

**NAME**

**bcg\_merge** – translation of a partitioned BCG graph into one single BCG graph

**SYNOPSIS**

**bcg\_merge** [*options*] *input* [.pb~~g~~] [*output* [.bc~~g~~]] [*global\_opt*] [*instance\_opt*]

**DESCRIPTION**

**bcg\_merge** is usually invoked after the **distributor**(LOCAL) tool of **CADP** for distributed state-space generation.

**bcg\_merge** takes as input a partitioned BCG graph *input* [.pb~~g~~], usually produced by **distributor**(LOCAL) and encoded in the PBG format (see the **pb~~g~~**(LOCAL) manual page for details about the PBG format), and produces one single BCG file *output* [.bc~~g~~] that combines the various graph fragments listed in *input* [.pb~~g~~].

If *output* [.bc~~g~~] is not specified on the command line, the output file will be named *input*.bc~~g~~.

These graph fragments are themselves encoded as BCG files and usually located on different machines. The list of these machines is given in a grid configuration file (see the **gcf**(LOCAL) manual page for details about the GCF format) referenced in *input* [.pb~~g~~].

**bcg\_merge** renumbers the states contained in the BCG files listed in the input PBG file, in such a way that the states of the output BCG file are assigned contiguous numbers and that the initial state is assigned number 0.

**OPTIONS**

The following *options* are currently supported:

**-compress, -register, -short, -medium, -size, -uncompress**

These options control the form under which the BCG graph *output*.bc~~g~~ is generated. See the **bc~~g~~**(LOCAL) manual page for a description of these options.

**-clean** After generating the output BCG file, erase the input PBG file together with its BCG graph fragments and their corresponding BCG dynamic libraries. Not a default option.

**-fast** When building *output* [.bc~~g~~], sort, by increasing source state numbers only, the transitions belonging to the file fragments listed in *input* [.pb~~g~~]. This is a faster sort than the default one, which sorts transitions lexicographically, by increasing source state numbers, then by increasing target state numbers; however, this faster sort may generate a larger BCG file than the lexicographic sort. Not a default option.

**-monitor**

Open a window for monitoring in real-time the generation of the BCG file *output*.bc~~g~~. Not a default option.

**-parse | -unparse**

These options can be used to control label parsing in the output BCG graph (see the **bc~~g~~\_write**(LOCAL) manual page for a technical discussion about label parsing). By default, or if option **-parse** is present, label parsing is enabled. If option **-unparse** is present, label parsing is disabled.

- stat** Display various statistics obtained during graph merging, such as the number of cross-border transitions in the partitioned BCG graph, and the network traffic with the remote machines. Not a default option.
- tmp** This option specifies the directory in which temporary files are to be stored. See the **bcg(LOCAL)** manual page for a description of this option.

Finally, the list of options *global\_opt* and *instance\_opt* can be used for last-minute changes to the settings of the grid configuration file referenced in *input[.pbg]*. See the **distributor(LOCAL)** manual page for a description of these options.

## ENVIRONMENT VARIABLES

See the **bcg(LOCAL)** manual page for a description of the environment variables used by all BCG application tools.

## EXIT STATUS

Exit status is 0 if everything is alright, 1 otherwise.

## AUTHORS

Version 1.0 of **bcg\_merge** was developed by Irina Smarandache-Sturm (INRIA/VASY) in 2000.

Version 2.0 of **bcg\_merge** was developed by Radu Mateescu (INRIA/VASY) in 2001.

Version 3.0 of **bcg\_merge** was developed by Nicolas Descoubes (INRIA/VASY) in 2003, then revised by Damien Bergamini and Hubert Garavel (both at INRIA/VASY) in 2004.

## OPERANDS

<i>input.pbg</i>	partitioned BCG graph (input)
<i>output.bcg</i>	resulting BCG file after merging (output)

## FILES

The binary code of **bcg\_merge** is available in `$CADP/bin.'arch'/bcg_merge`.

During the execution of **bcg\_merge**, dynamic libraries corresponding to the BCG files (i.e., the graph fragments) listed in the input PBG file may be generated if necessary, and stored on the remote machines.

## SEE ALSO

**bcg(LOCAL)**, **gcf(LOCAL)**, **pbg(LOCAL)**, **distributor(LOCAL)**, **pbg\_cp(LOCAL)**, **pbg\_info(LOCAL)**, **pbg\_mv(LOCAL)**, **pbg\_rm(LOCAL)**

Additional information is available from the CADP Web page located at <http://cadp.inria.fr>

Directives for installation are given in files `$CADP/INSTALLATION_*`.

Recent changes and improvements to this software are reported and commented in file `$CADP/HISTORY`.

## BUGS

Please report bugs to [cadp@inria.fr](mailto:cadp@inria.fr)

**BIBLIOGRAPHY**

[GMS01] Hubert Garavel, Radu Mateescu, and Irina Smarandache. Parallel State Space Construction for Model-Checking. In Matthew B. Dwyer, ed, Proceedings of the 8th International SPIN Workshop on Model Checking of Software (SPIN'01), Toronto, Canada, LNCS 2057, pp. 217-234, May 2001. Revised version available as INRIA Research Report RR-4341, December 2001. Available from <http://cadp.inria.fr/publications/Garavel-Mateescu-Smarandache-01.html>

[GMB+06] Hubert Garavel, Radu Mateescu, Damien Bergamini, Adrian Curic, Nicolas Descoubes, Christophe Joubert, Irina Smarandache-Sturm, and Gilles Stragier. DISTRIBUTOR and BCG\_MERGE: Tools for Distributed Explicit State Space Generation. In Holger Hermanns and Jens Palberg, eds., Proceedings of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06), Vienna, Austria, LNCS 3920, pp. 445-449, March-April 2006. Available from <http://cadp.inria.fr/publications/Garavel-Mateescu-Bergamini-et-al-06.html>

[GMS12] Hubert Garavel, Radu Mateescu, and Wendelin Serwe. Large-scale Distributed Verification using CADP: Beyond Clusters to Grids. Electronic Notes in Theoretical Computer Science, vol. 296, pp. 145-161, August 2012. Available from <http://cadp.inria.fr/publications/Garavel-Mateescu-Serwe-12.html>