

MOC standard

1.1 extension / ASCII MOCs

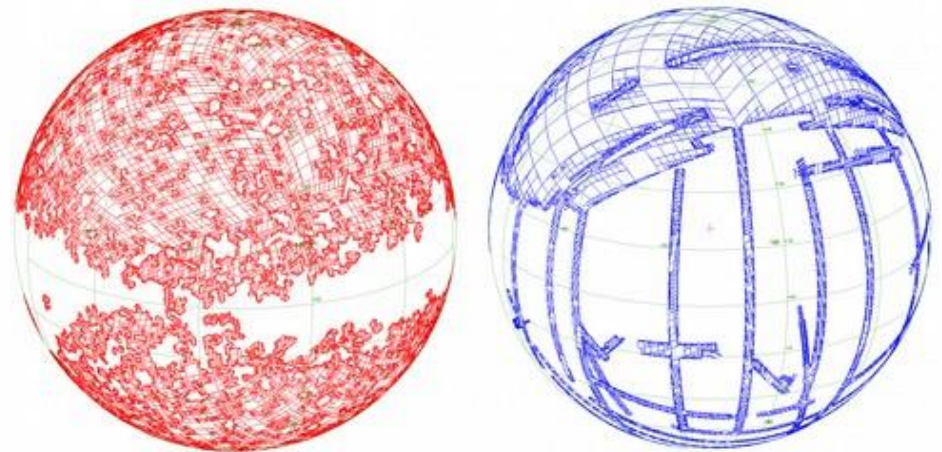
College Park Interop meeting
8 to 10 November 2018

Pierre Fernique
and others contributors



□ What's a MOC ?

- A **simple** and **efficiente** method for describing a sky region
- Principle : MOC = list of **HEALPix cell indexes** of the region, grouped hierarchically



□ MOC story

- IVOA **recommendation** since 4 years (june 2014)
- Good **success**: more & more usages, libs, algos, and tools
- It has been adopted by developers as a **generic tool** for manipulating any kind of regions (even very accurate regions, observation footprints, spatial index, ...)
- **One serialization**: FITS (= binary table of HEALPix index)
- **Alternate** ASCII and JSON serialization syntaxes **just suggested**



International
Virtual
Observatory
Alliance

MOC – HEALPix Multi-Order Coverage map Version 1.0 IVOA Recommendation 2 June 2014

This version:
I.O. Recommendation 2014-06-02

Previous version(s):
None

InterestWorking Group:
Applications: <http://www.ivoa.net/wiki/bin/view/IVOA/IvoaApplications>

Editor:
Pierre Fernique

Authors:
Thomas Boch, Tom Donadson, Daniel Durand, Pierre Fernique, Will O'Mulane,
Martin Reneke, Mark Taylor

Abstract

This document describes the Multi-Order Coverage map method (MOC) to specify arbitrary sky regions. The goal is to be able to provide a very fast comparison mechanism between coverage maps. The mechanism is based on the HEALPix sky tessellation algorithm. It is essentially a simple way to map regions of the sky into hierarchically grouped predefined cells.

1

3 Help for implementing

3.1 ASCII MOC

In general the FITS encoding described in section 2 should be used for exchange of MOCs. However, if it is required to write a MOC as an ASCII string (for a web form, for debugging, ...) it is suggested to use one of the following syntaxes:

3.1.1 JSON syntax

A JSON MOC **may** be written following this syntax:

```
{ "order":[npix,npix,...], "order":[npix, npix...], ... }.
```

Example of a JSON MOC

```
{"1": [1, 2, 4], "2": [12, 13, 14, 21, 23, 25]}
```

3.1.2 ASCII string syntax

An ASCII string MOC **may** be written following this syntax:

```
order/npix,npix,... order/npix,npix.
```

The usage of a range operator is allowed in the list of npix using the dash ("-") as a separator: lownpix-highnpix.

Warning: In this basic simple ASCII string format only the values **may** be not sorted, and the MOC **may** be not well-formed.

Example of a ASCII string MOC

```
1/1, 3, 4 2/4, 25, 12-14, 21
```


□ A new usecase

- M.Deimleitner:
« I would like to use MOC to describe roughly the space coverage of each VO resource in the VO registry »
- Consequence: we need to associate MOCs and VO registry records

□ Face to the reality


- But: **binary** MOC (FITS) **not really usable** inside **ASCII streams** (XML, ASCII file...)
- The solution: **normalize the ASCII MOC** syntax
 - Con: 2 serializations never help for interoperability
 - Pro : Will allow to use MOC directly in the VO registry, or in a VOTable result, or in a ADQL query...

□ 2 years ago => Trieste 2016

- We had a similar question, but concerning JSON MOC
=> At this time we decided to keep the standard as it was.
- **Today**, our usecase requires ASCII serialization as it would be reused in **other IVOA standard documents** (registry, ADQL?...)

□ Toward a MOC 1.1 REC

- The proposal: just **move** the **ASCII MOC syntax** from informative section to **normative section**.
- **Discussion** on the App mailing list has already started:
=> **no contradictor** presently
- A **WD 1.1 is ready, can be published**
=> If there is no opposition, **could be ready for PR** in a few weeks



International
Virtual
Observatory
Alliance

**MOC – HEALPix Multi-Order Coverage map
Version 1.1
IVOA Working Draft 5 Nov 2018**

This version:
1.1: Working Draft 2018-11-05
Previous version(s):
1.0: Recommendation 2014-06-02

Interest/Working Group:
Applications: <http://www.ivoa.net/wiki/bin/view/VOA/IvoaApplications>

Editor:
Pierre Fernique

Authors:
Pierre Fernique, Thomas Boh, Tom Donadson, Daniel Durand, Will O'Mullane,
Martin Reinecke, Mark Taylor

Abstract
This document describes the Multi-Order Coverage map method (MOC) to specify arbitrary sky regions. The goal is to be able to provide a very fast comparison mechanism between coverage maps. The mechanism is based on the HEALPix sky tessellation algorithm. It is essentially a simple way to map regions of the sky into hierarchically grouped predefined cells.

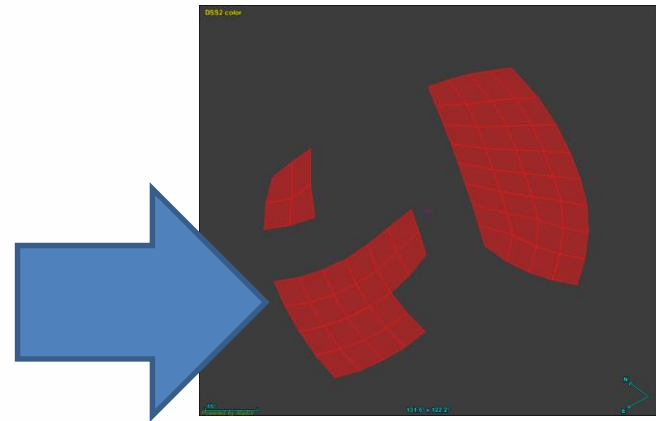
1

□ Good news !

- **Already implemented** in various tools/API:
 - Java MOC API,
 - HEALPix official lib,
 - Aladin Desktop,
 - Hipsgen,
 - ...

```
Command draw moc 1/1,3,4 2/4,25,12-14,21
```

● DSS ● DSS red ● Pan-STARRS ● SDSS ● 2MASS ● GALEX ● AKA



Reactions ? Discussion ?



*International
Virtual
Observatory
Alliance*

MOC – HEALPix Multi-Order Coverage map Version 1.1

IVOA Working Draft 5 Nov 2018

This version:

1.1: Working Draft 2018-11-05

Previous version(s):

1.0: Recommendation 2014-06-02

Interest/Working Group:

Applications: <http://www.ivoa.net/wiki/bin/view/IVOA/IvoaApplications>

Editor:

Pierre Fernique

Authors:

Pierre Fernique, Thomas Boch, Tom Donaldson, Daniel Durand, Wll O'Mullane,
Martin Reinecke, Mark Taylor

Abstract

This document describes the Multi-Order Coverage map method (MOC) to specify arbitrary sky regions. The goal is to be able to provide a very fast comparison mechanism between coverage maps. The mechanism is based on the HEALPix sky tessellation algorithm. It is essentially a simple way to map regions of the sky into hierarchically grouped predefined cells.

1