

Automatic insertion of detailed gamakam notation into raw notations (in Gaayaka format)

(Help file for using AddGamakam program)

1. The program (AddGamakam.exe) written by M. Subramanian, the author of Carnatic Music software Rasika, enables automatic insertion of notation for gamakams in raw (plain notes) in Gaayaka format, based on the Raagam characteristics. To play this notation Gaayaka program is required. A slightly upgraded version of Gaayaka is now made available and has the facility to easily exchange alternative notations for certain gamakams. The Addgamakam program is found to work well for varnams or swarajathis which are accurately notated and reasonably well for krithis whose basic notation is adequately accurate. To have an idea of the problems and limitations of automatic insertion of gamakam notation see http://compmusic.upf.edu/system/files/static_files/21-M. Subramanian-2nd-CompMusic-Workshop-2012_0.pdf which is a paper presented at the 2nd Computer Music Workshop held in July, 2012 at Istanbul, Turkey.

1.1 Licence: The programs are freeware. There are no guarantees about their functioning or the accuracy of the musical phrases. Please use at your own risk.

2. Installation and running: The programs work with all versions of MS-Windows upto Windows7 (32 or 64 bit). There is no installation file.

GaayakaAddGamakam.zip file contains AddGamakam.exe and its support files as well as the new version of Gaayaka (Gka7.exe) and its support files. Create a new folder (say) 'AddGamakam' in your PC. Unzip (extract/copy) the contents of the GaayakaAddGamakam.zip file into the new folder. The zip file has some folders which will have to be copied/extracted into the new folder as such. However, do not extract these files into Rasika folder if you already have Rasika in your system

2.2 You will now have 3 .exe files in the folder. AddGamakam(.exe), Gka7(.exe) and ChkGmkData(.exe) By clicking Gka7(.exe) you can run the new Gaayaka. By clicking AddGamakam(.exe) you can open plain notation files, convert them into notation with gamakam (for the files in Raagams for which Raagam definition files with extension .gkd are provided) and invoke Gaayaka with the converted notation loaded into it. The ChkGmkData.exe is meant for checking gamakam data files made by the user or files produced by modifying the given files.

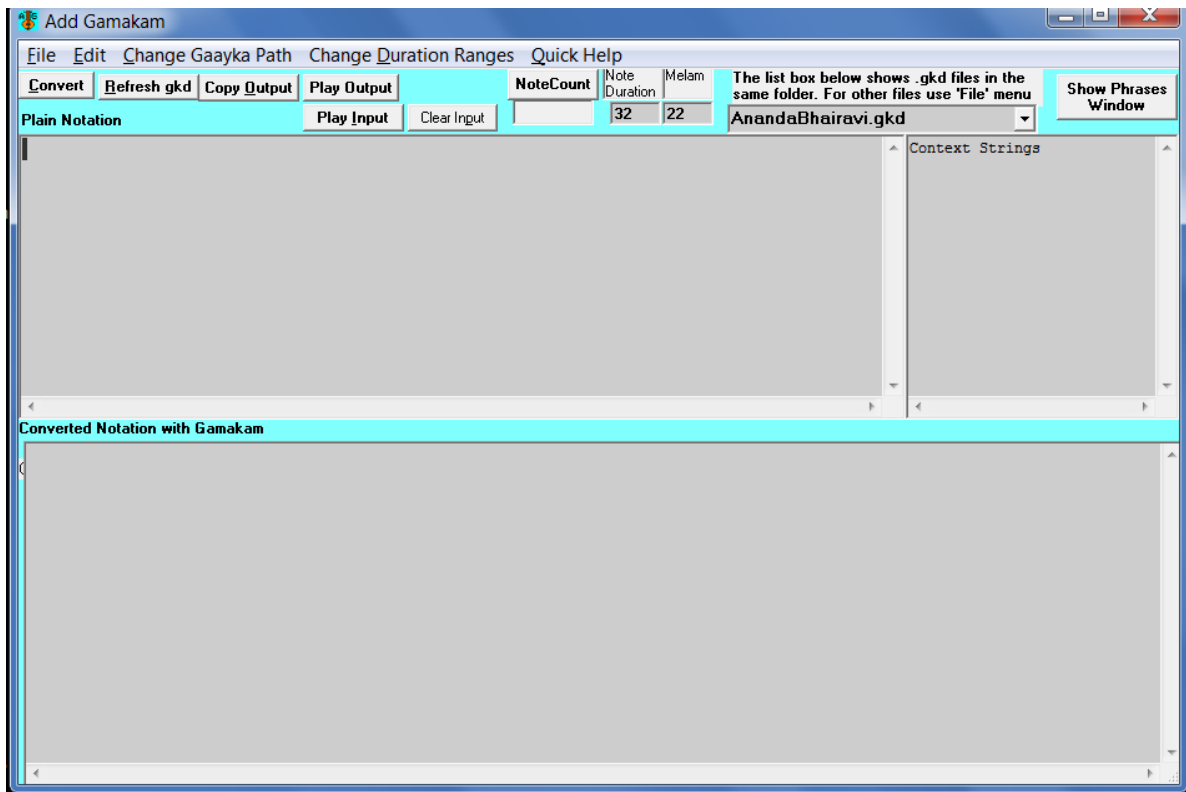
2.3 The sample Gaayaka files included in the RasikaV2 package are also included in the zip file in the folder 'RegularGkaFiles'. These are manually written files and not generated by AddGamakam program. They can be played and examined to understand how Gamakams can be notated.

2.4 If you are using Windows Vista or Windows7, you have to run each of the programs as 'Administrator' for the first time. Otherwise you may get error messages. So before starting to use AddGamakam.exe, run Gaayaka as

Administrator once, close it and then start AddGamakam program as Administrator.

3. Adding gamakam notation: The best way to proceed would be to first type the plain notation in Gaayaka for which you may use the new Gaayaka program. Choose only songs in the 16 Raagams for which gamakam files are included. In course of time I will add more files (and may also update some files given here). Set the Melam and Note Duration and play it to see whether the Tempo is OK and whether you have marked phrases with “-” symbol. Generally the “-” symbol is to be put where there are consonants in the lyric which is given with the notation. You can also copy the lyric in square brackets “[]” under the notation, align them and enter the “-” symbols in the notation. Where there are very long straight notes, suitable silences are to be added (ex. Saa;;; is too long make it Saa; - ;;;. Check the NoteCount to see that the count covers correct number of notes. You can play the raw notation file to see whether the tempo and phrase separations are OK. Save this Gaayaka file with a suitable name to indicate that it is a plain notation file (similar to the names in the folders of different Raagams in the unzipped folder).

3.1 Now run AddGamakam program by double clicking AddGamakam.exe in Windows Explorer. The screen shot of the AddGamakam program window is below:



3.2 Open the plain notation file using the File >Open dialog. The plain notation will appear in the top window.

[You can also type notation in the input window and convert it, but it is better to first create the plain notation file in Gaayaka and check it for correctness in terms of

number of notes, note duration, melam etc. If you prefer to type directly, then you have to enter Note Duration and select the Raagam. You can test it by clicking 'Play Input' which will open Gaayaka with the notation that you have entered]

3.3 Use Combo List box under 'The list box below shows' to load the required Raagam gamakam definition file (files with extension .gkd). This list will show the .gkd files in the same folder. Initially the first .gkd file (in alphabetical order) in the folder will be default gamakam definition file and shown in the list box. (You can also choose the gamakam data file using the menu File > OpenGamakamDataFile which will list all the .gkd files in the current folder, but also enables you to choose .gkd files in any other folder). At present the Raagam name is not included in the .gka file header and only Melam number is included. If you open a .gka file and the Raagam shown in the drop down combo box is of a different Melam you will get a warning, but if the Melam is same but Raagam is different you will not get any warning and the converted notation will not have correct gamakams. You have to choose the correct Raagam .gkd file.

3.4 You can insert gamakams only if there is a gamakam definition file. But sometimes the gamakam definition file of a Raagam in the same melam without much difference in gamakams may work (like Aabhogi using SriRanjani.gkd file -even this requires some tweaking for 'da' in aarohanam), but it will not work in most cases - for instance for Begada, Sankarabharanam.gkd file will not work.

3.5 Once notation is available in the top window and required Raagam .gkd file is chosen, click 'Convert'. The revised notation will appear in the bottom text window. (If some comment or lyric in [] is inserted in the middle of a note you will get a warning. The comment/lyric will get shifted to the end of the note.) You can also select some part of the raw notation in the top window and convert that alone. Ensure that the portion selected has no unbalanced brackets () or []. You can ignore the output in the top right window ('context strings'). This is required if you want to create your own gamakam definition file or edit one.

3.6 If you had opened a .gka file, Thaalam information if any will be incorporated in the temporary file created for playing (see later). If you had entered notation in the AddGamakam program window then you have to set Thaalam in the output opened in Gaayaka.

4. Playing the Output: You can play the output either by copying it (click 'Copy Output') and pasting it into Gaayaka. If you use this procedure you have to set the correct note duration and Melam as in the original file and play it. It is easier to play the output by directly clicking 'Play Output' in which case, the output is saved as 'AgTemp.gka' and the Gaayaka program (new Gka7.exe) in the same folder will open with the output loaded and the note duration and Melam automatically set. When you try it for the first time you may get a message box that Gaayaka program could not be found and asking whether you would like to locate it click 'yes' and then a file dialog box will open to enable you to indicate the location of the Gaayaka program. You can choose Gka7.exe which will be in the same folder and which has provision to exchange alternative Gamakam notations. You can also choose your

existing Gaayaka program (gaayaka6.exe or older) if you already have Rasika. When you close AddGamakam you will be asked to confirm that the path to the Gaayaka file indicated by you may be retained.

4.1 Once the converted notation is in Gaayaka, you can play it, edit it if required and save it with a suitable name. If you do not save it, it will be overwritten, when you use 'Play Output' next time. You can also click 'Change Gaayaka Path' to indicate the position of the Gaayaka program.

5. Note that note duration makes a lot of difference in the details of Gamakam applied. You have to choose correct duration for the raw notation by trial and error. Also if you get a warning that the melam of the Raagam differs from the one entered in the melam window, then check whether you have chosen the correct melam as in the Gamakam definition file. In some cases like Begada the books may give melam as 28 or 29. You have to use the melam number given in Rasika, Raagam module. (The melam number is available at the end of each Raagam definition .gkd file which can be viewed by opening the file in Notepad or other plain text editor).

5.1 You may also find that some gamakam notations in the output are enclosed in pairs of '@' symbols followed by notation in square brackets[]. These are alternatives possible. In the new Gaayaka these can be changed automatically by using the command button on top right which will open a menu. If you play the output as such, the first set of alternatives will be played as the second set is enclosed in square brackets. After choosing the required alternatives you can remove all the second alternatives and '@' characters using second item of the menu opened by the command button.

6. Results: Gamakam is a very complex and subjective thing. The gamakam used in a note is sensitive to the context and to the note duration. To make it manageable some sacrifice has been made in the accuracy. If you enter notation from a book, convert it by auto insertion of gamakam and find that the result is not up to the mark, it may not only be because of the above factor, but because the notation in the book itself is quite different from the currently sung version of the song. In addition, accurate use of "-" to mark consonants of the lyric is required for correct feeling of the lyric. This is often the single factor leading to unacceptable output. Another possible reason is that in the raw notation often, there is over-simplification of the kaalapraamaanam or thaalam structure.

6.1 The basic point to be remembered is that if a krithi in bare notation is given to two musicians, their interpretation and application of gamakams could be quite different and there may be no agreement as to what is 'correct'. Another aspect is the fact that a set of gamakams for a Raagam working well for one krithi may not work well for another (for instance the Begada krithis Vallabha and Naadopasana). To cover all styles and interpretations will not only be a laborious task but the user will face the problem of choosing. What is attempted is the common or average approach. However, the user can modify the Raagam definition file to suit his taste or modify the output. The program reduces the labour of making a .gka file from scratch with all gamakams.

7. Raagams included: Raagam gamakam definition files included: Aarabhi, Begada, Bhairavi, Dhanyaasi, Kaanada, Kalyaani, Kaambodi, Kharaharapriya, Madhyamaavathi, Panthuvaraali, Sahaana, Sankaraabharanam, Saaveri, Shanmukhapriya, Sriranjani, SuddhaDhanyaasi. Some of them are incomplete and you may get plain unchanged notes wherever gamakam notation is not provided for a particular note.

8. Possible problems:

8.1 Sometimes you may get the error “Phrase exceeds 20 notes – split suitably” when you play the revised notation although the original notation was OK. This is because the gamakam introduces additional notes (anuswarams). You have to split the original notation suitably.

8.2 The gamakam definition files contain mostly data for 4 duration ranges. Notes of longer duration range are split into 2 notes of smaller duration and corresponding gamakam notation is inserted. This may not always be suitable. In some case notation for 5th range ('nee' in Dhanyasi) has been added.

The following pages are for those who would like to create their own gamakam definition files or edit the available ones.

9. How it works.

9.1 The program first creates a 'context string' for each note (which are displayed in the “context strings” window). Essentially the context string has information of the note name, its duration range, whether the previous/succeeding notes are higher, lower or same. It also contains information of the actual duration of the note, the actual preceding/succeeding notes, position in the phrase etc. The context strings are displayed on the top right pane. (The position of the note in the chosen part of the file, the number of notes in the phrase and then one line each for the context string of the note). The following example explains the different parts of the context string.

da in the phrase 'ni da pa' with note duration of 36 (36/100 sec.)

D2HLCNP101.000-001003=36.000

Position	Character (in example)	Meaning
1	D	First letter of the note (da in the example)
2	2	Duration range (There are 6 ranges 0 to 5)
3	H	Direction of the previous note (H-higher, L-Lower, R-Same, Z-None)
4	L	Direction of the next note (same convention as above)
5	C	Position of the note in the phrase A-Alone, B-Beginning, C-Inside, E-End

6	N	First letter of the previous note (ni in the example)
7	P	First letter of the next note (pa in the example)
8	1	Octave (sthaayi - 0 lower, 1 middle, 2 upper)
9-14	01.000	Note size (ex da is 1, (da,) is 1.5, daa is 2)- Here 1
15	-	(Dummy)
16-18	001	Starting position in phrase
19-21	003	Ending position in phrase (after blanks if any)
22	=	(Dummy)
23-28	36.000	Actual duration of the note in 1/100 secs.

The first letter of the note (positions 1, 6 and 7) is normally in upper case (whatever the octave may be). If the note is an anyaswaram of a Bhaashaanga Raagam it will be in lower case).

Example 'paa da ni Saa' in Bhairavi:

P3ZHB0d102.000-001004=72.000
d2LHCPN101.000-005008=36.000
N2LHCdS101.000-009011=36.000
S3LZEN0202.000-012015=72.000

In the context string of the second note above, da is Chathusruthi dhaivatham which is the anyaswaram in Bhairavi and so it is in lower case. This is also reflected in the previous context string at position 7 and next context string in position 6 which both have lower case d instead of D.

In the case of a first note in a phrase, the previous note positions indicate the last note of the previous phrase and in the case of the last note in the phrase the next note positions refer to the first note of the next phrase. Instead of H,L or R in the 3rd or 4th positions, Z is used if there is no note or there is silence (i.e. start or end of the song or silence). In such cases for the actual note (positions 6 and 7) Q implies silence and 0 implies start or end of music.

There is some redundancy in the context string. The present system enables shorter Raagam gamakam definition files and faster processing since in most cases (about 90%) the first 4 characters are sufficient to define the gamakam.

Normally you have to bother only about the first 4 characters and further refinement in some cases will need the next 3 characters. The rest is only for the program to rebuild the gamakam included notation with correct duration and octave.

(When you convert raw notation using the program, the context strings for each note appears on the top right window. You may use this if you want to create a new Raagam gamakam definition file or edit one).

10. The Raagam's gamakam definition file

The gamakam definition .gkd file has one line each for all possible combinations of the first 4 characters of the context string, followed by a '#' and then the actual

gamakam string as in the example below.

```
D3LH#((pa.(da)pa.(da)))
```

The lines are arranged together for each note and note duration range and in alphabetical order for the first 2 characters. The program stops reading after it encounters a different note line is encountered and so the need for alphabetical order 'DGMNPRS' and then the ascending order of note duration range. The last line of the gamakam definition file contains the melam number for the present. Eventually Raagam name with standardised spelling has to be incorporated both in the gamakam definition file and the .gka file and Gaayaka program itself.

In the above example if there is no other line starting with D3LH in the gamakam definition file, the program will pick up the gamakam notation ((pa.(da)pa.(da))) . As you may see **the notation is exactly one note long** after taking into account the brackets and **is always in lower case** (madhya sthaayi) unless there is an anuswaram in the next sthaayi ex. (Sa (ni Sa)) for ni. As the gamakam phrase in the .gkd file is always one note long and in middle octave, the program will modify this to bring to the correct duration and octave. For instance if in the program the note is 'dAA then the two outer brackets above will be removed by the program and as it is in the lower octave the notes will become (pA.(dA)pA.(dA)).

In some cases the previous note or next note may influence the gamakam. In such cases additional lines will be there as in

```
D3LH#((pa//Sa\daa))#((/Sa\da)da)
D3LH__N#da
```

This is Kaambhodhi da. It may be held with 'nokku' in 'pa da Sa' but held without gamakam in 'da ni [da]'. The second line above shows 'N' after 2 underscores. This line will be used if the succeeding note is 'ni'. Here we do not bother about the position of the note in the phrase and so put an underscore '_' to skip the comparison. Again we are not bothered about the previous note and so put another underscore. Similar special definitions can be given for previous note, next note, position in phrase etc. and where some of these do not affect the gamakam we put a _ for skipping. For instance D2LH__S will be used for da if it is followed by sa, whatever the position in phrase or the previous note might be. 2 underscores are put to skip the position and previous note.

Where 2 alternatives are acceptable the second one is given with a '#' after the first phrase as in the first line in the above example.

If one of the alternatives is a plain note it can also be put as in

```
M3HL#ma#((da)\ma.)
```

If there is no definition for a particular group of first four characters (either missing 4 letter group or no '#' and phrase after the 4 letter group), the program will use the

original note as such.

The first alternative will be given enclosed by a pair of '@' and the next alternative by '[' and ']'. Thus playing the entire converted notation will play the first set of alternatives. Gka7.exe program of Gaayaka has a command button and menu to quickly exchange the 2 alternatives.

10.1 Janta

In the case of sa and pa the program will automatically insert simple janta (a crushed lower note of lower pitch of the same note, for a very short duration) in the second note of the pair. In the case of other notes ri,ga,ma,da and ni if simple janta is needed the symbol 'J' is put in the data base as in

G2RH#J

But often when a note is repeated simple janta will not work (as in 'ni nee' of Bhairavi or 'ga gaa' of Dharbaar). In such cases the gamakam definition has to contain the detailed notation. For instance the 2 nishaadams in 'ni-nee-da-pa' of Bhairavi are given gamakam as below:

```
N3ZR#((daa*(Sa) \da*.))#(((da*nee \daa*ni<<<daa)))  
N3RL#(((Sa)\da*.(ni*)da.))#(((ni)\da*.(ni*)da.))
```

10.2 'Gamakams' for sa and pa

Though these notes are not oscillated, they may be reached by a Jaaru as in the case of Aanandabhairavi reaching sa from ga. If this is required, lines can be added like S2HZ_G#\sa or 'orikai' added as in 'paa maa' of Sahaana (where usual notation omits the short anuswaram da) P2ZL__m#((paa,(da)))#pa. Here the plain alternative is also included.

11. The duration ranges are (in 1/100 secs.)

- 0 <= 12
- 1 >12 and <= 32
- 2 >32 and <= 48
- 3 >48 and <= 72
- 4 >72 and <= 120
- 5 >120

Generally range 0 requires no gamakam. But in special cases if required the data base can have them. If the data base contains no definitions for range 5, the program will split the phrase into 2 and apply the ranges for 2 and 3 and combine them into one phrase. This works in many cases but may not work in all cases.

12. A template .gkd file is included under the name FullTemplate.gkd and can be used to create new .gkd files by the brave and devoted ! Copy this and rename the copy as the <new raagam>.gkd file. One easy way is to take a song given with

Rasika software, create another file with bare notation (closely following the notation with gamakam, but simplifying and putting plain notes), convert it in small parts using this program. If the definition file has no definitions at all the output will be same as input. Then use the context strings and corresponding gamakams in the original Gaayaka notation (supplied with Rasika software) to create the data base. Such a data base will have lots of gaps and can be filled slowly by trying more songs or phrases in that Raagam.

12.1 If you want to edit a .gkd file supplied, make a back up copy. If you edit it after opening it in AddGamakam.exe then **click 'Refresh gkd' button** to enable the program to read again the gamakam definition file (which is read once and copied to an array when you choose a .gkd file)

Whether you edit a .gkd file or create a new one, you can check that the gamakam notation is 1 unit duration in all cases and also whether you have left anything blank after #, or whether brackets are balanced, by using the program ChkGmkData.exe which is included in the zip file. (This program has to be run Administrator when used first) It will show number of errors and the corresponding lines. It will also show number of empty lines (i.e those having 4 characters like D2LH) which can be ignored. Examples of errors are given below:

D1RZ 12 – BR D1RZ is the 12th line in the file and the brackets are not matched.

D2RZ 28 - 0 - D2RZ is the 28th line but it has no notation after # and so the duration is 0

G2HH 95 - .875 - In the 95th line G2HH, the phrase duration is .875 instead of 1.

13. For experimenting, provision has been made to change the duration ranges by clicking on the 'Change Duration Ranges' Menu. This opens a window showing the existing ranges and provision to enter revised ranges. The ranges will be effective only for the current session unless you save them when quitting the program in response to a message box which appears on Exit.

14. The command button 'Show phrases window', when clicked will open a new window below the context string window. The phrases in the input notation are shown separated by a #.

15. Also see

<http://gamakam.tripod.com>

which is the web site on this subject and gives links to other web sites on gamakam. A copy of article by M.Subramanian published in Sangeet Natak (referred to in the web site) on the issues involved in automatic insertion of gamakams can be sent to anyone interested. Link to the paper read on the subject in the Istanbul Computer Workshop is given in para 1 above. A slide set of a presentation on the subject with audio (including audio of auto converted files) is available at

<http://carnatic.heliohost.org>

For queries and suggestions email to manianms@yahoo.com