

# AO Mobile Media Architecture (Release 4)

## COMPONENT: PhotoCopy

### Provided Interface: ManageCopy

```
viewPhoto(); // "View"
copyPhoto(); // "Copy"
saveCopyPhoto(); // "Save Photo"
```

### Provided Interface: InitPhotoView

```
setPhotoScreen();
```

## COMPONENT: FavouritePhoto

### Provided Interface: ManageFavourite

```
setFavouriteCommand(); // "Set Favourite" //PhotoListScreen.ControlPhoto
setFavourite(); // "Set Favourite" //BaseController.ControlPhoto
viewFavouriteCommand(); // "View Favourites" //PhotoListScreen.ControlPhoto
viewFavourite(); // "View Favourites" //BaseController.ControlPhoto
append(); //BaseController.ManagePhotoList
```

### Required Interface: ManageImage

```
updateImage(ImageData image);
ImageData getImageInfo(String imageName) throws ImageNotFoundException, NullAlbumDataReference;
```

## COMPONENT: PhotoSorting

### Provided Interface: CountPhotoView

```
viewPhoto(); // "View" //BaseControler.ControlPhoto
```

### Provided Interface: SortPhoto

```
sortPhotosCommand(); // "Sort by Views" //PhotoListScreen.ControlPhoto
sortPhotos(); // "Sort by Views" //BaseController.ControlPhoto
append(); //BaseController.ManagePhotoList
```

### Required Interface: ManageImage

```
updateImage(ImageData image);
ImageData getImageInfo(String imageName) throws ImageNotFoundException, NullAlbumDataReference;
```

## COMPONENT: ExceptionHandling

### Provided Interface: HandleExceptions

```
ImageData[] getImages(String recordName) //AlbumData.ManagePhotoInfo
deleteImage(String imageName, String storeName); //AlbumData.ManagePhotoInfo
resetImageRecordStore();//AlbumData.ManagePhotoInfo
ImageData getImageInfo(String imageName); //ImageAcessor.PersistPhoto
loadAlbums(); // ImageAcessor.PersistPhoto
addImageData(String photoname, String path, String albumname); //ImageAcessor.PersistPhoto
ImageData[] loadImageDataFromRMS(String recordName); //ImageAcessor.PersistPhoto
boolean deleteSingleImageFromRMS(String storeName, String imageName); //ImageAcessor.PersistPhoto
deletePhotoAlbum(String albumName); //ImageAcessor.PersistPhoto
```

### COMPONENT: AlbumListScreen

#### Required Interface: ControlAlbum

```
newPhotoAlbum(); //"New Photo Album"
deleteAlbum(); //"Delete Album"
```

#### Required Interface: ControlPhotoList

```
viewAlbum(); // "Select"
```

### Provided Interface: ManageAlbumList

```
append(String albumName);
delete(int i);
repaintListAlbum(String[] names);
```

### COMPONENT: AddPhotoToAlbumScreen

#### Required Interface: ControlPhoto

```
saveNewPhoto(); //"Save Add Photo"
cancel();
```

### Provided Interface: ManagePhoto

```
String getPhotoName();
String getPath();
setLabePhotoPath(String label);
setPhotoName(String photoName);
```

### COMPONENT: NewLabelScreen

#### Required Interface: ControlLabel

```
saveLabel(); // "Save"
cancel();
```

### Required Interface: ControlAlbum

```
saveNewPhotoAlbum(); // "Save"  
cancel();
```

### Provided Interface: ManageLabel

```
String getLabelName();  
int getFormType();
```

## COMPONENT: PhotoListScreen

### Required Interface: ControlPhoto

```
viewPhoto(); // "View"  
newPhoto(); // "Add"  
deletePhoto(); // "Delete"  
editLabel(); // "Edit Label"  
back(); // "Back"
```

### Provided Interface: ManagePhotoList

```
append(String imageLabel);
```

## COMPONENT: PhotoViewScreen

### Required Interface: ControlCopy

```
back(); // "Back"
```

## COMPONENT: BaseController

### Required Interface: ManageAlbumList

```
append(String albumName);  
repaintListAlbum(String[] names);
```

### Required Interface: ManagePhotoInfo

```
String[] getAlbumNames();
```

### Provided Interface: HandleCommand

```
postCommand(Command command);
```

## COMPONENT: PhotoController

### Provided Interface: ControlPhoto

```
viewPhoto(); // "View"
```

```
newPhoto(); // "Add"  
saveNewPhoto(); // "Save Add Photo"  
deletePhoto(); // "Delete"  
editLabel(); // "Edit Label"  
saveLabel(); // "Save"  
cancel(); // "Cancel"  
back(); // "Back"
```

### Provided Interface: ManageImage

```
updateImage(ImageData image);  
showImageList(String recordName);  
ImageData getImageInfo(String imageName) throws ImageNotFoundException, NullAlbumDataReference;
```

### Required Interface: ManageLabel

```
String getLabelName();
```

### Required Interface: ManagePhoto

```
String getPhotoName();  
String getPath();
```

### Required Interface: ManagePhotoInfo

```
addNewPhotoToAlbum(String label, String path, String album) throws InvalidImageDataException, PersistenceMechanismException;  
deleteImage(String imageName, String storeName) throws PersistenceMechanismException, ImageNotFoundException;  
Image getImageFromRecordStore(String recordStore, String imageName) throws ImageNotFoundException, PersistenceMechanismException;  
ImageData getImageInfo(String imageName) throws ImageNotFoundException, NullAlbumDataReference;  
boolean updateImageInfo(ImageData oldData, ImageData newData) throws InvalidImageDataException, PersistenceMechanismException;
```

### Provided Interface: HandleCommand

```
postCommand(Command command);
```

### Required Interface: HandleCommand

```
postCommand(Command command);
```

## COMPONENT: AlbumController

### Provided Interface: ControlAlbum

```
newPhotoAlbum(); // "New Photo Album"  
deleteAlbum(); // "Delete Album"  
saveNewPhotoAlbum(); // "Save"
```

### Required Interface: ManageAlbumList

```
append(String albumName); (already exists)  
delete();  
repaintListAlbum(String[] names);
```

### Required Interface: ManageLabel

```
String getLabelName();
int getFormType();
```

### Required Interface: ManagePhotoInfo

```
createNewPhotoAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deletePhotoAlbum(String albumName) throws PersistenceMechanismException;
String[] getAlbumNames();
resetImageData() throws PersistenceMechanismException;
```

### Provided Interface: HandleCommand

```
postCommand(Command command);
```

### Required Interface: HandleCommand

```
postCommand(Command command);
```

## COMPONENT: PhotoViewController

### Provided Interface: ControlCopy

```
back(); // "Back"
copyPhoto(); // "Copy"
saveCopyPhoto(); // "Save Photo"
```

### Provided Interface: InitPhotoView

```
setPhotoScreen();
```

### Required Interface: ManagePhoto

```
String getPhotoName();
String getPath();
setLabePhotoPath(String label);
setPhotoName(String photoName);
```

### Required Interface: ManageImage

```
showImageList(String recordName);
```

### Required Interface: ManagePhotoInfo

```
ImageData getImageInfo(String imageName) throws ImageNotFoundException, NullAlbumDataReference;
copyPhotoToAlbum(String photoname, ImageData imageData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
```

### Required Interface: HandleCommand

```
postCommand(Command command);
```

## COMPONENT: PhotoListController

### Provided Interface: ControlPhotoList

```
viewAlbum(); // "Select"
```

### Required Interface: ManagePhotoList

```
append(String imageLabel);
```

### Required Interface: ManagePhotoInfo

```
ImageData[] getImages(String recordName) throws UnavailablePhotoAlbumException;
```

### Provided Interface: HandleCommand

```
postCommand(Command command);
```

### Required Interface: HandleCommand

```
postCommand(Command command);
```

## COMPONENT: AlbumData

### Provided Interface: ManagePhotoInfo

```
copyPhotoToAlbum(String photoname, ImageData imageData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addNewPhotoToAlbum(String label, String path, String album) throws InvalidImageDataException, PersistenceMechanismException;
createNewPhotoAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deleteImage(String imageName, String storeName) throws PersistenceMechanismException, ImageNotFoundException;
deletePhotoAlbum(String albumName) throws PersistenceMechanismException;
String[] getAlbumNames();
Image getImageFromRecordStore(String recordStore, String imageName) throws ImageNotFoundException, PersistenceMechanismException;
Hashtable getImageInfoTable();
ImageData[] getImages(String recordName) throws UnavailablePhotoAlbumException;
resetImageData() throws PersistenceMechanismException;
ImageData getImageInfo(String imageName) throws ImageNotFoundException, NullAlbumDataReference;
boolean updateImageInfo(ImageData oldData, ImageData newData) throws InvalidImageDataException, PersistenceMechanismException;
```

### Required Interface: PersistPhoto

```
addImageData(String photoname, ImageData imageData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addImageData(String photoname, String path, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
createNewPhotoAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deletePhotoAlbum(String albumName) throws PersistenceMechanismException;
boolean deleteSingleImageFromRMS(String storeName, String imageName) throws PersistenceMechanismException, ImageNotFoundException,
NullAlbumDataReference;
String[] getAlbumNames();
ImageData getImageInfo(String imageName) throws ImageNotFoundException, NullAlbumDataReference;
loadAlbums() throws InvalidImageDataException, PersistenceMechanismException;
ImageData[] loadImageDataFromRMS(String recordName) throws PersistenceMechanismException, InvalidImageDataException;
Image loadSingleImageFromRMS(String recordName, String imageName, int recordId) throws PersistenceMechanismException;
resetImageRecordStore() throws InvalidImageDataException, PersistenceMechanismException;
```

```
boolean updateImageInfo(ImageData oldData, ImageData newData) throws InvalidImageDataException, PersistenceMechanismException;
```

## COMPONENT: ImageAcessor

### Provided Interface: PersistPhoto

```
addImageData(String photoname, ImageData imageData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addImageData(String photoname, String path, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
createNewPhotoAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deletePhotoAlbum(String albumName) throws PersistenceMechanismException;
boolean deleteSingleImageFromRMS(String storeName, String imageName) throws PersistenceMechanismException, ImageNotFoundException,
NullAlbumDataReference;
String[] getAlbumNames();
ImageData getImageInfo(String imageName) throws ImageNotFoundException, NullAlbumDataReference;
loadAlbums() throws InvalidImageDataException, PersistenceMechanismException;
ImageData[] loadImageDataFromRMS(String recordName) throws PersistenceMechanismException, InvalidImageDataException;
Image loadSingleImageFromRMS(String recordName, String imageName,      int recordId) throws PersistenceMechanismException;
resetImageRecordStore() throws InvalidImageDataException, PersistenceMechanismException;
boolean updateImageInfo(ImageData oldData, ImageData newData) throws InvalidImageDataException, PersistenceMechanismException;
```

### Required Interface: ManagePhotoInfo

```
Hashtable getImageInfoTable();
```