



# S2MRCPBG

## S2QL USER MANUAL

---

### MRCPBG QUICKLOOK USER MANUAL

	Name	Company	Date	Signature
Written by:	Grégory Cazanave	Magellium	16/01/2012	
Verified by:	Grégory Cazanave	Magellium	16/01/2012	
Approved by:	Philippe Martimort	ESA/ESTEC		

Document reference:	S2MRCPBG-MU-012-MAG
Issue.revision:	2.0
Issue date:	16/01/2012
Customer:	ESA/ESTEC
ITT reference:	4000103130/11/NL/JA

## Distribution list

	Name	Company	Copies
<b>Recipients:</b>	Philippe Martimort	ESA/ESTEC	1 digital
	Riccardo Duca	ESA/ESTEC	1 digital
	Olivier Colin	ESA/ESRIN	1 digital
	Ferran Gascon	ESA/ESRIN	1 digital
	Borja Lopez Fernandez	ESA/ESRIN	1 digital
	Enrico Cadau	ESA/ESRIN	1 digital
<b>Internal copy:</b>	Customer folder	Magellium	1
	Project leader	Magellium	1 (electronic)
	Development team	Magellium	1 (electronic)

## Document change record

Issue	Rev.	Date	Nature of change	Observations
1	0	18/11/2011	Document creation for S2MCPBG, taking account of KO presentation, PM1, PM2, PM3, PM4, Change Request 1.	
2	0	16/01/2012	Update with the description of the encapsulator program of the QL function.	

## Table of Contents

<b>1. Introduction.....</b>	<b>4</b>
1.1. Scope of this Document.....	4
<b>2. Installation procedure.....</b>	<b>5</b>
2.1. Objective.....	5
2.2. Software and hardware environment.....	5
2.3. Installing the software.....	5
<b>3. Using the QuickLook function.....</b>	<b>7</b>
3.1. Objective.....	7
3.2. Operating mode .....	7
<b>4. Using the encapsulation program.....</b>	<b>9</b>
4.1. Objective.....	9
4.2. Operating mode .....	9

---

# 1. Introduction

---

## 1.1. Scope of this Document

The main purpose of this document is to guide users through the installation of the S2 QuickLook software on an operational workstation.

It then goes on to explain how to use the QuickLook function in Sentinel 2 context and the encapsulation software specifically developed for it.

---

## 2. Installation procedure

---

### 2.1. Objective

This chapter explains how to install the QuickLook software.

---

### 2.2. Software and hardware environment

The QuickLook software is designed to run in the following environments:

Identifier	Processor	OS	RAM	Compiler	Shell
Xeon	Intel Xeon 2.8 GHz 64 bits	LINUX RedHat 5	2 GB	icc v10.1	csH

---

### 2.3. Installing the software

Unarchive the TAR file QuickLook.tar retrieved from the CD by executing the following command:

```
tar xvf QuickLook.tar
```

The QuickLook directory is then created, containing two sub-directories:

- include: contains the quicklook.h file,
- src: contains the source of QuickLook library and the source of an example of an encapsulation program.

Go to the src directory and execute the command:

```
make all
```

to build all the executables.

The executable file "ExtractionAlbum" and the dynamic library "libQL.so" are then created in the "bin" directory, which is at the same level in the directory tree as the selected "src" directory.

The entire directory tree can also be cleaned up by executing a "clean" command:

```
make clean
```

Before executing the "ExtractionAlbum" program, first check that the operating system knows the path to the library "libQL.so". There are two ways of doing this:

- Copy the file "libQL.so" to the operating system's directory of shared dynamic libraries (usually "/usr/lib/"),
- Add the path that leads to the file "libQL.so" to the environment variable "LD\_LIBRARY\_PATH". If, for instance, the library is to be stored next to the executable "ExtractionAlbum", the current directory can be added to the environment variable as follows:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:.
```

---

## 3. Using the QuickLook function

---

### 3.1. Objective

This chapter explains how to use the QuickLook function, located in the dynamic library "libQL.so":

```
extern int QuickLook (unsigned char *Packet_In, unsigned int * QL_Out, int  
nbcol, int shln, int mirror);
```

---

### 3.2. Operating mode

In order to work properly, the QuickLook function requires four parameters:

- unsigned char \* Packet\_In,
- unsigned int \* QL\_Out,
- int nbcol,
- int spkln.

"Packet\_In" contains a complete data packet, i.e.:

- the "primary header" (Packet Identification, Packet Sequence Count, Packet Length),
- the "secondary header",
- the bitstream data,
- the CRC.

Each packet from a given bitstream must therefore be retrieved and stored in "Packet\_In" (as an unsigned char) and applied to the QuickLook function.

"QL\_Out" then contains the output of the QuickLook function.

The array has to be previously allocated by the caller with a size of  $2 \times \text{nbcol} / 8$  integers (representing two lines of  $\text{nbcol} / 8$  columns of the full resolution image).

Remark: Two lines of  $\text{nbcol} / 8$  columns is because the QuickLook function decompresses only a part of the BF to produce the QL\_Out image. The compressed image uses 3 level DWT with 16 lines BOLs. The reducing factor at each level is 2, so  $16 / 2^3 = 2$  lines and  $\text{nbcol} / 2^3$  ( $\text{nbcol} / 8$ ) columns.

The "QL\_Out" table contains the pixels on a 12-bit dynamic range, with one pixel per integer, in the storage format of the machine executing the program (little or big endian).

Therefore, it is necessary to know the endianness of the machine executing the program (little endian for Xeon) in order to save the data correctly. (note that the ExtractionAlbum program takes care of this).

The "nbcol" parameter refers to the number of columns of the full resolution image (the full resolution image is the image which was compressed resulting in the bitstream file).

The "shln" parameter refers to the size of the secondary header in words (a word is equivalent to two bytes).

The "mirror" parameter is a flag (value 0 for false or 1 for true) to produce a vertically mirrored output image.

The return value of the QuickLook function is an integer that can take these different values:

- 0: successful execution,
- 1: error detected in the field CesureBF,
- 3: the sum of NPN+CesureBF is greater than 17,
- 4: the capacity of the number of S bits generated is exceeded by a type 2 event.



---

## 4. Using the encapsulation program

---

### 4.1. Objective

This chapter explains how to use the encapsulation program "ExtractionAlbum"

Remark: This software is a specifically modified version of the original encapsulation program example distributed with the generic QuickLook library. It has numerous additional checks during initialization phase and during actual processing phase which were not present in the former version thus introducing a little overhead and therefore a (non significative) loss of performances.

---

### 4.2. Operating mode

In order to work properly, ExtractionAlbum requires five parameters:

- Input file,
- Output file,
- First scene to be decompressed,
- First BOL to be decompressed,
- Number of BOLs to be decompressed.

"Input file" is the path to the bitstream file to be decompressed.

"Output file" is the path to the output image, its file extension has to be one of the following 3:

- lum: to produce a LUM formatted image (Warning: this format cannot be used to write the output image in an Unix pipe file).
- raw: to produce a RAW formatted image,
- hdr: to produce a RAW formatted image and an associated HDR/ENVI file.

"First scene to be decompressed" is the first scene to be decompressed, it starts at 1.

"First BOL to be decompressed" is the number of the first BOL in the scene that will be decoded by the QuickLook function, it starts at 1.

"Number of BOLs to be decompressed" is the number of BOLs that will be decoded by the QuickLook function (special value -1 means to decode all the BOLs).

The return value of the ExtractionAlbum software is 1 if any error stopped the execution, 0 in case of success.

Remarks:

- The mirror flag used to determine if the decoded QuickLook image has to be vertically mirrored is computed from the APID field in the bitstream.

- The number of columns passed to the QuickLook function is also computed from the APID field.
- The zero stuffing of missing BOLs is always activated.
- The secondary header length passed to the QL function is fixed to 54 bytes (Sentinel2 default parameter).
- The length of the scenes in number of BOLs is computed from the APID field in the bitstream.
- The scenes duration is set to its default value for Sentinel2: 3.5s

Note: There is no synchronization mechanism in the software, therefore if a packet size, read in the bitstream, is different from the actual size of the packet, the behaviour of the software is undefined.