

QCMATRIX API (Fortran part)

```
type, public :: QcMat
  private
  type(C_PTR) :: c_mat = C_NULL_PTR
end type QcMat
```

```
function QcMatGetExternalMat(A, ..., A_ext) result(ierr)
  integer :: ierr
  type(QcMat), intent(in) :: A
  ... ..
  type(LANG_F_MATRIX), pointer, intent(inout) :: A_ext
  type(C_PTR) c_A
  ierr = f03_api_QcMatGetAdapterMat(A%c_mat, ..., c_A)
  call c_f_pointer(c_A, A_ext)
end function QcMatGetExternalMat
```

QCMATRIX API (C part)

```
typedef struct {
  QInt dim_block;
  RealMat **blocks;
} QcMat;
```

```
QErrorCode f03_api_QcMatGetAdapterMat(QVoid **A,
                                     ...,
                                     QVoid **f_A)

  QcMat *c_A;
  RealMat *A_adapter;
  QErrorCode ierr;
  c_A = (QcMat *)(*A);
  ierr = QcMatGetAdapterMat(c_A, ..., &A_adapter);
  ierr = AdapterMatGetExternalMat(A_adapter, f_A);
  A_adapter = NULL;
  return ierr;
}
```

```
typedef struct {
  QVoid *f03_mat;
  QBool external_mat;
} RealMat;
```

QCMATRIX Fortran Adapter (C part)

```
QErrorCode AdapterMatGetExternalMat(RealMat *A, QVoid **f_A)
  *f_A = (QVoid *)A->f03_mat;
  return QSUCCESS;
}
```