over marked rests> by Urs Ganz

<Layoutfix> by Peter Becker

**NOTATION SECTION**

Covering:

Note formatting

Score formatting

Graphics <Accidental adjuster> by Peter Becker

Automatic overlapping accidentals adjustment

<Voice split2> by Hartmut Ring

Close score to open score

<Format everything> by Paul Villiger

<Format – roughout layout> by Peter Becker

<Bar_numbering> by Hartmut Ring

<Octave change> by Andreas Hertzog

<Transpose an instrument> by Paul Villiger

<Rhythm synch> by Paul Villiger

<Enharmonicus> by Andreas Hertzog

<Voice split> by Andreas Herzog

<Swing notes> by Paul Villiger

<Voice cocktail> by Andreas Herzog

-Chord description expansion> by Paul Villiger

-Chord expansion> by Paul Villiger

Splitting up chords

<Flag lyric notes> by Hartmut Ring

<Reiteration bars> by Hartmut Lemmel

<Tremolo bars> by Paul Villiger

<Style editor> by Paul Villiger

<Short to open score> by Andreas Herzog

-The Stylist> by Andreas Herzog

-LetsSing> Andreas Herzog

-Score append> by Bernd Jungman

-Score synthesis> by Hans H. Lampe

CapCML

-Voice import> Paul Villiger

-StaveEditor> by Paul Villiger

-Systems lock> by Paul Villiger

-Format parallel staves > by Paul Villiger

-BagPiper> by Andreas Herzog

-LittleDrummerBoy> by Andreas Herzog

-Rhythm notation for guitar> by Andreas Herzog

-Vocal range> by Peter Becker

-CueCopy> by Hartmut Lemmel

Copying cue notes

<Dots in multivoices> by Paul Villiger Removing superfluous dots in multivoice staves

This script removes dotting in the lower voice when dotted notes of equal value are of the same pitch.

<Rest centerer> by Paul Villiger

<Rest mover> by Paul Villiger

<Rest manager> by Andreas Herzog

<Rest paddler> by Paul Villiger

Fill with rests

<Rest summarizer> by Peter Becker
<table>
<thead>
<tr>
<th>Section</th>
<th>Tool Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Condense individual rests to multiple bar rests</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>&lt;Beam over rests throughout the score&gt; by Peter Becker</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>&lt;Beam over grace notes&gt; by Urs Ganz</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>Grace notes in beamed groups</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>&lt;Stem direction&gt; by Peter Becker</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>Stem direction</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>&lt;Stem length set globally&gt; by Paul Villiger</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Stem length adjustment</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>&lt;Extensions&gt; by Hartmut Lemmel</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>&lt;Beam division&gt; by Paul Villiger</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>&lt;Symbol to note cloner&gt; by Paul Villiger</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Linking symbols to notes</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>&lt;Articulation signs group&gt; by Paul Villiger</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>&lt;Group graphics&gt; by Andreas Herzog</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>&lt;Brackets for guitar diagrams&gt;</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>&lt;Graphic notes&gt; by Paul Villiger</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>&lt;Slurs dash them all&gt; by Hartmut Ring</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>&lt;Ties to slurs to ties&gt; by Paul Villiger</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>&lt;Slurs stems to heads&gt; by Paul Villiger</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>Adapt lyric slurs in multi-voiced songs</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>&lt;Accent spacing&gt; by Hartmut Lemmel</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>&lt;Slur fix&gt; by Peter Becker</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>&lt;PerformanceMark centerer&gt; by Andreas Herzog</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>&lt;PerformanceMark standardization&gt; by Andreas Herzog</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>&lt;Graphic realignment&gt; by Paul Villiger</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>&lt;Coloured Notes&gt; by Lutz Haase</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>&lt;Graphic multiple bars rests&gt; by Paul Villiger</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>&lt;Graphic multiple bars rests&gt; by Paul Villiger</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>&lt;Pedal marks&gt; by Paul Villiger</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>Pedals markings (pedal brackets)</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>&lt;Repeat box alignment&gt; by Paul Villiger</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>Aligning repeat boxes</td>
<td>77</td>
</tr>
<tr>
<td>TEXT SECTION – COVERING LYRICS, FONTS, CHORD DESCRIPTIONS, FINGERING</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;Page object master&gt; by Andreas Herzog</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>&lt;Page object resize&gt; by Paul Villiger</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>&lt;Font Replacer&gt; by Hartmut Lemmel</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>&lt;Lyrics to text converter&gt; by Andreas Herzog</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>&lt;Letter and numbers&gt; by Paul Villiger</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>&lt;Text_to_behind_notes&gt; by Hartmut Ring</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>&lt;Font – capella replacer&gt; by Paul Villiger</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>To see the new fonts please restart capella.</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>&lt;Figured bass&gt; by Andreas Herzog</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>&lt;Chord reformatter&gt; by Paul Villiger</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>&lt;Find and replace&gt; by Andreas Herzog</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>&lt;Fingering&gt; by Andreas Herzog</td>
<td>14</td>
</tr>
<tr>
<td>FILE MANAGEMENT AND FILE FORMATS SECTION</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;CapellaToMusicXML&gt; by Bernd Jungmann</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>&lt;XML code displayer&gt; by Paul Villiger</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>&lt;CapellaToCapX converter&gt; by Paul Villiger</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>&lt;HTML-export&gt; by Hartmut Ring</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>&lt;Score to bitmap exporter&gt; by Paul Villiger</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>&lt;Score cataloguer&gt; by Paul Villiger</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>&lt;Print all files in folder&gt; by Paul Villiger</td>
<td>9</td>
</tr>
<tr>
<td>TABLATURE SECTION</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;DiatonicHarmonica_A&gt; by Peter Becker</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>&lt;SwissOrgan tablature&gt; by Radolf von Salis</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>&lt;ClubOrgan&gt; by Peter Becker</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>&lt;Ziach A&gt; by Peter Becker</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>The button accordion</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>&lt;Guitar tab&gt; by Andreas Herzog</td>
<td>12</td>
</tr>
</tbody>
</table>
This script produces harmonica tabulature

The fingering cannot be transposed any more - a backup copy of the original score is therefore saved.

MISCELLANEOUS SECTION

Miscellaneous and fonts

Classical Notation Fonts
Ornamental fonts
Scripts included with capella
Gallery Extras
An introduction to the capella scripts

These scripts have been created by capella customers to meet their specific requirements. Whilst we believe that they are all effective and safe to use, Software Partners can take no responsibility for them.

Tips:
♦ Some of these scripts are now incorporated into capella so it is worth checking if they are there before installing them.
♦ If you look through the whole of the manual you may well find useful scripts that help you in ways you had not thought about.
♦ The dialogue box bitmaps in this guide are taken from the German guide and are in German. Most of the scripts, however, are translated and with English dialogues.
♦ Script names are put in <> brackets to make them easier to locate.

With the advent of capella2004 and its new programming interface it became possible for everyone to adapt capella to their particular needs. The majority of users do not avail themselves of this opportunity but a handful of programmers have written some most useful scripts. The authors of capella have been very pleased to welcome new programmers into the ‘capella family’.

Some scripts correct program errors of the software, others facilitate use in certain areas or make a special representation or notation possible. A useful script is helpful for everyone. Some scripts have been listed under Star Scripts because we feel that they will be particularly welcome. We can now easily enter a triplet when the first element is a rest and we can beam across a rest. “LayoutFix.py” is very useful as it corrects most layout errors that can occasionally occur after editing a capella score. When we were producing this manual we were continually running and rerunning scripts and this caused capella to misdraw the score on several occasions. Running LayoutFix instantly put the score back to normal.

In these English language versions of the scripts we at Software Partners have done our best to anglicise them. However where a script is aimed at an uncommon instrument – a harmonica for example – we may not have found the correct word in translation. Any feedback on this will be most welcome.

Getting at the scripts

To simplify downloading we have collected the translated scripts together into a single self installing program which when you run it will create a scipt folder on your C drive or wherever you wish. The scripts can then be copied into the capella script folder so that they will appear under available scripts when you are in capella. Currently the capella script folder is MY DOCUMENTS/CAPELLA/SCRIPTS/USER SCRIPTS.
STAR SCRIPTS – THE ONES WE WERE DELIGHTED TO SEE!
<Triplet fix> by Peter Becker

Triplet bracket fix
This script fixes a long-standing capella problem.

If the first element of a triplet is a rest, the triplet bracket has difficulty in forming correctly. We have shown an example of a triplet bracket formed without the script and also with the script.

The triplet bracket may still need to be manually adjusted for position. Mark the triplet bracket and use the direction arrows on the PC keyboard.
<Beam over marked rests> by Bernd Youngman
Extending beaming over marked rests
With this script beams can be extended over rests by simply highlighting the notes or chords over which the beam should extend.

- all marked notes and rests must be of equal duration
- beginning and end of the marked area must be a note or a chord.
<Beam over marked rests > by Urs Ganz
This script beams notes over a rest and changes rests to graphic rests. It also replaces mute notes with graphic rests.

Mark the relevant notes and run the script.

Conditions:
• there must be at least one note before the rest
• the beams over the rest can range in value from quaver to 64th notes
• dotted rests are recognised

Without this script …
If you manually make a beamed note invisible and replace it with a graphical representation of a rest, capella does not get the length of beam correct. The examples below show you the problems that can arise. Using the script gives you correct beaming.
<Layoutfix> by Peter Becker

A really useful script to fix layout problems in capella. When you rerun scripts or repeatedly undo changes, capella can misdraw the score. This script corrects such layout problems. In addition you can choose from 5 different standard layout styles.

The script calls on the settings already in capella under FORMAT and SCORE [shift]+[ctrl]+[p].
Below are some of the errors that may not be corrected and may require manual intervention through the SORE FORMAT dialogue box.

To adjust the following errors put the cursor in front of the note and open the note spacing dialogue box:

- notes pushed together
- bars wrongly represented
- notes appear too far apart
- the beam on semiquavers is too short

Both sliders can be tried. Sometimes it is enough just to click on OK. For some errors the script will only work after saving and reloading the score.
NOTATION SECTION
Covering:
Note formatting
Score formatting
Graphics
<Accidental adjuster> by Peter Becker
Automatic overlapping accidentals adjustment

This script corrects overlapping accidentals that occur in multi-voice staves. It moves the lower accidental sideways, away from its note.

• the position of the notes is adjusted when two notes of the same pitch occur in two voices. The note position is also adjusted if notes forming an interval of a second occur in two voices.
• if the notes are dotted, the dot will be made invisible in the lower voice.
<Voice split2> by Hartmut Ring
Close score to open score

This script is included in the capella standard release.

All the multi voice staves in an open score are split into single voice staves. To run the script open the score and call up the script.
**<Format everything> by Paul Villiger**
This script can change all of the major layout settings through its single dialogue window. You can also apply the changes to every capella score in any specified folder by ticking a check box. Please ensure that your scores are closed and that they are all backed up elsewhere before you use this option.

In the Title Composer and Footnote fields you can have multiple lines by adding the “|” character where you want a line end. On many PCs you get this by holding the right shift key and pressing the “\” key on the keyboard.

![Partitur Layout](attachment:image.png)
The script is aimed at quickly bringing an extracted part into a presentable layout. It does not work with multivoice staves.

- Correct score view |
  - Remove brackets |
  - Combine whole bar rests |
  - Reset rests (rest height) |
  - Make invisible rests visible |
  - Remove instrument names |
  - Reset spacing |
  - Correct stem direction |
  - Correct stem length

Switching between portrait and landscape requires you to save and reload the score for the change to take effect.
<Bar_numbering> by Hartmut Ring
This script is included in capella.

With this script bar numbers can be set with any frequency and position.

Existing bar numbers from this script can be removed by running the script
Bar_numbers_delete.py
<Octave change> by Andreas Hertzog

Script for shifting notes by an octave

The script changes the notes for the entire score. If notes are to remain unchanged then tick the penultimate check box.

Playback in capella can also be changed (with the bottom check box).
<Transpose an instrument> by Paul Villiger

With this script individual staves (e.g. only B flat clarinet) or the whole score can be transposed. The current key and position of the cursor are transferred to the dialogue, along with the target settings.
Results for an E flat instrument

To use this script enter the pitch played and the target pitch

**B flat**- Clarinet > **E flat**- Clarinet / **C** Clarinet > **F** French horn / **D** – Waldhorn > **B flat** Clarinet

*Note:* For the time being, the Score Template must still be changed.
<Rhythm synch> by Paul Villager

This script synchronizes the rhythm of two voices, or all voices with one.
<Enharmonicus> by Andreas Herzog

This script allows the user to swap accidentals at will.

The script swaps accidentals in cases where notes have been ‘spelt’ wrongly. Several notes can be modified at the same time, the whole score can be modified or one voice only can be modified.

Without this score it is necessary to modify each note individually.
<Voice split> by Andreas Herzog

A script to split a single stave, for example after MIDI import, into a piano (treble and bass) layout.

Application

• First a new stave must be added to the score in the score template.
• Put the cursor into the appropriate stave and run the script. This is recognized automatically by the script as the source voice.
• The stave holding the split off voice is automatically created underneath.

Select split point and the clef and key.

Insert note descriptions with the <WhatsYourName.py> script. These note descriptions can also be removed using this script. These descriptions are of a note’s pitch and octave.
<Swing notes> by Paul Villiger

Notes are converted into ternary or dotted rhythm notes and back again if required.

The transformation takes place based on crotchets and quavers. The following ranges can be selected:
- simple cursor -- current voice in the stave
- notes marked in a stave - > marked notes
- several marked staves - > current system
- several marked systems - marked > systems
- whole score

Ternary Swing – swing basis ¼
<Voice cocktail> by Andreas Herzog

This script combines the notes from multi voice staves to form chords and changes the stave to a single voice stave.

Put the cursor into the stave and run the script. Select stem direction.

Note:

- With irregular division it is not always possible to combine the voices without errors.
- Noteheads are not always oriented correctly on the stem. See below.

To correct this, please save the score and open again.
Expanding a chord description

A chord description over a note in one stave (for example the treble stave) will have the notes of the chord laid out on the other stave of the line of music (in this example, the bass stave).

The following functions are available:

- expand chords on the first stave
- put together chords of the whole system
- produce new stave for chords
- delete chord stave
- reduce chords back to a single note
- delete chord descriptions on the chord stave
- clean up chord stave; text, lyrics, diagrams
- select treble or bass clef
- correct the pitch
- determine the stave

Tips:

- this script produces chords which you might wish to lay out differently according to the musical context.
- anacrusis are converted and must be adapted manually in the chord stave
- bars without a time signature are treated as if 4/4
<Chord expansion> by Paul Villiger

Splitting up chords
Chords are re-written as separate notes on individual staves.

• the number of staves is derived from the maximum number of noteheads.
• the highest note is written to the first stave.
• missing notes are replaced by the lowest note or by a rest.
• the whole score or an individual voice can be converted.

Restriction: The score or the stave may not be multi-voiced.
<Flag lyric notes> by Hartmut Ring
This script is contained in capella2004

Beams and flags that are appropriate to the syllables of the accompanying lyrics are created.

The note is flagged if it is linked to a lyric syllable.

The note is beamed if:
• no syllable is linked to it
• there is a hyphen after the syllable
• there is a melisma (indicated by a long underscore) after the syllable.

The next note will be flagged if there is a syllable under it

*Exception*: if a melisma continues over more than two notes.

1. Aus drei - en scho - neo Bla - nen - lein will ich ein Bursch - lip
   
2. Je - sus die scho - ne Ti - ka - pan will ich am er - sten
   
3. Ihr En - leut, wo - ßt ihr auch eins han, ich will euch ei - nos
<Reiteration bars> by Hartmut Lemmel
Reiterated notes

This script produces notes with reiteration bars (tremolo bars).
You can specify:
• produce from single notes
• divide into single notes
• number and value of notes to be specified in the dialogue window
Selectable range:
• whole score
• marked range
• group of notes behind the cursor
An example score ‘Reiteration_Demo.cap’ is available with these scripts.
This script lets you add tremolo instructions as shown here to notes. Until now it has only been possible to add short tremolo bars to individual note stems.

Input settings:

- number of tremolo bars
- number of long tremolo bars
- tremolo bars over or under notehead
- tremolo bar to note stem
- distance from notehead
- space between tremolo bars
- apply to the whole score

The first note determines the duration, and all following notes are without value as regards bar filling. Notes in between are invisible and serve as placeholders. The tremolo bars can be moved with [N] + up/down arrows. If the score layout changes please recall the script and repeat the instructions.

Note: The staves containing the notes must be justified to ensure that the tremolo bars line up.
<Style editor> by Paul Villiger
This script provides a dialogue box method of creating new styles in the capella.dat file. When importing new styles through importing a text file, existing definitions in capella.dat are overwritten. The script replaces all necessary entries in capella.dat. The styles are arranged alphabetically, with capella standard always in first place.

Klaus Meglitsch has many elegant kinds of notation on his homepage.

To use an existing style written by someone else, click on the Style Import button and import it from the folder you saved the text file into. You may also need to install a special font if this is used in the style. This can be done through the Windows Add Fonts routine.
You can also modify an existing style and save it with a new name.

New styles are only available after restarting capella.
<Short to open score> by Andreas Herzog
This script converts a two stave SATB score to a range of open score combinations, not just a separate stave per voice.
<The Stylist> by Andreas Herzog

With the help of this script you can assign a certain style to a score. Only the styles that match the dialogue box names in your capella.dat can be activated. You can create new styles either with the <capella style editor> script or with Windows Notepad by directly typing out the settings in capella.dat. Details of how to do this are in the capella manual.
<LetsSing> Andreas Herzog
This script can extract the singing lines from a stave of chords.

The highest note is selected in this example. You can select the highest notes or the lowest notes.
<Score append> by Bernd Jungman

This script joins score with different Template Systems.

Before two scores are merged they are shown so that they can be matched interactively.

First open the appropriate score using file / exporting / CapXML as capx - files store.
Scores
Make sure that the title on the first score is only visible on the first page and is anchored to the page.
If you wish to retain the second title its anchor must be moved from the page to a note, otherwise it will not be merged.

Template System
The Template System is extended upwards

* The sequence of the lines can be determined in the dialogue box.
<Score synthesis> by Hans H. Lampe
CapCML
This script adds single voices or scores together.
Application:
- Select a file

- Export each single voice, and/or the entire score as a CapXML file into the target file
- Under file /exporting /CapXML open the first voice in capella
- Run the script
- Enter the file name of the next voice (without capx - extension)

Note:
• the number of bars must be the same in all voices
• the stave name is added to the template as a numerically ascending number
<Voice import> Paul Villiger
This script can merge a number of single voices from another score and the Template System is extended.

Application:
- Open the target score
- Run the script
- Select the score and then the voice to import

Note:
It is not possible to import an active score into capella, and for this reason if you import a score which is open it will be closed without warning.
<StaveEditor> by Paul Villiger

With this script staves can be:

- renamed, swapped, copied, duplicated and deleted
- merged: A + B -> A

An example of its use to rearrange an imported long shapeless MIDI file is shown on the next page.

Changes can be for the current line or the whole score. Any bracket and barline changes must be made through the Score Template.
Who has not suffered this? An imported Midi or Scan import can look like this.
First step number the staves so that when you swap or merge them you can trace your actions more easily.

For the example there are 2 'simple' systems from a very long score - they are not sequential:

- in system 1 are staves 2 - 5
- in system 2 are staves 7 – 10
- first staves 1 and 2 are swapped
• also revise staves 3 - 5
• staves 7 to 10 from 2 can be deleted
• merge staves 1 with 7 / stave 2 with 8 etc
• the two lines of music should now be better arranged.

• now delete staves 5 and 6
• then we are already nearly finished

As a last step the template system must be revised:
• delete redundant staves
• rename staves
<Systems lock> by Paul Villiger

This script adds a through barline to the end of a line of music.
<Format parallel staves > by Paul Villiger

This script can set up two parallel staves and also let you enter negative numbers to close staves up or overlap staves.

Application:

Take the line of music that you want on the right and set an inset to the correct horizontal position. FORMAT/SYSTEM and left margin.

Run the script and enter a negative number in the appropriate direction to raise or lower the stave to alongside the preceding stave.

![Notenzeilen Abstand](image-url)
This script automatically formats notes for bagpipe music.

Normally with bagpipe notation grace notes are represented as 32nds. If the melody is already in 32nds then the grace notes are set as 64ths. The grace note is small, with the stem up and without value. The other notes are automatically aligned downwards.
<LittleDrummerBoy> by Andreas Herzog
A simple script for percussion notation.
If the notehead is not correctly represented due to incorrect note length, it will be replaced by a graphic version.
<Rhythm notation for guitar> by Andreas Herzog
This script can create a stave in rhythm guitar style.

capella - input

Em9 Em9 Em9 Fm9 Fm9 Em9 Em9

Em9 Em9 Em9 Fm9

Em9 Em9 Em9 Fm9 Em9

Em9 Em9 Em9 Fm9
<Vocal range> by Peter Becker

The vocal range of a score is shown in small notes at the start of the line of music.

Kartoffelkantate

To use:
the first line of music must correspond to the layout in the Score Template.

In the first line of music you can enter the notes showing the vocal range:
• at the start of the line
• after the key signature
• before the time signature - all staves in the same key
• before the time signature - a stave in C major - with transposing instruments
• before the line of music

Before the time signature - for transposing instruments, if the music is in the key of C

To show the vocal range before the clef you should inset the line of music using the dialogue box which comes up under Format / Stem. Put a fixed value in the relevant box.
The first line of music is not complete – it requires the vocal range indication notes in the second system

* use Ctrl + C to copy or Ctrl + X to cut
* run the script
* insert the line again with Ctrl + V

Amen
<CueCopy> by Hartmut Lemmel
Copying cue notes

Application:
1) mark the passing notes in the score and start the CueCopy.py script.
2) with the cursor in the target position in a voice, start CuePaste.py. Now an additional voice will be provided, which is filled up to the cursor position with invisible rests. Subsequently the small formatted notes are inserted into the score. See CueDemo.cap which can be downloaded from the Paul Villiger homepage.
<Dots in multivoices> by Paul Villiger
Removing superfluous dots in multivoice staves

This script removes dotting in the lower voice when dotted notes of equal value are of the same pitch.
<Rest centerer> by Paul Villiger

With two voices the rest is centred if the same rest occurs in both upper and lower voices

To use:
Run the script to adjust the whole score.
Restriction: When there are several upper and/or lower voices the last voice will be used.
<Rest mover> by Paul Villiger

This script ensures that rests are always centred.

- rests can be shifted vertically in upper and lower voices.

- rests can be set to the centre in upper and lower voices

- rests can be made invisible

Restriction: The rest must be in an upper or a lower voice.
<Rest manager> by Andreas Herzog
This script condenses rests together.

- the maximum length of rests is selectable
- condensed rests can be shown in order of length

You can decide whether or not to use dotted rests.
*<Rest padder> by Paul Villiger*

**Fill with rests**

This script fills incomplete voices with rests.

Because voices are not always continuous throughout an entire score it can be useful to fill up all remaining bars of the line of music, or even the whole score, with rests. For example, if you are going to use the automatic wrap around feature all voices must be alike in all lines of music.

With this script you can add a voice and rests to the whole score, to the current line of music, to the current stave, to the current voice, or in the current voice before or after the cursor position.
The rests can be invisible.
<Rest summarizer> by Peter Becker
Condense individual rests to multiple bar rests
This script will condense all the whole bars rests that are adjacent when the cursor is anywhere in the score.

With this script one can condense multiple bar rests, as shown below.
<Beam over rests throughout the score> by Peter Becker

Beaming across quaver and semiquaver rests

This script beams across quavers and semiquavers. It will go through the whole score.

At the moment the following time signatures are supported:


The beam starts on the note.

There are selectable beaming patterns.

Note

- The author comments that the script may upset the score layout, so please check that you have the results you want before saving your work.
- We have found with this script that if you ‘undo’ and rerun it can extend the note stems off the page. Justifying the staves seems to prevent this.
Grace notes in beamed groups
Grace notes are created with flags and without the diagonal stroke.
• All marked notes must be of equal length
• there must be a note or a chord at the beginning and end of the marked area
• values from quaver to 32\textsuperscript{nd} notes are allowed
• grace notes must be without value
• this can change the value and alignment of the beamed group of notes
Stem direction after voice

Stems down

stem direction
‘depending upon situation’

When voices are exported from a multi-voiced stave they keep the stem direction instructions that were appropriate when the voices were in the original score. In the new score these directions will usually be inappropriate. This script corrects the setting back to ‘depending upon situation’. Forced stem directions can be reinstated if desired.
Stem length adjustment

Stems in upper and lower voices can be shortened with this script. The change is made to all notes beyond a specified stave line up from the middle stave line in the current voice and for the whole score.

In the dialogue box:

• upper or lower voice can be defined
• the cut line can be specified, that is the stave line above or below the middle line beyond which stems will be shortened.
• the minimum stem length can be specified

Restriction: The script does not have an effect on beamed notes. This feature might come in a later revision.
<Extensions> by Hartmut Lemmel
Handling beamed groups of notes

- only divide beamed groups which contain notes whose value is smaller than a default
- split beam between large and small size notes
- dialogue settings are remembered
- press Esc to cancel
- notes without value are also included

Note:
We have found with this script that if you ‘undo’ and rerun it can extend the note stems off the page. Justifying the staves seems to prevent this.
<Beam division> by Paul Villiger

With this script arbitrary bar divisions on a quaver basis can be entered.

Method:

• mark everything [CTRL]+[A] and under FORMAT/SYSTEM/GENERAL select WHOLE BAR BEAMS
• mark everything and under FORMAT/BEAMING select Automatic
• call up the script and enter Beam Division.
Linking symbols to notes

Graphic objects, note colours and notehead shapes on a reference stave will be distributed throughout the whole score to matching notes.

- in chords, if you assign a non standard notehead then all the notes of the chord will be the same, e.g. percussion noteheads.
- when notes are coloured the chord will receive the colour of the lowest note
- graphic objects – with chords only the lowest note is considered
- with cloned notes pitch and sign are currently considered
- graphic objects are linked to script tags

Reference stave setup

In the example below the three examples of recorder fingering are allocated to the notes F, E and D. When the script comes across an F, an E or a D in the target score it will anchor the appropriate graphic to that note. To create the reference stave simply start a new score in capella and put your range of notes on the stave. Then anchor the appropriate symbols to each of the notes. Save the stave. When you run the script from your target score the script will ask for the name of the capella reference file, which you created earlier.

• single notes on a stave with their corresponding images
• fingering, graphic objects etc. provide a note line with the appropriate individual notes and placing
• notehead colour, with chords receiving the colour of the lowest note
• save the reference stave

To use:
• open a score

• call up the script

• select the required options and press OK

• open the saved reference stave

• the script will be run.
<Breathmark reviser> by Andreas Herzog
The script offers the choice of 2 different breath characters which it can format throughout the score, [ , ] or [ ' ].
**<Articulation signs grouper> by Paul Villiger**

With capella it is not possible to set different articulation characters over a note – except for staccato/tenuto. This script does it by converting them to text. An additional benefit is that they are now also more easily repositioned.

![Articulation signs example](image)

The articulation characters can be set as follows

**with a main voice**
- automatically at the top
- above the stave or below the stave

**With upper voices**
- only over the stave

**With lower voices**
- only under the stave

**To use:**
- mark the note in capella add the first articulation sign in the usual way FORMAT/NOTES/RESTS.
- Run the script which will convert this accidental into a graphic.
- Repeat the two steps above to enter all the accidentals for this note.

**Exception:** Staccato/Tenuto, which can be combined in capella

![Staccato/Tenuto example](image)

The script cannot always determine the stem direction of beamed notes. If any entries are wrong, as the first accidental below, then confirm the stem direction manually by using the capella shortcut [i]+the up or down arrow with the cursor in front of the note.
You can convert the whole of the score’s articulation signs in one go if you select whole score.
<Group graphics> by Andreas Herzog
This script groups graphics which are anchored to the same note.

Put the cursor in front of the note and run the script.
<Brackets for guitar diagrams>
- the vertical bracket can be used as a damper/clamp for guitar diagrams
- the left open sided horizontal bracket can also be used
<Graphic notes> by Paul Villiger

Graphic notes can be produced using this script. The original notes are made invisible and replaced by graphical symbols of the notes. Tied notes are removed and the combined note duration will be represented by a single larger note duration. The score plays correctly, as capella plays the invisible notes.

Mensural notation
In Mensural notation notes are often tied over a barline. The script combines the notes as shown below.

```
\[\text{Image of mensural notation}]
```

Multiple dotting
capella supports single and double dotting. This script allows up to quadruple dotting.

```
\[\text{Image of multiple dotting}]
```

This script will do the following:
• Produce double dotting
• Replace the first note by triple or quadruple dotting
• Replace the second note with a graphic note of half or quarter value

Graphic grace notes
Script for grace notes within beamed groups of notes.
Quaver grace notes receive a diagonal bar, creating an acciaccatura (or ‘crushed note’).
• enter the notes, highlight the relevant note and press [SHIFT]+[CTRL]+[Y] to make the note small and add a diagonal stroke
• call up and run the script

Restriction:
Dotted grace notes are not supported.

```
\[\text{Image of graphic grace notes}]
```
<Slurs dash them all> by Hartmut Ring

This script is included in capella.

The script can create dashed slurs from normal slurs for the whole score.

- the script can sometimes produce layout errors
<Ties to slurs to ties> by Paul Villiger

Ties are converted into slurs and slurs are converted into ties. The script handles slurs at the end of lines.

Both chords /notes must be of the same pitch.

The script can apply to the entire score.

Note: The shape of slurs may need to be corrected manually.
Adapt lyric slurs in multi-voiced songs

In songs with two or more voices capella draws the slur at the stem end of the note. This script will redraw the slurs at the noteheads.

The following conditions apply:

- the notes must lie in an upper or a lower voice
- the notes must not be beamed
- the slurs must be over two notes only
<Accent spacing> by Hartmut Lemmel
Through a dialogue box slurs can be adjusted in the following way:

• Delete tenuto marks

• Offset staccato marks from slurs that have been automatically realigned by capella. This action can only be done once after automatic slur alignment.

• Offset other articulation marks. Only the articulation marks at the start of the slur will be modified (i.e. changed into text objects).

• Align slurs at cue notes

• Align slurs with changed staves. This action can only be done once after automatic slur alignment.

An example, Bindebogen_ausrichten.cap by Hartmut Lemmel, is available on the homepage of Paul Villiger.

Restriction: The script functions correctly only if the voice was originally created with the command to orient stem direction ‘Depending on situation’. 
<Slur fix> by Peter Becker

If a line of music is broken under a slur the slur is not pulled to the end of the line. The remainder of the slur is not transferred to the start of the next line. This script will pull the slur to the line end and start a new slur on the new line of music which will continue to the appropriate note.

Note: At present the slur might still need some manual adjustment.
<PerformanceMark centerer> by Andreas Herzog
This script centres dynamic symbols exactly in the centre between two lines

Application:
• Put the cursor into the upper of the two staves of a piano system
• Run the script – if all italic text fields are to be centred, tick the checkbox.
• With 3-stave systems put the cursor in the second stave and run the script again
<PerformanceMark standardization> by Andreas Herzog
This script lets you reorganize all selected types of performance mark to a specified distance above or below the stave.
<Graphic realignment> by Paul Villiger
This script provides a simple way to realign graphics throughout the score.

- Vertical position is selectable in each case
- Crescendo and decrescendo can be separately identified for movement
- Transposable symbols are left justified to the note or positioned over the first note in the bar
- The selected objects can be deleted
- The change can be for the whole score or in all staves of the line containing the cursor.
<Coloured Notes> by Lutz Haase

Some instruments are particularly suited to coloured notes and for children this is often a very helpful feature.

• free colour selection
• the colours (as portions of red, green and blue) are represented from 0 to 255
• the colour you are mixing is shown
• you can change the colours any time you run the script
• the following keyboard keys are supported
• on the screen click to increase in steps of 10
• using the cursor changes the value by 1
• for direct note entry type in any number from 0 to 255 and press <ENTER>

Important
• the colour of chords will be based on the colour of the lowest note in the chord
• at present it is not possible to save the colour selection.

Summ, summ, summ,
capella produces a multiple bar rest with quite a thick horizontal bar. Using this script you can change the thickness of this bar and its length.
• the distance from the barline can be set to be either proportional or fixed.
• the rest can be removed by rerunning the script
• the script affects the entire score.

With this script multiple bar rests can be adapted to the width of the bar. The capella multiple bar rest is set on invisible and the new multiple bar rest is inserted as a graphic. Existing graphical rests are replaced.
<Pedal marks> by Paul Villiger

Pedals markings (pedal brackets)

The script produces a pedal mark under marked notes.

Mark the range of notes and run the script

• the ends of the pedal mark can be selected.
• the pedal marks can be computed over the whole score.

Special effects:

• the end of the pedal mark is drawn up to the next break in the bar
• the pedal mark is drawn to the end of the stave
• the pedal marks must be marked by moving through the score using the TAB if they are to be deleted

Note:

The pedal marks are formed as polygons and therefore it is not possible to mark them with the mouse. To delete them tab through the stave. Each time the tab key is pressed the next mark is highlighted and can be deleted with the DELETE key.
<Repeat box alignment> by Paul Villiger

Aligning repeat boxes

The beginning and end of repeat boxes are aligned to the barlines.

Run the script and the repeat boxes for the entire score will be aligned.

Note: The brackets are formed as polygons and therefore it is not possible to mark them with the mouse. To delete them tab through the stave. Each time the tab key is pressed the next mark is highlighted and can be deleted with the DELETE key.
This script lets you align all objects anchored on the page in a way specified through
the dialogue box, e.g. left right, centre or to an X/Y coordinate.

The texts of different scores can be positioned accurately in the same place.

Application:

When you run the script each page object will appear in the dialogue box in turn.
When you press the OK button its setting will be remembered, and when you have set
all of them the page will be redrawn.
<Page object resize> by Paul Villiger
The Page Object was introduced with capella2004. These are objects anchored to the page rather than to notes. They can now be resized independently of the stave size. In previous versions of capella page objects were always anchored to notes and rescaled with the score. This script also lets you relink page objects to the score/stave size.

The options are to rescale and link the object to the current stave size as if it were capella2002, to link the object to the current stave size and then vary if stave is changed, or to link the object to an arbitrary stave size entered through the dialogue box.
<Font Replacer> by Hartmut Lemmel
Certain character fonts can be replaced
With single texts, character fonts and sizes can be replaced globally.

The number of changed objects is indicated
**<Lyrics to text converter> by Andreas Herzog**

This script exports lyrics into text fields. If nine verses are present, the lyrics remain.

**Options:**
- show the lyrics in text fields
- verse numbering is selectable
- verses can all be in a text box, or can be distributed individually
- the anchorage of the text fields is selectable

**Application:**
- Put the cursor anywhere in the stave or voice where the lyrics are anchored and run the script.
- You can decide where the text box will be anchored, for example on the page or on a particular note.
- You can specify whether you want multiple verses from the stave to be put into a single text box or separate text boxes. Also specify how paragraphs are to be set out.

**Note:**
When laying out the page please make sure that the anchor points of the text fields are within two systems of the text field. In order to move the text fields manually you need to put the cursor at the beginning of a stave and press the TAB key to jump along the stave to each anchored item until you reach your text box. This is the easiest way of finding out where things are anchored.
<Notentexter> by Andreas Herzog

With this script the note text of a line and/or all lines can be formatted.
<Letter and numbers> by Paul Villiger

Letters and numbers can be set with various options.

Note: If the setting *above barline* is ticked then regardless of which note the cursor is by, the text will be put above the left hand barline.
<Text_to_behind_notes> by Hartmut Ring

This is an original capella script.

All RTF text fields are set behind the notes in Windows 2000 and Windows XP

\textit{très net et très sec.}

\textit{très net et très sec.}
<Font – capella replacer> by Paul Villiger

This script replace the capella font symbols in the score, e.g dynamics. This script links another capella font to the score.

It can also change the plain text font associated with the capella character set.

The fonts are changed for the whole score.

Note: The style definitions are read from the capella.dat file. It therefore follows that the capella.dat file must be accessible and correct.

Style: capella standard font: capella standard

Style: New Year Font: Handwritten

To see the new fonts please restart capella.
<Figured bass> by Andreas Herzog

With this script one can enter figured bass.

• the font size can be changed
• the entire score can be changed
• the position under the stave can be excluded from the global change
• underlines can be changed in length and thickness
<Chord reformatter> by Paul Villiger
This script reformats chord descriptions throughout a score or for marked notes.

- different chord types are selectable
- changes between B and H are shown
- Minor keys shown in small letters
• super and subscript plain text accompanies a transposable chord description
• produce transposable chord descriptions for lyric/plain text lines

To produce chord descriptions from lyrics see the examples below.
Put the chord descriptions in a line of lyrics

Run the script and select this option

Delete the lyric line using the script

Select a font

Note: The script only shows scripts already in the score plus Times and Arial. If you add plain text in the normal capella way and select another font then this script will have the font available.

Convert Plain Text to chord description is not currently available.
Ach du leidiger Henker, was hab ich dir getan, daß du bist meinem Leben so gram und stößt mich in das Wasser.

„Auf rufet mir alle Fischer daher, sie sollen fischen bis ins Schwarze Meer, daß sie mein feins Lieb suchen, ja suchen.“
**<Fingering> by Andreas Herzog**

Script for inserting fingering.

The line and column distances are freely selectable in a dialogue field.

- promissory note symbols for organ
- slurs for mute finger change
- individual fingering elements can be grouped
- Font size, slurs and line thickness are selectable
- fingering can be set over or under the stave
- to make changes to the whole score make sure that the cursor is placed before a note with fingering, *before* running the script.

Some examples of the corrections with minus numbers.

\[
\begin{array}{ccc}
0 & -2 & 0.5 & -2 & 0.8 & -0.8 \\
\end{array}
\]

vertikaler Abstand \hspace{1cm} horizontaler Abstand \hspace{1cm} Bogenshöhe
FILE MANAGEMENT AND FILE FORMATS SECTION
This script translates a capella file into a MusicXML file, which can be imported into Finale, or Sibelius or other programs.

CapToMusic includes three support files Rtf2TxtAndFont.py, Rtf2Txt.py, and RtfParser.py by Loic Fejoz. The first is modified by Bernd Jungman.

Also contained is the SetupC2M.py script by Bernd Jungmann, which implements a persistence mechanism for variables like globalExportToSibelius. All four files are to be unpacked into the capella scripts directory.

If you start this script from capella it will translate the active score. You can also start Python with this script from the commandline:

```python
python CapToMusic.py -l# -f FilenameWithoutExtension -o DestFile
```

If DestFile is not given, FilenameWithoutExtension is taken as destination file.

-`l#` denotes the language for messages sent by the program.

This script supports:

5 English (US)
1 German

You can also call it from the Python-shell:

```python
>>> import CapToMusic
```

FilenameWithoutExtension.capx will be translated to DestFile.xml and saved in the same directory. If a file of the same name is present, it will be overwritten.
<XML code display> by Paul Villiger
A small aid for script developers. A note and the attached objects are indicated in XML format.

```xml
<Tiny_XML>
  <chord>
    <duration base="1/4">4</duration>
    <heads>
      <head pitch="C5"/>
    </heads>
  </chord>
</Tiny_XML>
```
<CapellaToCapX converter> by Paul Villiger
This script converts cap files in capX format and vice versa

This script converts all files of a folder
  - capella files in capX format
  - capX files in capella format
  - if desired, the script can create a subfolder
<HTML-export> by Hartmut Ring
This script is contained in the original capella and available from the Plugin menu. When run, this script converts a score to GIF files (one per page) with an index page.
<Score to bitmap exporter> by Paul Villiger
This script converts all capella files in a folder into the specified bitmap format. The parameters are determined by means of a dialogue box. The bitmap files are saved in a subfolder.
<Score cataloguer> by Paul Villiger

This script provides a listing of all capella files in a folder as an HTML file. The HTML file "score listing HTML" is saved in the same folder. The score name is a hypertext link so you can click on it to load the score.

<table>
<thead>
<tr>
<th>Title</th>
<th>Composer / Origin</th>
<th>Notes/Staff</th>
<th>Tenor/Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSP</td>
<td>Text: Max Dingler, Weise: Willi Kraus</td>
<td>Tenor 2</td>
<td>D-Dur</td>
</tr>
<tr>
<td>'s letzte Blaard</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D'Zimmereleut</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Das Busserl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Der Fensternock</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heumahder - Granz</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mei Glück is a Hütterl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wohl in de Wiedachwink</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dateiname: Partiturverzeichnis.html
Dateityp: Alle Dateien (*)
<Print all files in folder> by Paul Villiger
With this script entire capella folder contents can be printed.

Note: Unfortunately it is not possible to set Wildcards with the file dialogue of capella. For this reason the filter must be queried in a second step. The listing is selected in the first dialogue.
Specify in the second dialogue box the files to be printed.

E.g.: D*. *

If no file is found a reference appears

Examples of possible filters are shown below.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>*</code></td>
<td>all</td>
<td><code>*.*</code></td>
</tr>
<tr>
<td><code>d*.*</code></td>
<td>files begin with 'd'</td>
<td><code>d*.*</code></td>
</tr>
<tr>
<td><code>?</code></td>
<td>position indicated</td>
<td><code>A???</code></td>
</tr>
<tr>
<td><code>[ seq ]</code></td>
<td>containing from the sequence</td>
<td><code>0[12]?_*.*</code></td>
</tr>
<tr>
<td><code>[^ seq ]</code></td>
<td>not containing from the sequence</td>
<td><code>[^0123456789]*. *</code></td>
</tr>
</tbody>
</table>
TABLATURE SECTION
This script is for tablature writing.

**Dankschön**

Text Franz Fuchs, Melodie Volker Schöbitz

- The process requires three scripts to be run: DiatonicHarmonica_A, B and C
- output is a single melody stave with tablature below
- Unplayable notes can be shown in red or suppressed
- Ziach.ttf is required and installed by the Software Partners script file in the Windows Fonts folder.
<SwissOrgan tablature> by Radolf von Salis

This script produces fingering for the Swiss organ type B (b-it-As).

Detailed guidance on this script is given on Peter Becker’s homepage. The chord formatting script <Chord reformatter> by Paul Villiger will help in relaying out chord descriptions.

Application:
- use a capella file with a single stave score. This may contain chords.
- Notes must be in the B flat – E flat – A flat range.
  Transpose and/or change the octave to ensure that the score is in the E to G range.
- after some seconds fingering appears with the heading Schwyzeroergeli (Swiss organ) B
- unplayable notes are shown in red. You can select undo in the script and rerun, followed by further editing to remove the unplayable notes.
- sharpened notes are shown with a cross notehead
- minims are shown with a circled cross notehead
- notes over a thin line should be ‘pulled’
- notes over a thick line should be ‘pushed’
- unplayable notes are red
<ClubOrgan> by Peter Becker

This script works like the Ziach script.

**Application:**

- a score with a single voice stave - this may contain chords
- transposing chord symbols – use the transposing chord symbols from capella
- Start the three-part Club Organ scripts successively
- script A: transposes the notes into different key/clef C - F or Bb
- script B: adds fingering
- script C: changes the notes. To correct the printing of beams and insert accompaniment
- playable notes can be suppressed or shown in red
- the last note is not reliably picked up
< Ziach A> by Peter Becker
The button accordion

The script converts notes into the tabulature of the styrian accordian. It comprises three separate scripts that you run in order – Ziach A, Ziach b, Ziach C. The script is quite detailed and as the audience in the UK could be very small we have not translated this. Please let us know of interest out there.

Information about the script can be found in Ziach (button accordion) tab guide.pdf which should be in your script folder.

- International language support in German or English through the files
- Ziach_MSG deu.txt - Ziach_MSG eng.txt. The copy Ziach_MSG eng.txt from software Partners is already renamed to Ziach_MSG.txt
- Ziach.ttf: the Font with the tabulature symbols. The Software Partners script program will have saved this to your Windows fonts folder.
- Die_Ziach_DEF.txt bass definitions
• your own bass definitions can be provided
• the bass definitions must be copied to this file (Die_Ziach_DEF.txt)
• this file must in the path C:\Programme\capella software\capella 2004\scripts. If you have a UK copy of capella you will need to edit the path in the script appropriately.

The Ziach A: Script for preparation for conversion to fingering.

• Choose the bass stave
• Transpose for a three and/or a four keyboard harmonica in ADG,C voices
• Specify whether playing is on squeeze or expand
Blaue Luft, Blunzenduft
Wanderlied-Parodie


A Blun-ges, dann flüßt er's samt der Haut. A Heiß, wie die Würstln bra-t D in der ha-ßn

In-ge, gib mar als a Stück's A geht net al-las A dei.n.

Vom Vater gelemt und als Kind oft gesungen.

Die_Ziach_A:  

Auswahl des Bass-Systems

- Bass-System:
  - 4-Reihen: Steinisch nach F. Fuchs
  - 4-Reihen:
  - 4-Reihen:
  - 4-Reihen:
  - 4-Reihen:
  - 4-Reihen:
  - 4-Reihen:
  - 4-Reihen:
  - 3-Reihen, 2l. bei Maurer S.: 158
  - 3-Reihen:
  - 3-Reihen:
  - 3-Reihen:
  - 3-Reihen:
  - 3-Reihen:
  - 3-Reihen:
  - 3-Reihen:
  - 3-Reihen:
Chords which cannot be played are suppressed in the fingering

- only suppress the non playable notes of the chord
- or represent as red notes in the entire chord.
- non playable notes remain in the score
- playable notes are converted

The Ziach B script translates the score into fingering

- it is possible to correct clef, key and octave pitch.
- lyrics can be indicated or suppressed.
Blaue Luft, Blunzenduft
Wanderlied-Parodie

4 Reihen
Statisch nach F. Fuchs


B Hunger hat, dann C füt-ter's samt der B Haut. B Re-ta, wie die B War-sta tra-ten C in der ha-lm


Fritz Fuchs
Buchbergasse 63
A-1410 Klosterneuburg
Tel. 02243/35314
The Ziach C script makes optical corrections and produces an accompanying voice.
Note: If a stave is not required you can delete it with Ctrl+Del.

If the stave is deleted by using the backspace key this will cause capella to crash when implementing the Ziach C script.
<Guitar tab> by Andreas Herzog
This script produces tabulature for guitar, based on an original stave line.
Bass guitar

E Bass with five strings

- You can select the number of strings so the script can be used for bass guitar
- The script can change the current cursor stave or the entire score
Fonts can be specified as below and changed later by rerunning the program.

The open chords of the guitar

The range can be added to.
The use of the Capo tasto and/or empty strings is considered.

- the key on the TAB line can be set with letters or numbers on the strings
- for double string instruments the appearance is selectable
  - red
  - prevent
  - ignore
  - slurs can be added to cue notes
• Octave clefs can be set

You can revise individual TAB numbers by clicking on the TAB line and running the script. Put the cursor in front of the appropriate number in the TAB line and run the script.

You can change individual TAB settings by marking TAB letter.

Note. Ask to work on all voices after you have used this option.
<Lute tabulature> by Stefan Thierfeld

This script produces French lute tablature as below.
• all voices which will transfer into the lute tablature must be in a single stave line.
• they are transferred to a selectable number of strings in French notation.

• symbols for note length are inserted over the strings.

You can select spacing and layout through a dialogue box.

An example score "Ravenscroft_ThreeRavens_lute.cap" is in the script folder.
<Harmonica tablature> by Andreas Herzog
This script produces harmonica tabulature
**<Chromatic accordion tab> by Radolf von Salis**

Transformation of notes into fingering for the chromatic button accordion.

This script covers all major and minor keys.

- Normal fingering between the lines corresponds to the outside button row and the second internal button row.

- Fingering on the lines corresponds to the secondary outer button row and the internal button row.

- Sharpened notes (always between the lines) correspond to the middle button row.

---

**Application:**

Open a capella score with a single stave. It can contain chords.

- normal fingering between the lines
  - corresponds to the outside and the secondary internal button row
- normal fingering on the lines
  - corresponds to the secondary outside and internal button row
- cross notes - always between the lines
  - corresponds to the middle button row
The fingering cannot be transposed any more - a backup copy of the original score is therefore saved.
<Manuscript paper> by Paul Villiger

This script produces manuscript paper.

You can select from the following:
- Margins
- Stave spacing
- Number of staves
- Clef, also percussion clef or no clef
- Five or one line staves
- Landscape/Portrait

The settings are held in memory until you tick the END checkbox. Then the design is available to be printed.
<Playback time> by Hartmut Ring

This script is included in the standard capella release.

This script shows you the time it will take for your score to play back at the specified metronome setting. It shows you the metronome setting.
<Chord arpeggiation> by Hartmut Ring
This script is included in the standard capella release.

- put the cursor on the left hand side of the chord
- run the script
- all notes of the chord are played individually, from the bottom up
<What's Your Name> by Andreas Herzog

With this script capella’s internal note designations can be made visible.

Alternatively the normal note designation can be selected. The script can be also used with the harmonica scripts, in order to provide new allocations for harmonica (only for script programmers).
<How long is the note> by Andreas Herzog

With this script one can put text fields showing the note length over the notes.
<Score database> by Lutz Haase
Please note that we have only roughly translated this as it is not related directly to producing scores. We hope what is below will be enough for anyone who wants to see if the script is what they want.

The script is a basic version in working condition, which can be adapted to each user as desired. Lutz Haase has written very detailed documentation for this script and we recommend that you read it carefully. We have not tested this script at Software Partners.

- start the data base
- use new data base or open an existing one
- over the Button new entry the data seize
- with start search the data searched
- files can be copied
- the expression effected into a RTF file and can by user be still worked on here.
- a direct expenditure over the printer is not intended
- colours, arrangement, text and the number of the Button and fields can by each user be determined.
- the data base is not bound at capella - it can be used also for other purposes.
- assign an appropriate icon if desired

- for exact instruction refer to Lutz Haase
Start script and select database.

- externally with doubleclick - here a preparation is necessary in the file options
- in capella - the script please in the file script in capella2004 placing.
Change to the data base

- right mouse click on the file and click Edit. Select IDLE select
  - open two windows
  - make changes in in the window with the script
  - click and with the key < F5 > the script start and thus the data base open
- in the script in the last line before window Main loop () a # set
- then can the data base be arranged new
- is settled everything, which # remove again, so that the data base again with doubleclick or in capella be opened can.

- colour selection in the IDLE with the entry
  >>> from tkColorChooser import *
  >>> askcolor ()
  - register the result in the script
<table>
<thead>
<tr>
<th>Variable</th>
<th>Wert</th>
<th>Erläuterung</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGTitel</td>
<td>&quot;#95890C&quot;</td>
<td>Hintergrund des Kopfbereiches</td>
</tr>
<tr>
<td>VGTitel</td>
<td>&quot;#E8E0D0&quot;</td>
<td>Text des Kopfbereiches</td>
</tr>
<tr>
<td>BGFarbe</td>
<td>&quot;#b66f60&quot;</td>
<td>Hintergrund der Anwendung und Label</td>
</tr>
<tr>
<td>VGFarbe</td>
<td>&quot;#ffffff&quot;</td>
<td>Textfarbe der Label</td>
</tr>
<tr>
<td>BGFeld</td>
<td>&quot;#ffffff&quot;</td>
<td>Hintergrund des Feldes ohne Cursor</td>
</tr>
<tr>
<td>BGAktuell</td>
<td>&quot;#c5f297&quot;</td>
<td>Hintergrund des gerade aktuellen Feldes</td>
</tr>
<tr>
<td>VGFeld</td>
<td>&quot;#0000ff&quot;</td>
<td>Text der Eingabefelder</td>
</tr>
<tr>
<td>BGAktuell</td>
<td>&quot;#00af00&quot;</td>
<td>Cursorfarbe</td>
</tr>
<tr>
<td>BGSchalter</td>
<td>&quot;#95890c&quot;</td>
<td>Hintergrund des Buttons</td>
</tr>
<tr>
<td>VGSchalter</td>
<td>&quot;#ffffff&quot;</td>
<td>Text auf dem Button</td>
</tr>
</tbody>
</table>
Miscellaneous and fonts

<Transposable_symbols> by Hartmut Ring
This script produces transposable chord descriptions. It is an original capella script which can be run via the PLUGIN menu and "C 7".

• Put the cursor in front of the note

• a suggested chord description based on the note is put in the entry field which you can change

• if you enter a different chord description it must start with:
  - the letters A to G
  - then a #, b (flat) or m
  - any description such as sus must be shown after a space.
This script provides a headline, footer and up to two headers

In the dialogue window right-justified it can be able substitute symbols for page numbers and be entered instrument designations for each of these lines texts with variable text size to be entered
- left justified
- centered
- right justified
Substitute symbols for page numbers and instrument designations can be entered
Classical Notation Fonts

Meisterwerk
by Klaus Meglitsch

Engraver
by Klaus Meglitsch

capella 2
This was the standard font with capella 2.x. The difference between this and the current capella3.ttf is the narrower notehead.

Extreme round
by Andreas Herzog

Capella cursive
by Andreas Herzog
Ornamental fonts

The following heads can be represented, as the head form is changed:

1. Bells = normal notes
2. Nikolaus boots / socks = rhombus
3. crosses = cross
4. zipfelmuetzen = triangles
5. gifts = square
New Year song
By Andreas Herzog

capella summer
By Andreas Herzog

Gently rounded

Church singing

Original freehand
Renaissance

capella jazzy
<table>
<thead>
<tr>
<th>Scripts included with capella</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chord symbols</strong></td>
</tr>
<tr>
<td><strong>Chord play</strong></td>
</tr>
<tr>
<td><strong>Slurs dashed</strong></td>
</tr>
<tr>
<td><strong>Slurs dotted</strong></td>
</tr>
<tr>
<td><strong>Coloured notes</strong></td>
</tr>
<tr>
<td><strong>capella version</strong></td>
</tr>
<tr>
<td><strong>HTML export</strong></td>
</tr>
<tr>
<td><strong>Automatic lyrics</strong></td>
</tr>
<tr>
<td><strong>Signature check</strong></td>
</tr>
<tr>
<td><strong>Play back time</strong></td>
</tr>
<tr>
<td><strong>Voice extraction</strong></td>
</tr>
<tr>
<td><strong>Bar numbers</strong></td>
</tr>
<tr>
<td><strong>Bar number removal</strong></td>
</tr>
<tr>
<td><strong>Text behind notes</strong></td>
</tr>
</tbody>
</table>
### Gallery Extras

<table>
<thead>
<tr>
<th align="center">}</th>
<th align="center">1. Bracket to join verses by Klaus Mehlitsch – Klammer.cag</th>
</tr>
</thead>
<tbody>
<tr>
<td align="center">:---:</td>
<td align="center">2. Symbol for notation of psalms by Klaus Mehlitsch – Psalmodie.cag</td>
</tr>
<tr>
<td align="center">-</td>
<td align="center">Broken barline</td>
</tr>
<tr>
<td align="center">*</td>
<td align="center">Asterisk</td>
</tr>
<tr>
<td align="center">/</td>
<td align="center">Accent</td>
</tr>
<tr>
<td align="center">→</td>
<td align="center">Accent movement</td>
</tr>
<tr>
<td align="center">_</td>
<td align="center">Cadence start</td>
</tr>
<tr>
<td align="center">/</td>
<td align="center">Flexa</td>
</tr>
<tr>
<td align="center">---</td>
<td align="center">Quavers to triplets ternaer.cag</td>
</tr>
<tr>
<td align="center">---</td>
<td align="center">Quavers to triplets. ternaer.cag</td>
</tr>
</tbody>
</table>