

# RMIB Meteosat Second Generation Chain

Nicolas Clerboux and Geert de Sadelaeer

November 21, 2008

## **MSG-RMIB-GE-TN-0040, Version 2.3**

(added informations about EKU account)

### **Contents**

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Scope of this document . . . . .	3
1.2	System overview . . . . .	3
<b>2</b>	<b>DVB system</b>	<b>4</b>
2.1	Data reception . . . . .	4
2.2	Data transfer . . . . .	5
2.3	Various . . . . .	5
2.4	Possible improvements . . . . .	5
<b>3</b>	<b>Processing and distribution system (stratus)</b>	<b>7</b>
3.1	Description . . . . .	7
3.2	Decompression . . . . .	7
3.3	Concatenation . . . . .	8
3.4	“belgium” extraction . . . . .	9
3.5	”Europe” extraction . . . . .	9
3.6	Implementation . . . . .	10
3.7	Known data usages . . . . .	11
3.8	Possible improvements . . . . .	11

<b>4</b>	<b>Archiving and retrieval system (tsunami)</b>	<b>12</b>
4.1	Description . . . . .	12
4.2	Archiving . . . . .	12
4.3	Data retrieval . . . . .	13
4.4	Belgian area archive . . . . .	13
<b>5</b>	<b>Annexes</b>	<b>14</b>
5.1	Details of the EKU account (2 EKU) . . . . .	14
5.2	Persons to be contacted in case of problem . . . . .	14
5.3	IP addresses of the computer . . . . .	14
5.4	Segmentation of the Meteosat Field of view . . . . .	15
5.5	Content of file “recv-channel.ini “ . . . . .	16
5.6	Code of “transfer_tsunami.pl “ . . . . .	17
5.7	Code of “transfer_stratus.pl “ . . . . .	18
5.8	Code of “transfer_stratus_rss.pl “ . . . . .	19
5.9	Code of “transfer.pl “ . . . . .	20
5.10	Code of “duplicate.pl“ . . . . .	22
5.11	What to do in case of failure of the /scratch disk . . . . .	23
5.12	How to stop the Rapid Scan Service (RSS) . . . . .	24

## 1 Introduction

### 1.1 Scope of this document

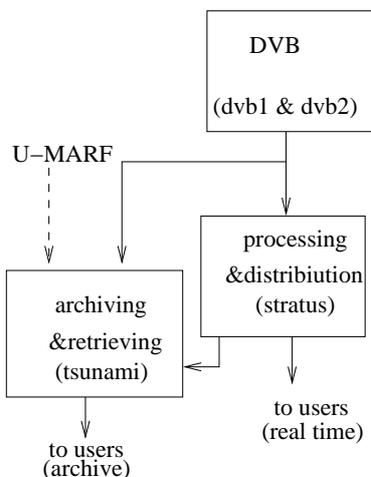
This document describes the implementation of the Meteosat Second Generation (MSG) data stream and processing at the RMIB. The goal of the document is twofold:

1. to allow the “watchman”/operator to take appropriate action in case of problem during the night or the week-end,
2. to provide to the users of the Meteosat data some practical information on “how” to access the data (where, format, timeliness, ...).

This document does not describe the use of the data by the various users (TV, Hawk, Nowcasting SAF,...). It only deals with the part which is common to most of the users.

### 1.2 System overview

As shown on the figure hereafter, 3 main subsystems can be identified:



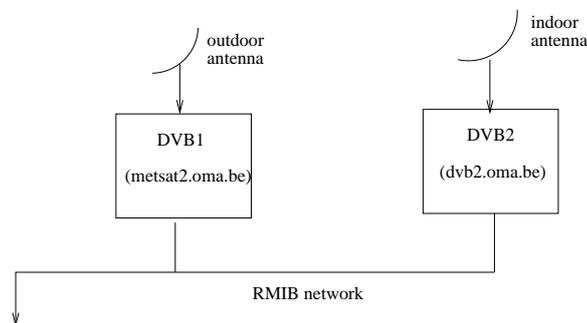
1. The DVB system performs the EUMETCast data reception and dispatching with a high reliability thanks to a full redundancy.
2. The real-time processing and distribution system performs the data decompression, the generation of various types of image and makes them available to the near real-time users (TV, forecasters, nowcasting SAF, RGP SEVIRI processing, CM-SAF, ...).
3. In parallel, the EUMETCast data is also transferred “as-it” to the file server of the section 11 for archiving. The data retrieval for the “non real time users” (RGP SEVIRI processing, CM-SAF, Aerosol team, MUSTI project, Hydrology-SAF, ...) is also performed on this file server.

These 3 subsystems are described in the next sections.

## 2 DVB system

### 2.1 Data reception

The reception system is located at the “Labo” and consists in 2 identical DVB reception and dispatching systems working in parallel. One system is fed through an outdoor antenna while the second uses an indoor antenna. The first is supposed to receive a better signal while the second is less sensitive to rain or snow deposit as well as to misalignment due to wind gust. For both system, a Dell PC running under Windows-XP is used but there is only 1 screen, 1 keyboard and 1 mouse for the 2 systems.



The technical description of EUMETCast, the system installation and setting (frequencies, ...), the registration process and the installation of the EKU (EUMETCast Key Unit) is out of the scope of this document but is fully described in:

*“EUMETCast, EUMETSAT’s Broadcast System for Environmental Data”, Technical Description, EUM TD 15 (available on the EUMETSAT web-site).*

The DVB software and the tq@-TELLICAST client software must run on the computer and have been specified in the “start-up” menu to be restarted automatically in case of PC reboot. The systems are configured to store the received data in the local directories:

```

c:/tellique/to_tsunami
c:/tellique/to_stratus
c:/tellique/to_stratus_rss
c:/tellique/to_duplicate
  
```

These directories act as buffers for further transfer to stratus (near real time) or tsunami (archive). The buffer 'to\_stratus\_rss' contains the EUMETSAT "Rapid Scan Service" data to be transferred to stratus, but in a specific directory (we do not want to mix the operational full-disk and rapid scan Northern hemisphere data). As the Tellique software can not write a received data in these 2 directories, a directory 'to\_duplicate' is used for the data to be transferred on the 'stratus' and 'tsunami' destinations. The configuration is realized via the file 'recv-channel.ini' (the content of this file is given in annex §5.5).

## 2.2 Data transfer

On each DVB computer 3 PERL scripts are running: one script per destination computers and one script to duplicate the data received in `c:/telligence/to_duplicate`.

```
c:/telligence/transfer_tsunami.pl
c:/telligence/transfer_stratus.pl
c:/telligence/duplicate.pl
c:/telligence/transfer_stratus_rss.pl
```

These 4 scripts must be running on each DVB computers, so they are specified in the “start-up” directory. If some of the scripts are not running, the data will rapidly grow in the buffer directories with catastrophic consequences. Note that there is neither “watchdog” nor protection against multiple simultaneous executions of a same script. The code of these scripts are given in annex.

The transfer mechanism is as follow. First the data to transfer to a destination (tsunami, stratus or stratus\_rss) are listed. For this a time-out is implemented to do not transfer data older than 15' to stratus(\_rss) and older than 24 hours to tsunami. Older data are simply removed from the buffer.

Then, a FTP session is opened to the destination and the data are transferred. Before transferring a data it is checked that this data has not been already transferred by another DVB PC. This is done by checking the existence of a log file on the destination server. If the remote log exists (meaning the data was already transferred by another DVB) no FTP transfer is realized and the data is simply removed from the buffer directory. Otherwise, the remote log file is first created, the data is then transferred to the destination and removed locally.

The `duplicate.pl` script simply copies the data received in `c:/telligence/to_duplicate` into the 2 tsunami and stratus buffer directories.

## 2.3 Various

It is possible to perform a remote reboot of the DVB1 system using a dual tone phone. Indeed, the power supply for this PC can be stopped and restarted by calling the 02/XXX.XX.XX and entering the code 6011.

It is possible to do remote access to the DVB computers using VNC (program 'vncviewer') with the user name 'msg\_op' and the corresponding password (starts with "vl" and ends with "o1").

## 2.4 Possible improvements

1. To change the name of the first DVB computer from “metsat2” to “dvh1”
2. For data routing on the network, it may be interesting to use a private address for stratus destination computer (this is done for tsunami).

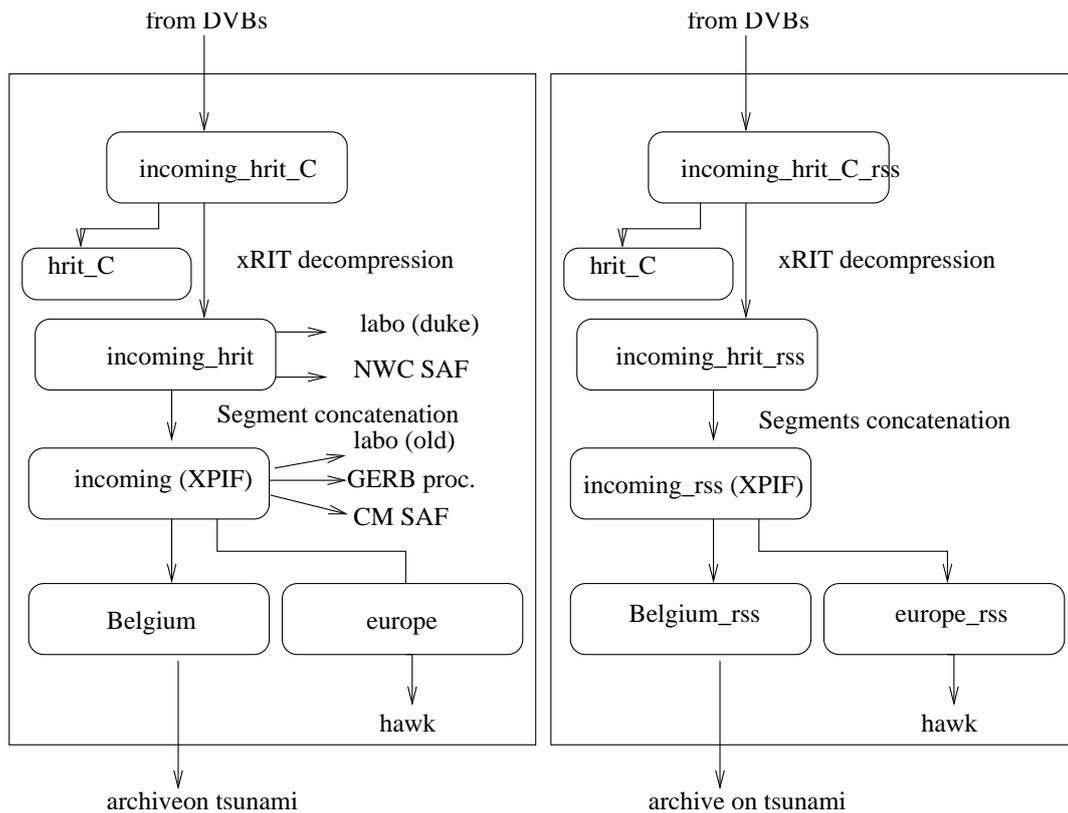
3. Implement “watchdog” (for example using the Windows “cron”) and protection against multiple simultaneous executions of a same script.
4. It can be interesting that a key of the labo is available (for example at the weather office) so that staff member of section 11 can access the room out of the standard work hours (late afternoon or early morning).
5. The VNC connection to 'metsat2' is only passive (we see the screen but we cannot use neither the keyboard nor the mouse). The reason for this is unknown.

### 3 Processing and distribution system (stratus)

#### 3.1 Description

This subsystem is running on “stratus”, one of the UNIX server of the RMIB computer center. It receives EUMETCast data from the DVB system using FTP as described in the previous section. The segments are first decompressed and then concatenated (merged) to produce full-disk images. Finally, rectangular extraction is performed for the “belgium” and “europe” area. At each of these stages, the data remains available to the users during a limited time period. The “near real-time” users can use either the segments or the full disk images but we promote the use of the segments for users who don’t need the full disk images (hawk, TV, ...).

Two chains operate in parallel: a first one to process the full-disk data from the operational satellite (currently Met-9) and a second one to process the “Rapid Scan Service (RSS)” data. The RSS is normally performed with the backup MSG satellite (currently Met-8).



#### 3.2 Decompression

The compressed data arrives from the DVB in the following SCRATCH directory:

```

    /scratch/msg_op/incoming_hrit_C          [remain 0h]
    /scratch/msg_op/incoming_hrit_C_rss     [remain 0h]
  
```

The segments are then decompressed (if the original data is not compressed a simple copy is performed) and the result is put in the directory:

```
$HOME/incoming_hrit           [remain 7h]
$HOME/incoming_hrit_rss       [remain 2h]
```

where \$HOME is the home directory of the user "msg\_op" on stratus:

```
$HOME = stratus:/home/msg_op = /map/msg/msg_op = /msg_data/msg_op
```

The original (compressed) data is then moved (mv) in:

```
/scratch/msg_op/hrit_C       [remain 2h]
```

As indicated, the uncompressed segments remain available to the users for a time period of 7 hours after the decompression (2h for RSS data). The compressed data can be accessed during 2h in the `hrit_C` directory. Some data are not decompressed and are directly moved to `hrit_C` as we do not expect to have usage of them. It is the case of: (i) the LRIT SEVIRI data, (ii) the EUMETSAT daily log file, (iii) the southern HRIT HRV segments 1 to 17.

The wavelet decompression is performed using the EUMETSAT `xRITDecompress` software. The C++ code source is available from EUMETSAT (with license agreement) and some minor modifications (mainly in the inclusion of the header files) have been done to compile on HP-UX. In average, the decompression time (user+sys) of a segment is about 1 s using 1 CPU of the everest server. As each day there are about 9100 HRIT segments to decompress, the cumulated decompression time is about 2.5 hours out of 24 hours (with 1 everest CPU).

### 3.3 Concatenation

When all the needed segments for one image have been received and decompressed, the image is created by concatenation of the segments. For each SEVIRI repeat cycle (15'), 12 XPIF images are created using the RMIB `hrit2xpif` concatenation program. The resulting images are located in (assuming MSG2 is perational and MSG1 performs RSS):

```
$HOME/incoming/YYYYMMDDhhmm_MSG2_SEVIRI_XXX.XPIF   [remain 12h]
$HOME/incoming_rss/YYYYMMDDhhmm_MSG1_SEVIRI_XXX.XPIF [remain 2h]
```

where XXX is the channel identifier (VIS006, VIS008, IR\_016, IR\_039, WV\_062, WV\_073, IR\_087, IR\_097, IR\_108, IR\_120, IR\_134 or HRV). The XPIF format is proprietary of VCS<sup>1</sup>. The HRV image only contains the upper window (3008 upper lines). For the RSS images most of the field of view is 'black' as only the 2 upper segments are used.

<sup>1</sup>The XPIF format is described in t"2met! XPIF. Extended Processed Image File Architecture", Reference Manual, Issue 1.2, Septembre 2002. Available from the VCS web site.

In normal conditions, the XPIF image is available about 14' to 16' after the repeat cycle time (which corresponds to the start of 12' image acquisition process). So, the XPIF image is made available to the users only about 1' to 5' after the western-europe observation. The same delay is bserved for the RSS.

For each SEVIRI repeat cycle, the MPEF (EUMETSAT Meteorological Product Extraction Facility) creates and disseminates an interesting cloud mask in the form of 6 LRIT segments. This data is concatenated using the RMIB `clm_lrit2pgm` program and the result is available in the file:

```
$HOME/incoming/YYYYMMDDhhmm_MSG2_MPEF_CLM.pgm [remain 12h]
```

Currently the cloud mask is not extracted for the RSS (can be done if needed).

### 3.4 “belgium” extraction

The “belgium” area is a 300\*600 pixels areas defined by the coordinates (ul.x=1650, ul.y=200, lr.x=2250, lr.y=500, with the upper-left pixel at x=y=0). This area is the same as the “BEL” area on which the NWCSAF software is operated and is illustrated on figure (1). For each repeat cycle, this area is extracted from the full disk image as well as from the MPEF cloud mask. A RGB composite image is also created at the HRV resolution from the visible images. All these data remain available for 24 hours in the files:

```
$HOME/belgium/YYYYMMDDhhmm_MSG2_SEVIRI_XXX_BEL.rof [remain 24h]
$HOME/belgium/YYYYMMDDhhmm_MSG2_MPEF_CLM_BEL.pgm [remain 24h]
$HOME/belgium/YYYYMMDDhhmm_MSG2_SEVIRI_VIS_BEL.jpg [remain 24h]
```

Similar extraction is performed for the RSS (assuming RSS is performed with MSG1)

```
$HOME/belgium_rss/YYYYMMDDhhmm_MSG1_SEVIRI_XXX_BEL.rof [remain 8h]
$HOME/belgium_rss/YYYYMMDDhhmm_MSG1_SEVIRI_VIS_BEL.jpg [remain 8h]
```

After that, the Belgian data (operational and RSS) remain available to the users on the hard drive on the file server of the section 11 (tsunami). The ROF image format is proprietary of the RMIB section 11 and is described in:

```
http://gerb.oma.be/seviri/seviri\_bel
```

### 3.5 ”Europe” extraction

The “europe” area is a 1854\*926 pixels areas defined by the coordinates (ul.x=930, ul.y=98, lr.x=2784,lr.y=1024). The area is the same as the “EURO” area over which the NWC SAF software is operated and is illustrated on figure (1). For each repeat cycle, the SEVIRI data is extracted for this area, calibrated and converted in reflectance (visible channels) or in brightness temperature (thermal channels). The cosine of Solar Zenith Angle(COSSZA) image is also computed. The resulting floating point images are available for 5 hours in

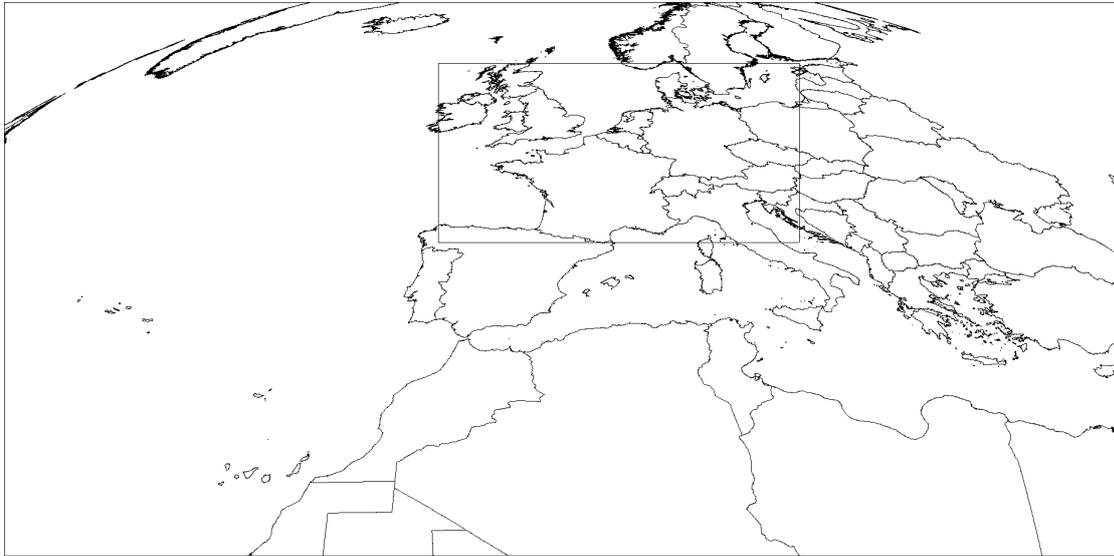


Figure 1: The RMIB “europe” and “belgium” areas in the Meteosat field-of-view.

```
$HOME/europe/YYYYMMDDhhmm_EUROPE_REFCT_XXXXXX.rof [remain 5h]
$HOME/europe/YYYYMMDDhhmm_EUROPE_BTEMP_XXXXXX.rof [remain 5h]
$HOME/europe/YYYYMMDDhhmm_EUROPE_COSSZA.rof [remain 5h]
```

This data allows a fast and easy creation of the color composite images (reflectance, thermal, water vapor, air mass, severe convection, ...) to be visualized in Hawk.

### 3.6 Implementation

The various operations described before are executed via 3 bash scripts:

**MSG\_chain1.bat** in charge of:

- the segments decompression (op. and RSS),
- the segments concatenation to create the full-disk XPIF images (op. and RSS),
- the segments concatenation to create the full-disk cloud mask images (op. only).

**MSG\_chain2.bat** in charge of:

- the belgian area extraction (op. and RSS),
- the creation of the 1km color composite reflectance image for the belgian area (op. and RSS),
- the europe extraction and conversion (op.),

- the creation of the color composite images for HAWK (op.),
- the extraction of the data for the planetarium (op.),

**MSG\_clean.bat** in charge of:

- the cleaning of the data received by the DVB systems and the associated 'transfer log',
- the cleaning of the data created by the MSG\_chain1.bat script
- the cleaning of the data created by the MSG\_chain2.bat script

These scripts are launched by the cron daemon (with protection to avoid multiple simultaneous execution) each 1' for MSG\_chain1.bat and MSG\_chain2.bat and 30' for MSG\_clean.bat . The scripts (as well as all the other scripts and binaries) are located in the directory:

```
$HOME/bin
```

Starting from this file, it is relatively easy to follow the successive operations (decompression, concatenation, belgium and europe extraction, removing old data, ...). The "cronfile" is located in the same directory. So, if the cronfile were lost (or in case of switch to a backup server):

```
crontab $HOME/bin/cronfile
```

In case of problem on the everest file server, we provide a script that stops the processings (remove crontab and kill the processes), removes the data in the `incoming_hrit_C` directory and restart the processings (install the cronfile). This script is located in:

```
$HOME/bin/MSG_restart.bat (to be checked before use!!!)
```

### 3.7 Known data usages

The list of user that access the MSG data on stratus is:

- the **SAF nowcasting** is running locally and access the 3 upper HRIT segments,
- the **labo** performs a copy of the 3 upper HRIT segment on the 'duke' computer.
- the **GERB team** download all but the 3.9 and HRV XPIF images to the 'typhoon' computer using the FTP,
- the **Climate SAF team** download the VIS006 XPIF images to the 'fog' computer using the FTP,
- the **HAWK Color Composite** is running locally and access the EURO data,
- the **tsunami** is archiving the belgian area data.

### 3.8 Possible improvements

1. To use HRIT segments instead of the XPIF for the belgian and europe extraction.

## 4 Archiving and retrieval system (tsunami)

### 4.1 Description

This subsystem is running on “tsunami”, the RAID file server of the RMIB Section 11. It receives the XRIT segments of the operational MSG satellite via the DVB system by FTP. In average 8.5 GB are transferred each day in the directory:

```
tsunami:/archive/seviri/incoming_xrit_C [remain 40']
```

After 40', the interesting data are sorted in the "repeat cycle directory" for further tape archiving:

```
tsunami:/archive/seviri/MSG2_YYYYMMDD/XRIT_YYYYMMDDhhmm
```

The data with a time that does not correspond to a standard SEVIRI repeat cycle (i.e. ending with 00, 15, 30 or 45) are sorted in the repeat cycle directory just before. Some XRIT data are not sorted but are deleted because we don't want to archive them. It is the case of:

- the Meteosat-6 and -7 Indian Ocean Data Cover (IODC) data,
- the SEVIRI LRIT images because we have the corresponding higher quality HRIT data.

In case of reception problem, the archive manager (Patrick Vandermeulen) orders the missing data to the UMARG (the EUMETSAT archive) in order to have an archive as complete as possible.

### 4.2 Archiving

When about 12 days of data is available, the data manager (Patrick Vandermeulen) performs the data transfer on LTO-1 tape (100GB), checks the tape and lists its content, and finally removes the corresponding data from the disk. The listing of the archived data is available on:

```
tsunami:/home/seviri/TapeContent/LTO/LTO_XXX.txt
```

where XXX is the tape number. The following web page give the archive status for a given day as well as the tape number where the data are stored:

```
http://gerb.oma.be/seviri/Status/sev\_status\_YYYYMMDD.html
```

### 4.3 Data retrieval

On user's request, the data is retrieved from the press the segments using the EUMETSAT software (we provide binaries for Windows, Linux as well as HP-UX) and perform the concatenation if full disk image is needed (we provide the C program for concatenation). The user can also ask the operator to perform thesetapes. The user can then decompress the segments using the EUMETSAT software (we provide binaries for Windows, Linux as well as HP-UX) and perform the concatenation if full disk image is needed (we provide the C program for concatenation). The user can also ask the operator to perform these operations.

### 4.4 Belgian area archive

The belgian area adta extracted on the stratus computer are archived on tsunami in the directories

```
/tsunami/seviri/SEVIRI_BEL/SEVIRI_BEL_YYYYMMDD  
/tsunami/seviri/SEVIRI_BEL/SEVIRI_BEL_YYYYMMDD_RSS
```

respectively for the operational satellite and the RSS satellite. The data are the same as described in §3.4, except they are compressed using bzip2.

## 5 Annexes

### 5.1 Details of the EKU account (2 EKU)

Geert De Sadelaer  
Company: Royal Met Institute of Belgium

Username: RMIBelgium2  
Password: CCQHSS6ZwQ

EKU numbers:  
00550  
00551

### 5.2 Persons to be contacted in case of problem

The persons hereafter should be informed in case of interruption of the Meteosat image service for a time period of more than about 2 hours:

forecast@oma.be -> the weather office forecasters  
labo@oma.be -> the labo (Alain, Michiel,...)  
Frank.DEBOOSERE@VRT.BE -> Frank Deboosere  
ops.cmsaf@oma.be -> operation team for the CM-SAF  
gerb@oma.be -> operation team for the GERB

with copy to:

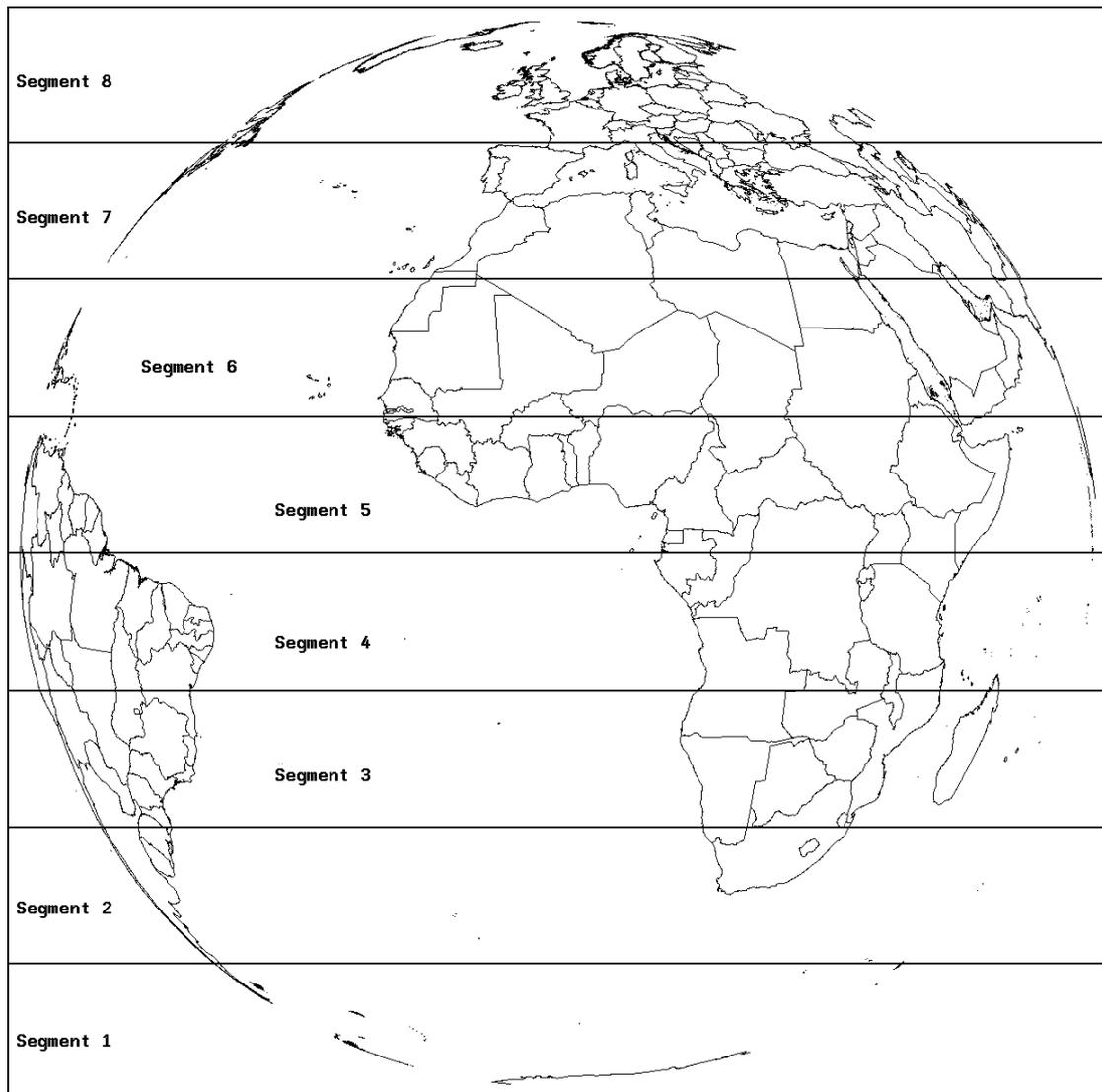
desa@oma.be -> Geert De Sadelaer  
ferauge@oma.be -> Christophe Ferauge  
nic@oma.be -> Nicolas Clerbaux (mobile phone is +32473423364)  
patvdm@oma.be -> Patrick Vandermeulen  
ui@oma.be -> the user interface  
Steven.Dewitte@oma.be -> head of Observations Dept.  
Henri.Malcorps@oma.be -> the director

### 5.3 IP addresses of the computer

metsat2.oma.be : 192.168.15.139  
dvb2.oma.be : 192.168.15.166  
everest.oma.be : 193.190.249.236  
gurung.oma.be : 193.190.249.248  
stratus.oma.be : 193.190.249.239  
tsunami-p.oma.be : 192.168.15.246  
tsunami.oma.be : 193.190.249.234 (not used for the FTP)

## 5.4 Segmentation of the Meteosat Field of view

The figure hereafter shows the position of the 8 segments of the “standard” SEVIRI images. The HRV images have 24 segments which are not illustrated here. Each standard segment covers 3 HRV segments.



## 5.5 Content of file “recv-channel.ini “

```
# TelliCast only: Definition for Channel Subscription of local Receiver
```

```
# EARS
```

```
[EUMETSAT Data Channel 1]  
target_directory=C:\tellique\to_stratus  
temp_directory=C:\tellique\receiving
```

```
# HRIT SEVIRI Images data (operational)
```

```
[EUMETSAT Data Channel 2]  
target_directory=C:\tellique\to_duplicate  
temp_directory=C:\tellique\receiving
```

```
# Multi-service Channel (LRIT operational, IODC, DCP, MDD, MPEF, OSI-SAF products, GOES, M
```

```
[EUMETSAT Data Channel 3]  
target_directory=C:\tellique\to_duplicate  
temp_directory=C:\tellique\receiving
```

```
# Land SAF and OSI SAF KNMI wind
```

```
[SAF-Europe]  
target_directory=C:\tellique\to_stratus  
temp_directory=C:\tellique\receiving
```

```
# Multi-service Europe
```

```
[EUMETSAT Data Channel 4]  
target_directory=C:\tellique\to_stratus  
temp_directory=C:\tellique\receiving
```

```
# HRIT - RSS
```

```
[EUMETSAT Data Channel 5]  
target_directory=C:\tellique\to_stratus_rss  
temp_directory=C:\tellique\receiving
```

```
# LRIT - RSS
```

```
[EUMETSAT Data Channel 6]  
target_directory=C:\tellique\to_stratus_rss  
temp_directory=C:\tellique\receiving
```

```
# DWDSAT
```

```
[DWDSAT]  
target_directory=C:\tellique\to_stratus  
temp_directory=C:\tellique\receiving
```

## 5.6 Code of “transfer\_tsunami.pl”

```
#!/usr/bin/perl -w
#
# Pearl script that performs the transfer of EUMETCAST data
# from a DVB computer to the archive file server "tsunami".
#
# The "time-out" for this transfer is 3 days.
#
#
# AUTHOR: Nicolas Clerbaux , RMIB, February 2006
#
use FileHandle;
use File::Basename;
use File::Copy;
use Net::FTP;
use strict;

# Settings for the FTP transfer
my $REMOTE_STR = "tsunami";           # only for display
my $REMOTE     = "193.190.249.234";   # the IP address of stratus.oma.be
my $LOGIN      = "gerb";              # user name on tsunami
my $PASSWD    = "XXXXXXXX";          # password for gerb on tsunami
my $FTP_DIR    = "/archive/incoming_xrit_C"; # remote directory
my $FTP_SUF    = ".TEMP2";           # suffix used during the FTP transfer

# Local settings
my $LISTING    = "list_tsunami.txt";  # name of the temporary file that contains the list of data to transfer
my $SLEEPTIME = 3;                   # sleep in the main loop
my $MAX_AGE    = 259200;              # do not transfer file older than 3 days
my $DIR_IN     = "c:/telligence/to_tsunami"; # EUMETCAST incoming directory
my $DELAY      = 1;                  # delay of transfer (1 =operationnel, 61 = backup)

# Main infinite loop ...
while ( 1 )
{
    print "Transfer to $REMOTE_STR :\n" ;

    # 1. Create the listing
    my @list = ( );
    foreach my $file (<$DIR_IN/*>)
    {
        # "age" of the file in second, basename and local log name
        my $file_time = time - (stat( $file ))[9] ;
        my $bfile     = basename($file) ;

        # file too young -- skip
        if ( $file_time < $DELAY )
        {
            # print " $bfile [too young!]\n" ;
            next;
        }

        # file too old -- do not transfer and remove the data
        if ( $file_time > $MAX_AGE )
        {
            print " $bfile [too old!]\n" ;
            unlink( $file );
            next;
        }

        # add to the list of files to transfer
        push @list, $file;
    }

    # 2. Write the listing on the disk
    my $HANDLE = new FileHandle ;
    $HANDLE->open( $LISTING , "w" ) ;
    print $HANDLE (join("\n", @list, ""));
    $HANDLE->close ;

    # 3. Call the "transfer.pl" script to transfer this list
    if( @list == 0 )
    {
        #print "No data to transfer\n" ;
    }
    else
    {
        system( "transfer.pl $REMOTE $LOGIN $PASSWD $FTP_DIR $FTP_SUF $LISTING" );
    }

    # 4. Sleep
    sleep ( $SLEEPTIME );
} # end of the main infinite loop
```

## 5.7 Code of “transfer\_stratus.pl”

```
#!/usr/bin/perl -w
#
# Pearl script that performs the transfer of EUMETCAST data
# from a DVB computer to the real-time file server "stratus".
#
# There is a "time-out" for this transfer: the data older than
# 30' are not transferred.
#
# AUTHOR: Nicolas Clerbaux , RMIB, February 2006
#
#
use FileHandle;
use File::Basename;
use File::Copy;
use Net::FTP;
use strict;

# Settings for the FTP transfer
my $REMOTE_STR = "stratus";           # only for display
my $REMOTE = "193.190.249.239";      # the IP address of stratus.oma.be
my $LOGIN = "msg_op";                 # user name on stratus
my $PASSWD = "XXXXXXX";              # password for msg_op on stratus
my $FTP_DIR = "/scratch/msg_op/incoming_hrnt_C"; # remote directory
my $FTP_SUF = ".TEMP2";               # suffix used during the FTP transfer

# Local settings
my $LISTING = "list_stratus.txt";     # name of the temporary file that contains the list of data to transfer
my $SLEEPTIME = 3;                   # sleep in the main loop
my $MAX_AGE = 1800;                   # do not transfer file older than 900'
my $DIR_IN = "c:/telligence/to_stratus"; # EUMETCAST incoming directory
my $DELAY = 31;                       # delay of transfer (1 =operationnel, 61 = backup)

# Main infinite loop ...
while ( 1 )
{
    print "Transfer to $REMOTE_STR :\n" ;

    # 1. Create the listing
    my @list = ( );
    foreach my $file (<$DIR_IN/*>)
    {
        # "age" of the file in second, basename and local log name
        my $file_time = time - (stat( $file ))[9] ;
        my $bfile = basename($file) ;

        # file too young -- skip
        if ( $file_time < $DELAY )
        {
            # print " $bfile [too young!]\n" ;
            next;
        }

        # file too old -- do not transfer and remove the data
        if ( $file_time > $MAX_AGE )
        {
            print " $bfile [too old!]\n" ;
            unlink( ${file} );
            next;
        }

        # add to the list of files to transfer
        push @list, $file;
    }

    # 2. Write the listing on the disk
    my $HANDLE = new FileHandle ;
    $HANDLE->open( $LISTING , "w" ) ;
    print $HANDLE (join("\n", @list, ""));
    $HANDLE->close ;

    # 3. Call the "transfer.pl" script to transfer this list
    if( @list == 0 )
    {
        print "No data to transfer\n" ;
    }
    else
    {
        system( "transfer.pl $REMOTE $LOGIN $PASSWD $FTP_DIR $FTP_SUF $LISTING" );
    }

    # 4. Sleep
    print "sleeping\n" ;
    sleep ( $SLEEPTIME );
} # end of the main infinite loop
```

## 5.8 Code of “transfer\_stratus\_rss.pl”

```
#!/usr/bin/perl -w
# (...)

use FileHandle;
use File::Basename;
use File::Copy;
use Net::FTP;
use strict;

# Settings for the FTP transfer
my $REMOTE_STR = "stratus (rss)";           # only for display
my $REMOTE     = "193.190.249.239";        # the IP address of stratus.oma.be
my $LOGIN      = "msg_op";                 # user name on stratus
my $PASSWD     = "XXXXXXXX";              # password for msg_op on stratus
my $FTP_DIR    = "/scratch/msg_op/incoming_hrnt_C_rss"; # remote directory
my $FTP_SUF    = ".TEMP2";                # suffix used during the FTP transfer

# Local settings
my $LISTING    = "list_stratus_rss.txt";   # name of the temporary file that contains the list of data to transfer
my $SLEEPTIME  = 3;                        # sleep in the main loop
my $MAX_AGE    = 1800;                     # do not transfer file older than 900'
my $DIR_IN     = "c:/telligence/to_stratus_rss"; # EUMETCAST incoming directory
my $DELAY      = 31;                       # delay of transfer (1=operationnel, 61 = backup)

# Main infinite loop ...
while ( 1 )
{
    print "Transfer to $REMOTE_STR :\n" ;

    # 1. Create the listing
    my @list = ();
    foreach my $file (<$DIR_IN/*>)
    {
        # "age" of the file in second, basename and local log name
        my $file_time = time - (stat( $file )) [9] ;
        my $bfile     = basename($file) ;

        # file too young -- skip
        if ( $file_time < $DELAY )
        {
            # print " $bfile [too young!]\n" ;
            next;
        }

        # file too old -- do not transfer and remove the data
        if ( $file_time > $MAX_AGE )
        {
            print " $bfile [too old!]\n" ;
            unlink( $file );
            next;
        }

        # segments 5 and 6 are not transferred
        my $seg = substr($bfile, 35, 11);
        if ( $seg eq "-000005___-" || $seg eq "-000006___-" || $seg eq "-000015___-" || $seg eq "-000016___-" || $seg eq "-000017___-" || $seg eq "-000018___-" )
        {
            print " $bfile [rm segment]\n" ;
            unlink( $file );
            next;
        }

        # add to the list of files to transfer
        push @list, $file;
    }

    # 2. Write the listing on the disk
    my $HANDLE = new FileHandle ;
    $HANDLE->open( $LISTING , "w" ) ;
    print $HANDLE (join("\n", @list, ""));
    $HANDLE->close ;

    # 3. Call the "transfer.pl" script to transfer this list
    if( @list == 0 )
    {
        print "No data to transfer\n" ;
    }
    else
    {
        system( "transfer.pl $REMOTE $LOGIN $PASSWD $FTP_DIR $FTP_SUF $LISTING" ) ;
    }

    # 4. Sleep
    print "sleeping\n" ;
    sleep ( $SLEEPTIME );
} # end of the main infinite loop
```

## 5.9 Code of “transfer.pl”

```
#!/usr/bin/perl -w
#
# Pearl script that performs the FTP transfer of a list of
# EUMETCAST files to a remote computer. This script is
# called by "transfer_stratus.pl" and "transfer_tsunami.pl".
#
# Usage :
#   transfer remote  uname  passwd  dir  ftp_suf  list
#
# Where:
# - "remote" (0) : is the FTP destination (IP address)
# - "uname" (1) : is the user name for FTP login
# - "passwd" (2) : is the password for FTP login
# - "dir" (3) : is the remote directory for the FTP
# - "ftp_suf" (4) : is the suffix for the FTP (.TEMP for example)
# - "list" (5) : is the file that contains the list of data to transfer
#
# AUTHOR: Nicolas Clerbaux , RMIB, February 2006
#
#
use FileHandle;
use File::Basename;
use File::Copy;
use Net::FTP;
use strict;

# 1. if not exist, create an empty file
if ( ! -e "logpattern.txt" )
{
    open( LFILE, "> logpattern.txt " );
    close(LFILE) ;
}

# 2. Open and read the list of the files to be transferred
my $HANDLE = new FileHandle ;
$HANDLE->open( $ARGV[5] , "r" ) ;
my @list = <$HANDLE>;
$HANDLE->close ;
chomp(@list);

# 3. Sorting the listing according to the file's modification time
print " Sorting the list ...\n" ;
my %cache;
$cache{$_} = -M for @list;
@list = sort { $cache{$b} <=> $cache{$a} } @list;

# 4. Open the FTP connection
my $ftp = Net::FTP->new( $ARGV[0], Timeout => 30)    or die "FTP $ARGV[0] : connect error ";
$ftp->login($ARGV[1],$ARGV[2])                    or die "FTP $ARGV[0] : invalid user name or password";
$ftp->cwd($ARGV[3])                               or die "FTP $ARGV[0] : cannot change remote directory";
$ftp->binary()                                    or die "FTP $ARGV[0] : cannot change to binary transfer type";

# 5. Loop on the files to transfer
foreach my $file ( @list )
{
    # 5.1 build the remote log name
    my $bfile = basename($file) ;
    my $remote_log = "TransferLogs/${bfile}.log" ;

    # 5.2 test existence of the remote log. if the remote
    # log exists (meaning the data was already transferred
    # by an other DVB system) the data is removed
    my $size = $ftp->size($remote_log) ;
    if ( defined($size) )
    {
        print " $bfile [rlog exist]\n" ;
        unlink( $file );
        next;
    }

    # 5.3 Create the remote log before starting the transfer
    my $rc = $ftp->put("logpattern.txt", $remote_log );
    if ( $rc ne $remote_log )
    {
        print "ERROR: cannot create remote log !!!\n";
    }

    # 5.4 perform the FTP transfer in a temporary file ( ARGV[4] is the FTP_SUF)
    my $tmp_file = "${bfile}$ARGV[4]";
    $rc = $ftp->put($file,$tmp_file );

    # 5.6 if successful delete the file
}
```

```
if ( $rc eq $tmp_file )
{
    $rc = $ftp->rename( $tmp_file,$bfile );
    print " $bfile [ftp OK] \n" ;
    unlink( ${file} );
}
else
# 5.5 in case of FTP problem, try to remove the remote log and exit
{
    print " $bfile [ftp KO] \n" ;
    # $rc = $ftp->delete( $tmp_file );
    $rc = $ftp->delete( $remote_log );
    last ;
}
}

# close the connection
$ftp->quit();
```

## 5.10 Code of “duplicate.pl”

```
#!/usr/bin/perl -w
#
# This simple PERL script simply duplicate the data received by the
# DVB system in the directory "c:/tellique/to_duplicate" into the
# Buffers directory for the transfer to the near real-time server
# (stratus/everest) and to the archive server (tsunami).
#
# AUTHOR: Nicolas Clerbaux -- RMIB
#

use FileHandle;
use File::Basename;
use File::Copy;
use strict;

my $DIR_IN      = "c:/tellique/to_duplicate" ; # EUMETCAST Input directory
my $SLEEPTIME  = 10;                          # sleep time in the main loop

my $DIR_TSUNAMI = "c:/tellique/to_tsunami" ;   # Buffers directory to tsunami
my $DIR_STRATUS = "c:/tellique/to_stratus" ;   # Buffers directory to stratus

# main infinite loop
my $i=0;
while ( 1 )
{
    foreach my $file (<$DIR_IN/*>)
    {
        my $bfile      = basename($file) ;

        print " $bfile [duplicating]\n" ;

        # create a copy of the file and move original and copy to the buffers directories...
        copy( $file , "${bfile}.COPY" );
        move( "${bfile}.COPY", "$DIR_TSUNAMI/${bfile}" );
        move( $file, "$DIR_STRATUS/${bfile}" );
    }
    sleep( $SLEEPTIME );
}
}
```

## 5.11 What to do in case of failure of the /scratch disk

The processing and distribution system will be interrupted in case of failure of the /scratch disk on stratus or in case of switch of the package from everest to illuin (illuin does not have /scratch disk, TBC). In this case, the simplest thing to do is to create a symbolic link:

```
/scratch -> /home
```

and to create the following directories in the home directory, in order to use the S.A.N. instead of the failling /scratch disk

```
rm /home/msg_op/incoming_hrit_C
rm /home/msg_op/hrit_C

mkdir /home/msg_op/incoming_hrit_C
mkdir /home/msg_op/incoming_hrit_C/TransferLogs
mkdir /home/msg_op/hrit_C
mkdir /home/msg_op/tmp_safnwc
```

If it is not possible to create the symbolic link (one needs to have administrator right), it is suggested to create the directories as before and to change the destination directory in the 'transfer\_stratus.pl' script on the 2 DVB systems in the labo. Destination should change from

```
my $FTP_DIR = "/scratch/msg_op/incoming_hrit_C"; # remote directory
```

into

```
my $FTP_DIR = "/home/msg_op/incoming_hrit_C"; # remote directory
```

In this case, the /scratch must be replace by /home in the 2 scripts:

```
/msg/msg_op/bin/xritDecomp.bat
/msg/msg_op/bin/MSG_clean.bat
```

## 5.12 How to stop the Rapid Scan Service (RSS)

The first possibility is to modify the 'recv-channels.ini' files **ON THE 2** DVB reception computers (dvh2 and metsat2). We just have to comment the lines with:

```
# HRIT - RSS
[EUMETSAT Data Channel 5]
target_directory=C:\tellique\to_stratus_rss
temp_directory=C:\tellique\receiving

# LRIT - RSS
[EUMETSAT Data Channel 6]
target_directory=C:\tellique\to_stratus_rss
temp_directory=C:\tellique\receiving
```

In this case the RSS data will not be received anymore and therefore no data will be transferred on stratus.

Another possibility is to keep the DVB reception and FTP transfer alive, but to do not uncompress the RSS data on stratus. To do this, the call to `xritDecomp_rss.bat` in the script `MSG_chain1.bat` should be commented.