RCS.Carbon.CPP.DLL

Version 30 November 2022

This document describes the RCS.Carbon.CPP.DLL C++ engine interface

RETURN FORMAT

All public functions return this format

```
Result=success
Format=lines
Data=tag1=value1
tag2=value2
…
```

Result= can be success or fail

Format= indicates the format of what follows Data=. It can be id, lines, tree, strings, string … still settling on these

Data= is followed by the data, possible several lines

In all examples below the initial Result=..Format=..Data= is not shown.

RCSSYSTEM (CMD ITEM PARAM)

cmd can be:

GetVersion

Item,Param blank

Return is version number

GetBuild

Item,Param blank

Return is build date

AppPath

Item,Param blank

Returns the full path to application

GetFreeLicence

Item=email (optional),Param blank

Obtains a free licence, which has a few restrictions but gives you access to the full api and the RCS public jobs. Return is details of the licence.

LoginId

Item=userid, Param=password

Obtains a full desktop licence that can also be used online.

Return is details of the licence.

LogoutId

Item,Param blank

Reduces the user count on the current userid in the licence database.

GetLicenceName

Item=name, Param=password

Obtains a full online licence. Usually this is an Analyst licence so DP functions will be unavailable.

GetCustomerList

Item,Param blank

Returns the list of Cloud Customer accounts available to the current licence.

GetJobList

Item=CloudCustomerAccount, Param blank

Returns the list of jobs available to the current licence under a particular CloudCustomerAccount.

GetVartreeList

Item=CloudCustomerAccount, Param jobname

Returns the list of vartrees available to the current licence under a particular CloudCustomerAccount/job.

---- not documented? ----

GetUser

Item=userid, Param=blank

Return licence details for a user id.

SetDebug

Item=path eg "d:\local\temp\rcsdebug.txt"

Sets the path for writing a debug file.

Any calls in code thelib->WriteDebug() will write here.

Turn off by setting debug to ""

OpenLocalContext (path)

Path = job path eg "c:\ruby\jobs\demo"

This is the starting point for a local job. It takes one parameter, the job path, and returns a session id that will be required for all other calls.

Return is the session id.

OpenAzureContext (customer, job)

Customer=CloudCustomerAccount (eg rcsruby), Job = jobname

This is the starting point for an online job. It takes two parameter, the customer account and job name, and returns a session id that will be required for all other calls.

Return is the session id.

Close Context (id)

ID = session id from an open call

Closes the job/context making the session unknown.

Variables (id, cmd, item, param)

ID is the session id from an Open call

CMD can be:

GetVarTree

Item, param blank

Returns the vartree details in TREE format text

```
0- .Periods
1- Count
1- Month
1- Year
0- .Demographics
1- Gender Gender
1- Region 1- Location
```

VarMeta

Item = varname, Param blank

Returns variable codeframe(s) including nets and arithmetic.

A simple variable like

```
0+ Codeframes
1+ gender
2- 1=Male
2- 2=Female
0+ MetaData
1- Type=Simple
1- Desc=Respondent Gender
1- Base0=cwf
1- DataMJD=0
1- Cases=10000
```

… a hierarchic like

```
0+ Brand
1- 1=BrandX
1- 2=BrandY
1- 3=BrandZ
0+ Statement
1- 1=Well-Known
1- 2=Trustworthy
1- 3=Easy to get
1- 4=High quality
1- 5=Good reputation
1- _net(1/3)=Mix1
1- #c8+c9+c10=Mix2
0- MetaData
1- Desc=Brand Image
1- Base0=cwf
1- Base1=cwf
1- Type=Hierarchic
1- DataMJD=0
1- Cases=10000
```

VarFrames

Item = variable, param blank

Returns the names of codeframes in the variable

A simple variable like

```
gender
```

… a hierarchic like

```
Brand
Statement
```

VarFrame

Item = frame (region, bim.brand …), param blank

Returns the codes, nets and arithmetic for a specific codeframe. A simple name is like ‘region’ and a hierarchic like ‘bim.brand’

```
1=NE
2=SE
```

```
3=SW
4=NW
_net(1;4)=North
_net(2;3)=South
_net(1;2)=East
_net(3;4)=West
#sum#(1/2)+sum#(3/4)=sumtest
#avg#(1/2)=avgtest
#cmn=Mean
```

Construct

Item = variable name, Param blank

This reads a MET file and runs the construction if available.

Returns a message 'constructed ok'

Generate

Item = variable name, Param blank

This reads generate info in [Generate], [DateGen] or [Grid] sections in the MET and run the first available to generate construct lines for a subsequent Construct call. You don't need to say if it is "lines" "dates" or "grid", only one of the sections will be filled out so the variable knows which sort it is.

Returns a message 'generated ok'

ConfirmVar

Item = variable name, Param blank

This ensures the variable is available and in the vartree.

Returns a message 'confirmed'

ListVarTrees

Item, param blank

Lists the vartrees available in the current job

Export

Item = varlist, Param = settings

Varlist is a comma separated list eg "age,gender,region" – must be simple vars but can be multiresponse.

Settings is a \n separated list of settings such as

```
Format=csv
FirstLineIsVars=true
CodeLabels=true
BinaryMulti=true
QuoteText=true
QuoteNumeric=true
Missing=*
FileName=
```

If FileName is supplied output is written to that and return is "ok", otherwise Return is text file containing case data.

Format can be csv, tsc, tableau.

Tableau export asserts BinaryMulti=true, Delimiter=comma, and requires FileName, which is treated as a filestem, writing three files

filestemCodes.csv (with CodeLabels false)
filestemLabels.csv (with CodeLabels false)
filestemMeta.csv (Questions, Question Text)

Import

Item = source, Param = settings

Source is input source file eg d:\\1jobs\\demo\\testimport.csv

Currently supports tsv and csv with no settings.

Coding

Item=\n delimited settings

First setting Type can be

```
AutoCodeToSingle
AutoCodeToMulti
MatchReference
DecodeToText
```

Other settings

```
Input
Output
Reference
KeepPuncChars
Fuzz1
Fuzz2
Fuzz3
Fuzz1Length
Fuzz2Length
```

Return is a success message with timing.

ValidateExp

Item=filter or weight expression

This checks the expression is sensible and all mentioned variables are available.

ReadZCD

Item = varname

A test call to confirm zcds read correctly.

GENTAB (ID, TOP, SIDE, FLT, WGT, SPEC, DSP)

Params: id, top, side, filter, weight, specflags, displayflags

Return is display tab text in various formats

This could have been a call under the general Table function but has so many parameters it's worth standing alone.

Top, side, filter, weight are standard GenTab syntax.

The DisplayFlag Output.Format can be

```
tsv
csv
ssv
oxt
diamond
none
```

Specflags - \n separated list, can be:

```
Casefilter=case(1)\n
InitAsMissing=true\n
ExcludeNE=false\n
PadHierarchics=false\n
ArithOverStats=false\n
```

DisplayFlags - \n separated list, can be:

See full list of display props in appenbdix

TABLE(ID, CMD, PARAM)

ID is the session id from an Open call

CMD can be:

GetSpecDefaults

Param blank

This returns the default spec properties \n separated.

```
Casefilter=case(1)\n
InitAsMissing=true\n
ExcludeNE=false\n
PadHierarchics=false\n
ArithOverStats=false\n
```

GetProps

Param can be "All" or \n separated list of display flag names

Return format is \n separated list of display flags

SetProps

Param is \n separated list of flags

If flags include Output.Format other than None the return is the table in that format.

Format

Param = output format "tsv", "csv", "ssv", "html", "oxt", "diamond"

Result is table in that format

RawTabDiagnostics

No parameters.

Each line is rawtab in memory: spec then miliseconds

```
RawTabs Diagnostic
```

```
Gender Region {} [] 104
Age Region {} [] 90
Gender UBA {} [] 149
Age UBA {} [] 151
```

ClearRawTabs

No parameters.

Clears RawTabs from memory

Timings

No parameters.

Returns table run times and rawtab diagnostics

SaveCBT

Param = job relative save name

Saves current report in CBT format

LoadCBT

Param = job relative save name

Loads current report from CBT format

ScriptToTree

Param = script spec for axis

Converts script spec for an axis to the text form of a tree

Eg

```
Param = {age(1)},gender(cwf;*)
```

result

```
[filter=age(1)]
(codeframe=gender%cwf)
code=cwf
code=%1
code=%2
()
```

TreeToScript

Param = tree form of axis axis

Reverse of ScriptToTree

LoadAxis

Param = axis name in job/Specs folder

Loads .axs file and returns script form

OUTPUT(ID, CMD, PARAM1, PARAM2)

For outputting a number of tables to the same file or string.

Command can be:

Open

Param1 = format "tsv", "csv", "ssv", "html"

Param2 is destination job relative file path (eg docs\test.tsv)

Param2 can be empty – result will be tables in one string.

This opens a file and does any other preparation.

Table

No parameters.

Output the current table

Line

Param1 = text

Send the passed line, or many lines \n delimited, to the output.

Close

No parameters.

Close the file or return the assembled string output.

Message

No parameters.

Return any message from the process – useful if a call fails.

An example sequence for writing a number of tables to an html file might be:

```
string s = Receive(Output(ID, "open", "htm", "docs\demo.htm"));
s = Receive(GenTab(ID, "gender", "region", "", "",sp, dp ));
s = Receive(Output(ID, "table"));
s = Receive(Output(ID, "line", "testing interpolation"));
s = Receive(GenTab(ID, "age", "region", "", "",sp, dp ));
s = Receive(Output(ID, "table"));
s = Receive(Output(ID, "close"));
```

FILES(ID, CMD, FNAME, PARAM)

Command can be:

ReadFile

Param1 is job relative file path eg "CaseData/Age.met"

Return data is contents of utf8 file

WriteFile

Param1 is job relative file path eg "CaseData/Age.met"

Param2 is text to write

CODE FRAGMENTS FOR C# DRIVER

Main Form DLL links

All DLL functions have to be declared with a DLLImport prefix

```
public partial class RubyDLLTestForm : Form
{
[DllImport(@"RCS.Carbon.CPP.dll", CallingConvention =
CallingConvention.StdCall, CharSet = CharSet.Unicode)]
static extern IntPtr RCSSystem(string cmd, string item, string
param);
[DllImport(@"RCS.Carbon.CPP.dll", CallingConvention =
CallingConvention.StdCall, CharSet = CharSet.Unicode)]
static extern IntPtr OpenLocalContext(string vtrpath);
[DllImport(@"RCS.Carbon.CPP.dll", CallingConvention =
CallingConvention.StdCall, CharSet = CharSet.Unicode)]
static extern IntPtr OpenAzureContext(string connect, string job);
[DllImport(@"RCS.Carbon.CPP.dll", CallingConvention =
CallingConvention.StdCall, CharSet = CharSet.Unicode)]
static extern IntPtr CloseContext(string id);
[DllImport(@"RCS.Carbon.CPP.dll", CallingConvention =
CallingConvention.StdCall, CharSet = CharSet.Unicode)]
static extern IntPtr GenTab(string id, string name, string top,
string side, string filter, string weight, string sflags, string
dflags);
[DllImport(@"RCS.Carbon.CPP.dll", CallingConvention =
CallingConvention.StdCall, CharSet = CharSet.Unicode)]
static extern IntPtr Variables(string id, string cmd, string item,
string param);
[DllImport(@"RCS.Carbon.CPP.dll", CallingConvention =
CallingConvention.StdCall, CharSet = CharSet.Unicode)]
static extern IntPtr Testing(string id, string cmd, string item,
string param);
[DllImport(@"RCS.Carbon.CPP.dll", CallingConvention =
CallingConvention.StdCall, CharSet = CharSet.Unicode)]
static extern IntPtr Table(string id, string cmd, string param);
[DllImport(@"RCS.Carbon.CPP.dll", CallingConvention =
CallingConvention.StdCall, CharSet = CharSet.Unicode)]
static extern IntPtr Files(string id, string cmd, string fname,
string param);
[DllImport(@"RCS.Carbon.CPP.dll", CallingConvention =
CallingConvention.StdCall, CharSet = CharSet.Unicode)]
static extern IntPtr Output(string id, string cmd, string param1 =
null, string param2 = null);
```

A wrapper function to organise DLL returns

```
string Receive(IntPtr p)
{
return Marshal.PtrToStringUni(p);
Confidential. Copyright Red Centre Software 2020-21 Page 12 of 12
}
```

… used like this

```
string s = Receive(RubySystem("GetVersion", ""));
```

General Sequence

Login

```
string s = Receive(RCSSystem("LoginId", id, psw));
```

Open job

Start with a local context (ie select job)

```
string result = Receive(OpenLocalContext (jobpath));
```

jobpath is the job directory eg "c:\ruby\jobs\demo"

The result will be

```
Result=success
Format=id
Data=randomid
```

Extract the data string, which will be used as ID in all subsequent calls.

Get the Vartree

```
string s = Receive(Variables(ID,"GetVarTree", "RubyLib",""));
```

Run a Table

```
string s = Receive(GenTab(ID, top, side, flt, wgt, sflg, dflg));
```

Change Display Properties

```
Table(ContextID, "SetProps", props);
```

Redisplay Table

```
string s = Receive(Table(ID, "Format", "tsv"));
```

APPENDIX – DISPLAYPROPS

```
Titles.Name.Visible=True
Titles.Name.Font.Name=
Titles.Name.Font.Size=10
Titles.Name.Font.Style=Regular
Titles.Name.Font.HAlign=Left
Titles.Name.Font.VAlign=Top
Titles.Name.Font.Color=255,0,0,0
Titles.Name.Font.Back=255,255,255,255
Titles.Name.Font.Wordwrap=False
Titles.Top.Visible=True
Titles.Top.Font.Name=
Titles.Top.Font.Size=10
Titles.Top.Font.Style=Regular
Titles.Top.Font.HAlign=Left
Titles.Top.Font.VAlign=Top
Titles.Top.Font.Color=255,0,0,0
Titles.Top.Font.Back=255,255,255,255
Titles.Top.Font.Wordwrap=False
Titles.Side.Visible=True
Titles.Side.Font.Name=
Titles.Side.Font.Size=10
Titles.Side.Font.Style=Regular
Titles.Side.Font.HAlign=Left
Titles.Side.Font.VAlign=Top
Titles.Side.Font.Color=255,0,0,0
Titles.Side.Font.Back=255,255,255,255
Titles.Side.Font.Wordwrap=False
Titles.Filter.Visible=False
Titles.Filter.Font.Name=
```

```
Titles.Filter.Font.Size=10
Titles.Filter.Font.Style=Regular
Titles.Filter.Font.HAlign=Left
Titles.Filter.Font.VAlign=Top
Titles.Filter.Font.Color=255,0,0,0
Titles.Filter.Font.Back=255,255,255,255
Titles.Filter.Font.Wordwrap=False
Titles.Weight.Visible=False
Titles.Weight.Font.Name=
Titles.Weight.Font.Size=10
Titles.Weight.Font.Style=Regular
Titles.Weight.Font.HAlign=Left
Titles.Weight.Font.VAlign=Top
Titles.Weight.Font.Color=255,0,0,0
Titles.Weight.Font.Back=255,255,255,255
Titles.Weight.Font.Wordwrap=False
Titles.Status.Visible=False
Titles.Status.Font.Name=
Titles.Status.Font.Size=10
Titles.Status.Font.Style=Regular
Titles.Status.Font.HAlign=Left
Titles.Status.Font.VAlign=Top
Titles.Status.Font.Color=255,0,0,0
Titles.Status.Font.Back=255,255,255,255
Titles.Status.Font.Wordwrap=False
Titles.Labelling.Script=True
Titles.Labelling.Codes=True
Titles.Labelling.Name=False
Titles.Labelling.Desc=True
Columns.Groups.Size=30
Columns.Groups.Visible=True
Columns.Groups.Font.Name=
Columns.Groups.Font.Size=10
Columns.Groups.Font.Style=Regular
Columns.Groups.Font.HAlign=Middle
Columns.Groups.Font.VAlign=Middle
Columns.Groups.Font.Color=255,0,0,0
Columns.Groups.Font.Back=255,255,255,255
Columns.Groups.Font.Wordwrap=False
Columns.Groups.Labelling.Name=False
Columns.Groups.Labelling.Desc=True
Columns.Labels.Width=74
Columns.Labels.Height=67
Columns.Labels.Visible=True
Columns.Labels.Font.Name=
Columns.Labels.Font.Size=10
Columns.Labels.Font.Style=Regular
Columns.Labels.Font.HAlign=Middle
Columns.Labels.Font.VAlign=Top
Columns.Labels.Font.Color=255,0,0,0
Columns.Labels.Font.Back=255,255,255,255
Columns.Labels.Font.Wordwrap=False
Columns.Labels.Labelling.Name=False
Columns.Labels.Labelling.Desc=True
Columns.Letters.Font.Name=
Columns.Letters.Font.Size=10
Columns.Letters.Font.Style=Regular
Columns.Letters.Font.HAlign=Middle
Columns.Letters.Font.VAlign=Middle
Columns.Letters.Font.Color=255,0,0,0
Columns.Letters.Font.Back=255,255,255,255
```

```
Columns.Letters.Font.Wordwrap=False
Columns.Sort.Active=False
Columns.Sort.Increasing=False
Columns.Sort.Ungrouped=False
Columns.Sort.Band=First      (XSortBand)
Columns.Sort.Type=Value       (XSortType)
Columns.Sort.Key=0
Columns.Hide.Active=False
Columns.Hide.Missing=False
Columns.Hide.Empty=False
Rows.Groups.Size=23
Rows.Groups.Visible=True
Rows.Groups.Font.Name=
Rows.Groups.Font.Size=10
Rows.Groups.Font.Style=Regular
Rows.Groups.Font.HAlign=Left
Rows.Groups.Font.VAlign=Top
Rows.Groups.Font.Color=255,0,0,0
Rows.Groups.Font.Back=255,255,255,255
Rows.Groups.Font.Wordwrap=False
Rows.Groups.Labelling.Name=False
Rows.Groups.Labelling.Desc=True
Rows.Labels.Width=81
Rows.Labels.Height=20
Rows.Labels.Visible=True
Rows.Labels.Font.Name=
Rows.Labels.Font.Size=10
Rows.Labels.Font.Style=Regular
Rows.Labels.Font.HAlign=Left
Rows.Labels.Font.VAlign=Top
Rows.Labels.Font.Color=255,0,0,0
Rows.Labels.Font.Back=255,255,255,255
Rows.Labels.Font.Wordwrap=False
Rows.Labels.Labelling.Name=False
Rows.Labels.Labelling.Desc=True
Rows.Letters.Font.Name=
Rows.Letters.Font.Size=10
Rows.Letters.Font.Style=Regular
Rows.Letters.Font.HAlign=Left
Rows.Letters.Font.VAlign=Top
Rows.Letters.Font.Color=255,0,0,0
Rows.Letters.Font.Back=255,255,255,255
Rows.Letters.Font.Wordwrap=False
Rows.Sort.Active=False
Rows.Sort.Increasing=False
Rows.Sort.Ungrouped=False
Rows.Sort.Band=First         (XSortBand)
Rows.Sort.Type=Value       (XSortType)
Rows.Sort.Key=0
Rows.Hide.Active=False
Rows.Hide.Missing=False
Rows.Hide.Empty=False
Cells.Key.Font.Name=
Cells.Key.Font.Size=10
Cells.Key.Font.Style=Regular
Cells.Key.Font.HAlign=Left
Cells.Key.Font.VAlign=Top
Cells.Key.Font.Color=255,0,0,0
Cells.Key.Font.Back=255,255,255,255
Cells.Key.Font.Wordwrap=False
Cells.Bases.Font.Name=
```

```
Cells.Bases.Font.Size=10
Cells.Bases.Font.Style=Regular
Cells.Bases.Font.HAlign=Left
Cells.Bases.Font.VAlign=Top
Cells.Bases.Font.Color=255,0,0,0
Cells.Bases.Font.Back=255,255,255,204
Cells.Bases.Font.Wordwrap=False
Cells.Frequencies.Visible=True
Cells.Frequencies.Font.Name=
Cells.Frequencies.Font.Size=10
Cells.Frequencies.Font.Style=Regular
Cells.Frequencies.Font.HAlign=Left
Cells.Frequencies.Font.VAlign=Top
Cells.Frequencies.Font.Color=255,0,0,0
Cells.Frequencies.Font.Back=255,255,255,255
Cells.Frequencies.Font.Wordwrap=False
Cells.ColumnPercents.Visible=False
Cells.ColumnPercents.Font.Name=
Cells.ColumnPercents.Font.Size=10
Cells.ColumnPercents.Font.Style=Regular
Cells.ColumnPercents.Font.HAlign=Left
Cells.ColumnPercents.Font.VAlign=Top
Cells.ColumnPercents.Font.Color=255,0,0,0
Cells.ColumnPercents.Font.Back=255,206,255,255
Cells.ColumnPercents.Font.Wordwrap=False
Cells.RowPercents.Visible=False
Cells.RowPercents.Font.Name=
Cells.RowPercents.Font.Size=10
Cells.RowPercents.Font.Style=Regular
Cells.RowPercents.Font.HAlign=Left
Cells.RowPercents.Font.VAlign=Top
Cells.RowPercents.Font.Color=255,0,0,0
Cells.RowPercents.Font.Back=255,255,242,242
Cells.RowPercents.Font.Wordwrap=False
Cells.Stat1.Visible=False
Cells.Stat1.Font.Name=
Cells.Stat1.Font.Size=10
Cells.Stat1.Font.Style=Regular
Cells.Stat1.Font.HAlign=Left
Cells.Stat1.Font.VAlign=Top
Cells.Stat1.Font.Color=255,0,0,0
Cells.Stat1.Font.Back=255,255,255,255
Cells.Stat1.Font.Wordwrap=False
Cells.Stat2.Visible=False
Cells.Stat2.Font.Name=
Cells.Stat2.Font.Size=10
Cells.Stat2.Font.Style=Regular
Cells.Stat2.Font.HAlign=Left
Cells.Stat2.Font.VAlign=Top
Cells.Stat2.Font.Color=255,0,0,0
Cells.Stat2.Font.Back=255,255,255,255
Cells.Stat2.Font.Wordwrap=False
Cells.PercentSign.Visible=True
Cells.MissingAsZero=False
Cells.MissingAsBlank=False
Cells.ZeroAsBlank=False
Cells.BlankAsChar=False
Cells.BlankChar=False
Cells.PercentsAsProportions=False
Cells.ShowRedundant100=False
Significance.Visible=False
```

```
Significance.Type=SingleCell      (XSigType)
Significance.PropStat=Z
Significance.MeanStat=T
Significance.StatsParam=
Significance.LetterSequence=
Significance.Letters=None           (XLetters)
Significance.AppendLetters=False
Significance.Headcount=Weighted    (XSigHeadcount)
Significance.MeanPoolVariance=False
Significance.PropPooledEst=False
Significance.ContinuityCorr=False
Significance.SkipBase30=True
Significance.SkipCell5=True
Significance.Letters64=False
Significance.OneTailed=False
Significance.SigLevel1.Threshold=95
Significance.SigLevel1.Font.Name=
Significance.SigLevel1.Font.Size=10
Significance.SigLevel1.Font.Style=Regular
Significance.SigLevel1.Font.HAlign=Left
Significance.SigLevel1.Font.VAlign=Top
Significance.SigLevel1.Font.Color=255,255,0,0
Significance.SigLevel1.Font.Back=255,255,255,255
Significance.SigLevel1.Font.Wordwrap=False
Significance.SigLevel2.Threshold=90
Significance.SigLevel2.Font.Name=
Significance.SigLevel2.Font.Size=10
Significance.SigLevel2.Font.Style=Regular
Significance.SigLevel2.Font.HAlign=Left
Significance.SigLevel2.Font.VAlign=Top
Significance.SigLevel2.Font.Color=255,0,0,255
Significance.SigLevel2.Font.Back=255,255,255,255
Significance.SigLevel2.Font.Wordwrap=False
Significance.SigLevel3.Threshold=80
Significance.SigLevel3.Font.Name=
Significance.SigLevel3.Font.Size=10
Significance.SigLevel3.Font.Style=Regular
Significance.SigLevel3.Font.HAlign=Left
Significance.SigLevel3.Font.VAlign=Top
Significance.SigLevel3.Font.Color=255,0,255,0
Significance.SigLevel3.Font.Back=255,255,255,255
Significance.SigLevel3.Font.Wordwrap=False
Decimals.Frequencies=0
Decimals.Percents=2
Decimals.Statistics=1
Decimals.Expressions=2
Output.Format=tsv              (XOutputFormat)
Corner.Priority=Side           (XPriority)
```

Enums:

XOutputFormat. none, tsv, csv, ssv, htm, oxt, diamond
XPriority Side,Top
XSigType SingleCell,ColumnGroups,RefColumn,RowGroups,RefRow
XSigStat Z,T
XSigHeadcount Weighted,Unweighted,EffectiveBase
XLetters None,Column,Row
XSortBand First,Freq,ColPC,RowPC
XSortType Value,Code,Label