

GABC/NABC real-time rendering software specification

0. Introduction and definitions

[INTRO_GABC]

GABC is the language in which the scores of Gregorian Chant in square notation are most commonly described. An example of such square note notation can be found [here](#). GABC also supports the addition of so-called “rhythmic signs” to scores in square notation. The same example, with rhythmic signs added, can be found [here](#).

NABC is an extension of the GABC language that supports the addition of ancient neumes. Ancient neumes are glyphs found in the earliest musical manuscripts, before the invention of the square notation (and, in turn, modern musical notation). Our previous example, with ancient neumes added, can be found [here](#).

GABC and NABC source files are plain text files with the .gabc extension, consisting of a header with metadata, and a body with GABC or NABC code. The source code corresponding to our examples can be found in Annex 2. Many other examples can be found via <https://gregobase.selapa.net/nabc.php>.

GABC language is best documented [here](#).

NABC language is best documented [here](#).

#

[INTRO_GREGORIO]

Gregorio is a piece of software that compiles .gabc files, including those using the NABC extension to the GABC language, into non-human-readable TeX files (with the .gtex extension), with the purpose of ultimately rendering the score as part of a LuaLaTeX document.

Gregorio is not very well documented, developer-wise, but this should have no bearing on the software specified in this document..

Its usage is simply “gregorio source_file.gabc”.

#

[INTRO_GREGORIOTEX]

GregorioTex is a LaTeX package that defines all the macros necessary for the inclusion of .gtex files obtained from Gregorio, into a LuaLaTeX document. This largely consists of:

- defining the fonts to be used (the note heads, staff lines, clefs, and all other signs, are characters of a specific font, and a whole score is a paragraph in this font)
- defining many distances: the distance between notes, between staff lines and lyrics, and so forth.
- defining special characters, either found in Unicode like U+211F, or not, like \Abar.

Gregoriotex is best documented [here](#).

#

[INTRO_PURPOSE]

The purpose of this document is to specify the functions of a real-time renderer for GABC and NABC. It is foreseen that this software would run client-side in a browser, and be written in javascript. The purpose of such software is essentially the proofreading of scores during the process of writing the corresponding source code in NABC. The rationale is that GABC/NABC compilation and rendering through LaTeX is extremely long and that mistakes are therefore detected long after being made, because compilations are few and far between in the NABC+LaTeX typesetting process.

#

[INTRO_STRUCTURE]

The first part of this document describes plain-GABC-only rendering functions, and outlines general requirements. The requirements in this part of the document are mostly, but not entirely, covered by an existing piece of software by Benjamin Bloomfield, called `jpgabc` and most commonly known as “Chant Tools” (specifically, the “transcription tool”). It is free software, the source code is found [here](#), and a web instance of `jpgabc` is found [here](#). It is anticipated that software meeting the present specification could be a fork of `jpgabc`.

The second part of this document describes NABC specifics, and is the core part of the work that will be performed on a fork of `jpgabc`.

Annex 1 is the compliance matrix of the latest (sept. 2022) version of `jpgabc` relative to the present specification document, and proposes different scopes for further work.

Annex 2 is the source code of the examples found above.

#

[DEF_SOFTWARE]

The Software is the piece of software, both source code and executable, or either depending on context, that meets the specifications set in this document.

#

[DEF_USER]

The User is any person running the Software in its intended manner.

#

[DEF_CODE]

“Code” designates NABC code, i.e. GABC source code with NABC extensions, unless “plain GABC” is specified.

#

[DEF_DROPCAP]

A drop cap is a larger letter, the first letter of the lyrics of a score, that takes up the whole height of the staff and displaces the beginning of the score by the width necessary to fit it.

See the examples given above.

#

[DEF_SCORE]



A rendered, human-readable musical score, corresponding to a specific GABC source code.

#

[DEF_GLYPH]

A NABC glyph is the joining of one or several characters from the Gregall and Grelaon fonts (or another such font), which themselves are vectorizations of signs found in ancient manuscripts.

A NABC glyph corresponds to what musicologists call “a neume”.

For example, the NABC glyph described in NABC by “pesu2lsc2” is described as a neume called “pes subbipunctis with celeriter on top”, and is the joining of the  character with a  character above it.

#

1. GABC real-time rendering

[REQ_CONTEXT]

The Software shall run in the latest version of Google Chrome.

The Software should run in any modern web browser supporting Javascript.

#

[REQ_FRONTEND]

The Software must not make constant transactions with a backend, but run client-side. However, user-triggered requests can happen (e.g. loading a font, loading a template).

#

[REQ_UI_STRUCT]

The Software’s user interface shall comprise a source code area, a rendering area, and a configuration area.

{Note: jgabc currently does not allow any configuration and has no configuration area.

However, Source&Summit, a rewrite of jgabc which is not free software, has it: [here](#).}

#

[REQ_UI_CODE_MODES]

The source code area of the Software’s UI shall be in one of two modes: “Separate text boxes” and “Integrated GABC”.

#

[REQ_UI_CODE_SEP]

In “Separate text boxes” mode, the user shall be allowed to input the score lyrics in a text area, and the note heights in another text area.

{Note: this is for plain GABC. For NABC, there shall be a third box, cf.

REQ_UI_CODE_SEP_NABC}

#

[REQ_UI_CODE_INTEGRATED]

In “Integrated GABC” mode, the user shall be allowed to input code in a single text area in the source code area of the UI.

#

[REQ_UI_CODE_SWITCH]

The user shall be allowed to switch between the UI “Separate text boxes” and “Integrated GABC” modes by clicking a button. Switching between modes must not result in changes to the rendered score.

{Note: currently, this is broken in jgabc: switching between modes and then updating the rendering does have side effects.}

When switching from “Separate text boxes” to “Integrated GABC”, the lyrics and note heights shall be merged into GABC code. The opposite operation shall be performed when switching back.

#

[REQ_UI_PARAMS]

The user shall be allowed to define the width, height, and left, right, top and bottom margins of the rendering area.

{Note: these parameters correspond to the same parameters from the geometry package when compiling and rendering code in LaTeX.}

The user should be allowed to define the presence or absence of a drop cap.

The user should be allowed to define the lyrics font size.

The user should ideally be allowed to define all GregorioTex-specific distances that are user-definable in LaTeX and documented in §2.4 of [GregorioRef.pdf](#).

#

[REQ_UI_CONFIG_AREA]

The parameters referenced in REQ_UI_PARAMS shall be accessible in the configuration area.

#

[REQ_CONFIG_SAVE_RESTORE]

The current state of the configuration shall be saved client-side.

The saved state of the configuration shall be restored when the Software is loaded again.

#

[REQ_CONFIG_SHARE]

The user shall be allowed to save the current state of the configuration to their hard drive as a file, and the software shall be able to import such a file.

{Note: this is useful if a lot of distances are configurable, to share configs across a project's contributors}

[REQ_RENDERING]

The Software shall convert plain GABC code, input in the source code area by the user, into a rendered score in square note notation.

#

[REQ_FAITHFULNESS]

The rendered score shall be visually identical to the score that would be rendered by Gregorio+LaTeX from the same source code.

{Note: this requirement cannot be complied with exactly; jgabc as it stands has several non-compliances with the behavior of Gregorio+LaTeX.}

{Note: this requirement becomes more important once distances are customizable, because the point of fine-tuning distances in jgabc is to not have to do it all over again by trial and error in Gregorio+LaTeX.}

#

[REQ_RESPONSIVENESS]

Changes made in the source code area shall be reflected in the rendering area within one second, for all scores with source code less than 2000 characters long.

This delay shall be less than three seconds, for all scores with source code less than 10000 characters long.

This is assuming mid-range consumer-grade modern computer hardware.

#

2. NABC real-time rendering

[REQ_RENDERING_NABC]

The Software shall convert complex NABC code, input in the source code area by the user, into a rendered score in square note notation with ancient neumes.

#

[REQ_UI_NABC_SWITCH]

The user shall be allowed to switch on and off the rendering of NABC by clicking a button.

#

[REQ_FAITHFULNESS_NABC_GLYPH]

{Note: some complex glyphs (with praepuncta/subpuncta and LS) are prepackaged into single characters of the Gregall or Grelaon font. Some need several characters to be assembled.}

When determining which character(s) to display for a given complex NABC glyph, the Software shall use the same algorithm as GregorioTex, as found [here, in the LUA language](#).

#

[REQ_FAITHFULNESS_NABC_HEIGHT]

When rendering NABC glyphs, the Software shall use the same increment of height between -ha-, -hb-... -hm- (the different possible vertical positions of a NABC glyph) as GregorioTex does.

#

[REQ_FAITHFULNESS_NABC_SPACE]

When rendering NABC glyphs, the Software shall render the '/' character (resp. the '' character) using the same increment (resp. decrement) of horizontal position as GregorioTex does.

{Note: this also corresponds to a distance that should be user-configurable per REQ_UI_PARAMS. If it is not configurable, the default shall be used as documented in GregorioRef.pdf}

#

[REQ_FAITHFULNESS_NABC_LS]

When rendering NABC glyphs, the Software shall position the significant letters (-lsxy- part of the complex glyph code) in the same position relative to the base glyph as GregorioTex does.

#

[REQ_UI_CODE_SEP_NABC]

{Note: this requirement extends REQ_UI_CODE_SEP.}

In “Separate text boxes” mode, a third text area shall allow the user to enter NABC glyphs corresponding to the lyrics and square notes of the two other text areas.

#

[REQ_UI_PARAMS_NABC]

From the configuration area, the user shall be allowed to:

- select the font used for rendering NABC glyphs (Grelaon or Gregall, which are part of the GregorioTex package);
- define the distance ‘abovelinestextraise’, which controls the average height of NABC glyphs above the staff, which is fine-tuned individually using -hx- (cf. REQ_FAITHFULNESS_NABC_HEIGHT);
- define the size of the font used to render NABC glyphs.

#

Annex 1. Compliance matrix

C = Compliant; NC = Non-Compliant; PC = Partially Compliant, cf. comments

Requirement	Current jgabc	Minimal NABC scope	Full scope	Comments
REQ_CONTEXT	C	C	C	
REQ_FRONTEND	C	C	C	
REQ_UI_STRUCT	PC	C	C	jgabc has no config
REQ_UI_CODE_MODES	C	C	C	
REQ_UI_CODE_SEP	C	C	C	
REQ_UI_CODE_INTEGRATED	C	C	C	
REQ_UI_CODE_SWITCH	PC	PC	C	A detailed list of non-compliances, i.e. rendered bits of code that break when switching modes in jgabc, may be established later. Fixing those would be very good but not a priority.
REQ_UI_PARAMS	NC	PC	C	jgabc only has drop cap config. No geometry. Geometry is part of the minimal NABC scope. Defining all distances is less of a priority.
REQ_UI_CONFIG_AREA	N/A	C	C	
REQ_CONFIG_SAVE_RESTORE	C	C	C	jgabc has no config for the transcription tool (the one that interests us), but has a locally-saved config for the psalm tool. This should be kept and expanded to have the newly configurable parameters.

REQ_CONFIG_SHARE	NC	NC	C	useful only if REQ_UI_PARAMS is C
REQ_RENDERING	C	C	C	
REQ_FAITHFULNESS	PC	PC	C	Cf. note in the req. All continued development, rather than actively fixing GregorioTex compliance in jgabc, should aim at fixing it in whichever areas development happens.
REQ_RESPONSIVENESS	C	C	C	
REQ_RENDERING_NABC	NC	C	C	
REQ_UI_NABC_SWITCH	NC	C	C	
REQ_FAITHFULNESS_NABC_GLYPH	NC	C	C	
REQ_FAITHFULNESS_NABC_HEIGHT	NC	PC	C	In a first step of development, something close enough would be ok.
REQ_FAITHFULNESS_NABC_SPACE	NC	PC	C	In a first step of development, something close enough would be ok.
REQ_FAITHFULNESS_NABC_LS	NC	PC	C	In a first step of development, something close enough would be ok.
REQ_UI_CODE_SEP_NABC	NC	C	C	
REQ_UI_PARAMS_NABC	NC	C	C	

Annex 2. Example source code

Example 1: GABC without rhythmic signs.

%%

```
(c4) DUM(e/fff) sanc(ed~)ti(gh)fi(g)cá(hj)tus(j) *() fú(j)e(j)ro(jjj) in(h) vo(h!iwj/kjj)bis,(ji) (;) con(g)gre(gh)gá(h)bo(hg/hgh) vos(e) (,) de(ed)
u(g)ni(hj)vér(iij)sis(hk) ter(jkjj)ris:(ji) (:) et(j) ef(j)fún(jj//jjh~)dam(ih) su(g)per(fg/hg/hi) vos(h) (;) a(gh/ji)quam(jvvlHiw!jvlH)
mun(gh!ivHGh)dam,(hg) (:) et(h) mun(g)da(h)bí(hkj)mi(kjjh)ni(h) (;) ab(g) óm(gh/ihj)ni(ghGFef)bus(fe) (,) in(ef~)qui(ef)na(e)mén(e!fh)tis(h!iwj)
ves(gfgvFD)tris:(d) (:) et(gj~) da(jv//jjj)bo(g) vo(h)bis(f) (,) spí(fh)ri(g)tum(hggf/ghg) no(egff)vum.(fe) <i>T.P.</i> (::) Al(e)le(f)lú(gh)ia,(g) (,)
al(gh)le(gh!iwjvlHG)lú(ge/fgFE)ia.(e) <i>Ps.</i> (::) Be(g)ne(hj)dí(j)cam(j) Dó(j)mi(j)num(j) in(j) om(k)ni(j) tem(j)pó(ih)re:(jjj) *(:) sem(ig~)per(hj)
laus(j) e(j)jus(j) in(jji) o(hg)re(h) me(i)o.(gh) (::) Gló(g)ri(hj)a(j) Pa(j)tri.(j) (::) E(j) u(jji) o(hg) u(h) a(i) e.(gh) (::)
```

Example 2: GABC with rhythmic signs

%%

```
(c4) DUM(e./fff) san(ed~)cti(gh)fi(g)cá(hj)tus(j) *() fú(j)e(j)ro(jjj) in(h) vo(h!iwj/kjj)bis,(ji..) (;) con(g)gre(gh)gá(h)bo(hg/hgh) vos(e) (,) de(ed)
u(g)ni(hj)vér(iij)sis(hk) ter(jkjj)ris : (ji..) (:) et(j) ef(j)fún(jj//jjh~)dam(i_[oh:h]h) su(g)per(fg/hg/hi) vos(h.) (;) a(gh/ji)quam(jvvlH'iw!jvlH)
mun(gh!ivHG'h)dam,(hg..) (:) et(h_) mun(g)da(h)bí(hkj)mi(kjjh)ni(h.) (;) ab(g) ó(gh/ihj)mni(gh_G'_FE'f)bus(fe..) (,)
in(ef~)qui(ef)na(e)mén(e!f'h)tis(h!iwj) ve(g_[oh:h]fgvFD)stris : (d.) (:) et(gj~) da(jv.jjj)bo(g') vo(h)bis(f.) (,) spí(fh)ri(g)tum(hggf/ghg)
no(egff)vum.(fe..) <i>T. P.</i> (::) Al(e)le(f)lú(gh){ia} ,(g.) (,) al(gh)le(g.h!iwjvlH'G)lú(g_[oh:h]e/fgF~'E~){ia}.(e) <i>Ps.</i> (::) Be(g)ne(hj)dí(j)cam(j)
Dó(j)mi(j)num(j) in(j) o(k)mni(j) tém(j)po(ih)re : (jjj) *(:) sem(ig~)per(hj) laus(j) e(j)jus(j) in(jji) o(hg)re(h') me(i)o.(gh..) (::) Gló(g)ri(hj)a(j) Pa(j)tri.(j)
 (::) E(j) u(jji) o(hg) u(h') a(i) e.(gh..) (::)
```

Example 3: NABC, along GABC with rhythmic signs

%%

```
(c4) DUM(e./fff//ta/tshh) sanc(ed~|cl~)ti(gh|pe)fi(g|ta)cá(hj|pe)tus(j|vi) *()
fú(j|vi)e(j|vi)ro(jjj|/tshh) in(h|ta) vo(h!iwj/kjj|`qlppt1`clhhsthi)bis,(ji..|cl-) (;)
con(g|ta)gre(gh|pe)gá(h|vi)bo(hg/hgh|//cl!po) vos(e.|vi) (,)
de(ed|clsc2) u(g|vi)ni(hj|pe)vér(iij|sa)sis(hk|peShi) ter(jkjj|tahh/pohilsc2)ris:(ji..|cl-hh) (:)
et(j|vi) ef(j|vi)fún(jj//jjh~|//ds-hi//ds>hi)dam(i_h|cl-) su(g|vi)per(fg/hg/hi|//pohhpp2vi>hj) vos(h.|vi) (;)
a(gh|j|//clsc2pp2)quam(jvvlHiw!jvlH|//bvhsut2lst2//ql-hhsu1sux1) mun(gh!ivHGh|//vi-hhpp2su2/vi)dam,(hg..|cl-) (:)
et(h_|vi>lstm2) mun(g|ta)da(h|vi)bí(hkj|/tohi)mi(kjjh|clShi/cl-hh)ni(h.|vi) (;)
ab(g|ta) óm(gh/ihj|/po>hhpp2)ni(gh_G'_FEf|//toSsu2/vi)bus(fe..|cl-) (,)
```

in(ef~|pe~)qui(ef|vslsc2)na(e|ta)mén(e!fh|sa)tis(h!iwj|vi-qlhh) ves(g_fgvFD|/cl-//cihhlsc2)tris:(d.|ta) (:)
et(gj~|pe~) da(jv.jjj|vi-//tshi)bo(g|ta) vo(h|vi)bis(f.|ta) (,)
spí(fh|peS)ri(g|ta)tum(hggf/ghg|/cl!pr////to) no(egff|/tosthh)vum.(fe..|/cl-) <i>T.P.</i>(:)
Al(e)le(f)lú(gh)ia,(g.) (,) al(gh)le(g.h!iwjv|HG)lú(g_e/fgFE)ia.(e.) (::Z)
<i>Ps.</i> Be(g)ne(hj)dí(j)cam(j) Dó(j)mi(j)num(j) in(j) om(k)ni(j) tém(j)po(ih)re:(jjj) (:)
sem(ig~)per(hj) laus(j) e(j)jus(j) in(jji) o(hg)re(h) me(i)o.(gh..) (::)