

Linux サーバー版 GAMESS インストールマニュアル

2019/1/27

本マニュアルの目的

本マニュアルでは、単独ユーザが独占的に Linux サーバー (CentOS 6.6) を使用して GAMESS ジョブを実行するための環境構築方法、および Winmostar のリモートジョブ機能による計算手順を示しています。つまり複数ユーザが共同使用する計算サーバー等は対象外です。また単独ノードでの複数コアによる並列計算のみを想定しており、複数ノードにまたがる並列計算は対象としておりません。そのため計算環境構築は全てユーザのホームディレクトリ配下で行うことを想定しています。共用サーバー上での環境構築および複数ノードによる環境構築方法につきましては別途お問い合わせください。

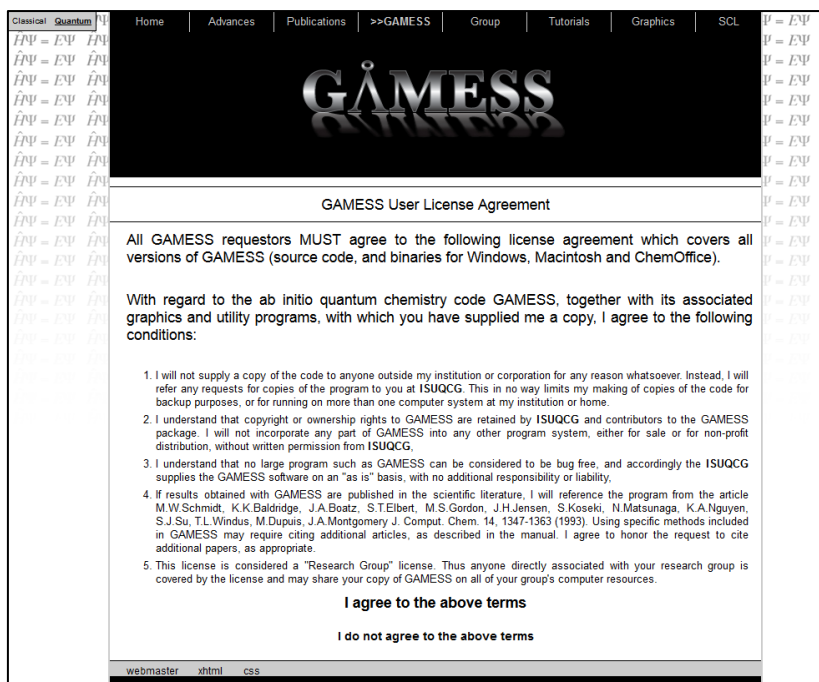
なお、本マニュアルでは Linux サーバー上でジョブスケジューラ TORQUE¹と数学ライブラリ ATLAS²が使用できる状態であることを仮定しています。また以下の命名としています。

サーバーホスト名 : remote_server

ユーザ名 : winmostar_user

1. GAMESS のサイト内の[GAMESS User License Agreement]サイトにブラウザ (IE など) を用いてアクセスする。³

http://www.msg.ameslab.gov/gameSS/License_Agreement.html



同意できる場合は、「I agree to the above terms」をクリックする。次の画面が表示される。

¹ <http://www.adaptivecomputing.com/products/open-source/torque/> yum 等を用いて導入する。

² <http://math-atlas.sourceforge.net/> yum 等を用いて導入する。

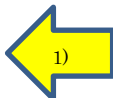
³ 本手順は、GAMESS のバージョン「GAMESS VERSION = 18 AUG 2016 (R1)」に基づいて記述されている。それ以外のバージョンについては手順が異なることがある。

GAMESS Registration System

Request Download

Fields in bold face type are required.

Email address:



Select desired programs below

Pre-compiled Binary Distributions

Distributions tested on their platform:

- GAMESS version December 5, 2014 R1 for Apple MacOS X
- GAMESS version May 1, 2013 R1 for Microsoft Windows
- GAMESS version Sep 6 2001 for MacOS 7-9
- GAMESS (VigyanCD based on an obsolete version of GAMESS) version 1.0 beta for 32 bit x86 compatible PC
- GAMESS for ChemOffice version 6-9 (Does not include ChemOffice or work with more recent versions) version June 2003 for Microsoft Windows

Source Code Distributions

There is only one combined source code version of GAMESS, containing versions to compile on any of the machines listed below. You will be downloading only one tar file, so pl survey of the actual usage of different machine types, so please check all of the machines which you do actually intend to use.

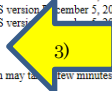
Distributions well-tested on their platform:

- GAMESS version December 5, 2014 R1 for 32 bit (x86 compatible) under Linux
- GAMESS version December 5, 2014 R1 for 64 bit (x86_64 compatible) under Linux with gnu compilers
- GAMESS version December 5, 2014 R1 for 64 bit IA64-x86_64 under Linux with Intel compilers
- GAMESS version December 5, 2014 R1 for Apple MacOS X
- GAMESS version December 5, 2014 R1 for AXP under Tru64 Linux (sold as Digital, Compaq, or HP)
- GAMESS version December 5, 2014 R1 for Cray X1 under Unicos
- GAMESS version December 5, 2014 R1 for Cray XT
- GAMESS version December 5, 2014 R1 for Fujitsu PrimePower under Solaris OE
- GAMESS version December 5, 2014 R1 for HP PA-RISC or Intel under HP-UX
- GAMESS version December 5, 2014 R1 for IBM BlueGene
- GAMESS version December 5, 2014 R1 for IBM PowerPowerPC processors under AIX and Linux
- GAMESS version December 5, 2014 R1 for IBM SP under AIX
- GAMESS version December 5, 2014 R1 for Microsoft Windows HPC Server 2008 using PGI compilers
- GAMESS version December 5, 2014 R1 for NEC-SX series
- GAMESS version December 5, 2014 R1 for Sun Opteron or UltraSPARC under Solaris
- GAMESS-LIBCCHEM version December 5, 2014 R1 for NVIDIA GPU's



Distributions for older systems (extra effort may be needed to run)

- GAMESS version December 5, 2014 R1 for Cray T3E under Unicos
- GAMESS version December 5, 2014 R1 for Cray vector machine under Unicos
- GAMESS version December 5, 2014 R1 for SGI MIPS under Irix



Request submission may take a few minutes. Please only click the Submit button once.

1) Email address を入力する

2) Linux 版(64 bit)を入手の場合は GAMESS versionfor 64 bit (x86_64 compatible) under Linux with gnu compilers にチェックを入れる

3) Submit Request をクリックする

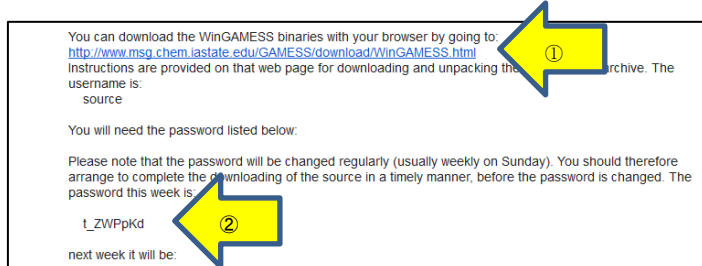
以下のような受付完了の画面が表示される

GAMESS Registration System

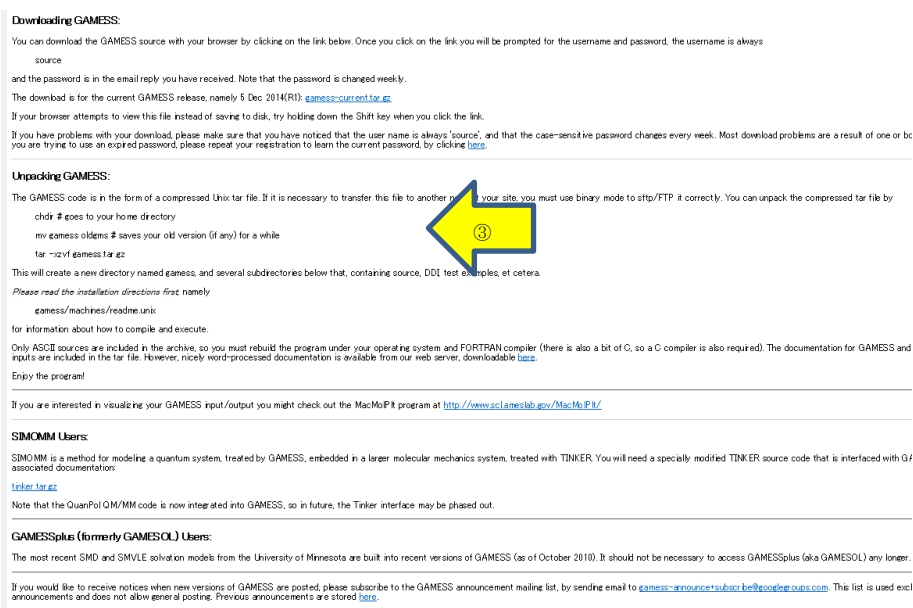
Request Accepted

Your download request has been accepted. E-mail has been sent to your address containing instructions for downloading the requested software. If these email(s) do not arrive in the next few days, contact games@si.msg.chem.iastate.edu Note that the email will have a from address of gamses@si.msg.chem.iastate.edu and a reply to address of gamses@si.msg.chem.iastate.edu. If you do not receive the email reply within the next hour please check your spam folder or otherwise verify that the email reply has not been filtered by your email software.

2. 以下のようなメールが届く（届くまでに数日かかることがある）。送られて来たメールの指示に従って GAMESS をダウンロードする。



- ① ダウンロードリンク先をクリックする。下図に示すサイトが立ち上がる。
② ダウンロードするためのパスワードをメモしておく。



- ③ **games-current.tar.gz** をクリックする。ユーザ名とパスワードの入力を要求されるので、以下のように入力する。

ユーザ名： source

パスワード：(メールに記述されたもの)

- ④ **games-current.tar.gz** を ftp など linux サーバーに転送する。
⑤ 圧縮ファイルを解凍する。

tar xzvf games-current.tar.gz

3. configuration を実行する。

- ① configuration を起動する。

```
$ cd ~/games
```

```
$ ./config
```

- ② コマンドを入力するための別ウインドウを立ち上げたのち、[return]を押す。

```
[winmostar_user@remote_server]$ ./config
This script asks a few questions, depending on your computer system,
to set up compiler names, libraries, message passing libraries,
and so forth.
You can quit at any time by pressing control-C, and then <return>.
```

Please open a second window by logging into your target machine, in case this script asks you to 'type' a command to learn something about your system software situation. All such extra questions will use the word 'type' to indicate it is a command for the other window.

After the new window is open, please hit <return> to go on.

- ③ マシン種を指定する（本例では、linux64 としている）。なお、マシンがわからない場合は、別ウインドウで'uname -a' コマンドにより調べる。

```
GAMESS can compile on the following 32 bit or 64 bit machines:
axp64 - Alpha chip, native compiler, running Tru64 or Linux
cray-xt - Cray's massively parallel system, running CNL
hpux32 - HP PA-RISC chips (old models only), running HP-UX
hpux64 - HP Intel or PA-RISC chips, running HP-UX
ibm32 - IBM (old models only), running AIX
ibm64 - IBM, Power3 chip or newer, running AIX or Linux
ibm64-sp - IBM SP parallel system, running AIX
ibm-bg - IBM Blue Gene (Q model), these are 64 bit systems
linux32 - Linux (any 32 bit distribution), for x86 (old systems only)
linux64 - Linux (any 64 bit distribution), for x86_64 or ia64 chips
        AMD/Intel chip Linux machines are sold by many companies
mac32 - Apple Mac, any chip, running OS X 10.4 or older
mac64 - Apple Mac, any chip, running OS X 10.5 or newer
sgi32 - Silicon Graphics Inc., MIPS chip only, running Irix
sgi64 - Silicon Graphics Inc., MIPS chip only, running Irix
sun32 - Sun ultraSPARC chips (old models only), running Solaris
sun64 - Sun ultraSPARC or Opteron chips, running Solaris
win32 - Windows 32-bit (Windows XP, Vista, 7, Compute Cluster, HPC Edition)
win64 - Windows 64-bit (Windows XP, Vista, 7, Compute Cluster, HPC Edition)
winazure - Windows Azure Cloud Platform running Windows 64-bit
type 'uname -a' to partially clarify your computer's flavor.
please enter your target machine name: linux64
```

- ④ ダウンロードした GAMESS セットアップファイル群が存在している directory を指定する。

```
Where is the GAMESS software on your system?
A typical response might be /u1/mike/games,
most probably the correct answer is /home/winmostar_user/games

GAMESS directory? [/home/winmostar_user/games] /home/winmostar_user/games
```

- ⑤ GAMESS の実行環境を作成する directory を指定する。

```
Setting up GAMESS compile and link for GMS_TARGET=linux64
GAMESS software is located at GMS_PATH=/home/winmostar_user/games

Please provide the name of the build location.
This may be the same location as the GAMESS directory.

GAMESS build directory? [/home/winmostar_user/games] /home/winmostar_user/games
```

- ⑥ GAMESS の実行バイナリモジュールのバージョン番号を指定する（任意でよい。ここでは 00 としている。この場合、実行バイナリモジュールのファイル名は「game00.x」となる。）

Please provide a version number for the GAMESS executable.
This will be used as the middle part of the binary's name,
for example: gamess.00.x

Version? [00] Please provide a version number for the GAMESS executable.
This will be used as the middle part of the binary's name,
for example: gamess.00.x

Version? 00

- ⑦ FORTRAN コンパイラーを指定する。ここでは無償利用が可能な `gfortran` を指定する。⁴

Linux offers many choices for FORTRAN compilers, including the GNU compiler suite's free compiler 'gfortran', usually included in any Linux distribution. If gfortran is not installed, it can be installed from your distribution media.

To check on installed GNU compilers, for RedHat/SUSE style Linux,
type 'rpm -aq | grep gcc' for both languages,
and for Debian/Ubuntu style Linux, it takes two commands
type 'dpkg -l | grep gcc'
type 'dpkg -l | grep gfortran'

There are also commercial compilers, namely Intel's 'ifort', and Portland Group's 'pgfortran', and Pathscale's 'pathf90'.
The last two are not common, and aren't as well tested.

type 'which gfortran' to look for GNU's gfortran (a good choice),
type 'which ifort' to look for Intel's compiler (a good choice),
type 'which pgfortran' to look for Portland Group's compiler,
type 'which pathf90' to look for Pathscale's compiler.
Please enter your choice of FORTRAN: `gfortran`

- ⑧ `gfortran` のバージョンを指定する。バージョンは別ウインドウで ``gfortran -v`` とタイプして確認する (`gfortran` の場合)。

`gfortran` is very robust, so this is a wise choice.

Please type 'gfortran -dumpversion' or else 'gfortran -v' to detect the version number of your gfortran.
This reply should be a string with at least two decimal points, such as 4.1.2 or 4.6.1, or maybe even hyphens like 4.4.2-12.
The reply may be labeled as a 'gcc' version, but it is really your gfortran version.

Please enter only the first decimal place, such as 4.6 or 4.8: `4.4`

- ⑨

Alas, your version of gfortran does not support REAL*16,
so relativistic integrals cannot use quadruple precision.
Other than this, everything will work properly.
hit <return> to continue to the math library setup.

<return>押す。

- ⑩ 数学ライブラリを指定する。無償利用したい場合は ATLAS の環境構築を実施した上で (`yum install atlas-devel` で入る) `atlas` を指定する。⁵ さらに ATLAS のインストールディレクトリ

⁴ `gfortran` 存在の有無は ``which gfortran`` とタイプして確かめられる。もしくは Intel compiler を購入し環境が整っている場合は `ifort` を指定してもよい。

⁵ ``ls /usr/lib64/atlas`` とタイプして正常にリストアップされれば ATLAS の環境が存在していることになる。

を指定する。

```
Linux distributions do not include a standard math library.

There are several reasonable add-on library choices,
    MKL from Intel           for 32 or 64 bit Linux (very fast)
    ACML from AMD           for 32 or 64 bit Linux (free)
    ATLAS from www.rpmfind.net for 32 or 64 bit Linux (free)
and one very unreasonable option, namely 'none', which will use
some slow FORTRAN routines supplied with GAMESS.  Choosing 'none'
will run MP2 jobs 2x slower, or CCSD(T) jobs 5x slower.

Some typical places (but not the only ones) to find math libraries are
Type 'ls /opt/intel/mkl'           to look for MKL
Type 'ls /opt/intel/Compiler/mkl'  to look for MKL
Type 'ls /opt/intel/composerxe/mkl' to look for MKL
Type 'ls -d /opt/acml*'            to look for ACML
Type 'ls -d /usr/local/acml*'      to look for ACML
Type 'ls /usr/lib64/atlas'         to look for Atlas

Enter your choice of 'mkl' or 'atlas' or 'acml' or 'none': atlas
Where is your Atlas math library installed?  A likely place is
    /usr/lib64/atlas
Please enter the Atlas subdirectory on your system: /usr/lib64/atlas
Math library 'atlas' will be taken from /usr/lib64/atlas

please hit <return> to compile the GAMESS source code activator
Please enter the Atlas subdirectory on your system: /usr/lib64/atlas
Math library 'atlas' will be taken from /usr/lib64/atlas
Source code activator was successfully compiled.
```

- ⑪ mpi 環境での並列実行を行わないので、sockets を設定する。⁶

```
please hit <return> to set up your network for Linux clusters.

If you have a slow network, like Gigabit Ethernet (GE), or
if you have so few nodes you won't run extensively in parallel, or
if you have no MPI library installed, or
if you want a fail-safe compile/link and easy execution,
    choose 'sockets'
to use good old reliable standard TCP/IP networking.

If you have an expensive but fast network like Infiniband (IB), and
if you have an MPI library correctly installed,
    choose 'mpi'.

communication library ('sockets' or 'mpi')? sockets
```

- ⑫ nVIDIA の GPU を使用しない場合は LIBCCHEM を使わないため no を指定する。⁷

```
64 bit Linux builds can attach a special LIBCCHEM code for fast
MP2 and CCSD(T) runs.  The LIBCCHEM code can utilize nVIDIA GPUs,
through the CUDA libraries, if GPUs are available.
Usage of LIBCCHEM requires installation of HDF5 I/O software as well.
GAMESS+LIBCCHEM binaries are unable to run most of GAMESS computations,
and are a bit harder to create due to the additional CUDA/HDF5 software.
Therefore, the first time you run 'config', the best answer is 'no'!
If you decide to try LIBCCHEM later, just run this 'config' again.

Do you want to try LIBCCHEM?  (yes/no): no
```

- ⑬ 下記メッセージが出力されれば configuration は正常終了している。

⁶ 複数 node を用いた並列実行環境の構築は、本マニュアルでは想定していない。個別に問い合わせのこと。

⁷ GPU を用いた実行環境の構築は、本マニュアルでは想定していない。個別に問い合わせのこと。

```
Your configuration for GAMESS compilation is now in
/home/winmostar_user/games/install.info
Now, please follow the directions in
/home/winmostar_user/games/machines/readme.unix
[winmostar_user@remote_server]$
```

configuration が正常終了すると 'install.info' に情報が書き込まれる。

4. Distributed Data Interface (DDI) 環境の構築

- ① 並列実行に必要となる DDI library のコンパイルを行う。

```
$ cd ~/games/ddi
```

```
$ ./compddi >& compddi.log &
```

- ② compddi.log を less コマンドなどで開き、再終行に `DDI compilation ended successfully` となっていることを確認する。

```
[winmostar_user@remote_server]$ less compddi.log
unset echo
Finished compiling: finalize_tasks.o

Compiling ddikick object: kickoff_local.o
gcc -DLINUX -m64 -O3 -fstrict-aliasing -I./ -Dgetarg=_gfortran_getarg_i4 -
Diargc=_gfortran_iargc -DDDI_SOC -DUSE_SYSV -DMAX_SMP_PROCS=32 -
DMAX_NODES=1024 -I./src -c kickoff_local.c
unset echo
Finished compiling: kickoff_local.o

Compiling ddikick object: kickoff_remote.o
gcc -DLINUX -m64 -O3 -fstrict-aliasing -I./ -Dgetarg=_gfortran_getarg_i4 -
Diargc=_gfortran_iargc -DDDI_SOC -DUSE_SYSV -DMAX_SMP_PROCS=32 -
DMAX_NODES=1024 -I./src -c kickoff_remote.c
unset echo
Finished compiling: kickoff_remote.o

Compiling ddikick object: kickoff_pbs.o
gcc -DLINUX -m64 -O3 -fstrict-aliasing -I./ -Dgetarg=_gfortran_getarg_i4 -
Diargc=_gfortran_iargc -DDDI_SOC -DUSE_SYSV -DMAX_SMP_PROCS=32 -
DMAX_NODES=1024 -I./src -c kickoff_pbs.c
unset echo
Finished compiling: kickoff_pbs.o

Linking DDI kickoff program, ddikick.x
gcc -DLINUX -m64 -O3 -fstrict-aliasing -I./ -Dgetarg=_gfortran_getarg_i4 -
Diargc=_gfortran_iargc -DDDI_SOC -DUSE_SYSV -DMAX_SMP_PROCS=32 -
DMAX_NODES=1024 -o ./ddikick.x ddikick.o libddikick.a -lpthread
unset echo

Don't forget to move ddikick.x up one directory level,
by a 'mv ddikick.x ..' command.

DDI compilation ended successfully.
(END)
```

- ③ games/ddi 配下に生成した ddikick.x を一つ上の階層に移動する。

```
mv ddikick.x ..
```

5. コンパイルを実行する。

```
$ cd ~/games
```

```
$ ./compall >& compall.log &
```

コンパイルにはしばらく時間がかかる（5分程度）。終了したら'compall.log'を注意深く閲覧しコンパイルエラーが出力されていないか確認のこと。

6. リンクを実施する

```
$ cd ~/gamess
```

```
$ ./lked gamess 00 >& lked.log &
```

実行ファイルの'gamess.00.x'が生成する。また'lked.log'を閲覧しエラーメッセージが出力されていないか確認のこと。⁸

7. gamess/ 配下の rungms の内容を2箇所修正する

①

```
set SCR=/scr/$USER
set USERSCR=/u1/$USER/scr
set GMSPATH=/u1/mike/gamess
```

↓

```
set SCR=$HOME/scr
set USERSCR=$HOME/scr
set GMSPATH=$HOME/gamess
```

8. .bashrc に次の一行を追加する。

```
export PATH=$HOME/gamess:$PATH
```

9. Winmostar から GAMESS をリモートジョブ実行する。

- ① [Winmostar GAMESS 基礎編チュートリアル](#)の内容に従い操作を進めるが、キーワード設定ウインドウで **Run** ボタンを押さずに **OK** ボタンを押す。
- ② その後、[ユーザマニュアルのリモートジョブの実行手順](#)に従って操作を行い、**Get All Files** ボタンを押してファイルを取得するところまで実行する。
- ③ 再び [Winmostar GAMESS 基礎編チュートリアル](#)の内容に戻り、ローカルジョブの時と全く同じ操作方法で結果解析を行う。

以上

⁸ ここで指定するバージョン（本例では 00）は、GAMESS 開発元の提供バージョンとは関係無い。GAMESS の提供バージョンは、output file に印字される。