

# Package ‘pbdSLAP’

June 14, 2021

**Version** 0.3-1

**Date** 2021-06-12

**Title** Programming with Big Data -- Scalable Linear Algebra Packages

**Depends** R (>= 3.6.0), methods, rlecuyer, pbdMPI (>= 0.3-1)

**LazyLoad** yes

**Copyright** See 'pbdSLAP/inst/ScaLAPACK\_LICENSE.txt' for the files in 'src/BLACS/', 'src/PBLAS/', 'src/REDIST/', 'src/ScaLAPACK/', and 'src/TOOLS/'.

**Description** Utilizing scalable linear algebra packages mainly including 'BLACS', 'PBLAS', and 'ScaLAPACK' in double precision via 'pbdMPI' based on 'ScaLAPACK' version 2.0.2.

**SystemRequirements** 'OpenMPI' (>= 1.5.4) on Solaris, Linux, Mac, and FreeBSD. 'MS-MPI' (Microsoft HPC Pack 2012 R2 MS-MPI Redistributable Package) on Windows.

**License** Mozilla Public License 2.0

**URL** <https://pbdR.org/>

**BugReports** <https://github.com/snoweye/pbdSLAP/issues>

**MailingList** Please send questions and comments regarding pbdR to RBigData@gmail.com

**NeedsCompilation** yes

**Maintainer** Wei-Chen Chen <wccsnow@gmail.com>

**Author** Wei-Chen Chen [aut, cre],  
Drew Schmidt [aut],  
George Ostrouchov [aut],  
Pragneshkumar Patel [aut],  
Brian Ripley [ctb] (Solaris & Mac)

**Repository** CRAN

**Date/Publication** 2021-06-14 14:20:02 UTC

## R topics documented:

pbdSLAP-package . . . . .	2
SLAP Grid Information . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

pbdSLAP-package	<i>Programming with Big Data – Scalable Linear Algebra Packages</i>
-----------------	---

---

### Description

pbdSLAP utilizes scalable linear algebra packages mainly including BLACS, PBLAS, and ScaLAPACK in double precision via pbdMPI based on ScaLAPACK version 2.0.2.

### Details

Package:	pbdSLAP
Type:	Package
License:	MPL2
LazyLoad:	yes

This package requires **pbdMPI** and MPI system. The main purpose of **pbdSLAP** is to provide several scalable linear algebra packages containing double precision libraries for **pbdDMAC** or other useful packages.

### Author(s)

Wei-Chen Chen <wccsnow@gmail.com>, Drew Schmidt, George Ostrouchov, and Pragneshkumar Patel.

### References

Programming with Big Data in R Website: <https://pbdr.org/>  
 ScaLAPACK Website: <http://www.netlib.org/scalapack/>  
 ScaLAPACK Block Cyclic Data Distribution Website: <http://icl.cs.utk.edu/lapack-forum/viewtopic.php?f=5&t=4922>

### Examples

```
## Not run:
### Under command mode, run the demo with 2 processors by
### (Use Rscript.exe for windows system)

mpiexec -np 2 Rscript -e "demo(gridinfo,'pbdSLAP',ask=F,echo=F)"

## End(Not run)
```

## Description

These functions initialize a grid of pbdSLAP, assign the information to a global object, and free the grid.

## Usage

```
slap.init.grid(nprow, npcol = 1, ictxt = 0)
slap.exit.grid(ictxt)
slap.finalize(quit.mpi = FALSE)
```

## Arguments

<code>nprow</code>	number of row processors.
<code>npcol</code>	number of column processors.
<code>ictxt</code>	the grid id
<code>quit.mpi</code>	if finalize MPI.

## Details

This function arranges all processors in a (`nprow * npcol`) grid and the grid will map the big data matrix.

## Value

`slap.init.grid` assigns a global object `__grid_info_0` for `ictxt = 0` containing the grid information. `slap.exit.grid` free the grid. `slap.finalize` free all memory.

## Author(s)

Wei-Chen Chen <[wccsnow@gmail.com](mailto:wccsnow@gmail.com)>, Drew Schmidt, George Ostrouchov, and Pragneshkumar Patel.

## References

Programming with Big Data in R Website: <https://pbdr.org/>

ScaLAPACK Website: <http://www.netlib.org/scalapack/>

ScaLAPACK Block Cyclic Data Distribution Website: <http://icl.cs.utk.edu/lapack-forum/viewtopic.php?f=5&t=4922>

**Examples**

```
## Not run:  
### Under command mode, run the demo with 2 processors by  
### (Use Rscript.exe for windows system)  
  
mpiexec -np 2 Rscript -e "demo(gridinfo,'pbdSLAP',ask=F,echo=F)"  
  
## End(Not run)
```

# Index

- \* **package**

- pbdsLAP-package, [2](#)

- \* **programming**

- SLAP Grid Information, [3](#)

pbdsLAP (pbdsLAP-package), [2](#)

pbdsLAP-package, [2](#)

SLAP Grid Information, [3](#)

slap.exit.grid(SLAP Grid Information),  
[3](#)

slap.finalize(SLAP Grid Information), [3](#)

slap.init.grid(SLAP Grid Information),  
[3](#)