

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Bringing Healpix and MOC in TAP

Grégory Mantelet (ARI/ZAH)

May 16, 2017

This preliminary study has been done in collaboration with the CDS (P. Fernique, T. Boch, G. Landais, L. Michel, A. Oberto, F-X. Pineau).

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

Quick reminder

Healpix & MOC quick reminder

- ▶ Healpix is a hierarchical tessellation of the sky.
 - ▶ **scheme** ('nested' (generally) or 'ring')
 - ▶ **order** within [0;29]
- ▶ Derived products:
 - ▶ **Healpix map** = FITS binary table of 2 columns: Healpix index and a value -> 2-D histogram
 - ▶ *Requirements*: all Hpx indices at the same order
 - ▶ *Formats*: FITS (*defined by the Healpix community*)
 - ▶ **MOC** = Multi-Ordered Coverage map = list of Healpix indices possibly at different Hpx order
 - ▶ *Requirements*: nested scheme, indices with their resp. order
 - ▶ *Formats*: FITS (*non-standard: JSON and ASCII*)

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

- References
- Proposed ASCII
serialization
- PEG grammar for
MOC-ASCII

Usage in TAP

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

Possible usage in TAP

1. Get Healpix-related data (i.e. SELECT):
 - 1.1 Healpix map
 - 1.2 MOC (*2 possible solutions*)
2. Filter dataset (i.e. WHERE):
 - 2.1 by Healpix index
 - 2.2 by MOC (*3 possible solutions*)

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Quick reminder

Usage in TAP

**1. Get
Healpix-related
data**

- a. Create an Healpix map
- b. Create a MOC

2. Filter dataset

- a. ...by Healpix index
- b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

1. Get Healpix-related data

Quick reminder

Usage in TAP

1. Get
Healpix-related
data

- a. Create an Healpix map
- b. Create a MOC

2. Filter dataset

- a. ...by Healpix index
- b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

a. Create an Healpix map

1a. Create an **Healpix** map - 1/2

```
SELECT ivo_healpix_index(7, POINT('', ra, dec))
        AS hpx_index,
        COUNT(*) AS density
FROM tycho2
GROUP BY hpx_index
```

See the [ADASS poster](#) - M. Taylor, G. Mantelet, M. Demleitner (and the corresponding [paper](#)).

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

1a. Create an Healpix map - 1/2

```
SELECT ivo_healpix_index(7, POINT('', ra, dec))
        AS hpx_index,
        COUNT(*) AS density
FROM tycho2
GROUP BY hpx_index
```

See the [ADASS poster](#) - M. Taylor, G. Mantelet, M. Demleitner (and the corresponding [paper](#)).

Requirements:

- ▶ **new function:**
 - ▶ `ivo_healpix_index(order, POINT) -> BIGINT`
 - ▶ `ivo_healpix_index(order, ra, dec) -> BIGINT`

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

1a. Create an **Healpix** map - 1/2

```
SELECT ivo_healpix_index(7, POINT('', ra, dec))
        AS hpx_index,
        COUNT(*) AS density
FROM tycho2
GROUP BY hpx_index
```

See the [ADASS poster](#) - M. Taylor, G. Mantelet, M. Demleitner (and the corresponding [paper](#)).

Requirements:

- ▶ **new function:**
 - ▶ `ivo_healpix_index(order, POINT) -> BIGINT`
 - ▶ `ivo_healpix_index(order, ra, dec) -> BIGINT`

Optionally:

- ▶ `ivo_healpix_center(order, hpx_index) -> POINT`

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

1a. Create an **Healpix** map - 2/2

Note:

Can be a pretty long execution on large tables and/or with an high Healpix order. However, a database index on a such expression could improve significantly the execution time.

WARNING:

The resulting table will **lack two important pieces of information** about the Healpix indices: the Healpix **scheme** and **order**.

-> **Solution: a UNIQ value? (cf MOC) UCD? UType? Smth else?**

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

Quick reminder

Usage in TAP

1. Get
Healpix-related
data

- a. Create an Healpix map
- b. Create a MOC

2. Filter dataset

- a. ...by Healpix index
- b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

b. Create a MOC

1b. Create a MOC - 1/2

Solution 1: *new function moc_agg*

```
SELECT moc_agg(7, POINT('', ra, dec)) AS mymoc  
FROM tycho2
```

...

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

1b. Create a MOC - 1/2

Solution 1: *new function moc_agg*

```
SELECT moc_agg(7, POINT('', ra, dec)) AS mymoc  
FROM tycho2
```

...

Requirements:

- ▶ **new ADQL function:** `moc_agg(order, POINT) -> MOC` and/or `moc_agg(order, hpIndex) -> MOC`
- ▶ **new VOTable datatype:** `type="VARCHAR"` and `(xtype="adql:MOC" or xtype="MOC")`
- ▶ **add DB aggregate function ; "easy" for PostgreSQL, but what about the other DBMS?**

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

- References
- Proposed ASCII
serialization
- PEG grammar for
MOC-ASCII

1b. Create a MOC - 2/2

Solution 2: *new entry point /moc*

```
MOC_ORDER = 7
MOC_RA    = ra
MOC_DEC   = dec
QUERY     = SELECT ra, dec FROM tycho2 ...
```

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

1b. Create a MOC - 2/2

Solution 2: *new entry point /moc*

```
MOC_ORDER = 7
MOC_RA     = ra
MOC_DEC    = dec
QUERY      = SELECT ra, dec FROM tycho2 ...
```

Requirements:

- ▶ **new entry point:** */moc*
- ▶ **new parameters:** MOC_ORDER and (MOC_INDEX or (MOC_RA and MOC_DEC)).

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

1b. Create a MOC - 2/2

Solution 2: *new entry point /moc*

```
MOC_ORDER = 7
MOC_RA    = ra
MOC_DEC   = dec
QUERY     = SELECT ra, dec FROM tycho2 ...
```

Requirements:

- ▶ **new entry point:** /moc
- ▶ **new parameters:** MOC_ORDER and (MOC_INDEX or (MOC_RA and MOC_DEC)).

Optionally:

- ▶ `FORMAT = 'FITS'` by default (*maybe more later*)
- ▶ `MAXREC` ignored (*but the ADQL keyword TOP should still be possible*)
- ▶ `ORDER BY` ignored (*for performance reasons*)

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

Quick reminder

Usage in TAP

1. Get
Healpix-related
data

- a. Create an Healpix map
- b. Create a MOC

2. Filter dataset

- a. ...by Healpix index
- b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

2. Filter dataset

Quick reminder

Usage in TAP

1. Get
Healpix-related
data

- a. Create an Healpix map
- b. Create a MOC

2. Filter dataset

- a. ...by Healpix index
- b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

a. ...by Healpix index

2a. Filter dataset by **Healpix index**

```
SELECT *  
FROM tycho2  
WHERE ivo_healpix_index(7, POINT('', ra, dec))  
      IN (12,23,68,69,70)
```

Quick reminder

Usage in TAP

1. Get
Healpix-related
data

- a. Create an Healpix map
- b. Create a MOC

2. Filter dataset

- a. ...by Healpix index
- b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

2a. Filter dataset by **Healpix index**

```
SELECT *
FROM tycho2
WHERE ivo_healpix_index(7, POINT('', ra, dec))
      IN (12,23,68,69,70)
```

Requirements:

- ▶ **new function:** `ivo_healpix_index(order, POINT)`
and/or `ivo_healpix_index(order, ra, dec)`

Optionally:

- ▶ `ivo_healpix_center(order, hpx_index) -> POINT`

- Create an Healpix map
- Create a MOC

- ...by Healpix index
- ...by MOC

b. ...by MOC

Quick reminder

Usage in TAP

1. Get
Healpix-related
data

- a. Create an Healpix map
- b. Create a MOC

2. Filter dataset

- a. ...by Healpix index
- b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

2b. Filter dataset by **MOC** - 1/4

Principle: *new region: MOC*

```
WHERE 1=CONTAINS(POINT('', ra, dec), <MOC>)
```

Requirements:

- ▶ **extension of** CONTAINS and INTERSECTS in order to support MOC regions
- ▶ **DBMS support of MOC** in SELECT, CONTAINS and INTERSECTS
- ▶ [...]

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

2b. Filter dataset by **MOC** - 2/4

Solution 1: *STC-s or DALI serialization*

```
WHERE 1= CONTAINS(POINT('', ra, dec),  
                  MOC('2/12-20 5/60')) -- STC-s
```

OR

```
REGION('MOC 2/12-20 5/60')) - STC-s
```

OR

```
REGION('2/12-20 5/60')) - DALI
```

Requirements:

- ▶ [...]
- ▶ **new function:** MOC(STC/s) or MOC(DALI-ASCII)

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

- References
- Proposed ASCII
serialization
- PEG grammar for
MOC-ASCII

2b. Filter dataset by MOC - 3/4

Solution 2: *UPLOAD of a VOTable table with one (or more) MOC column(s)*

```
SELECT t.*
FROM tycho2 AS t
JOIN TAP_UPLOAD.mymoc AS m
  ON 1=CONTAINS(POINT('', t.ra, t.dec), m.moc1);
```

In this example: TAP_UPLOAD.mymoc is a normal table with a column named moc1 of type 'VARCHAR' and xtype 'adql:MOC' or 'MOC'.

Requirements:

- ▶ [...]
- ▶ type="VARCHAR" and (xtype="adql:MOC" (STC/s) or xtype="MOC" (DALI region))

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

2b. Filter dataset by **MOC** - 4/4

Solution 3: *UPLOAD of a MOC in FITS format*

```
SELECT t.*  
FROM tycho2 AS t  
JOIN TAP_UPLOAD.mymoc AS m  
  ON 1=CONTAINS(POINT('', t.ra, t.dec), m.moc);
```

The idea is that the uploaded FITS will actually be represented in TAP_UPLOAD as a one-cell table (column named moc)

Requirements:

- ▶ [...]
- ▶ allowing the UPLOAD of MOC-FITS

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

Proposal

Bringing Healpix
and MOC in TAP

Grégory Mantelet
(ARI/ZAH)

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII



Proposal - 1/2

ADQL extension:

- ▶ **Addition of a new geometry**
 - ▶ MOC(VARCHAR) -> MOC (*the string is an ASCII serialisation*)
- ▶ **Update some geometrical functions**
 - ▶ AREA
 - ▶ CONTAINS
 - ▶ INTERSECTS
- ▶ **Addition of Healpix and MOC functions**
 - ▶ moc_agg(order INT, hpxIndex BIGINT) -> VARCHAR
 - ▶ moc_agg(order INT, position POINT) -> VARCHAR
 - ▶ ivo_healpix_index(order INT, position POINT) -> BIGINT
 - ▶ ivo_healpix_index(order INT, ra DOUBLE, dec DOUBLE) -> BIGINT
 - ▶ ivo_healpix_center(order INT, hpxIndex BIGINT) -> POINT
 - ▶ ivo_healpix_uniq(order INT, hpxIndex BIGINT) -> BIGINT

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References
Proposed ASCII
serialization
PEG grammar for
MOC-ASCII

Extension of TAP:

- ▶ **Addition of a MOC datatype**
 - ▶ STC/s (`xtype="adql:MOC"`) or DALI syntax (`xtype="MOC"`)
- ▶ **New entry point:** `/moc`
- ▶ **SELECT of a MOC** returns a string serialization of the MOC
- ▶ **UPLOAD of a VOTable** with a **FIELD** of datatype MOC
- ▶ *UPLOAD of FITS/FITS-MOC?*

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

What should be discussed

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

What should be discussed

- ▶ **VO-Registration:** easily possible for /moc, but what about the support of MOC region?
- ▶ **Standardisation of another serialization for MOCs:**
 - ▶ ASCII: STC-S or DALI? (*cf next slide for a standard serialization proposal*)
 - ▶ Should there be a more compact non-ASCII serialization?
- ▶ **How to convey the Hpx scheme and Hpx order of an Hpx index or of an entire MOC?**
 - ▶ UCD?
 - ▶ Datatype?
 - ▶ While waiting for a DM: an apart VOTable GROUP?

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

Appendices

Bringing Healpix
and MOC in TAP

Grégory Mantelet
(ARI/ZAH)

Quick reminder

Usage in TAP

1. Get
Healpix-related
data

- a. Create an Healpix map
- b. Create a MOC

2. Filter dataset

- a. ...by Healpix index
- b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII



References

Bringing Healpix
and MOC in TAP

Grégory Mantelet
(ARI/ZAH)

Quick reminder

Usage in TAP

1. Get
Healpix-related
data

- a. Create an Healpix map
- b. Create a MOC

2. Filter dataset

- a. ...by Healpix index
- b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII



References

- ▶ **TAP 1.0 (REC):**
<http://www.ivoa.net/documents/TAP/20100327/>
- ▶ **ADQL 2.0 (REC):** <http://www.ivoa.net/documents/cover/ADQL-20081030.html>
- ▶ **Healpix:** <http://healpix.sourceforge.net/documentation.php>
- ▶ **MOC 1.0 (REC):** <http://www.ivoa.net/documents/MOC/20140602/index.html>
- ▶ **STC-S 1.0 (WD):** <http://www.ivoa.net/documents/STC-S/index.html>
(see 5.2)
- ▶ **DALI 1.1 (PR):** <http://www.ivoa.net/documents/DALI/20170412/index.html> (see 3.3.5 to 3.3.7)

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

Proposed ASCII serialization

Proposed ASCII serialization

```
MOC <mocItem> ( <mocItem>)*
```

, where:

- ▶ `<mocItem>` is defined as: `<hpxOrder>/<hpxIndices>`
- ▶ `<hpxOrder>` must be an integer between 0 and 29 (included).
- ▶ `<hpx-indices>` is a comma separated list of either Healpix indices (i.e. an integer between 0 and $12*4<hpxOrder>$) or ranges of consecutive Healpix indices.

...

Quick reminder

Usage in TAP

1. Get
Healpix-related
data

- Create an Healpix map
- Create a MOC

2. Filter dataset

- ...by Healpix index
- ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

Proposed ASCII serialization - 2/3

...

The format for a range must be:

`<minHpxIndex>-<maxHpxIndex>`

, where `<minHpxIndex>` strictly less than `<maxHpxIndex>`.

Example: *MOC 1/1,3,4 2/4,25,12-14,21*

Healpix indices and Healpix orders can be provided in any order, but it is anyway recommended to list them in ascending order for easier human readability.

It is also recommended, but not required, to represent Healpix cells at the lowest order as possible in order to have a string as short as possible (example: "MOC 2/92-95" could be written "MOC 1/23").

...

- a. Create an Healpix map
- b. Create a MOC

- a. ...by Healpix index
- b. ...by MOC

Proposed ASCII serialization - 3/3

...

However, **the maximum order of the MOC must always be indicated**. If no cell is listed at this order, it must be indicated with an empty list of cells.

Example: *MOC 10/63-65,87 13/*

Final note: DALI or STC-s?

This syntax can be used exactly as such for STC-s.

Example: *MOC 10/63-65,87 13/*

For DALI, just removing the prefix "MOC " should be enough.

Example: *10/63-65,87 13/*

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

PEG grammar for MOC-ASCII

Bringing Healpix
and MOC in TAP

Grégory Mantelet
(ARI/ZAH)

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII



PEG grammar for MOC-ASCII - 1/2

```
/* THE ASCII SERIALIZATION OF A MOC
 * MUST start with "MOC" (case insensitive)
 * and be followed by one or more MOC_ITEMS,
 * all separated by at least one space. */
MOC_ASCII = "MOC"i __ MOC_ITEM ( __ MOC_ITEM)*

/* A MOC-ITEM
 * MUST start with an Healpix order, a '/'
 * and zero or more Healpix indices.
 *
 * Theoretically, ONLY one MOC-ITEM may be composed of
 * ONLY an Healpix order. This item would then represent
 * the maximum order of the MOC, even if no Healpix index
 * of this order is part of the MOC. */
MOC_ITEM = HPX_ORDER '/' HPX_INDICES?

/* AN HEALPIX-ORDER
 * MUST be an integer between 0 and 29 (included). */
HPX_ORDER = [12]?[0-9]
```

...

Quick reminder

Usage in TAP

1. Get
Healpix-related
data

- Create an Healpix map
- Create a MOC

2. Filter dataset

- ...by Healpix index
- ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII



PEG grammar for MOC-ASCII - 2/2

```
...

/* HEALPIX-INDICES
 * CAN be:
 *   - a single HEALPIX-INDEX (e.g. '2398'),
 *   - a range (e.g. '1-9')
 *   - or a list of single indices and/or ranges
 *     (e.g. '1,3,6-98,101'). */
HPX_INDICES = HPX_INDEX (('-'/'',') HPX_INDEX)*

/* AN HEALPIX-INDEX
 * MUST be a long integer number (e.g. '2398').
 *
 * If different from the digit zero, the number
 * should NOT start with zero
 * (e.g. the number '012' should not be considered
 * as an Healpix index). */
HPX_INDEX = ('0' / [1-9][0-9]+)

/* Mandatory spaces (at least one) */
__ = ' '+
```

Bringing Healpix
and MOC in TAP

Grégory Mantelet
(ARI/ZAH)

Quick reminder

Usage in TAP

1. Get Healpix-related data
 - a. Create an Healpix map
 - b. Create a MOC
2. Filter dataset
 - a. ...by Healpix index
 - b. ...by MOC

Proposal

What should be
discussed

Appendices

References

Proposed ASCII
serialization

PEG grammar for
MOC-ASCII

