Version 1.0 of the Q5Cost library

Anthony Scemama, Antonio Monari, Celestino Angeli, Stefano Borini, Stefano Evangelisti, and Elda Rossi

LCPQ-IRSAMC CNRS, Toulouse (France)
scemama@irsamc.ups-tlse.fr

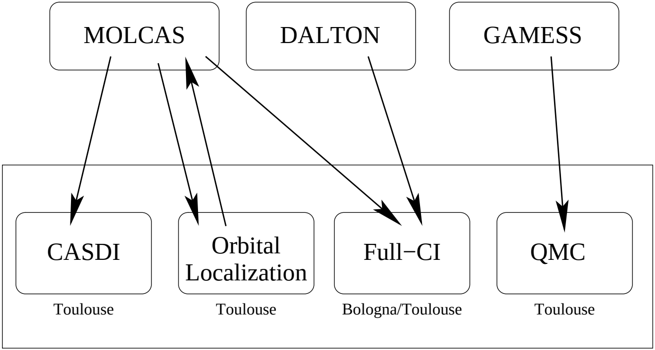
30 June 2008

Presentation of the talk

1. Context
   - Motivations
2. Q5Cost
   - The File format
   - The Q5Cost API
   - The Q5Cost Package
3. Conclusions

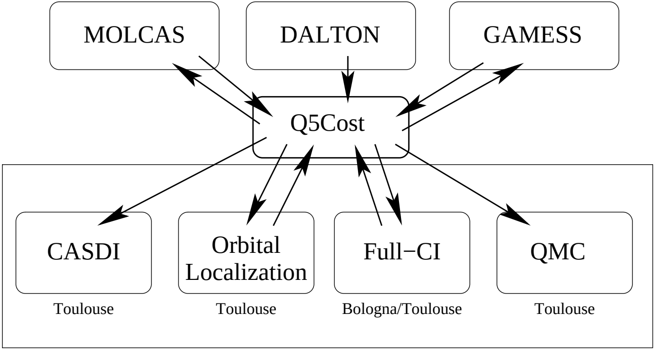
The need for a common file format

Situation in Toulouse a few years ago...



The need for a common file format

Situation in Toulouse now



Presentation of the talk

1. Context
   - Motivations
2. Q5Cost
   - The File format
   - The Q5Cost API
   - The Q5Cost Package
3. Conclusions

Motivations

- Our codes read files generated from other codes which change from one version to the other.
- The structures of these files are difficult to understand
- We should have as many interfaces as pairs of codes $(N(N-1))$
- With a common file format, we need as many interfaces as codes (N)

Motivations

- Our codes read files generated from other codes which change from one version to the other.
- The structures of these files are difficult to understand
- We should have as many interfaces as pairs of codes $(N(N-1))$
- With a common file format, we need as many interfaces as codes (N)

Motivations

- Our codes read files generated from other codes which change from one version to the other.
- The structures of these files are difficult to understand
- We should have as many interfaces as pairs of codes ($N(N-1)$)
- With a common file format, we need as many interfaces as codes (N)

Motivations

- Our codes read files generated from other codes which change from one version to the other.
- The structures of these files are difficult to understand
- We should have as many interfaces as pairs of codes $(N(N-1))$
- With a common file format, we need as many interfaces as codes (N)

Goal

For quantum chemistry codes, we need :

- Structured files (easily understandable)
- Machine-independent files
- Compressed files
- Easy access to the data in Fortran

Goal

For quantum chemistry codes, we need :

- Structured files (easily understandable)
- Machine-independent files
- Compressed files
- Easy access to the data in Fortran

Goal

For quantum chemistry codes, we need :

- Structured files (easily understandable)
- Machine-independent files
- Compressed files
- Easy access to the data in Fortran

Goal

For quantum chemistry codes, we need :

- Structured files (easily understandable)
- Machine-independent files
- Compressed files
- Easy access to the data in Fortran

Presentation of the talk

1 Context
- Motivations

2 Q5Cost
- The File format
- The Q5Cost API
- The Q5Cost Package

3 Conclusions

What is Q5Cost ?

- A file format designed for quantum chemistry
- A Fortran API to access the data
- Some tools for manipulating these files.

What is Q5Cost ?

- A file format designed for quantum chemistry
- A Fortran API to access the data
- Some tools for manipulating these files.

What is Q5Cost ?

- A file format designed for quantum chemistry
- A Fortran API to access the data
- Some tools for manipulating these files.

Presentation of the talk

1. Context
   - Motivations
2. Q5Cost
   - The File format
   - The Q5Cost API
   - The Q5Cost Package
3. Conclusions

The file format

Q5Cost use HDF5 technology

- Binary data
- Structured in a way similar to a filesystem
- Machine-independent
- Compression (with zlib)
- Very good I/O performance

The file format

Q5Cost use HDF5 technology

- Binary data
- Structured in a way similar to a filesystem
- Machine-independent
- Compression (with zlib)
- Very good I/O performance

The file format

Q5Cost use HDF5 technology

- Binary data
- Structured in a way similar to a filesystem
- Machine-independent
- Compression (with zlib)
- Very good I/O performance

The file format

Q5Cost use HDF5 technology

- Binary data
- Structured in a way similar to a filesystem
- Machine-independent
- Compression (with zlib)
- Very good I/O performance

The file format

Q5Cost use HDF5 technology

- Binary data
- Structured in a way similar to a filesystem
- Machine-independent
- Compression (with zlib)
- Very good I/O performance

The file format

Q5Cost use HDF5 technology

- Binary data
- Structured in a way similar to a filesystem
- Machine-independent
- Compression (with zlib)
- Very good I/O performance

The file format

5 Main groups :

- System (the molecular system)
  - molecular geometry, symmetry
  - nuclear repulsion energy, number of α and β electrons
- Basis (the basis set information)
  - Coordinate system (spherical/cartesian)
  - Gaussian contractions (exponents, coefficients,. . .)
- AO (the atomic orbitals information)
  - Symmetry-adatped LCAO on which the MOs are expressed
  - 1- and 2-electron integrals, overlap matrix
- MO (the molecular orbitals information)
  - Orbital energies, occupation numbers, symmetry, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - MO coefficients
- WF (the wave function information)
  - Determinants and coefficients
  - Total energy

The file format

5 Main groups :

- System (the molecular system)
  - molecular geometry, symmetry
  - nuclear repulsion energy, number of α and β electrons
- Basis (the basis set information)
  - Coordinate system (spherical/cartesian)
  - Gaussian contractions (exponents, coefficients,. . .)
- AO (the atomic orbitals information)
  - Symmetry-adatped LCAO on which the MOs are expressed
  - 1- and 2-electron integrals, overlap matrix
- MO (the molecular orbitals information)
  - Orbital energies, occupation numbers, symmetry, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - MO coefficients
- WF (the wave function information)
  - Determinants and coefficients
  - Total energy

The file format

5 Main groups :

- System (the molecular system)
  - molecular geometry, symmetry
  - nuclear repulsion energy, number of α and β electrons
- Basis (the basis set information)
  - Coordinate system (spherical/cartesian)
  - Gaussian contractions (exponents, coefficients,. . .)
- AO (the atomic orbitals information)
  - Symmetry-adatped LCAO on which the MOs are expressed
  - 1- and 2-electron integrals, overlap matrix
- MO (the molecular orbitals information)
  - Orbital energies, occupation numbers, symmetry, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - MO coefficients
- WF (the wave function information)
  - Determinants and coefficients
  - Total energy

The file format

5 Main groups :

- System (the molecular system)
  - molecular geometry, symmetry
  - nuclear repulsion energy, number of α and β electrons
- Basis (the basis set information)
  - Coordinate system (spherical/cartesian)
  - Gaussian contractions (exponents, coefficients,. . .)
- AO (the atomic orbitals information)
  - Symmetry-adatped LCAO on which the MOs are expressed
  - 1- and 2-electron integrals, overlap matrix
- MO (the molecular orbitals information)
  - Orbital energies, occupation numbers, symmetry, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - MO coefficients
- WF (the wave function information)
  - Determinants and coefficients
  - Total energy

The file format

5 Main groups :

- System (the molecular system)
  - molecular geometry, symmetry
  - nuclear repulsion energy, number of α and β electrons
- Basis (the basis set information)
  - Coordinate system (spherical/cartesian)
  - Gaussian contractions (exponents, coefficients,...)
- AO (the atomic orbitals information)
  - Symmetry-adatped LCAO on which the MOs are expressed
  - 1- and 2-electron integrals, overlap matrix
- MO (the molecular orbitals information)
  - Orbital energies, occupation numbers, symmetry, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - MO coefficients
- WF (the wave function information)
  - Determinants and coefficients
  - Total energy

The file format

5 Main groups :

- System (the molecular system)
  - molecular geometry, symmetry
  - nuclear repulsion energy, number of α and β electrons
- Basis (the basis set information)
  - Coordinate system (spherical/cartesian)
  - Gaussian contractions (exponents, coefficients,. . .)
- AO (the atomic orbitals information)
  - Symmetry-adatped LCAO on which the MOs are expressed
  - 1- and 2-electron integrals, overlap matrix
- MO (the molecular orbitals information)
  - Orbital energies, occupation numbers, symmetry, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - MO coefficients
- WF (the wave function information)
  - Determinants and coefficients
  - Total energy

Presentation of the talk

1. Context
   - Motivations
2. Q5Cost
   - The File format
   - The Q5Cost API
   - The Q5Cost Package
3. Conclusions

The Q5Cost API

- A set of fortran routines which encapsualte the HDF5 library calls → The users don't need to know HDF5
- All routine names can be calculated. Example :

```
Q5Cost_System_get_num_alpha(file_id,num_alpha,error)
--1--- --2--- -3- ----4---- ---5--- ----6---- --7--
```

  - 1) all routine names start with "Q5Cost"
  - 2) the group which contains the data
  - 3) set/get the data
  - 4) the name of the data to reach
  - 5) the ID of the file to use
  - 6) the variable in which to put the data (or the variable to write)
  - 7) an error code which is 0 upon success

The Q5Cost API

- A set of fortran routines which encapsualte the HDF5 library calls → The users don't need to know HDF5
- All routine names can be calculated. Example :

```
Q5Cost_System_get_num_alpha(file_id,num_alpha,error)
--1--- --2--- -3- ----4---- ---5--- ----6---- --7--
```

  - 1) all routine names start with "Q5Cost"
  - 2) the group which contains the data
  - 3) set/get the data
  - 4) the name of the data to reach
  - 5) the ID of the file to use
  - 6) the variable in which to put the data (or the variable to write)
  - 7) an error code which is 0 upon success

The Q5Cost API

- A set of fortran routines which encapsualte the HDF5 library calls → The users don't need to know HDF5
- All routine names can be calculated. Example :

```
Q5Cost_System_get_num_alpha(file_id,num_alpha,error)
--1--- --2--- -3- ----4---- ---5--- ----6---- --7--
```

  - 1) all routine names start with "Q5Cost"
  - 2) the group which contains the data
  - 3) set/get the data
  - 4) the name of the data to reach
  - 5) the ID of the file to use
  - 6) the variable in which to put the data (or the variable to write)
  - 7) an error code which is 0 upon success

The Q5Cost API

- A set of fortran routines which encapsualte the HDF5 library calls → The users don't need to know HDF5
- All routine names can be calculated. Example :

```
Q5Cost_System_get_num_alpha(file_id,num_alpha,error)
--1--- --2--- -3- ----4---- ---5--- ----6---- --7--
```

  - 1) all routine names start with "Q5Cost"
  - 2) the group which contains the data
  - 3) set/get the data
  - 4) the name of the data to reach
  - 5) the ID of the file to use
  - 6) the variable in which to put the data (or the variable to write)
  - 7) an error code which is 0 upon success

The Q5Cost API

- A set of fortran routines which encapsualte the HDF5 library calls → The users don't need to know HDF5
- All routine names can be calculated. Example :

```
Q5Cost_System_get_num_alpha(file_id,num_alpha,error)
--1--- --2--- -3- ----4---- ---5--- ----6---- --7--
```

  - 1) all routine names start with "Q5Cost"
  - 2) the group which contains the data
  - 3) set/get the data
  - 4) the name of the data to reach
  - 5) the ID of the file to use
  - 6) the variable in which to put the data (or the variable to write)
  - 7) an error code which is 0 upon success

The Q5Cost API

- A set of fortran routines which encapsualte the HDF5 library calls → The users don't need to know HDF5
- All routine names can be calculated. Example :

```
Q5Cost_System_get_num_alpha(file_id,num_alpha,error)
--1--- --2--- -3- ----4---- ---5--- ----6---- --7--
```

  - 1) all routine names start with "Q5Cost"
  - 2) the group which contains the data
  - 3) set/get the data
  - 4) the name of the data to reach
  - 5) the ID of the file to use
  - 6) the variable in which to put the data (or the variable to write)
  - 7) an error code which is 0 upon success

The Q5Cost API

- A set of fortran routines which encapsualte the HDF5 library calls → The users don't need to know HDF5
- All routine names can be calculated. Example :

```
Q5Cost_System_get_num_alpha(file_id,num_alpha,error)
--1--- --2--- -3- ----4---- ---5--- ----6---- --7--
```

  - 1) all routine names start with "Q5Cost"
  - 2) the group which contains the data
  - 3) set/get the data
  - 4) the name of the data to reach
  - 5) the ID of the file to use
  - 6) the variable in which to put the data (or the variable to write)
  - 7) an error code which is 0 upon success

The Q5Cost API

- A set of fortran routines which encapsualte the HDF5 library calls → The users don't need to know HDF5
- All routine names can be calculated. Example :

```
Q5Cost_System_get_num_alpha(file_id,num_alpha,error)
--1--- --2--- -3- ----4---- ---5--- ----6---- --7--
```

  - 1) all routine names start with "Q5Cost"
  - 2) the group which contains the data
  - 3) set/get the data
  - 4) the name of the data to reach
  - 5) the ID of the file to use
  - 6) the variable in which to put the data (or the variable to write)
  - 7) an error code which is 0 upon success

The Q5Cost API

- A set of fortran routines which encapsualte the HDF5 library calls → The users don't need to know HDF5
- All routine names can be calculated. Example :

```
Q5Cost_System_get_num_alpha(file_id,num_alpha,error)
--1--- --2--- -3- ----4---- ---5--- ----6---- --7--
```

  - 1) all routine names start with "Q5Cost"
  - 2) the group which contains the data
  - 3) set/get the data
  - 4) the name of the data to reach
  - 5) the ID of the file to use
  - 6) the variable in which to put the data (or the variable to write)
  - 7) an error code which is 0 upon success

Presentation of the talk

1. Context
   - Motivations
2. Q5Cost
   - The File format
   - The Q5Cost API
   - The Q5Cost Package
3. Conclusions

The package

- ./configure
- library and include files
- tests
- documentation (file format and API)
- auto-generated F77, C++ and Python bindings
- q5edit (interactive)
- q5dump

The package

- ./configure
- library and include files
- tests
- documentation (file format and API)
- auto-generated F77, C++ and Python bindings
- q5edit (interactive)
- q5dump

The package

- ./configure
- library and include files
- tests
- documentation (file format and API)
- auto-generated F77, C++ and Python bindings
- q5edit (interactive)
- q5dump

The package

- ./configure
- library and include files
- tests
- documentation (file format and API)
- auto-generated F77, C++ and Python bindings
- q5edit (interactive)
- q5dump

The package

- ./configure
- library and include files
- tests
- documentation (file format and API)
- auto-generated F77, C++ and Python bindings
- q5edit (interactive)
- q5dump

The package

- ./configure
- library and include files
- tests
- documentation (file format and API)
- auto-generated F77, C++ and Python bindings
- q5edit (interactive)
- q5dump

The package

- ./configure
- library and include files
- tests
- documentation (file format and API)
- auto-generated F77, C++ and Python bindings
- q5edit (interactive)
- q5dump

Presentation of the talk

1. Context
   - Motivations
2. Q5Cost
   - The File format
   - The Q5Cost API
   - The Q5Cost Package
3. Conclusions

Some Code communications

- GAMESS
- Dalton
- Molcas
- Full-CI (Bologna)
- Orbital localization (Toulouse)
- CASDI (Toulouse)
- QMC=Chem (Toulouse)
- Columbus (Vienna)
- Aces II (Budapest)
- Molekel

Acknowledgments

Q5Cost people :
E. Rossi, A. Emerson, G. Bendazzoli, A. Monari, S. Evangelisti, D. Maynau, M. Kállay, K. Baldridge, K. Ruud, R. Cimiraglia, C. Angeli, C. Pastore, V. Vallet , J.-P. Flament, A. Scemama, J. Sanchez-Marin , P. Szalay, A. Tajti, S. Borini, and more...
This project is partially funded by the COST-D37.