

PX-Job 3.2

General

- **PX-Job**: command line based PC-Axis file managing application
 - meant especially for routine process automation
 - the commands may be given directly from the command line or via specific command files (e.g. `bat` or `cmd` text files)
 - may be called from other applications
(the **return code** for a successful operation is 0)
- PX-Job uses the **same** source code as PX-Edit
 - the default answer to all user interface requests (e.g. when opening or saving files) is **OK**
- PX-Job has some functionalities that are not available in the PX-Edit interface
- PX-Job replaces the older *PX-EditBatch* and *PX-Tool* applications

Installation

- Three files are needed for basic use:
 - `PX-Job.exe` main program
 - `PX-Edit_32.paq` program code
 - `dyalog140rt_unicode.dll` interpreter dll
- Other installation files, which **may** be needed are
 - `zip.exe, unzip.exe` archive packer and opener
 - `PX-Edit_main_32.ini` main settings file
 - language files and other settings or control files
- PX-Job may be installed simply by copying the necessary files into one directory
 - updating is usually simple: just replace the old `paq` file with a newer one
 - Excel functionality needs Excel (or MS Office) to be installed

Command line syntax

PX-Job {job} [**in**] {out} {err} {copy} {meta} {set} {path} {log} {-} {!}

job job type (default: px)

in source directory, file, file list or list file (**mandatory**)

out output directory or file (default: source files)

err directory for erroneous files

copy directory for source files

meta metadata file or directory

set settings file

path common directory path

log log file

- options

! switches

Command line

- The command line consists of the program name `[path\]PX-Job[.exe]` and space separated parametre definitions (arguments)
 - the parametre ordering is free
 - the parametre codes are separated from the definition part by an equal sign (for example `job=csv`)
 - if the first parametre definition does not have the equal sign, it is interpreted as a `job` parametre
 - `in` parametre is **always** compulsory

job: batch type

px	standardized PC-Axis file (default)
csv	semicolon separated structured text file
exp	<i>eXplorer</i> file
htm	html file
sql	<i>simplePXsql</i> macro
txt	tab separated structured text file
xml	CoSSI/XML table
xls	Excel file
report	database report
translate	multilingualizer

- If `job` parametre is missing, and the first argument does not have the equal sign, it is interpreted as the batch type
- If the `job` cannot be deducted, it is interpreted as `job=px`

File types

- For some file types the default output (type, format and character coding) may be changed with switches (e.g. `-o` and `-t`)
- The default types need no extra switches
 - `px` standard PC-Axis file
 - `htm` html table with no specific colouring
 - `sql` all INPUT macros
 - `txt` WinANSI coded tab separated text file
 - `xls` Excel file (`xls` normally, `xlsx` for big tables)
 - `xml` CoSSI/XML table: XDF format

Parametres

in: source (mandatory) /1

- Either source directory, file, list file or comma-separated file list
 - if there are spaces in the definition, the parametre must be given in quotes (")
- Source directory
 - the directory name should end with backslash (\)
 - as default, only px files are read from the directory (see options `-i` and `-s`)
- Source file
 - the file name may include wildcard characters `?` and `*`
 - if there is a comma in the name, it must be replaced by the character `?`
 - if the file extension is `.list` (or `.lst`), the file is regarded as a *list file*

in: source (mandatory) /2

— List file

- a *list file* is a simple text file, which contains file names, one per row
- the lines that start with a semicolon are skipped (i.e. the file may include comments)
- if all the files are in the same directory, only the first name must contain the file path
- if all the files have a common path, it will be used with `s1` option

— File list

- the file list consists of comma-separated file names (no spaces allowed)
- if the file names contain commas (which is **not** recommended), the names may be separated by semicolons or the commas must be replaced by wildcard characters
- if all the files are in the same directory, only the first name must contain the file path

in: source (mandatory) /3

in =D:\path\source\	source directory
in =D:\path\source.xls	source file
in =D:\path*.xls	read all xls files
in =D:\path\ajolista.list	read all list file files
" in =D:\pa th\source.csv"	space in the name
in =D:\path1*.px,D:\path2*.csv	file list (with wildcards)
in =D:\path\a.px,b.px,c.px	file list (in the same directory)
in =D:\path\a,1.px;b,2.px	file list (commas in names)

out: result

- Output directory or file
 - the directory name should end with backslash
 - any non-existent directory will be created (see options `-o` and `-s`)
 - if there are spaces in the definition, the parametre must be given in quotes

out=D:\path\result.px output file

out=D:\path\target\ output directory

- If `out` is omitted, the output files are written in the source directories possibly overwriting existing source files

err: error directory

- When this setting is set, the erroneous source files will be copied in the designated directory (they will not be processed otherwise)
 - the directory name should end with a backslash
 - any non-existent directory will be created
 - if there are spaces in the definition, the parametre must be given in quotes
- as default, PX-Job is pretty sensitive to all errors with `err` parametre
 - the sensitivity level may be set with option `-e`
- source files are **not** deleted automatically
 - the expunge switch `!x` is used for file deleting

copy : archive directory

- The source files will be copied into the designated directory
 - the directory name should end with a backslash
 - any non-existent directory will be created
 - if there are spaces in the definition, the parametre must be given in quotes
 - if the `err` parametre is in use, it will be used in erroneous cases
 - the source files are **not** deleted automatically
 - the expunge switch `!x` is used for file deleting

meta: templates

- Setting of **template** directory or file
 - if there are spaces in the definition, the parametre must be given in quotes

meta=D:\path\result.pxk template file

meta=D:\path\templates\ template **directory**

- PX-Job will try to find an equivalent metadata source file (px or pxk) which has the same name as the result file
 - with or without trailing or leading parts (separated by underline characters)
 - name searching is not case sensitive
- the **meta** parametre is also used with multilingualizing (**job**=**translate**), reporting (**job**=**report**) and metadata injection (**-a1**, ..., **-a4**)

set: settings file

- Settings file for batch job
 - usually PX-Job uses the main settings file in the installation directory (`PX-Edit_main_32.ini`), if it is found
 - the `set` parametre may be used for using an additional settings file (like the personal settings file in PX-Edit)
 - it might be suitable e.g. for defining default keywords
 - if there are spaces in the definition, the parametre must be given in quotes

path: common directory path

- The common path setting for file/directory definitions in the command line
 - the file/directory settings that **start** with a backslash (\) will be prefixed with the `path` setting
 - if there are spaces in the definition, the parametre must be given in quotes
 - for example, the following input strings will be interpreted similarly:

```
in=D:\database\input\ out=D:\database\output\  
meta=D:\database\template\
```

```
in=\input\ out=\output\ meta=\template\  
path=D:\database\
```

log: keeping track

- The log file is used to record the messages shown during the job (in English)
 - all error messages, confirmations, input and output file names, etc. will be recorded
 - if there are spaces in the definition, the parametre must be given in quotes
 - PX-Job will also log the batch start and stop timestamp in the PX-Edit log (if it is in use)
 - the log feature may be switched off with !o
- The log directory must exist (i.e. it will not be created)
 - the default directory is the launch directory
 - the default name is `px-job_yyyymmdd.log`
- The file extension is `log`
 - if the file exists, the new log messages will be added to the file

Options and switches

Options start with dash, switches with exclamation mark

- Options and **switches** are used to fine-tune the batch
 - the options have (usually) multiple alternatives
 - the symbols after the options in this document:
`? / * / &` = one character/ string/ comma-separated list
 - the switches have two alternatives (states)
 - some of the option values may be given as switches (`-s1 = -s = !s`)
- The options and switches may be grouped, but
`-g, -i, -j, -l, -n` and `-v` have to be separate

For example

```
-o1 -icsv,txt -z -r -u+2 !x !s
```

may be shortened to

```
-o1ru+2z -icsv,txt !sx
```

Input

-e? : error sensitivity

- As default, PX-Job handles all input files
 - except those that have fatal errors in them
(e.g. the number of data cells does not equal the metadata)
- With the `err` parametre, PX-Job will not tolerate invalid tables, and all warnings or errors will make the job copy the files in the `err` directory
 - **-e1** normal sensitivity level (as without `err` parametre)
- Valueless option (`-e`) or switch (`!e`) equals to option **-e1**

-i&: input file types

- Source file extension definition
 - the default input files are px files, but it is possible to give a comma-separated extension list (do not add spaces)
- Wildcard characters ? and * are allowed
 - ixls only xls files will be read
 - ipx,xls both px and xls files
 - i* all files in the source directory

-s? : sub-directories

- As default, only the files in the source directory are handled
- The `-s` option is meant for situations, when there is need to go through the directory structure (starting from the source directory)
 - `-s1` reflect the directory structure in output, too
 - `-s2` write the output files in the output directory only (no sub-directories)
- Valueless option (`-s`) or switch (`!s`) equals to option `-s1`

-y* : freshness filtering

- y filters input files based on age in days (“youngness”)
- The definition format is `dd.h.min`
 - y14 not older than two weeks old files are read
 - y0.2 not older than two hours old files are read
 - y0.0.15 not older than fifteen minutes old files are read
- The *extended format* makes it possible to set the time window:
`dd.h.min+dd.h.min`
 - y2+10 not newer than ten minutes and older than two days
 - y2+1.0 not newer than one hour and older than two days
 - y2+1.0.0 not newer than one day and older than two days
 - y0+10 not newer than ten minutes

Input switches

- ! **a** try to open all Excel worksheets
 - usable e.g. for joining tables
 - all sheets will have the sheet number at the end of the output file name (except the first sheet)
- ! **b** bypass the default input string conversion (DOS > Unicode)
- ! **d** remove variables with only one value
- ! **i** show job progress information in the Task Bar balloon

Metadata

-a? : metadata injection

- Table metadata can be added, updated or removed with a control csv file (given with the `meta` parametre) and `-a` option
 - a1 `table` specific keywords
 - a2 `variable` specific keywords
 - a3 `value` specific keywords
 - a4 `cell` specific keywords
- The keywords *CODES*, *DATA*, *HEADING*, *HIERARCHIES*, *HIERARCHYNAMES*, *KEYS*, *LANGUAGES*, *STUB*, *PARTITIONED* and *VALUES* are not manageable, though
- The keyword contents are `lightly` checked (not as thoroughly as in px file reading)
- The control texts are not case-sensitive

-a? : the structure of the control file (csv)

- The first row of the file contains the column headers

<i>KEYWORD</i>	keyword name (mandatory)
languagecode	specific language code
variablename	variable name for variable and value specific keywords
valuetext	value text or code for value specific keywords
<i>VARIABLE</i>	variable names for cell specific keywords
replacetext	the text string to be replaced (if needed)
filename	the file name or mask (if needed)

- The column order is free
- Every row is regarded as a separate update instruction (either add, change or delete command)
 - empty cells are not handled

-a? : keyword and language columns

— *KEYWORD*

- mandatory (at least one is required)
- contains the new contents for the keyword
or the text string, which replaces the `replacetext` string
- empty cells are not handled
- **~ removes** the keyword (not a mandatory one, though)
- a new keyword will be added, but replacing an existing one needs to have the `-r` option as well
- *TIMEVAL* is treated differently: only the format of the new contents is taken in account, the final keyword will be based on the target variable

— `languagecode`

- the language code for the row (empty \Rightarrow all table languages)
- if the code = the main language of the table, the keyword will also be set to all other table languages, (if they are not set separately)

-a? : variable and value columns

- `variablename`
 - mandatory for variable and value specific keywords
 - the variable name may be either in the defined language (defined by `languagecode`) or in the main language
- `valuetext`
 - mandatory for value specific keywords
 - either `value text` in the defined language or in the main language or `value code`
- *VARIABLE*
 - mandatory for cell specific keywords
 - either value text in the defined or main language, value code, or * (= all)

-a? : replace and file filtering columns

- `replacetext`
 - the text string, which is to be replaced
 - may contain wildcard characters `*` and `?`
 - will bypass the possible `-r` option
- `filename`
 - table file name
 - the `px` extension is not needed
 - wildcard characters `*` and `?` are accepted
 - may contain (part of the) database path

-g& : group variables for combining

- The grouped variables will be joined as a **new** variable
- The variable name may be set with the `-v` option or it will be combined from the grouped variable
- The variable values and codes will be combined with the separator, which may be set with the `-p` option (the default is slash `/`)
 - the variable names are given in the main table language as a comma-separated list (the names cannot thus contain commas)
 - the *TIMEVAL* keyword will be fixed, if possible (see `!t` switch)

`-gYear,Month`

variables to be combined

`-g"First name", "Old name"`

quotes needed for the spaces

`-gHEADING`

combine all column variables

`-gSTUB`

combine all row variables

-h? : Statistics Finland specific

- These are reserved for Statistic Finland's internal use, they rely upon the database structure, table file names and some specific metadata settings
 - h0 generate *MATRIX* keywords based on the table file name
 - h1 standardize *CONTACT* keywords (e.g. new phone numbers)
 - h2 "-" for *Maaseutuindikaattorit* database
 - h3 "-" for *Kaupunki-indikaattorit* database
 - h4 *TABLEID* batch (meta-csv: table, contents)
 - h5 *VARIABLE-ID* batch (meta-csv: table, variable, contents)
 - h6 key figures batch (variable order: *Alue*, *Tiedot*, *Vuosi*)
 - h7 variable-value listings
 - h8 *eXist* update csv
 - h25 data value search in the database (default value is 25, may be changed with -v option), the result is a csv file

-m? : default metadata

- The default keywords are read from the `[Default]` section in the main settings file
 - default metadata will be added **after** any other metadata operations
 - **-m1** copy the missing metadata
 - **-m2** replace all possible metadata
- Valueless option (`-m`) or switch (`!m`) equals to option **-m1**

-n& : add a new variable

- The new variable will have one value text, which is copied from the file name
 - without path or file extension
 - underlines are converted to spaces
 - the variable name can be given for multilingual tables as comma-separated list (the language order must be standardized)
 - the variable names cannot have commas

-nInformation new variable name Information

"-nNew variable" quotes needed for the spaces

-nTieto, Information bilingual setting

-p* : combination separator

- The separator is used in variable combining (see the -g option)
- The default separator is a slash (/)

-p- use dash

-p:: use two colons

"-p = " quotes needed for the spaces

-r? : metadata replace level

- Used only with the `meta` parametre
 - r0** add all possible metadata (default)
 - r1** add and replace all possible metadata
 - r2** add all possible metadata and replace variable names
(if there are the same number of names)
 - r3** add and replace all possible metadata and replace
variable names, values and codes (use with caution!)
- Valueless option (`-r`) or switch (`!r`) equals to option **-r1**

-u* : timestamp update

— Add the *LAST-UPDATED* keyword

- u0 use the current date (i.e. when the batch is run)
- u20130713 set the date (use the default time)
- u20130713_13:30 set the full timestamp
- u+2 set the date two days later
 - if the date would be weekend, it will be changed to the next weekday
 - as default, the weekend is regarded as Saturday and Sunday (see option -w)
 - the default time is usually defined in the main settings file
- u+2_09:00 set the date and time in future

— Valueless option (-u) or switch (!u) equals to option -u0

-v* : combine variable name

- As default, the combined variable name will be made from the grouped variable names using the combine separator (see the `-g` and `-p` options)
 - `-vTime` set the new variable name as *Time*
 - `-vTime, Tid, Aika` set the new names for multilingual table (in the language order)
 - `"-vNew name"` quotes needed for the spaces

-w* : weekend skipping

- Defining the non-Western weekend (along with option -u)
 - the default weekend is regarded as Saturday and Sunday
 - needed, if there is need to change the weekend days or just bypass the default weekend skipping with the option -u
 - the weekday numbering: 0 (Monday) ... 6 (Sunday)
 - w45 set weekend as Friday and Saturday
 - w7 no weekend skipping

Database report

job=report: database reports

- The database report result is a csv file
- The following column headers are used:

<i>KEYWORD</i>	contents of the specified keyword
filecreate	file creation timestamp
filename	file name
filepath	the directory path of the file
filesize	file size in bytes
fileupdate	file change timestamp
languagecode	the language codes in the table
pathname	file path and name combined
tablesize	table size (rows) x (columns) = figures

report: general

- If there is no control csv file, the report includes the columns `filepath`, `filename`, `filesize`, `fileupdate`, `tablesize`, `languagecode`, `VARIABLES` and all **mandatory** keywords
- The `VARIABLES` column contains all the variables in the table (from *STUB* and *HEADING* keywords)
- The keywords *CODES*, *HEADING*, *KEYS*, *STUB* and *VALUES* cannot be included in the report
- The `!f` switch will print all the requested file information in the report even though there is no filtering information found (see the control csv structure)
- The controls are not case sensitive

report: the control csv structure /1

- The report contents may be tuned by a controlling csv file, which is given with `meta` parametre; the file may be created in PX-Edit (*Edit/Database/Report*)
 - The **first** column is a **control code**, which is either `0` (general) or `1..4` (the keyword level: table, variable, value or cell specific keyword accordingly)
 - The second column contains the control words or keywords
 - The general controls are either for information (`filename`, `filecreate`, ...) or filtering (`languagecode`, `variable`, `value` or `content`)
 - `languagecode` may be alone (all the table languages) or it may be used for filtering the needed languages (language codes in separate columns)
 - each language will add a new line in the report
- `0; languagecode` all languages
- `0; languagecode; en; sv` only English and Swedish metadata

report: the control csv structure /2

- `variable` may be alone (equals to all table variables) or it may be used for filtering the needed variables (variable names in separate columns)
 - each variable will add a new line in the report
- The variable names can be either in specified or main language
 - `0;variable` all variables (each in different line)
 - `0;variable;alue;region` filtering of the listed variables
 - `1;VARIABLES` all variable names in a single cell
- `value` defines the filtering value texts or codes
 - the value names can be either in specified or main language
 - `0;value;helsinki;091`

report: the control csv structure /3

- `content` defines the keyword content filtering (it must not be the whole text) for **all** keywords
`0;content;Tilastokesk;Statist`
- It is possible to define the filtering values for each keyword **separately**
`1;DESCRIPTION;kala;fisk;fish`
- The report creating will be **permissive**, i.e. if an individual keyword is filtered (and the corresponding cell left empty), other keyword values may still add a new line in the report

Table joining

-j &: table joining

- The joining option has to be **separate**
- When this option is in use, all the input files will be opened and grouped, and each group is joined to the **first** table (the only criteria: the **same** number of variables)
 - corresponds to *Edit/Join* function with multiple tables in PX-Edit
- The plain -j option joins the tables with default values
 - this special case may be denoted by switch !j

-j &: join options

- These joining options are closely related to the PX-Edit Join window
 - j_a replace all metadata (default: only missing ones)
 - j_r do not replace any metadata
 - j_e exact value text or code match with values
 - j_m do not try to match the variable names
 - j_n group the tables by file name without the last part
 - j_l do not create multilingual files, if possible
 - j_c do not use codes when matching values
 - j_o use only original values
 - j_s do not try to match values
 - j_t replace value texts
- There may be several settings in use at the same time (-j_{mo})

File saving

-b* : bypass the standard file name modification

- Fine tuning of the file names
 - b_ spaces will not be converted to underscores
 - b= uppercase characters will not be converted to lowercase
 - b~ accented characters will not be converted to ascii
- Multiple settings may be defined, if needed (-b~_ =)

-c? : character conversion

- Set the character conversion for text and px files
 - c0 WinANSI (default)
 - c1 Unicode (UTF-8)
- It is still possible to read the old DOS-ANSI formatted PC-Axis files, but DOS saving format is **not** supported any more
- Valueless option (-c) or switch (!c) equals to option -c1

-d* : dash conversion

- The dash characters (**not** for px file) may be converted to dot codes or zero
 - d0 change dashes to zeroes

– **f*** : fill item

- The fill item may be any dot code or zero
- The default value will be read from the main settings file (default: . .)
 - **f . . .** set fill code as three dots

-l&: output languages

- The language option is used either for defining the languages for the output tables or the languages to be added when using the `meta` parametre
 - the first language in the list becomes the main language, which will e.g. define the possible character conversions
 - the languages will be added, if there are suitable metadata in the source
 - the option makes PX-Job read all the available language files
- `-l en` use only **English** in the output file (if found),
if there is no *LANGUAGE* keyword, it will be set as *LANGUAGE*="en";
- `-l fi, sv, en` the output tables will have Finnish (main language),
Swedish and English (if possible)
- Valueless option (`-l`) or switch (`!l`) makes PX-Job read the language files
 - the txt strings needed for TITLE creation are included in PX-Job for languages `da, en, es, fi, fr, it, kl, no, pt, ru, sl, sv` and `uk`

-o? : output type

- px -o1 metadata part of the table (.pxk)
 -o2 sparse matrix format
- htm -o1 **PX-Edit cell colouring**
- sql -o1 only metadata INSERT macros
 -o2 only data INSERT macros
 -o3 bulk saving (data part in csv format)
 -o4 DROP, CREATE and all INSERT macros
 -o5 DROP, CREATE and metadata INSERT macros
- xls -o1 tables in xls format
- xml -o1 tables in XML/Cals format
 -o2 tables in XML/Keys format

-q* : decimal and thousand separators

— Formatting is used in XML/CalS saving only

- q, comma for decimals, no thousand separator
- q, . comma for decimals, dot for thousands
- q. , dot for decimals, comma for thousands
- q, _ comma for decimals, space for thousands
- q_ dot for decimals, space for thousands
- q, ~ comma for decimals, non-breaking space for thousands
- q~ dot for decimals, non-breaking space for thousands
- q, ' comma for decimals, apostrophe for thousands
- q' dot for decimals, apostrophe for thousands

-t* : variable titles

- Effects only `csv`, `htm`, `xls` and `xls` output
 - hierarchically arranged:
 - t0 texts only
 - t1 codes only
 - t2 codes and texts combined
 - t3 codes and texts separated
 - all values:
 - t10 texts only
 - t11 codes only
 - t12 codes and texts combined
 - t13 codes and texts separated

–z? : archive to zip files

- The separate `zip.exe` program is needed (included in the setup package)
- The archiving type depends on the `out` parametre
 - `file`: all files will be zipped in it
 - `-s` option copies the directory structure within the output file
 - `directory`: all output will be copied in individual archives
 - `-s` option copies the directory structure within the output directory
- If there is no `out` parametre, the files will be packed in the source directory
 - `-z2` option packs files in each directory in files (name = the directory)
 - `-z3` option works as `z2`, but the file names contain the directory paths (backslashes are replaced by underscores)
- Valueless option (`-z`) or switch (`!z`) equals to option `-z1`

Saving switches

- ! **f** print the file information always in the report (bypassing filtering)
 - forces handling all the files in metadata injection, too
- ! **g** add a single language code at the end of the file name
- ! **k** keep the file change date
- ! **n** add footnotes in the output table (`csv`, `htm`, `xls`)
- ! **o** do not write the log file
- ! **p** save using the screen decimal precision (**not** for `px` files)
- ! **q** quick data part copying from source file (in `px` job)
- ! **t** try to set the *TIMEVAL* when combining variables
- ! **v** file replace validation
 - only with `px` job and `out` directory parametre, no join or zipping defined
 - checks that there is no newer output file present in the output directory
- ! **x** delete (expunge) the source file(s) (use with caution!)
- ! **y** save changed tables only

Metadata editing

There are many ways to manage metadata

- Keyword **fetching** from template files
 - prepared px or pxk file is needed
 - the file does not have to be perfect, PX-Edit has to be able to open it

PX-Job in=... out=... **meta**=template.**pxk** ...

- Adding **default** keyword values from the settings file
 - prepared settings file is needed (with [Defaults] section)
 - useful in quick patches (for example setting **CONTACT** keyword)

PX-Job in=... out=... **set**=myset.**ini** -**m** ...

- Metadata **injection** with control csv files
 - separate csv files needed (creating is easy, e.g. with Excel)
 - versatile

PX-Job in=... out=... **meta**=control.**csv** -**a1** ...

Simple multilingualizing

translate: multilingualizing

- Creating the translation files (no `meta` parametre):
PX-Job `translate` in=D:\dbase\ out=D:\lang\ -s1
creates `.translate` files from each px file in the database
- The translation files contain (most of the) multilingual keywords
 - every keyword defines its own block (in brackets)
 - the texts inside the sections should be translated, and the language code set accordingly
 - the file may contain several language blocks
- Multilingualizing
 - the `meta` parametre defines the translation files/directory

PX-Job `translate` in=D:\dbase\ `meta`=D:\lang\ out=D:\outp\ -s
multilingualizes those px files in the database, for which the corresponding translation file is found

Examples

Database actions

PX-Job in=d:\dbase\ -s log=d:\log\checklog

- read all px files in the database d:\dbase\ (sub-directories, too), validate them and save back, write all actions in the log file

PX-Job in=d:\dbase\ -sy7 log=d:\log\checklog

- as before, but read only files, which are no older than one week

PX-Job **csv** in=d:\dbase\ out=d:\csvt\ -sc1 log=d:\log\csv

- convert all px files in the database d:\dbase as Unicode (UTF-8) csv-tables (according to the main language) to the directory d:\csvt

PX-Job csv in=d:\dbase\ out=d:\csvt\ -sc1 -1sv log=d:\log\csv

- as before, but now the metadata will be in Swedish (if there is sv language setting in the table) or in the table main language

Metadata enriching

```
PX-Job in=d:\input\ meta=d:\template\ log=d:\log\fetch
```

- read px files from the directory d:\input\
- fetch possible new **metadata** for each file from the directory d:\template according to the table name
- save the result files in the source directory (no out= parametre)

```
PX-Job in=d:\input\ meta=d:\template\ -r log=d:\log\fetch
```

- as before, but now **all** possible keywords will be copied

```
PX-Job in=d:\input\ set=d:\batch.ini -ms log=d:\log\set
```

- add to all px files in the database d:\dbase (with sub-directories) the new **default** keywords from the [Defaults] section in the settings file d:\batch.ini

```
PX-Job in=d:\input\ set=d:\batch.ini -m2s log=d:\log\set
```

- as before, but now the **existing** values will be replaced as well

Metadata injection: table specific keywords /1

PX-Job in=d:\dbase\ **meta**=d:\control\ctrl_a11 -sa**l** ...

- read the control file and **add new** keywords for each language in the database d:\dbase (sub-directories included)

PX-Job in=d:\dbase\ meta=d:\control\ctrl_a11 -s**r**a1 ...

- as before, but now the keywords will be **added or replaced**
- The semicolon-separated file structure is now as follows (ctrl_a11.csv):

NOTE ; languagecode

taulukkoalaviite ; fi

tablenote ; en

samma på svenska ; sv

Metadata injection: table specific keywords /2

PX-Job in=d:\dbase\ meta=d:\control\ctrl_a12 -sal ...

- the command is similar to the previous one
- The control file (ctrl_a12.csv) contains several control rows:
 - set the *NOTEX*-keyword for files, the name of which starts with 0?0
 - remove existing *PRESTEXT* keywords
 - replace text *kuolleet* to *menehtyneet* in the keywords *DESCRIPTION* and *CONTENTS*

	A	B	C	D	E	F
1	NOTEX	pretext	DESCRIPTION	CONTENTS	filename	replacetext
2	pika-alaviite				0?0*	
3		~				
4			menehtyneet	menehtyneet		kuolleet

Metadata injection: variable specific keywords

PX-Job in=d:\dbase\ meta=d:\control\ctrl_a21 -sa2 ...

- The control file (ctrl_a21.csv):
 - set the *NOTE* keyword for variables of different languages
 - set the *TIMEVAL* keyword for the variable *vuosi* ($A = \text{TLIST}(A)$)
 - works for multilingual tables, if the main language is Finnish (and the name of the variable is *vuosi*, *Vuosi*, *VUOSI*, etc.)

	A	B	C	D
1	NOTE	TIMEVAL	variablename	languagecode
2	muuttuja-alaviite		tienkäyttäjä	fi
3	bära bilbälte		tienkäyttäjä	sv
4	remember safety		road user	en
5		A	vuosi	

Metadata injection: value specific keywords

PX-Job in=d:\dbase\ meta=d:\control\ctrl_a31 -sa3 ...

- The control file (ctrl_a31.csv):
 - set the *VALUENOTE* keyword for values
 - value *Yöpymiset* in variable *Tiedot* (all languages)
 - value *Hotellit* in variable *Toimiala* (all languages)
 - value *Foreign* in variable *Country* (in English)
 - value code *MK02* in variable *Alue MK02* (in Swedish)

	A	B	C	D
1	VALUENOTE	variablename	valuetext	languagecode
2	arvoalaviite	tiedot	yöpymiset	
3	beware Bates!	Toimiala	Hotellit	
4	welcome	Country	Foreign	en
5	Velkua o Nådendal	alue	MK02	sv

Metadata injection: cell specific keywords

PX-Job in=d:\dbase\ meta=d:\control\ctrl_a41 -sa4 ...

— The control file (ctrl_a41.csv) :

— set the *CELLNOTEX* keyword

- the table must have the variables *toimialat*, *indeksit*, *kuukausi* and *vuosi* (or some subset of them)
- the setting will touch all values (*) for the variable *indeksit*, and the defined values for other variables (if found)

	A	B	C	D	E
1	toimialat	indeksit	kuukausi	vuosi	CELLNOTEX
2	50-52	*	7	1998	solualaviite

Table joining

```
PX-Job in=d:\dbase\tseries.px,"d:\in\monthly upd.xlsx" -j ...
```

- add data for a new month to the time series from a structured Excel table
- the Excel file name has to be in quotes because of the space in the name
- the original table has to be the first in the list

```
PX-Job in=d:\input\provinces.xlsx out=d:\result\package  
meta=d:\template\mk.pmk !ajz
```

- open all worksheets of the Excel file (they have to be structured): !a
- join the tables: !j
- add suitable **metadata** from the template file: meta=
- save the zipped (!z) result (provinces.px) as package.zip

Miscellaneous /1

PX-Job **report** in=d:\dbase\ out=d:\reports\ -s

- make a default database report and save it in the output directory as `px-report_timestamp.csv`

PX-Job in=d:\dbase\deaths\ out=d:\result\co_deaths.px

"-**n**Cause of death" !**j**

- add a new variable *Cause of death* in the tables (NB: quotes), the value texts will be taken from the file names: -n
- join the tables (with the same number of variables): !j
- save the result table as `co_deaths.px`

PX-Job in=d:\spain\ -**les** -s

- if needed, add the *LANGUAGE* keyword as **Spanish**, and create the *TITLE* keywords, too

Miscellaneous /2

PX-Job in=d:\dbase\ -gYear,Period -vTime -s !t

- **combine** (group) variables *Year* and *Period* as a variable *Time* in each table in the database where such variables are found
- try to set a suitable *TIMEVAL* expression for the combined variable
- if it succeeds, the variable codes are set according to the new *TIMEVAL*

PX-Job in=d:\olddb\ out=d:\newdb -jn -s -lfi,sv,en

- **join** monolingual tables in the database *olddb* as multilingual ones in the language order Finnish, Swedish and English to new location *newdb* with sub-directories
- the monolingual file names should have the language code as separate postfix (the main language files can lack the code)
e.g. *table1_fi.px*, *table1_sv.px*, *table1_en.px*, *table2.px*, *table2_sw.px*, *table2_en.px*, ...

Syntax changes to previous version

3.1

⇒

3.2

-a, -g, -n, -p, -x
-c0, -c1, -c2
-m0, -m1
<new>, -r0, -r1, -r2
-s0, -s1
-t1, -t0, -t11, -t10
-v0
-z0, -z1

!a, !g, !n, !p, !x
<off>, -c0, -c1
-m1, -m2
-r0, -r1, -r2, -r3
-s1, -s2
-t0, -t1, -t10, -t11
!v
-z1, -z2