

Euredit utility function: data_conv

1 Purpose

data_conv converts data into an Euredit XML file.

2 Specification

```
#include <euredit_sys.h>
```

```
void data_conv (char *dname, char *ifname, char *xmlfname, char *xslfname,
                char *xdrfname, long n, long m, long del_type, double v_bound,
                long head, char **v_name, long v_type[], long n_cat[],
                long cat_val[], long maxcat, long info[2])
```

3 Parameters

dname

Input: the name of the data set.

ifname

Input: the file name of the data file to be converted.

xmlfname

Input: the file name of the output Euredit XML data file.

xdrfname

Input: the file name of the Euredit output XDR associated with the XML file.

xslfname

Input: the file name of the Euredit output XSL associated with the XML file.

n

Input: the number of observations in the data.

Constraint: $n \geq 1$.

m

Input: the number of variables in the data.

Constraint: $m \geq 1$.

del_type

Input: indicates the delimiter used.

If **del_type** = 0 space (Note that with a space delimiter missing values must be indicated by a non-numeric character.)

If **del_type** = 1 tab.

If **del_type** = 2 comma.

Constraint: $0 \leq \text{del_type} \leq 2$.

v_bound

Input: the lower bound for a valid value (any value below **v_bound** will be treated as not applicable).

head

Input: the number of header lines in the data file.

v_name[m]

Input: the names of the variables.

v_type[m]

Input: the type of variable.

If **v_type**[*i*] = 1 continuous variable.

If **v_type**[*i*] = 2 categorical/nominal variable.

If **v_type**[*i*] = 3 categorical/ordinal variable.

Constraint: $1 \leq \text{del_type} \leq 2$.

n_cat[m]*Input:* the number of categories present if a categorical variable.**cat_val[m*maxcat]***Input:* the categories for the categorical variables. The categories for the i th variable are stored in **catval**[$i*\mathbf{maxcat}+j$] for $j = 1, 2, \dots, \mathbf{n_cat}[i]$.**maxcat***Input:* the maximum number of categories in any categorical variable.**info***Output:* information on the success of the function call.**info**[0] = 0: the function successfully completed its task.**info**[0] = i : the specification of the i th formal parameter was incorrect, $i = 1, 2, \dots, 16$.**info**[0] = 51: error parsing file.**info**[0] = 99: the function failed to allocate enough memory.**info**[1] contains additional information for system debugging.