esrcselect

January 12, 2017

Abstract

Selects sources from an EPIC product source list.

1 Instruments/Modes

Instrument	Mode
EPIC	Imaging

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

This task has been completely rewritten at version 2.0, with a new parameter interface; however since the general thrust of the task is the same, the same name has been retained. The previous functionality can be achieved with this new version, although several invocations on the 'new' esrcselect may be required to duplicate the result of the 'old'.

Source list tables produced by **eboxdetect** or **emldetect** generally contain more than 1 table row per source. If one wants to filter the list so as to discard all rows pertaining to a given source, depending on column values at just 1 of these rows, then one finds this rather difficult to do using the standard fits or **sas** tools. It is to perform such source-based filtering that **esrcselect** has been written.

There are two ways in which the list can be filtered, depending on the value of the parameter `style`:

1. `style`='filter': in this mode, the task accepts a straightforward selection expression from parameter `expression`.
2. `style`='sort': in this mode, the user provides an arithmetic expression involving column names via parameter `sortexpression`; the rows are sorted in increasing order of the result; finally the first `maxn` sources are retained.

Clearly, for an expression involving column values to be meaningful on a source-by-source rather than a row-by-row basis, only values from 1 row per source must be included in the expression. This requirement is the same regardless of whether the result of the expression is a logical value (`style`=‘filter’, `expression`) or a numerical value (`style`=‘sort’, `sortexpression`). So, how to identify the row? In source lists created by **eboxdetect** or **emldetect** there are columns named `ID_INST` and `ID_BAND`. For a given source, each row has a unique combination of these column values, hence they can be used to identify a row per source. Parameters `idinst` and `idband` are provided to allow this.

It is also necessary to know the name of a source ID column, ie a column which has a value unique to the source. The parameters `withidcol` and `idcol` allows the user to specify the name of this column. If `withidcol` is not set, the task looks first for a column called `BOX_ID_SRC`, then one called `ML_ID_SRC`; if neither is found, an error results.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints

tempset	no	dataset	tempset.ds	

The name of a temporary data set (for pipeline or other parallel useage).

insrclisttab	yes	table		

The dataset+table name of the input source list.

outsrclistset	no	dataset	out_src_list.ds	

The name of the output source list dataset.

style	no	string	filter	filter—sort

If `style`=‘filter’, parameter `expression` is read and the source list is filtered to retain only sources for which the result of `expression` is TRUE; if `style`=‘sort’, parameter `sortexpression` is read and evaluated, the sources are sorted in increasing order of the result and the first `maxn` sources retained. In either case, the expression is applied only to those rows which have the combination of `ID_INST` and `ID_BAND` specified via parameters `idinst` and `idband`.

expression	no	string		

This parameter is read if `style`=‘filter’. The source list is filtered to retain only sources for which the result of `expression` is TRUE. The expression is applied only to those rows which have the combination of `ID_INST` and `ID_BAND` specified via parameters `idinst` and `idband`.

sortexpression	yes	string		

This parameter is read if `style`=‘sort’. `sortexpression` is evaluated and the sources are sorted in increasing order of the result and the first `maxn` sources retained. `sortexpression` is evaluated only for those rows which have the combination of `ID_INST` and `ID_BAND` specified via parameters `idinst` and `idband`.

maxn	yes	int		0 <`maxn`

This parameter is read if `style`=‘sort’. See description of parameter `sortexpression`.

withidcol	no	bool	no	

If this is left at ‘no’, the task will look first for a column called `BOX_ID_SRC`, then one called `ML_ID_SRC`. If ‘yes’, the user can specify the column name via `idcol`.

idcol	yes	string		

This parameter is read if `withidcol`='yes'. It specifies the name of a column in the input source table. This column should be of data type INTEGER32 and its values should be unique to each source.

idinst	no	int	0	0 ≤`idinst`

The value of column `ID_INST` to which to apply either `expression` or `sortexpression`.

idband	no	int	0	0 ≤`idband`

The value of column `ID_BAND` to which to apply either `expression` or `sortexpression`.

5 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.

noSources *(error)*
The input list contained no sources.

noQualifyingSources *(error)*
The input list contained no sources with the specified combination of `idinst` and `idband`.

noSourcesSelected *(error)*
No sources passed the specified filtering expression.

badStyleValue *(error)*
The value of the `style` parameter was not recognized.

6 Input Files

1. Source list table with the following columns:
   - An ID column of data type INTEGER32. Each source must have a unique value of this column, and all rows pertaining to that source must have the same value. The column name is supplied via parameter `idcol`.
   - Columns `ID_INST` and `ID_BAND` (they may be of any integer data type). At most 1 row may have any given combination of `ID_INST`, `ID_BAND` and the `idcol` column.
   - All columns mentioned in the relevant parameter of `expression` or `sortexpression`.

7 Output Files

1. A copy of the input source list, minus the deleted rows.

8 Algorithm

```
Later...
```

9 Comments

10 Future developments

References