DRLVM Documentation Quality

Back to Documentation TODO

Introduction

This page gathers information on the current state and improvement progress for documentation autogenerated from DRLVM sources. You can view the current autogenerated documentation posted at Doxygen Starting page on our website. You can also regenerate the docs by parses local copies of source files (follow the same link for instructions).

Documentation Metrics

This section displays indicators for the quantity and quality of current autogenerated documentation.

Quantity of Documentation

This metric reflects the completion of documentation: scans resulting HTML documentation pages for each component, rates each page per doc completeness, and calculates the percentage of well-documented HTML pages.

• *	**Generated html files**			**Comments**	
Bundle	**Documented**	**Total**	**Completion**	• *	
Inter-component interfaces	76	267	29%	Q decrease 5%, increase in total file N(+56)	
Execution manager	15	44	36%	no change	
Copying/compacting GC	8	35	23%	no change	
Generational GC	7	41	16%	Q increase 1%, reduction in total file N(-3)	
Interpreter	16	26	62%	Q increase 19%	
Jitrino	15	59	28%	no change	
Porting layer	44	163	28%	Q decrease 1%, increase in total file N	
Thread manager	14	37	38%	no change	
VM core	72	542	14%	no change	
Class library support	6	9	66%	Q increase 6%, 1 undoc file less	
Component manager	10	23	43%	Q decrease 1%, 1 undoc file less	
Average	24	114	21%	Q increase 5%, reduction in total file N	

last update: 02-06-2008 this update measures against April'07 to show slow H2'07 progress

You can get the results for yourself. Generate documentation and run the scripts over it:

- Download check_doc_quality.sh and html_to_list.pl to the same directory
- Uncomment `convert_html_to_list` call at the end of `sh check_doc_quality.sh`
- Invoke `sh check_doc_quality.sh <documentation directory>`
- Check rating at `doc_quality/result.txt`

Quality of Documentation

The quality of comments in source files is measured by means of CCCC tool. See the attached cccc.html for a list of all scanned files and their quality. Red and yellow boxes represent entities with poor comments. Some fields have dashes instead of numbers, which means the tool has failed to estimate the quality. Check here for unabridged CCCC output.

We're planning to generate shorted ratings for DRLVM external interfaces and for VM core interfaces.

Ranking Algorithm

To rank the resulting HTML documentation for the Quantity Metric, do the following:

- Create a list of words from HTML files.
- Merge a dictionary of all words used in documentation.
- Remove a half of the most frequently used words from the dictionary - I believe they do not add much sense.
- Remove misspelled words (including identifiers) from the dictionary.
- Give a page +128 for each rare, correctly spelled word according to the dictionary.
- Divide to the sum of 128 and the total number of words on the page.

Doc Healthiness

Here we note the healthiness of generated docs by noting the amount of logged warnings and errors during doc generation. Things to note about this measurement:

- Warnings reflect errors in syntax and missing/redundant comments, and do not evaluate their quality; for example, an indication of author and revision instead of a class definition does not generate an error/warning, but does seem wrong
- Different warnings have different value, the relation between number of warnings and quality of docs is non-linear; e.g., there'd be just one warning "File not documented" instead of 10 warnings of smaller errors like wrong HTML syntax in a comment line or missing parameter definition
- Warnings are only printed for files that are documented; if no @file tag is present, Doxygen does not process it to print warnings; e.g., EM and Component Manager doc bundles are out of these measurements because file definitions are missing, though the code is sometimes well-documented

Bundle	**State**	**Comment**
Inter-component interfaces	yellow	some warnings
Execution manager	N/A	
Copying/compacting GC	green	almost none
Generational GC	green	almost none
Interpreter	green	almost none
Jitrino	green	almost none
Porting layer	green	almost none
Thread manager	green	almost none
VM core	red	many
Class library support	green	almost none
Component manager	N/A	

Back to Documentation TODO