

**CASA Task “exportasdm”  
Requirements and Design**

Doc #:  
Date: 2009-12-09  
Status: comments included, sent out for approval  
Page: 1 of 10

# CASA task “exportasdm” requirements and design

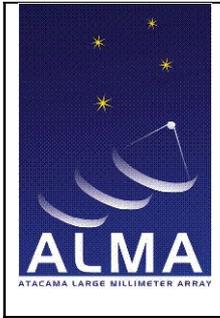
Doc number

Version: 1.0

Status: partially approved

2009-12-09

<b>Prepared By:</b>		
<b>Name and Signature</b>	<b>Organisation</b>	<b>Date</b>
Dirk Petry	ESO	12/04/09
<b>Approved by:</b>		
<b>Name and Signature</b>	<b>Organisation</b>	<b>Date</b>
Paola Andreani	ESO	12/09/09
Andreas Wicenc	ESO	12/09/09
Alan Bridger	UKATC	12/09/09
Lindsey Davis	NRAO	
Francois Viallefond/Michel Caillat	Obs. Paris Meudon	12/09/09
Nick Elias	NRAO	

	<p><b>CASA Task “exportasdm” Requirements and Design</b></p>	<p>Doc #: Date: 2009-12-09 Status: comments included, sent out for approval Page: 2 of 10</p>
--	--	---

## 1 Summary

A new task is to be developed for CASA tentatively scheduled for patch release 3.0.1, full version 3.1.

The three main use cases are:

- 1) enable tests of the archive with simulated data
- 2) enable tests of the online pipeline with simulated data
- 3) store simulated data permanently in the archive for science and other purposes

In all three cases, CASA exportasdm is needed since the CASA simulator only produces MSs. exportasdm is so-to-speak a back-end to the simulator. It is not a tool to submit ASDMs to the archive. The submission tool is to be tested with the ASDMs produced by exportasdm.

The implementation is going to take place in two steps: Version 1 (for release 3.0.1 in March 2010) will cover all requirements by the archive team, i.e. will deliver the exportasdm needed for archive testing. Version 2 (for release 3.1 in Summer 2010) will also cover additional requirements by the pipeline and simulations teams (mainly involves filling of calibration, WVR, and SwitchCycle tables in the ASDM).

The new task will take an existing Measurement Set (simulated or real) and translate it into an ASDM by filling those ASDM tables for which information exists in the MS.

The MS tables involved are: Main, Antenna, Data description, Feed, Flag\_cmd, Field, History, Observation, Pointing, Polarization, Processor, Source, Spectral\_Window, State, Syscal, Weather. For version 2, CASA calibration tables external to the MS will also be used.

The ASDM tables to be filled are (in the order they are read by CASA importasdm): Antenna, SpectralWindow, Polarization, DataDescription, Feed, Field, FlagCmd<sup>1</sup>, History<sup>1</sup>, ExecBlock, Pointing, Processor, Source, Weather, State, Main.

The tables among these which contain binary data are Main, Pointing, and Weather. This means exportasdm will need access to the library for writing binary ASDM data.

---

<sup>1</sup> not present in SDM Tables Short Description document, t.b.d. with Michel Caillat end of December.

	<p><b>CASA Task “exportasdm” Requirements and Design</b></p>	<p>Doc #: Date: 2009-12-09 Status: comments included, sent out for approval Page: 3 of 10</p>
--	--	---

This library is already part of the CASA code (it is used by CASA importasdm). Also for the non-binary data, the implementation will make use of the existing libraries which are already part of CASA and are presently used by the ASDM2MS filler.

Furthermore the ASDM tables ConfigDescription, PointingModel, Receiver, Scan, Station, SubScan (which are not read by CASA importasdm) need to be filled in version 1 since the archive needs to test queries on them. For these tables several details still need to be discussed with Francois Viallefond end of December 2009. This document will be updated then.

All UIDs for the references between tables of the output ASDM will be set using the scheme “uid://X0/X1/X<running>” where X0 (the archive ID) and X1 (the range ID) are provided as input parameters to exportasdm, and X<running> is assigned by exportasdm autonomously starting at 0 and giving a different number for each table.

The SBSummary Table will not be produced by exportasdm but will later be attached by external software when the ASDM is connected with a matching APDM.

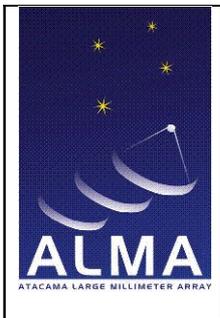
## **2 Overall design and data flow for version 1**

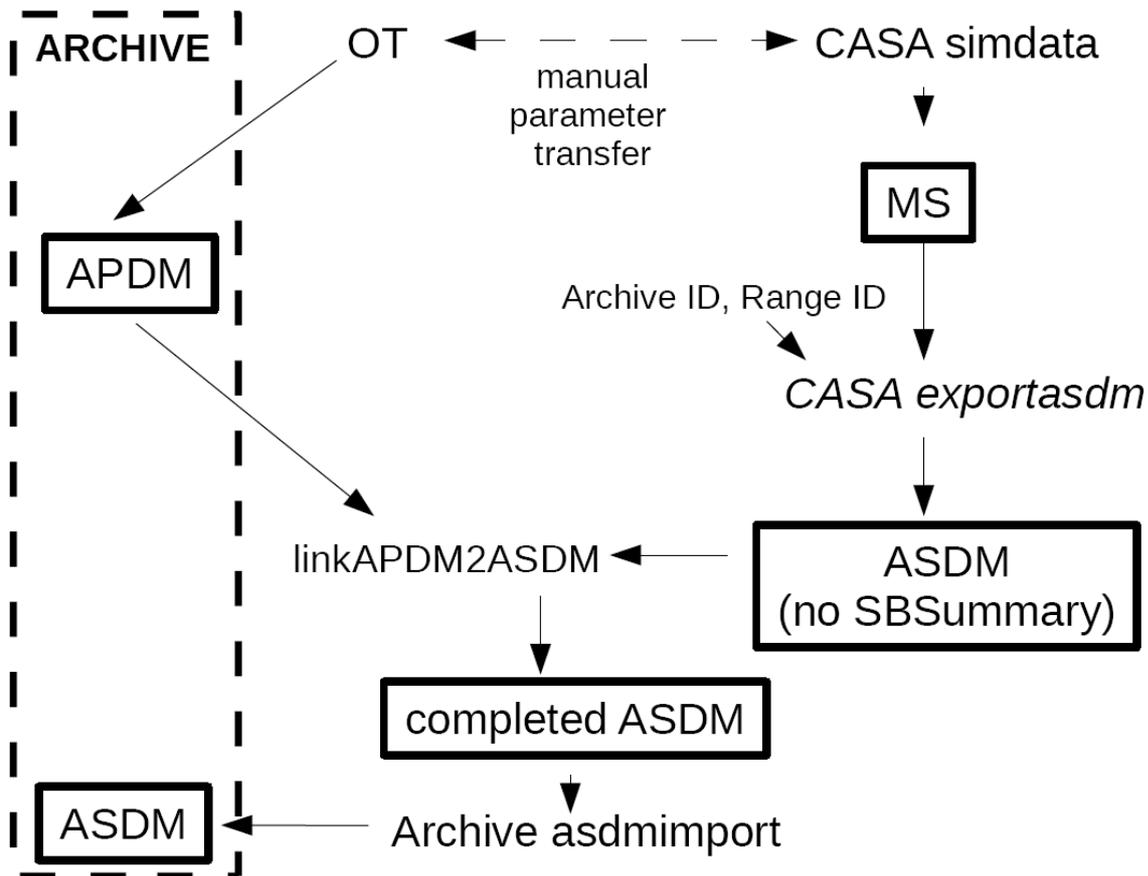
### **2.1 Data flow for exportasdm version 1**

The OT creates an APDM and a matching set of simulations is generated with CASA simdata. (Alternatively, if a simulated data set already exists, the OT is run to produce an APDM matching the simulation.) The matching is ensured by manual setting of the right parameters. The exportasdm task is then used to turn the simulated MS into an ASDM. That ASDM does not contain an SBSummary table and all UIDs have been set to the value uid://X0/X1/X<running> (see above). The thus generated APDM and ASDM are taken by another tool “linkAPDM2ASDM” which creates the SBSummary table and sets the UIDs to their final values.

The “linkAPDM2ASDM” tool will be developed by the archive team.

The OT needs to submit the APDM to the archive before linkAPDM2ASDM is run because only in the archive submission process the UIDs in the APDM get their final values which can then be transferred to the ASDM.

	<p><b>CASA Task “exportasdm” Requirements and Design</b></p>	<p>Doc #:  Date: 2009-12-09  Status: comments included, sent out for approval  Page: 4 of 10</p>
--	--	--

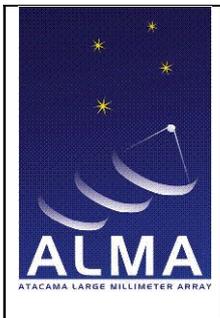


**Fig 1: Dataflow for version 1 of exportasdm.**

The Archive ID given to CASA exportasdm needs to follow the conventions for Archive ID assignment (see references). For simulations, an archive ID starting with the letter “S” should be used. The range ID is controlled by manual book keeping.

## 2.2 Data flow for exportasdm version 2

For version 2 the requirements are still being discussed. The dataflow will be extended only by adding the input of CASA calibration tables from which then ASDM calibration tables will be filled, probably the CalGain table.

	<p><b>CASA Task “exportasdm” Requirements and Design</b></p>	<p>Doc #:  Date: 2009-12-09  Status: comments included, sent out for approval  Page: 5 of 10</p>
--	--	--

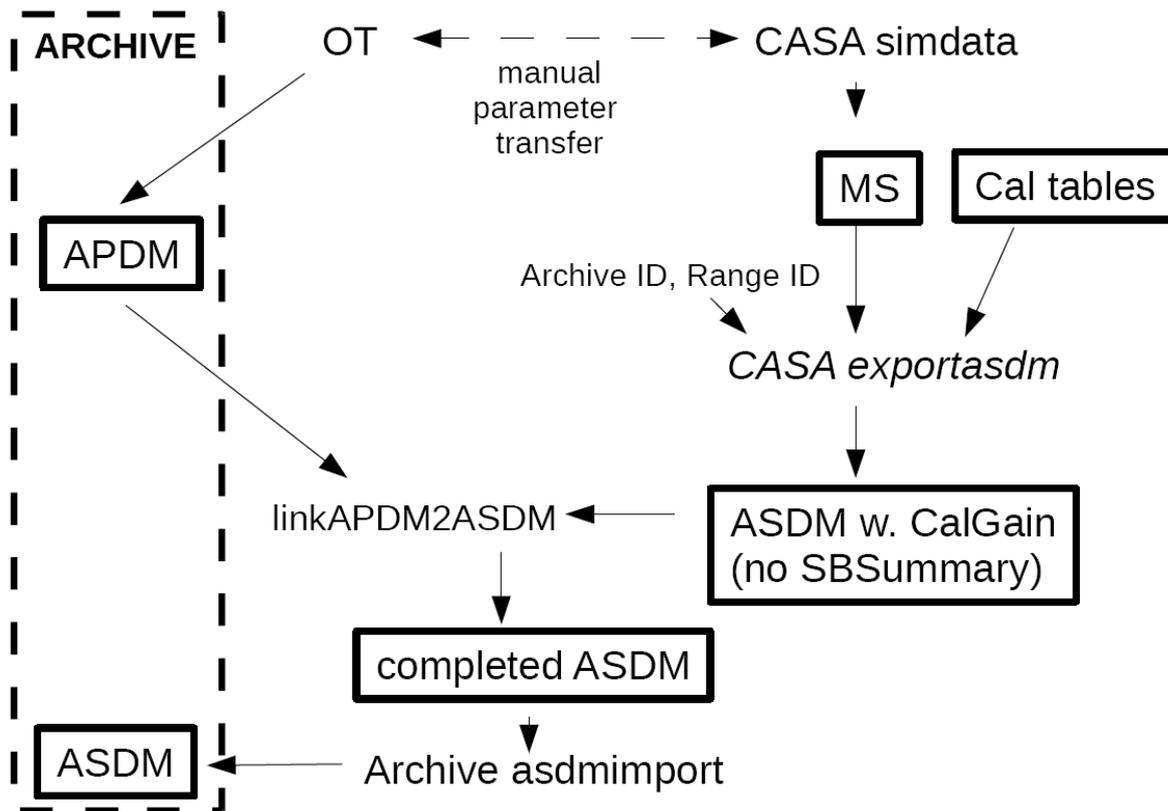


Fig. 2: Dataflow for version 2 of exportasdm.

### 2.3 Additional thoughts for version 2

Here we record several comments which were made to draft 1 of this document in order to keep them for the design of exportasdm version 2.

Should exportasdm use CASA calibration tables or other input to fill Cal tables in the ASDM? The MS itself does not contain any information which could be used to generate ASDM Cal tables.

Yes, this functionality should be put at least in the ms.toasdm tool. CASA can independently

- produce noise-free "perfect" ms,
- produce cal tables that can be used to corrupt said ms
- actually corrupt the visibilities and make new ms

So the option the output the first two items into ASDM would be useful.

	<p><b>CASA Task “exportasdm” Requirements and Design</b></p>	<p>Doc #: Date: 2009-12-09 Status: comments included, sent out for approval Page: 6 of 10</p>
--	--	---

The ASDM presently contains a place holder table, CalGain, for the calibration results obtained in Casa. This table may not be well designed for that intent because F. Viallefond never got feed-back from the CASA specialists of the calibration scheme. F.V. wishes very much to have this feed-back to properly design the ASDM table(s) for the calibration results obtained with CASA.

The use-case of having both telcal-produced cal table and CASA cal table side-by-side (and being able to compare) may be important. If two stub tables were present in the ASDM model for CalGain, then one could compare telcal's calibration, the pipeline calibration, and offline calibration or the corrupting gain. It may be an important use case to compare the pipeline's cal solution with the table that was used to corrupt the data in the first place (so-called "controlled corruption")

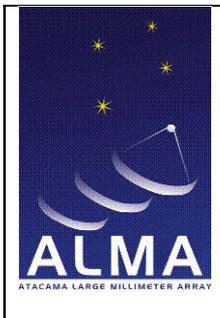
Another wrinkle in the design is that CASA has numerous different kinds of cal tables (corresponding to the different terms in the measurement equation). Putting their product into a single gain table would probably work mathematically but would not be easily reversible.

Some PI's might want to put user-reduced data into the archive, if they want the community to use this data. This does not happen automatically.

There has been mention at NAASC of re-reducing, or additional processing, etc e.g. combining several programs in an offline processing and then re-releasing that to the public. So there will likely be inputs to the archive other than simulation and the actual array.

Related improvements to CASA simdata:

- coordination of cal table I/O from casa as discussed above
- improvements to CASA:simdata to actually fill the weather table with information correlated with the (e.g. phase) corruptions applied to the data - this is what I think pipeline wants at some point, but is further out and will have to be negotiated against other NAASC priorities if I am to be the developer (which seems most likely)
- improvements to CASA:simdata to create synthetic WVR data consistent with corruptions applied to the visibilities. Again, this is not a ridiculous request given that simdata already uses an atmospheric delay screen and the aatm library but longer-term.

	<p><b>CASA Task “exportasdm” Requirements and Design</b></p>	<p>Doc #: Date: 2009-12-09 Status: comments included, sent out for approval Page: 7 of 10</p>
--	--	---

- more complex correlator modes understood by simdata (I think we'll be completely swamped with real data long before we get here, probably before we get to some of the other things too).

### 3 Task User Interface

The CASA task exportasdm will use the same nomenclature as importasdm. The short description given by `help exportasdm` in casapy will be as follows:

Convert a CASA visibility file (MS) into an ALMA Science Data Model.

Keyword arguments:

```
vis          -- MS name,
              default: none
```

```
asdm -- Name of output ASDM file (directory),
      default: none; example: asdm='ExecBlock3'
```

```
datacolumn -- specifies which of the MS data columns (DATA,
              CORRECTED_DATA, or MODEL_DATA) should be used as the
              visibilities in the ASDM, default: DATA
```

```
archiveid -- the X0 in uid://X0/X1/X<running>
            default: "S0"
```

```
rangeid -- the X1 in uid://X0/X1/X<running>
          default: "X1"
```

```
verbose      -- produce log output, default: True
```

```
showversion -- report the version of the ASDM class set,
            default: True
```

All UIDs throughout the output ASDM are set following the scheme `uid://X0/X1/X<running>` where `<running>` is a number which is different for each table. When setting the `archiveid`, the user should observe the “Archive Id Assignment Document” (see references).

	<p><b>CASA Task "exportasdm" Requirements and Design</b></p>	<p>Doc #: Date: 2009-12-09 Status: comments included, sent out for approval Page: 8 of 10</p>
--	--	---

## 4 Implementation

The implementation will follow the standard CASA scheme. A class MS2ASDM will be created in `code/msvis/implement/MSVis` to contain all necessary additional C++ code to perform the conversion. The class will use the necessary ASDM classes (already part of the CASA code base). In the ms tool, a new method ```toasdm``` will be created which will use the class MS2ASDM. Then the task ```exportasdm``` will be created based on the `ms.toasdm()` method.

The parameters of the ```toasdm``` method will be:

```
ms.toasdm(asdm, datacolumn, apdm, execblockuid, sbsummaryuid,
projectuid)
```

```
asdm -- Name of output ASDM file (directory)
      default: none; example: asdm='ExecBlock3'
```

```
datacolumn -- specifies which of the MS data columns (DATA,
CORRECTED_DATA, or MODEL_DATA) should be used as the
visibilities in the ASDM. default: DATA
```

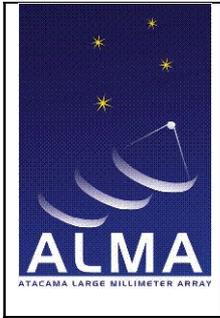
```
archiveid - the X0 in uid://X0/X1/X<running>
           default: "S0"
```

```
rangeid - the X1 in uid://X0/X1/X<running>
          default: "X1"
```

The method will operate on the MS currently attached to the ms tool with the usual messages if none is attached.

### 4.1 Tables with existing conversion example

The following tables are read by CASA `importasdm`: Antenna, SpectralWindow, Polarization, DataDescription, Feed, Field, FlagCmd, History, ExecBlock, Pointing, Processor, Source, Weather, State, Main. Hence there is an example in the `importasdm` code of how to convert them to the corresponding MS data. CASA `exportasdm` will use this example to derive how to fill the ASDM tables from the MS.

	<b>CASA Task “exportasdm” Requirements and Design</b>	Doc #: Date: 2009-12-09 Status: comments included, sent out for approval Page: 9 of 10
--	---	---

## 4.2 Tables requiring special treatment

The following tables are not read by CASA importasdm but need to be written by CASA exportasdm version 1: ConfigDescription, PointingModel, Receiver, Scan, Station, SubScan. Hence there is no inverse example of how to write them in importasdm. Their creation therefore has to be discussed in more detail. This will be done at a meeting in Paris end of December 2009 and this document will be revised accordingly.

## 5 Testing

The stakeholders (archive and pipeline) shall provide a set of at least five test cases each with the necessary input data.

The developer will create a unit test script for the tool method and the task based on these example cases and perform testing until he is convinced that the tool performs as expected.

He will then hand over the software to the stakeholders and they will perform tests as they see fit to confirm the correctness of the code.

	<p><b>CASA Task “exportasdm” Requirements and Design</b></p>	<p>Doc #: Date: 2009-12-09 Status: comments included, sent out for approval Page: 10 of 10</p>
--	--	--

## 6 Timeline

- 9 Dec 09: Finish detailed design and Python coding for version 1. Start implementation of C++ code.
  
- 21 Dec 09: Meet with Michel Caillat and Francois Viallefond to sort out possible problems of the implementation which have arisen until then and to decide how to fill the tables ConfigDescription, PointingModel, Receiver, Scan, Station, and SubScan.
  
- 11 Jan 10: Archive team provides all test data with a description of expected output.
  
- 1 Feb 10: Finish preliminary implementation. Start unit testing.
  
- 8 Feb 10: Hand over task version 1 to archive team for final testing.
  
- 15 Feb 10: Release of exportasdm to ALMA-internal users.
  
- Mar 2010: Release of exportasdm version 1 in CASA patch 3.0.1
  
- 15 Mar 10: Finish requirements for version 2 of exportasdm and start coding
  
- Apr 2010: Pipeline and simulations teams provide all test data with a description of the expected output.
  
- Jul 2010: Release of exportasdm version 2 in CASA release 3.1

## References

- Kemball, A.J. & Wieringa, M.H., 2000, “MeasurementSet definition version 2.0”, <http://aips2.nrao.edu/docs/notes/229.ps.gz>
  
- Viallefond, F. & Lucas, R., 2009, “SDM Tables Short Description v. 2009-03-03”, <http://almasw.hq.eso.org/almasw/pub/HLA/ASDImplementationFBT/SDMTablesV1.pdf>
  
- Wiecenec, A., et al., 2009, Archive ID Assignment Document (Wiki) <http://almasw.hq.eso.org/almasw/bin/view/Archive/ArchiveIdAssignment>