

## iMolpro 2025.2.0

### ウェブページ

<https://github.com/molpro/iMolpro>

### バージョン

2025.2.0

### ビルド環境

-

### ビルドに必要なファイル

- Chooser.py.diff

```
--- Chooser.py.org    2025-10-29 14:48:32.515824489 +0900
+++ Chooser.py       2025-10-29 14:49:00.013448729 +0900
@@ -19,6 +19,7 @@
 from ProjectWindow import ProjectWindow
 from WindowManager import WindowManager
 from settings import settings, settings_edit
+from pysjef import recent_project

class PushButton(QPushButton):
@@ -209,7 +210,7 @@
     item.widget().setParent(None)
     self.recent_project_box.layout().addWidget(QLabel('Open a recently-used project:'), 0, QtCore.Qt.AlignLeft)
     for i in range(1, max_items):
-        f = pymolpro.recent_project('molpro', i)
+        f = recent_project('molpro', i)
     if f:
         button = RecentProjectButton(f, i, self)
         self.recent_project_box.layout().addWidget(button, -1, QtCore.Qt.AlignLeft)
```

- RecentMenu.py.diff

```
--- RecentMenu.py.org 2025-10-29 14:49:04.987378227 +0900
+++ RecentMenu.py     2025-10-29 14:49:35.470946141 +0900
@@ -3,6 +3,7 @@

import pymolpro
from PyQt5.QtWidgets import QMenu, QAction
+from pysjef import recent_project

class RecentMenuAction(QAction):
@@ -32,7 +33,7 @@
     self.recentProjects.clear()
     self.clear()
     for i in range(1, max_items+1):
-        f = pymolpro.recent_project('molpro', i)
+        f = recent_project('molpro', i)
     if f:
         action = RecentMenuAction(self, self.windowManager, f)
         self.recentProjects.append((f, action))
```

- iMolpro.py.diff

```
--- iMolpro.py.org    2025-10-30 10:03:03.408257254 +0900
+++ iMolpro.py        2025-10-30 10:03:21.086087361 +0900
```

```

@@ -10,6 +10,7 @@
import os
import platform
import logging
+import re

if __name__ == '__main__':

@@ -61,7 +62,11 @@
    if 'CONDA_PREFIX' not in os.environ:
        os.environ['CONDA_PREFIX'] = os.path.dirname(os.path.abspath(__file__))
    elif 'PATH' in os.environ and 'SHELL' in os.environ:
-        os.environ['PATH'] = os.popen(os.environ['SHELL'] + " -l -c 'echo $PATH'").read() + ':' + os.environ[
+        if re.search( 'csh', os.environ['SHELL'] ):
+            myopts = " -c 'echo $PATH'"
+        else:
+            myopts = " -l -c 'echo $PATH'"
+        os.environ['PATH'] = os.popen(os.environ['SHELL'] + myopts).read() + ':' + os.environ[
            'PATH'] # make PATH just as if running from shell
except Exception as e:
    msg = QMessageBox()

```

- build.sh.diff

```

--- build.sh.org      2025-10-29 14:24:13.679985528 +0900
+++ build.sh      2025-10-29 14:25:43.049738279 +0900
@@ -2,11 +2,11 @@

pkgbuild=1
#dmg=1
-#tar=1
+tar=1
sh=1

if [ -z "$NOCONDA" ]; then
-conda install -c conda-forge -c defaults -y --file=requirements.txt python=3.12 scipy=1.11 || exit 1
+#conda install -c conda-forge -c defaults -y --file=requirements.txt python=3.12 scipy=1.11 || exit 1
conda remove -y pubchempy
pip install -I https://github.com/molpro/PubChemPy/archive/refs/heads/master.zip
gem install --user-install -n~/bin fpm
@@ -51,8 +51,8 @@
--add-data Molpro_Logo_Molpro_Quantum_Chemistry_Software.png:. \
--add-data README.md:. \
--add-data doc:./doc \
+ --add-data ${PYMOLPRO}/molpro_input.json:./pymolpro/ \
--add-data "${versionfile}":. \
- --add-data $builddir/molpro:./molpro \
$pyinstaller_opt \
iMolpro.py || exit 1
sed -i -e '$d' iMolpro.spec
@@ -135,25 +135,4 @@
if [ ! -z "$tar" ]; then
tar cjf dist/iMolpro-"${descriptor}".tar.bz2 -C dist iMolpro
fi
- if [ ! -z "$sh" ]; then
- prefix="/usr"
- echo '#!/bin/sh' > ${builddir}/preinstall
- echo "more <<'EOF'" >> ${builddir}/preinstall
- cat ./Package-README.md ./Package-license.md | sed -e 's/^##* */' -e 's/[//g' -e 's\[ *(/, /g' -e 's()/)/@/g' -e 's()///g' -e 's/@()/g' -e 's/^*///g' >>
${builddir}/preinstall
- echo "EOF" >> ${builddir}/preinstall
- echo "echo 'Accept license[yN]?'" >> ${builddir}/preinstall
- echo "exec 0</dev/tty" >> ${builddir}/preinstall
- echo "read response" >> ${builddir}/preinstall
- echo 'if [ x"$response" != xy -a x"$response" != xY ]; then kill $$ ; fi' >> ${builddir}/preinstall

```

```

- echo '#!/bin/sh' > ${builddir}/postinstall
-# echo 'env' >> ${builddir}/postinstall
- echo "ln -sf ${prefix}/libexec/iMolpro/iMolpro ${prefix}/bin/iMolpro" >> ${builddir}/postinstall
- if [[ "$version" =~ "[0-9]*\.[0-9]*\.[0-9]*" ]]; then true ; else version="0.0.0" ; fi
- echo version=$version
- for type in deb rpm ; do
-   rm -f dist/iMolpro-"${descriptor}".${type}
-   dash='-'; if [ $type = rpm ]; then dash='_'; fi
-   fpm -s dir -C dist -t ${type} -p dist/iMolpro-"${descriptor}".${type} -v "${version}" -n iMolpro --prefix=${prefix}/libexec --before-install
${builddir}/preinstall --after-install ${builddir}/postinstall iMolpro
- done
- fi
fi

```

- environment.yml

```

name: imolpro_build
channels:
- conda-forge
dependencies:
- python=3.12
- pyqt>=5.15
- pyqtwebengine>=5.15
- pyinstaller>=6
- pymolpro>=1.13.2
- pubchempy>=1
- chemspipy>=2
- scipy<1.13
- git=2.45
- libboost=1.84.0
- pysjef>=1.40.0
- certifi>=2024.8.30

```

## ビルド手順

```

#!/bin/sh

VERSION=2025.2.0
#INSTALL_DIR=/apl/imolpro/${VERSION}

BASEDIR=/home/users/${USER}/Software/iMolpro/${VERSION}
MINIFORGE_VER=25.3.1-0
MINIFORGE=Miniforge3-${MINIFORGE_VER}-Linux-x86_64.sh
MINIFORGE_URL=https://github.com/conda-forge/miniforge/releases/download/${MINIFORGE_VER}/${MINIFORGE}
IMOLPRO_GIT=https://github.com/molpro/iMolpro.git
IMOLPRO_TAG=refs/tags/${VERSION}

WORKDIR=/gwork/users/${USER}
WORKDIR_TMP=${WORKDIR}/tmp
WORKNAME=iMolpro

PATCH=${BASEDIR}/build.sh.diff
CONDAENV_YML=environment.yml
CONDAENV=${BASEDIR}/${CONDAENV_YML}
# https://github.com/molpro/iMolpro/issues/310
PATCH1=${BASEDIR}/Chooser.py.diff
PATCH2=${BASEDIR}/RecentMenu.py.diff
PATCH3=${BASEDIR}/iMolpro.py.diff

# -----
umask 0022
export LANG=C
export LC_ALL=C

module -s purge

```

```

cd ${WORKDIR}
# /tmp is not only for you!
export TMPDIR=${WORKDIR}_TMP}
mkdir -p ${WORKDIR}_TMP}

rm -f ${MINIFORGE}
if [ -d ${WORKDIR}/miniforge3 ]; then
  mv ${WORKDIR}/miniforge3 ${WORKDIR}/miniforge3-erase
  rm -rf ${WORKDIR}/miniforge3-erase &
fi

wget ${MINIFORGE_URL}
sh ${MINIFORGE} -b -f -p ${WORKDIR}/miniforge3
eval "$( ${WORKDIR}/miniforge3/bin/conda shell.bash hook )"

if [ -d ${WORKNAME} ]; then
  mv ${WORKNAME} ${WORKNAME}-erase
  rm -rf ${WORKNAME}-erase &
fi
git clone ${IMOLPRO_GIT} -b ${VERSION} ${WORKNAME}
cd ${WORKNAME}

patch -p0 < $PATCH
patch -p0 < $PATCH1
patch -p0 < $PATCH2
patch -p0 < $PATCH3
cp ${CONDAENV} ${CONDAENV_YML}

# create imolpro_build env
conda env create -f ${CONDAENV_YML}
conda activate imolpro_build

export PYMOLPRO=${WORKDIR}/miniforge3/envs/imolpro_build/lib/python3.12/site-packages/pymolpro
sh ./build.sh
# install dist/iMolpro-***.tar.bz2 into $INSTALL_DIR manually

```

上記実行し、iMolpro-2025.2.0-dirty.Linux.x86\_64.tar.bz2 を展開後、iMolpro/\_internal/libstdc++.so.6 を手動で置き換え

```
# cp /gwork/users/${USER}/miniforge3/envs/imolpro_build/lib/libstdc++.so.6.0.34 iMolpro/_internal/libstdc++.so.6
```

## メモ

- Chooser.py.diff と RecentMenu.py.diff は <https://github.com/molpro/iMolpro/issues/310> 対応のため
  - <https://github.com/pjknowles/iMolpro/commit/f2f5865bb0b6db2bcc040f0264295ad8ecb3ffbb>
- iMolpro.py.diff はログインシェルが csh の時起動時のエラーを回避するため
- requirements.txt を使って環境を作ると(conda install --file=requirements.txt)エラーが発生するため、environment.yml で env を作成している(conda env create)。
  - なお、requirements.txt を使わずに必要なパッケージを列挙してインストールする場合は成功する。--file で指定した場合とは依存解決の方法が異なっている？
- pymolpro/molpro\_input.json が起動時に必要なようだったので、追加している。
- rpm や deb ではあまり都合が良くないため、初めから tar のみで作成。
- libstdc++.so.6 を強制的に置き換えない場合は GLIBCXX 関連のエラーが発生した。詳細については未検証。
  - 環境依存の問題だと思われる。
- miniforge 環境はビルド時のみ必須。実行時は不要。
- molpro コマンドが PATH で見つければ連携可能。
  - ただし、RCCS の Open OnDemand デスクトップ環境ではパーミッションの関係で連携不可。
  - X11 転送でご利用ください。
- backend の設定を行っていないため、直接ジョブを投入することはできません。