



immask

January 12, 2017

Abstract

This task makes a mask image which includes only those pixels whose values obey the specified selection expression.

1 Instruments/Modes

The task is not XMM-specific.

2 Description

The task is designed to make a mask image (ie an image of boolean values) by applying a selection expression to the pixel values of an input image. These pixel values are assigned to a variable named 'VAL' in the selection expression. Hence for example if one wished to mark as TRUE all pixels of the input which exceeded 7.3 in value, one would call **immask** with **expression='VAL > 7.3'**. All the syntax provided by **selectlib** is available for this selection expression.

The output mask image is the same size as the input image. All attributes (keywords), including WCS keywords, but excluding keywords related to any data subspace (DSS) in the input image, are copied to the output image. The input image is expected to be in the primary array extension of the input dataset and only this extension is written to the output.

The data type of the output is not actually boolean, but 8-bit integer: the values are either 0 or 1.

3 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints
-----------	------	------	---------	-------------

imageset	yes	dataset	dummy_default	
-----------------	-----	---------	---------------	--

The input image dataset.

maskset	no	dataset	mask.ds	
----------------	----	---------	---------	--

The output mask dataset.



tempset	no	dataset	tempset.ds	
----------------	----	---------	------------	--

Name of temporary dataset used during the filtering operation.

tempimageset	no	dataset	tempimage.ds	
---------------------	----	---------	--------------	--

Name of a temporary cloned image dataset used while writing the output.

expression	no	string		
-------------------	----	--------	--	--

The expression used to generate the boolean output.

4 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

badArrayDataType (*error*)

The data type of the input image was not accepted.

5 Input Files

1. A FITS dataset containing a 2-dimensional image array in the primary extension. The datatype of the image may be 8-bit integer, 16-bit integer, 32-bit integer, 32-bit real or 64-bit real.

6 Output Files

1. A FITS dataset containing a 2-dimensional image array in the primary extension, the same size as the input array and containing all of the keywords (except those pertaining to DSS) of the input. The datatype of this image is 8-bit integer but its values are restricted to 0 and 1.

7 Algorithm

References