SPONSOR

CONTRACTOR

EUROPEAN COMMISSION – EUROSTAT UNIT C2

CONTRACT: 20102.2010.003-2010.520

CALL FOR TENDER: 2010/S 109-164439

NATIONAL ACCOUNTS METHODOLOGICAL AND PRACTICAL IMPROVEMENTS: NAMP C4 (LOT 10)

IMPROVED SPECIFIC IT TOOLS FOR THE COMPILATION OF NATIONAL ACCOUNTS

TECHNICAL NOTE OF BATCH JECOTRIM
DESIGN REPORT BATCH DEMETRA

Comment [BB1]: Que vient faire DEMETRA dans ce titre ?

Project Manager, Hendyplan:
François Libeau francois.libeau@hendyplan.com

Date: Revision 22 March 2013

Table of contents

Tables and figures

Annexes

Index	Description
1	By HENDYPLAN Input parameters from user needs analysis 001-InputParms.xls
2	By HENDYPLAN Sample of '.jba' file 002-BatchTestMulti.jba

Bibliography

Index	Description
1	By HENDYPLAN IMPROVED SPECIFIC IT TOOLS FOR THE COMPILATION OF NATIONAL ACCOUNTS –– 3RD INTERIM GENERAL REPORT NAMP.L10.M18_GeneralReport.pdf
2	By HENDYPLAN 3RD INTERIM TECHNICAL REPORT –– REPORT ON DESIGNOF NEW ECOTRIM TOOL NAMP.L10.M18_DesignReport.pdf
3	By EUROSTAT ECOTRIM Interface (Version 1.01) User Manual ECOTRIM 1.01 user Manuel.pdf

1. Introduction

The batch computation is the automation of run of steps one after the other, which is registered and which may be run as many times as necessary without having to declare again the conditions of execution.

The batch computation may be seen in 2 ways, the BATCH strictly speaking and the MULTI-BATCH by opposition to the interactive computation of action, using live the Graphical User Interface of the application to express the actions.

BATCH is the application of any of the benchmarking and temporal disaggregation methods on series but each computation is independent of the other. A BATCH run is then a series of steps for which there is no need of ordering: any step could be run at any place of the list.

MULTI-BATCH is the computation of steps one after the other but output of one step may be the input of another step making the sequence of treatment dependent of past executions. Then, order of the steps is important and the declaration of which data is input of which step is a bit trickier as results of other steps may be input of the current one.

JDemetra+ is the core environment where the application is developed.

There is no batch implementation in JEcotrim, strictly speaking.

HENDYPLAN explains in this document the design of the system and the extensions which will be necessary to make the implementation possible.

The MULTI-BATCH implementation requires:

- A batch language to stack in a single file the successive calls to methods; this file must be ‘callable’ by the software, understanding it has to run the specifications one after the other and differentiate the output of one step from the other;
- A rule in naming output variables as they may be input of other steps;
- A translator of the batch language to make it understandable by the JAVA code in JEcotrim; the Bank of Belgium developed a short template illustrating its views on how to plug the batch computation in JDemetra+, called ec.tstoolkit.algorithm.BatchProcessingTest.

This document explains how these components are structured in order to implement the functionality.

After consultation of Eurostat on 15th November 2012 and a confirmation by email on Fri 11/23/2012 12:17 PM, Batch files will be in XML format.

In [2] (see bibliography), a first level of design has been presented for this batch module. This position was connected to the unavailability of the NetBeans version of

JEcotrim. As this version arrived in the meantime, HENDYPLAN is revising in this report the design of this module to make it better integrated in JEcotrim and more sustainable. The batch mechanism will be plugged inside the JEcotrim interface[1].

2. Approach

A MULTI-BATCH is a declaration in XML language of multiple steps of computation

2.1. Description of a standard step

A step is a sequence of sub-steps like:

- HEADER providing the name of the method to be applied and the name of the step,
- SPECIFICATION declaring all parameters of computation for a given method; they are options of computation for the mathematical routines;
- INPUT, the declaration of origin and variables used as input to the method's computation;
- OUTPUT, the recollection of output produced by the method and the storage in an output database.

A variable coming from an external database is said exogenous. A variable coming from another step of the batch is said endogenous.

A name of step is important because it identifies where to take an endogenous variable once it is used as input of another step.

There are different levels in the way to name the variables in the system:

- The variable name (VName), i.e. the name of the time series as it appears in the external database; it may be a name for a series as input or a series as output;
- The system name (SysName), the name of the data object by convention in JEcotrim (ex.: YBench in Denton);

[1][1] Initially, i.e. before end of June 2012, Eurostat asked Hendyplan to develop the batch capabilities outside the JDemetra+ framework, using Eurostat IT standards, for an internal usage inside unit C2, as it was not planned to distribute it over the network of the ESS. But, now, with availability of NetBeans JDemetra+, the Batch facility will be accessible from JEcotrim and using as much as possible JDemetra+ features.

- The suffix (Suffix), i.e. the label which may be attached to the series keeping the memory of the method and parameters used to produce that series; an input data object has, in principle, no suffix.

NAMES will be the table keeping the track of the association for each DATA object in the following order [SysName, VName, Suffix].

2.2. *Declaration of a sequence of steps*

A MULTI-BATCH is a stack of steps like described before under the control of some common statements where global conditions of run are described; it is called the global step (GLOBAL). A final step called CLOSING is also gathering all statement used in a common way to close the batch session.

A MULTI-BATCH sequence looks then like:

- A GLOBAL Step with all characteristics of computation which are shared by all other steps.
- A sequence of computation steps
    - Step1 with sections in a standard way
        - Header (method and naming),
        - Specifications (for parameters of computation),
        - Input (external data involved),
        - Output (optional defining how to export results).
    - [... repeated for any number of computation steps ...]
- A CLOSING Step with instruction to close the process.

CLOSING has been introduced in a provisional way in this first version of JEcotrim. There is no statement inside, so far.

3. Definition of the batch language

This chapter describes the JEcotrim batch language and explains the structure of the XML files.

3.1. *Conventions*

The language contains keywords called statements, arguments and punctuation assembled in what we will call *instructions* which are surrounded by tags like XML requires in a standard way: see annex 1 with extensive examples.

In the following, each statement and component of language will be described according to some conventions.

Each chapter below is more or less a statement of the language. It is divided in standard sub-chapters,

- The format of the instruction,
- Indications about the algorithm to decode this instruction,
- The usage of this instruction,
- Examples.

The format sub-chapter explains the syntax of an instruction. There are some conventions:

- All mandatory keywords are in bold; ex.: <**specification**> means it opens an area of declaration of computation specification for a given method; it is not possible to write it in other way;
- Optional items are between square brackets; ex. [,] means at this place in the pattern, the comma is optional;
- Argument or any item which has to be replaced by something is in italic; ex.: *Varname* is an argument which has to be replaced by any specification according to the rules of the step declaration;
- Repetition of a sequence is followed by the continuation dashes; Ex.: "VARS= *varname* [*, varname*] [...]" means it is allowed to declare as many name of variables as needed, but at least one; the last continuation dashes mean no limitation in the list.

3.2. General syntax of a batch file

The general format of the XML file is as described below.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<batch>
```

XML standard header.

```
 <global>
  <outfile>OutputName</outfile>
  <outlog>LogName</outlog>
  <ow>OverwritingMode</ow>
 </global>
```

The GLOBAL STEP

```
  <dbxl>
    <VarCategory name = thisName>
            <file>filename</file>
            [<sheet>sheetname</sheet>]
     </VarCategory>
  </dbxl>
```

The DBXL STEP

```
 <steps>
  <step method=NameMeth name =StepName><!—Comment -->
   <specification>
        [SpecVal] [...]
   </specification>
   <input>
               [InputVal] [...]
   </input>
   <output>
               [OutputVal] [...]
   </output>
  </step>

[ ... Repeat <step> as needed ... ]

</steps>
```

The COMPUTATION STEPS one after the other

```
</batch>
```

Closing tag

The various arguments inside sections as introduced above are explained the STATEMENT chapters

A MULTI-BATCH file is a sequence of arguments, between opening and closing tags following the XML standard, according to the sequence as specified in Format above.

A STATEMENT is a keyword defining a given action.

There are categories of STATEMENTs matching with the sub-steps in a given step, i.e. HEADER, SPECIFICATION, INPUT or OUTPUT.

Table 1: list of statements in the BATCH language

Keyword	Description	Restriction of usage
GLOBAL category:		
OUTFILE	Declaration of the location of the output file	
OUTLOG	Declaration of the location of the log file	
OW	Overwriting behaviour in output file	
DBXL category:		
EXCELTS	Declaration of MS-Excel external databases of time-series	
EXCELMATRIX	Declaration of MS-Excel external databases of matrices	
HEADER category:		
STEP	Declaration of a STEP	
METHOD	Declaration of a method	
STEPNAME	Name allocated to the STEP of computation	

SPECIFICATION category:			
TYPEAGG	Type of temporal aggregation	In suffix	DEN, CL, LIT, FER, 2STEP
BENCH	Benchmarking estimation method	In suffix	DEN, 2STEP
HFM	High frequency disturbance model	In suffix	CL, LIT, FER
ARFLAG	Decision about AR parameter	In suffix	CL, LIT
EM	Estimation method		CL, LIT if ARFLAG = estimated
SCAN	Scanning option		CL, LIT if ARFLAG = estimated
GRIDSTEP	Number of steps		CL, LIT if SCAN = user-defined
PHI1	Scanning interval lower bound		CL, LIT if SCAN = user-defined
PHI2	Scanning interval upper bound		CL, LIT if SCAN = user-defined
ARFIX	AR parameter value		CL, LIT if ARFLAG = fixed (user-defined)
TOL	Tolerance		2STEP, RASPM
MAXITER	Maximum number of iterations		RASPM

INPUT category:		
AGGR **XREL** **…**	See chapter 3.5 for the detailed list of variable categories (*inCategory*).	Restrictions of usage are expressed in table 2.

OUTPUT category:		
DATABASE	Defines whether or not the newly calculated data is exported to output file.	
DETAILS	Defines whether or not all detailed information of the STEP computation is exported to output file.	

Details of arguments expected after the STATEMENTs are provided in the chapters below.

Examples

Examples of full MULTI-BATCH declaration are provided in chapter 4.

3.3. The GLOBAL STATEMENTs

3.3.1. OUTFILE STATEMENT - Name of common output file

FORMAT:

<**outfile**>*OutputName*<**/outfile**>

Where *OutputName* is the name of the output file.

Usage

OutputName is the name of the output file containing eventually its pathname, using the standard Ms-Windows conventions (either full path specification or relative path specification).

If <**output**> tag is not present, then no output file is produced.

An error is generated once the *OutputName* cannot be resolved.

Example

1. <output>m:\prodUnitC2\Q1Round\Prod2013Q1.xlsx</output>

With 1, the output file will be *m:* *prodUnitC2**Q1Round*\Prod*2013Q1.xlsx*.

3.3.2. OUTLOG STATEMENT – Name of the log file

FORMAT:

<**outlog**>*LogName*<**/outlog**>

Where *LogName* is the filename of the log file.

Usage

LogName is the name of the LOG file containing eventually its pathname, using the standard Ms-Windows conventions (either full path specification or relative path specification).

If <fl> tag is not present, then no log file is produced.

An error is generated once the *LogName* cannot be resolved.

Examples

1. <fl> m: \prodUnitC2\Q1Round\2013Q1.log</fl>

With 1, the log file will be *m: \prodUnitC2\Q1Round\2013Q1.log*.

3.3.3. OW STATEMENT - Overwriting behaviour

FORMAT:

<**ow**>*OverwritingMode*<***/ow***>

Where *OverwritingMode* =

- **append**,
- **overwrite**,
- **replace**.

Usage

User has to declare one option in the predefined ones:

- **append** means that if the output process intends to write in a sheet which already exists, the new dataset will be stored in a new sheet next to the initial one with an increment to avoid confusion (Ex. If S1.DENTON already exists, the process will create S1.DENTON(1) or S1;DENTON(2), etc, using the first incremental value which does not exist).

- **overwrite** means that if the output sheet already exists in the output workbook, the sheet is removed and replaced by a new one.
- **replace** means that if the output workbook already exists in the output directory, the whole workbook is removed and replace by a new one. Be careful with this mode if you have several steps in the batch file.

This feature allows the user to generate multiple snapshots of the same batch run in the same Ms-Excel file, with different versions of the same input dataset.

Example

1. <ow>0</ow>

With Example 1, the mode 'Append' is chosen.

3.4. The DBXL STATEMENT

FORMAT:

<**dbxl**>

<*VarCategory* **name** = *thisName*>

<**file**>*filename*</file>

[<**sheet**>*sheetname*</sheet>]

</*VarCategory*>

</**dbxl**> [...]

Where

VarCategory is in

- **excelts** if the dataset is loaded as a collection of time series,
- **excelmatrix** if the dataset is loaded as a matrix

thisName is the name of database as an alias for this MULTI-BATCH computation.

fileName is the filename of database as an Ms-Excel file, perhaps with a path specification.

sheetName is the name of the sheet which addressed by this computation.

Usage

The DBXL STATEMENT is preparing the list of potential sources of data for teh MULTI-BATCH computation. One sequence inside a given **dbxl** statement contains:

- A specification of the type of data object, like specified in format, mandatory,
- A *filename* specification which is also mandatory, following the Windows standards in terms of pathname and filename specifications,
- A *sheetname* specification to address data specifically in a given sheet inside *filename*;

It is possible to omit the declaration of **variable**, then, all variables in *filename* / *sheetname* are read as *VarCategory*.

It is possible to omit the declaration of **sheet**, then, all variables of all sheets in *filename* are read as *VarCategory*.

Example

1. Address the whole file

```
<dbxl>
  <excelts name="FullDB">
  <file>Flow.xlsx</file>
</dbxl>
```

This computation will consider for the MULTI-BATCH run all variables in all sheets.

2. Address only a sheet in the file

```
<dbxl>
  <excelts name="aggr">
        <file>Flow.xlsx</file>
        <sheet>Aggregated</sheet>
   </excelts>
  <excelts name="prel">
        <file>Flow.xlsx</file>
        <sheet>Related</sheet>
  </excelts>
</dbxl>
```

Two aliases are declared for the current batch. The run will take variables from the Aggregated and Related sheets. The rest of the variables and sheets are ignored.

3.5. The STEP STATEMENT

FORMAT:

<**step method**=*NameMeth* **stepname**=*StepName*><!-- comment -->

Where *StepName* is a string of characters in a-z, A-Z, 0-9 and '_'.

Where *NameMeth* is in:

• "den"	for Denton
• "cl"	for Chow-Lin
• "lit"	for Litterman
• "fer"	for Fernández
• "2step"	for 2-Step Procedure
• "raspm"	for Plus-Minus RAS

StepName is a name of step with any ASCII character in a..z, A...Z, 0...9 and '_'.

Comment, is a comment as a standard text between '< !--' and '-->' tags.

Usage

The set of authorized characters is restricted to avoid any ambiguity with delimiters in statement expression where lists may have to be separated, where especially variable names have to be declared (see input STATEMENTs). Only '_' is allowed at this purpose. Consequently, '_' will never be a separator in STATEMENT expressions.

Examples

1. <step method="den" stepname="DentonOnB1GM">
2. <step method="raspm" stepname=" RASPM_BESUIOT"> <!-- Run RASPM on the Belgian SUIOT -->

With 1, the section name will be DentonOnB1GM. In 2, the current section will be called RASPM_BESUIOT and the instruction is followed by a comment which will be ignored.

3.6. The INPUT sub-step STATEMENTs

FORMAT:

<In.category>FullVarname</In.category>

Where

- *In.category* =
    - **aggr** for aggregated series, only for Denton, Chow-Lin, Litterman, Fernández, 2-Step;
    - **xrel** for related series, only for Denton, Chow-Lin, Litterman, Fernández;
    - **prel** for preliminary series, for 2-Step;

    Then, only for the 2-Step procedure:

    - **z** for contemporaneous constraints;
    - **j** for the weight matrix of the contemporaneous aggregations;
    - **cv** for the column vector of coefficients of variation for variables reliability;

    Then, only for RASPM:

    - **xmat** for the matrix to be balanced;
    - **rowmarg** for the row marginal totals;
    - **colmarg** for the column marginal totals;
    - **fixedval** for the matrix addressing the location of cells in the high frequency time series which have to be fixed during the balancing mechanism (returned with their initial value)
- *FullVarname = prefixval.varname*

  Where *prefixval* is the name of the prefix for this variable,

  Varname is the name of a variable as it appears in the external database or as an output of a step.

Usage

There are as many *in.Category* instructions as a given METHOD is requiring.

There is always at least one variable in the list of *FullVarname* arguments. *NamePrefix* has to be an already declared alias or a step name which is before the current one.

It is possible to declare more than one INPUT statement in a given method. The pattern of XML will look like (using the DENTON case):

```
<input>
    <aggr>MyAnnualDB.GDP_A|MyAnnualDB.CONS_A|MyAnnualDB.I_A </aggr>
    <prel>MyQDB.GDP_Q|MyQDB.CONS_Q|MyQDB.I_Q </prel>
</input>
```

taking GDP_A, CONS_A, I_A from MyAnnualDB and GDP_Q, CONS_Q, I_Q from MyQDB.

Implicitly, there are associations of categories of variables according to the method which is invoked. The table below is expressing the different cases.

Table 2: Association of categories of variables by method

SysName	Shape	Size	Freq	Methods					
				Denton	Chow-Lin	Litterman	Fernández	Two-Step	RASPM
aggr	Time Series	N	LF	1	1	1	1	m >= 2	▨
xrel	Time Series	n	HF	1	k >= 1	k >= 1	k >= 1	▨	▨
prel	Time Series	n	HF	▨	▨	▨	▨	m	▨
z	Time Series	n	HF	▨	▨	▨	▨	k >= 1	▨
j	Matrix	(k, m)		▨	▨	▨	▨	0 or 1 (1)	▨
cv	Column vector	m		▨	▨	▨	▨	0 or 1 (1)	▨
xmat	Matrix	(n, m)		▨	▨	▨	▨	▨	1
rowmarg	Vector	n		▨	▨	▨	▨	▨	1
colmarg	Vector	m		▨	▨	▨	▨	▨	1
fixedval	Matrix	(n,m)		▨	▨	▨	▨	▨	0 or 1 (1)
Multi-computation				Yes	Yes	Yes	Yes	No	No

n	Nb of observations in High Frequency time series
N	Nb of observations in Low Frequency time series
m	Nb of time series in the system
k	Nb of associated series for a given reference. Ex. 1 AGGRetated series requires k XRELated series
(1)	j, cv and fixedval are optional
▨	Not relevant

Multi-computation: Means method may be invoked in a STEP computation for multiple groups of input.

From the table, reading it by column, it means the <input> section of the STEP declaration is expecting 2 categories, for Denton and regression-based methods, 3 to 5 categories of variables for the 2-Step procedure, 3 or 4 categories of variables for RASPM.

In most of the cases, there is always one variable involved by category in the association. For example, using Denton, if there one aggregated variable declared (AGGR category), there is one and only one preliminary variable declared in PREL. It is not the case for the regression-based methods where a single aggregated (AGGR) variable is associated to one or more related (XREL). The same feature also exists with the 2-STEP procedure. In the RASPM case, a matrix is optional (FIXEDVAL); here there is even no associated for this category.

There is also a mechanism of repetitive computation of a method, using the same set of computation parameters in <Specification>, for multiple groups. For example, with the example above, using GDP_A, CONS_A, I_A from MyAnnualDB and GDP_Q, CONS_Q, I_Q from MyQDB, the system is understanding it should repeat the computation for the pairs (GDP_A, GDP_Q), (CONS_A, CONS_Q), (I_A, I_Q). It is called the multi-processing. This loop on the computation is only possible if there are as many variables in the first input statement as in the second. It is only possible for Denton and for the regression-based methods once there is only one related variable.

Once there is an association of one AGGR variable for k (k>=1), the declaration will be through an AGGR with one variable and as many XREL variable as needed in the FullVarname list. In this case, there is no possible multi-computation.

The other methods are possible only as a single STEP computation, i.e. with only one variable per category.

3.7. The SPECIFICATION STATEMENTs

The various parameter options are behind individual STATEMENTs like presented in the table 1. All STATEMENTs are reviewed below.

3.7.1. TYPEAGG STATEMENT - Type of temporal aggregation

FORMAT:

<**typeagg**>*TypeAggregation*<**/typeagg**>

Where *TypeAggregation* =

- **flow** for Sum,
- **index** for Average,
- **last** for Sample of Last,
- **first** for Sample of first.

Restriction: Only for DEN, CL, LIT, FER, 2STEP

Usage

User has to declare one option in the predefined ones.

This parameter is present in the suffix (cf chapter 4.2).

Examples

1. <typeagg>last</typeagg>

With Example 1, the last in sample is selected to feed values to the disaggregation algorithm.

3.7.2. BENCH STATEMENT - Benchmarking estimation method

FORMAT:

<**bench**>*BenchmarkingMethod*</**bench**>

Where *BenchmarkingMethod* =

- **afd** for Additive First Differences,
- **asd** for Additive Second Differences,
- **pfd** for Proportional First Differences,
- **psd** for Proportional Second Differences,
- **st** for Squared preliminary series over the diagonal (Di Fonzo and Marini (2011) ST reconciliation technique), only for 2STEP,
- **qr** for Absolute preliminary series over the diagonal (Queneville and Rancourt (2005) QR reconciliation technique), only for 2STEP.

Restriction: Only for DEN, 2STEP. DEN uses only afd to psd. 2STEP uses all options.

Usage

User has to declare one option in the predefined ones.

For Denton, only the values *afd*, *asd*, *pfd* and *psd* are used, the default value being *pfd*.

For Two step, only the values *st* and *qr* are used, the default value being *st*.

This parameter is present in the suffix (cf chapter 4.2).

Examples

1. <bench>afd</bench>
2. <bench>st</bench>

With Example 1, the Additive First Differences method is selected only if method is Denton. With Example 2, the value is 'ST reconciliation technique' only if method is 2STEP.

3.7.3. HFM STATEMENT - High frequency disturbance model

FORMAT:

<**hfm**>*HFDisturb*</**hfm**>

Where *HFDisturb* =

- **with** for AR(1) disturbance with constant (CL) or ARIMA(0,1,1) disturbance with drift (LIT) or First difference regression with drift (FER),
- **without** for AR(1) disturbance without constant (CL) or ARIMA(0,1,1) disturbance without drift (LIT) or First difference regression without drift (FER).

Restriction: Only for CL, LIT, FER.

Usage

User has to declare one option in the predefined ones.

This parameter is present in the suffix (cf chapter 4.2).

Examples

1. <hfm>without</hfm>

With example 1, the choice done is AR(1) disturbance without constant if the method is Chow-Lin, ARIMA(0,1,1) disturbance without drift if the method is Litterman or First difference regression without drift if the method is Fernández.

3.7.4. ARFLAG STATEMENT - Decision about AR parameter

FORMAT:

<**arflag**>*ARFlag*</**arflag**>

Where *ARFlag* =

- **estim** for Estimated,
- **fixed** for Fixed by the user.

Restriction: Only for CL, LIT

Usage

User has to declare one option in the predefined ones.

If *estim* is chosen then the following tags are also needed: <em> and <scan> and the tag <arfix> is ignored.

If *fixed* is chosen then <arfix> is active and the following parameters are ignored: <em> and <scan>.

This parameter is present in the suffix (cf chapter 4.2).

Examples

1. <arflag>estim</arflag>
2. <arflag>fixed</arflag>

With Example 1, the user has chosen to estimate the AR parameter. He has to specify the parameters <em> and <scan> to decide how the AR parameter will be estimated.

With Example 2, the user has chosen to fix the AR parameter. The value of the AR parameter has to be specified through the tag <arfix>.

3.7.5. EM STATEMENT - Estimation method

FORMAT:

<**em**>*EstimMethod*<**/em**>

Where *EstimMethod* =

- **gls** for Minimum Generalized Least Squares,
- **ml** for Maximum Likelihood method.

Restriction: Only for CL, LIT when <arflag>estim</arflag>

Usage

User has to declare one option in the predefined ones.

If <arflag>fixed</arflag>, this parameter is ignored.

Example

1. <em>gls</em>

With Example 1, the user has chosen to the Minimum Generalized Least Squares method.

3.7.6. SCAN STATEMENT - Scanning option

FORMAT:

<**scan**>*ScanOption*<**/scan**>

Where *ScanOption* =

- **auto** for Automatic scanning,
- **user** for User-defined scanning.

Restriction: Only for CL, LIT when <arflag>estim</arflag>

Usage

User has to declare one option in the predefined ones.

If <arflag>fixed</arflag>, this parameter is ignored.

If *auto* is chosen, the parameters GRISDSTEP, PHI1 and PHI2 are fixed to GRIDSTEP=101, PHI1=-0.99 and PHI2=0.99.

Id *user* is chosen, the parameters GRIDSTEP, PHI1 and PHI2 take the values indicated by the tags <gridstep>, <phi1> and <phi2>.

Example

1. <scan>auto</scan>
2. <scan>user</scan>

With Example 1, the user has selected the 'Automatic scanning' option.

With Example 2, the user has selected the 'User-defined scanning' option and he has to use the tags <gridstep>, <phi1> and <phi2> according to his needs.

3.7.7. GRIDSTEP STATEMENT - Number of steps

FORMAT:

<**gridstep**>*NbStep*</**gridstep**>

Where *NbStep* is an integer in the range [11, 1999].

Restriction: Only for CL, LIT when <scan>user</scan>

Usage

User has to declare there the number of steps used for the estimation.

If <scan>auto</scan>, this parameter is ignored.

Example

1. <gridstep>1001</gridstep>

With Example 1, the user has decided that the estimation should use 1001 steps.

3.7.8. PHI1 STATEMENT - *Scanning interval lower bound*

FORMAT:

<**phi1**>*ScanLowerBound*</**phi1**>

Where *ScanLowerBound* is a double in the range [-0.999, 0.999].

Restriction: Only for CL, LIT when <scan>user</scan>

Usage

User has to declare there the user-defined value of the scanning interval lower bound.

If <scan>auto</scan>, this parameter is ignored.

Example

1. <phi1>-0.999</phi1>

With Example 1, the user has decided that scanning will start at -0.999.

3.7.9. PHI2 STATEMENT - *Scanning interval upper bound*

FORMAT:

<**phi2**>*ScanUpperBound*</**phi2**>

Where *ScanUpperBound* is a double in the range [-0.999, 0.999].

Restriction: Only for CL, LIT when <scan>user</scan>

Usage

User has to declare there the user-defined value of the scanning interval upper bound.

If <scan>auto</scan>, this parameter is ignored.

Example

1. <phi2>0.999</phi2>

With Example 1, the user has decided that scanning will end at 0.999.

3.7.10. ARFIX STATEMENT - AR parameter value

FORMAT:

<arfix>*ARValue***</arfix>**

Where *ARValue* is a double in the range [-0.999, 0.999].

Restriction: Only for CL, LIT when <arflag>fixed</arflag>

Usage

User has to declare there the user-defined value of the AR parmater.

If <arflag>estim</arflag>, this parameter is ignored.

Example

2. <arfix>0</arfix>

With Example 1, the user has decided that the AR parameter is fixed to 0.

3.7.11. TOL STATEMENT - Tolerance

FORMAT:

<tol>*Tolerance***</tol>**

Where *Tolerance* is a double in the range [0, 1].

Restriction: Only for 2STEP and RASPM

Usage

User has to declare there the tolerance.

Example

1. <tol>0.001</tol>

With Example 1, the user has chosen a tolerance of 0.001.

3.7.12. MAXITER STATEMENT - Maximum number of iterations

FORMAT:

<**maxiter**>*NbIter*<**/maxiter**>

Where *NbIter* is an integer in the range [11, 1999].

Restriction: Only for RASPM

Usage

User has to declare there the maximum number of iterations.

Example

1. <maxiter>1000</maxiter>

With Example 1, the user has decided that the process should use a maximum of 1000 iterations.

3.8. The OUTPUT STATEMENTs

There are few categories of data produced by a calculation step (outside GLOBAL and CLOSING):

- The main output series, i.e. the time series which are benchmarked, or temporarily disaggregated, or reconciled, or balanced;
- The derived series which are calculated from the main output series (ex.: growth rates, etc);
- Diagnostics, i.e. a collection of statistics following various econometric computations (regressions, other mathematical algorithms).

3.8.1. DATABASE STATEMENT – Export main output series

FORMAT:

<**database**>*BoolDatabase*<**/database**>

Where *BoolDetails* is one of the following options:

- **0** once main output series are not exported in external file,
- **1** once main output series are exported in external file.

Usage

The DATABASE STATEMENT expresses through 0 or 1 whether all main output series produced by a step are exported to external file.

Chapter 4 is explaining how an Ms-EXCEL is formatted in a standard way by JEcotrim.

Example

1. <database>0</database>

With Example 1, main output series are not produced in external file.

3.8.2. DETAILS STATEMENT – Export all results of a STEP computation

FORMAT:

<**details**>*BoolDetails*<**/details**>

Where *BoolDetails* is one of the following options:

- **0** once no detailed output is produced in external file,
- **1** once detailed output is produced in external file.

Usage

The DETAILS STATEMENT expresses through 0 or 1 whether all detailed data produced by a step is exported in external file. It accounts the main output series, the derived series (growth rates, etc), diagnostics, computation specifications (i.e. the parameter in the SPECIFICATION STEP) and the LOG file for the given step.

The following chapter is explaining how an Ms-EXCEL is formatted in a standard way by JEcotrim.

Example

1. <details>1</details>

With Example 1, all detailed output of the given STEP is produced in external file.

4. Definition of the format for output in Ms-Excel

4.1. Overview

It is assumed a MULTI-BATCH session output in Ms-EXCEL format is sent to a single file/workbook. If there is a need to dispatch the results in multiple Workbooks, the user should initiate multiple MULTI-BATCH files.

Once the output of a MULTI-BATCH is going to Ms-Excel, two groups of sheets are generated:

- The DATABASE sheet(s) where output data which has been benchmarked will be populated at a single place;
- The STEP sheet(s) receiving all results of the computation of a given STEP, including statistics and regressions' diagnostics.

There are naming conventions to present the various categories of output (time series, matrices, parameters, etc) and the multiple frequencies invoked by the MULTI-BATCH (ex.: MONthly to ANNual, Quarterly to ANNual, etc).

The following chapters are explaining the structure of the file and the standards inside.

4.2. The DATABASE Sheets

There are as many DATABASE Sheets as the number of periodicities invoked in this MULTI-BATCH.

<u>Convention on the sheet names</u>

The sheet will have a generic name like:

Output-*Freq*

Where 'Output' is a fixed name followed by *Freq*, the abbreviation of the frequency like reminded in the table below.

Table 3: Possible frequencies in JEcotrim

Frequency Abbreviation	Definition
A	Annual
H	Half-yearly
Q	Quarterly
M	Monthly

For example, coming back to the case where there are either Monthly to Annual and Quarterly to Annual computations, the output workbook will have three DATABASE sheets:

- Output-A
- Output -Q
- Output -M.

<u>Content</u>

The first line is the collection of output variables for this frequency. The first column is the list of dates as a common scale for all variables, i.e. starting at the earliest start date of the dataset, and ending at the latest date of the dataset.

The variable names are structured according to two parts, *Vname*[*Step#-Meth-Suffix*]

where

- *Vname* is the variable name like in the input; if it is not specified, it takes the *SysName* for the given method.
- *Step#* is the generic name of the step, i.e. its sequential number as appearing in the MULTI-BATCH file through the STEP STATEMENT; it has a shape like 'S1' for the first step, 'S2' for the second step, 'S3.1', then 'S3.2', 'S3.3' for the third step of the MULTI-BATCH where there are 3 groups to compute according to the convention in chapter3, etc.
- *Meth* is the method's abbreviation like in the METHOD STATEMENT;
- *Suffix* is the list of selected computation parameters of each method separated by a comma (in table 1, the parameters used in the suffix are indicated by the mention 'In suffix' in the third column).

Table 4: Combined computation parameters by method

METHOD	List of computation parameters present in suffix
[DEN]	TypeAggregation, BenchmarkingMethod, FirstDiffChoice
[CL]	TypeAggregation, HFDisturb, ARFlag
[LIT]	TypeAggregation, HFDisturb, ARFlag
[FER]	TypeAggregation, HFDisturb
[2STEP]	TypeAggregation, BenchmarkingMethod, FirstDiffChoice, Approach
[RASPM]	N/A

Since the only computation parameters of RASPM have numerical values, they are not present in the suffix.

Note: The generic name through the step number has been preferred to *Stepname* to generate a shorter chain of characters a single cell, despite it is less legible.

The variables which are exported here are the targeted ones and not the derived ones. For example, the growth rates of output variables generally computed by the methods are not relevant here. The targeted variables by method are listed below.

Table 5: Targeted variables by method

METHOD	*SysName* of the targeted variables
[DEN]	YBENCH
[CL, FER, LIT]	YDISAGG
[2STEP]	YOUT
[RASPM]	XBAL

Note that YBENCH and YDISAGG are single high-frequency time-series whereas YOUT and XBAL are systems of any number of high-frequency time-series.

4.3. The GLOBAL Sheet

The GLOBAL sheet, named as such in a fixed way, will recollect the value of the GLOBAL STATEMENT from the current MULTI-BATCH, and the LOG session which may have been generated by them.

In case of normal computation with no ERROR/WARNING event, the message 'No ERROR / WARNING in GLOBAL' will appear.

4.4. The STEP Sheets

There are as many sheets as the number of steps in the MULTI-BATCH including the expansion due to the invocation of groups.

Convention on the sheet names

The name of the sheets is built using:

Step#-*Method*

where *Method* is the short name of the method for this STEP.

Ex.: A MULTI-BATCH with 3 steps (DEN, RASPM, LIT), the 3d one being a sequential run of 3 groups of variables with the same processing parameters, will generate 5 STEP sheets S1-DEN, S2-RASPM, S3.1-LIT, S3.2-RASPM, S3.3-LIT.

Content

Looking at the sheet from the top to the bottom, the system generates:

- A Title,

- Tables of time series,
- A table of statistics
- A table of computation parameters (process METADATA).

The title is composed of:

- The generic STEP number,
- The *Stepname* as declared by the STEP STATEMENT,
- The variable name (as a *Vname*) like in the input, only if the method is Univariate.

Then, there are 2 tables of time series, one for the HF periodicity dataset on the left, and one for the LF periodicity dataset on the right. At each time, the date range is the maximum possible, considering series all together: the system identifies the earliest start date and the latest end date while building the time scale.

There are three lines of metadata at the top of each column, the *SysName*, followed by the *Vname*. A provisional line has been introduced below *Vname*, which may receive a variable description. This feature is not supported yet.

The STEP sheet helps the user to interpret the results. For this reason, some input data is repeated next to the output for easier analysis. From there, the user may post-compute any extra tables and graphs without linkage to other workbook and/or sheets. Once it is the case, the Vname is suffixed by '(I)'.

Then, the block of statistical results is appearing, in two columns, one for the label, one for the values.

Finally, the computation parameters, having an impact on the way algorithm has been invoked, is displayed, also in two columns, one for the name of the parameter, one for the value. All the computation parameters are present here, not only the ones present in the Suffix as appended to the variable name appearing at the top of each column of the DATABASE sheets.

The list of time series in the HF (resp. LF) tables as explained before is depending on the method. The table below is providing the details.