imbWBI.ConsoleTool Console

Reference of console commands available at* imbWBI.ConsoleTool Console *command line interface. Implementation class (CommandConsole) of (imbWBI.ConsoleTool, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null 1.0.0.0). Generated: Monday, November 19, 2018 - imbWBI Console tool [v0.3.1]. Author: Goran Grubić [Copyright (c) 2017-2018]

> *Generated with imbACE.Core (0.2.0.21)*

General info

Type 'help' and press enter to show list of all commands, 'exit' to close the console.

Command with default parameters: [command name]

For help on a command: [command name] ?

To be prompted for parameters: [command name] *

> *Commands available at the console root level.*

001 Abort

> *Should be called from executing script.*

> *It will abort the running script, and stay in the console shell*

Example :

```
Abort
```

002 Cls

> *Clear screen - flushing console output buffer*

> *Clears the current screen*

Alias*: Cls*

Example :

```
Cls
```

003 Execute [exe]

> *Basic automation facility: reads lines from the script file and executes it as it was typed by the console user. If filename parameter is * it will ask user to select script to load.*

> *It opens the specified script (.ace) file from the scripts folder and performs commands from the script*

Command arguments:

ID	Name	Type	Default	Comment
01	*filename*	*String*	*script.ace*	*Filename for script to execute*
02	*delay*	*Int32*	*5*	*Delay milliseconds between execution of each line/commands*
03	*repeat*	*Int32*	*1*	*Number of times to repeat the script*
04	*askConfirmation*	*String*		*Text of the Yes/No confirmation box, if left blank it will not ask user to confirm script execution*

Example :

```
Execute filename="script.ace";delay=5;repeat=1;askConfirmation="";
```

004 Exit [x]

> *Exits the current console run-loop*

> *If this is the main console of the application - it actually does nothing*

Example :

```
Exit
```

005 ExportHelp

> *Exports help file into current state project folder*

> *Writes a txt file with content equal to the result of help command*

Command arguments:

ID	Name	Type	Default	Comment
01	*filename*	*String*	*help.txt*	*help.txt*
02	*open*	*Boolean*	*True*	*trueValues: True,False*
03	*onlyThisConsole*	*Boolean*	*False*	*If true it will generate user manual only for this consoleValues: True,False*

Example :

```
ExportHelp filename="help.txt";open=True;onlyThisConsole=False;
```

006 Help [h]

> *Provides Type-specific help content for the console, lists of all supported commands, plugins, local variables...*

> *If help option is not specified, it will ask user for type of help should be displayed*

Command arguments:

ID	Name	Type	Default	Comment
01	*option*	*aceCommandConsoleHelpOptions*	*none*	*Type of help content to show, if not specified it prompts the userValues: none,brief,parameters,commands,modules,plugins,full*
02	*onlyThisConsole*	*Boolean*	*True*	*If true it will generate user manual only for this consoleValues: True,False*

Example :

```
Help option=none;onlyThisConsole=True;
```

007 None [none]

> *Pseudo instruction - does nothing actually*

Example :

```
None
```

008 Pause

> *It pause ACE script execution, optionally displays custom message and allows user to end the pause*

> *If wait set to -1 there will be no time limit, the user will have to stop it. It will beep in last 1/5 of wait period.*

Command arguments:

ID	Name	Type	Default	Comment
01	*wait*	*Int32*	*60*	*How log the pause may last? in seconds. If set to -1 there is no time limit*
02	*msg*	*String*		*Custom message to be displayed to user.*

Example :

```
Pause wait=60;msg="";
```

009 Process

> *Processing script into selected form*

> *It will load the specified script and process it into selected format*

Command arguments:

ID	Name	Type	Default	Comment
01	*script*	*String*	*script.ace*	*Filename of script file to process*
02	*format*	*commandLineFormat*	*explicitFormat*	*Format to process script intoValues: unknown,explicitFormat,implicitFormat,onlyCommand,onlyComment,emptyLine*
03	*askForFormat*	*Boolean*	*True*	*Prompt user to choose the formatValues: True,False*

Example :

```
Process script="script.ace";format=explicitFormat;askForFormat=True;
```

010 Quit [q]

> *Quiting the aceApplication this console is running in*

> *It will prompt the user to confirm, if specified by console configuration*

Example :

```
Quit
```

011 Reset

> *Creating blank work folder (job/project) after saving the current*

> *It will ask you for new job/state name and save the current state before cleaning the memory. If autorename is true it will make new name for new state if the specified one is already taken.*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*newproject*	*--*

Example :

```
Reset name="newproject";
```

012 Template

> *Uses template script file to dynamically create customized execution script*

> *It loads specified template script file and applies provided parameters to the {n} template placeholders*

Command arguments:

ID	Name	Type	Default	Comment
01	*templateName*	*String*	*word*	*Name of template file*
02	*parameters*	*String*	*2,SM-LSD,1,preloadLexicon*	*Comma separated values for parameters*
03	*saveScript*	*Boolean*	*True*	*trueValues: True,False*

Example :

```
Template templateName="word";parameters="2,SM-LSD,1,preloadLexicon";saveScript=True;
```

> *Below are plugin command references, given in appropriate format (with plugin instance name prefix, leading to command name).*

013 Plugin: wem

Class *imbWEM.Core.consolePlugin.crawlJobPlugin*

```
imbWBI.ConsoleTool Console.crawlJobPlugin
```

> *This is imbACE advanced console plugin for crawlJobPlugin*

1 wem.Crawler

> *Defines new instance of the specified crawler. LT_t defines link take per iteration, I_max is iteration limit, PL_max defines max. page loads, PS_c is count of selected pages at end.*

> *New crawler is attached to the AnalyticJobRecord and set as current on the state level*

Command arguments:

ID	Name	Type	Default	Comment
01	*classname*	*String*	*SM_LTS*	*Name of the crawler class*
02	*LT_t*	*Int32*	*1*	*Load take - number of parallel loads*
03	*I_max*	*Int32*	*100*	*Iteration number limit*
04	*PL_max*	*Int32*	*50*	*Page Loads limit*
05	*instanceNameSufix*	*String*		*Crawler name sufix*
06	*primLanguage*	*basicLanguageEnum*	*serbian*	*Primary languageValues: serbian,italian,english,german,russian,slovenian,unknown,serbianCyr,french,catalan,danish*
07	*secLanguage*	*basicLanguageEnum*	*english*	*Secondary languageValues: serbian,italian,english,german,russian,slovenian,unknown,serbianCyr,french,catalan,danish*

Example :

```
wem.Crawler classname="SM_LTS";LT_t=1;I_max=100;PL_max=50;instanceNameSufix="";primLanguage=serbian;secLanguage=english;
```

2 wem.CrawlJobEngineSettings [CJES]

> *Crawl Job Engine controls the parallel execution of the Crawl Job.*
> *Tdl_max defines max. minutes per one domain level crawl, Tll_max per single link load and TC_max defines number of parallel domain loads.*

> *This command sets the most important parameters of the Crawl Job execution. For Tdl_max and Tll_max value -1 means limit is off, for TC_max value -1 means auto management.*

Command arguments:

ID	Name	Type	Default	Comment
01	*TC_max*	*Int32*	*8*	*Maximum number of parallel DLC executing in the same moment*

ID	Name	Type	Default	Comment
02	*Tdl_max*	*Int32*	*50*	*Maximum minutes allowed for single DLC to run*
03	*Tll_max*	*Int32*	*20*	*Maximum minutes of single iteration allowed for a DLC before its termination*
04	*Tcjl_max*	*Int32*	*100*	*Maximum minutes for the complete Crawl Job execution*

Example :

```
wem.CrawlJobEngineSettings TC_max=8;Tdl_max=50;Tll_max=20;Tcjl_max=100;
```

3 wem.Job

> *AnaliticJob declares one experimental run, this is the first command to call in scripts with experiment definitions*

> *Creates new instance of ActivityJog and assigns it to the current state.*

Command arguments:

ID	Name	Type	Default	Comment
01	*jobName*	*String*	*job*	*Name of the Job to define*
02	*jobDesc*	*String*		*Description for the job*
03	*defaultStage*	*Boolean*	*True*	*If true it will prepare default crawler stage to execute crawl inValues: True,False*
04	*stampPrefix*	*String*		*Prefix at timestamp*
05	*stampCount*	*Int32*	*1*	*Stamp version count*

Example :

```
wem.Job jobName="job";jobDesc="";defaultStage=True;stampPrefix="";stampCount=1;
```

4 wem.OpenSession

> *Selects and preloads local index and Experiment session information. useJobSettings option will ignore other params and use Job definition*

> *It will set report output information and create or load local index*

Alias*: OpenSession*

Command arguments:

ID	Name	Type	Default	Comment
01	*experimentSession*	*String*		
02	*IndexID*	*String*		
03	*useJobSettings*	*Boolean*	*False*	*Values: True,False*
04	*crawlFolderNameTemplate*	*String*	***	

Example :

```
wem.OpenSession experimentSession="";IndexID="";useJobSettings=False;crawlFolderNameTemplate="*";
```

5 wem.Plugin

> *Allows additional execution customization by crawling plugin*

> *It will create instance of specified plug in and set it into proper collection*

Command arguments:

ID	Name	Type	Default	Comment
01	*plugin_classname*	*String*	***	*Proper name of the crawling plugin class*

Example :

```
wem.Plugin plugin_classname="*";
```

6 wem.Run [R]

> *Runs the current crawl job*

> *Starts crawl execution*

Example :

```
wem.Run
```

7 wem.SampleFile

> *Imports sample from text file*

> *Loads the file and adds domain urls from it into context's sample list*

Command arguments:

ID	Name	Type	Default	Comment
01	*path*	*String*	***	*path to file with samples, if * it will open dialog to select the file*
02	*inWorkspace*	*Boolean*	*True*	*if true, the file path is interpreted as relative to console workspaceValues: True,False*
03	*sampleName*	*String*		*Name of the sample list, if empty it will not change current sample list name*
04	*replace*	*Boolean*	*False*	*if set to true it will replace any existing samples in the listValues: True,False*
05	*skip*	*Int32*	*0*	*Number of entries to skip, from the imported file*
06	*limit*	*Int32*	*-1*	*If set above 0, it limits the total number of domains imported*
07	*debug*	*Boolean*	*True*	*if true it will report on link preprocessingValues: True,False*

Example :

```
wem.SampleFile path="*";inWorkspace=True;sampleName="";replace=False;skip=0;limit=-1;debug=True;
```

014 Plugin: bec

Class *imbWBI.IndustryTermModel.consolePlugin.becPlugin*

```
imbWBI.ConsoleTool Console.becPlugin
```

> *This is imbACE advanced console plugin for becPlugin*

1 bec.CompareDataSets

> *What is purpose of this?*

> *What it will do?*

Command arguments:

ID	Name	Type	Default	Comment
01	*datasetA*	*String*		*--*
02	*datasetB*	*String*		*--*
03	*runName*	*String*	*DataSets*	*--*
04	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
bec.CompareDataSets datasetA="";datasetB="";runName="DataSets";debug=True;
```

2 bec.CrawlScript

> *Creates script to perform (re)build MCRepository crawl*

> *Creates script that prepares crawler, crawl job and performs the crawl*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*ODPBusinessDistantTopics*	*What component of the model to crawl*
02	*clearRepo*	*Boolean*	*False*	*If true it will clear existing MC RepoValues: True,False*
03	*debug*	*Boolean*	*True*	*If true it will show some extra information during the processValues: True,False*
04	*autorun*	*Boolean*	*True*	*If true it will autorun created script without asking userValues: True,False*

Example :

```
bec.CrawlScript name="ODPBusinessDistantTopics";clearRepo=False;debug=True;autorun=True;
```

3 bec.DataSetGraphs

> *Constructs web site graphs for pointed dataset*

> *It will load dataset from the path and construct web site graphs*

Command arguments:

ID	Name	Type	Default	Comment
01	*runName*	*String*	*word*	*report directory name*
02	*datasetPath*	*String*		*path containing dataset to load*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
bec.DataSetGraphs runName="word";datasetPath="";debug=True;
```

4 bec.DataSetReport

> *Provides detailed statistics on dataset, when current BEC configuration is applied*

> *What it will do?*

Command arguments:

ID	Name	Type	Default	Comment
01	*runName*	*String*	*word*	*Name of the report folder*
02	*datasetPath*	*String*		*Path to dataset - when other than currently loaded should be reported about*
03	*reconstruct*	*Boolean*	*True*	*If true, it will perform recornstructive analysis of all urls in the datasetValues: True,False*

Example :

```
bec.DataSetReport runName="word";datasetPath="";reconstruct=True;
```

5 bec.ExperimentGroup

> *Sets active experiment group*

> *Creates subfolder in report directory*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*BEC*	*Name for the group*
02	*description*	*String*		*Optional description for the group*
03	*script*	*String*		*Script associated with this group*

Example :

```
bec.ExperimentGroup name="BEC";description="";script="";
```

6 bec.GroupDescription

> *Sets description for current experiment group*

Command arguments:

ID	Name	Type	Default	Comment
01	*description*	*String*		*Description for the experiment group*

Example :

```
bec.GroupDescription description="";
```

7 bec.HarvestDomainList

> *Runs crawler to harvest domain list categories currently loaded*

> *It will generate and run crawler script for each category*

Command arguments:

ID	Name	Type	Default	Comment
01	*path*	*String*	*word*	*--*
02	*steps*	*Int32*	*5*	*--*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
bec.HarvestDomainList path="word";steps=5;debug=True;
```

8 bec.ImportDomains

> *Import currently selected sample lists into imbIndustryTermProject category*

> *Takes categories from webDataset*

Command arguments:

ID	Name	Type	Default	Comment
01	*word*	*String*	*word*	*--*
02	*steps*	*Int32*	*5*	*--*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
bec.ImportDomains word="word";steps=5;debug=True;
```

9 bec.Init

> *Initialize project*

> *Loads the project or creates new*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*bec01*	*project name*

Example :

```
bec.Init name="bec01";
```

10 bec.OverviewReport

> *Provides overview spreadsheet report with cross-fold average F1 metrics for all experiments ran since console boot*

> *Generates an overview report for active experiment group*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*word*	*filename without extension for the overview report*

Example :

```
bec.OverviewReport name="word";
```

11 bec.RunGroupScript

> *Runs the script associated with experiment group*

Command arguments:

ID	Name	Type	Default	Comment
01	*overview*	*Boolean*	*True*	*if overview report should be createdValues: True,False*

Example :

```
bec.RunGroupScript overview=True;
```

12 bec.Save

> *Saves project*

> *What it will do?*

Example :

```
bec.Save
```

13 bec.ToolkitTest

> *What is purpose of this?*

> *What it will do?*

Command arguments:

ID	Name	Type	Default	Comment
01	*runName*	*String*	***	*Name of the experiment log*
02	*runComment*	*String*	*BEC experiment run*	*Comment on the test run*
03	*silentDatasetLoad*	*Boolean*	*True*	*Prevents printout during dataset loadValues: True,False*

Example :

```
bec.ToolkitTest runName="*";runComment="BEC experiment run";silentDatasetLoad=True;
```

015 Plugin: mcm

Class *imbMiningContext.imbMCManager*

```
imbWBI.ConsoleTool Console.MC Manager Plugin
```

> ***Manager plugin takes care about MC repositories, loads, creates and provides the instances.***

1 mcm.Close

> ***Saves and unselects the currently selected MC Repository***

> ***It will save all muttable variables of the repository and its content***

Command arguments:

ID	Name	Type	Default	Comment
01	*log_msg*	*String*		*Log message to write into repository log just before unselecting it, and after saving it*
02	*doReport*	*Boolean*	*True*	*if true, it will create report files for complete content of the currently selected repository: spreadsheets, folder content readmes etc.Values: True,False*
03	*debug*	*Boolean*	*False*	*If true, it will print out short report on content of the repository (if any)Values: True,False*

Example :

```
mcm.Close log_msg="";doReport=True;debug=False;
```

2 mcm.Open

> ***Opens or creates new MCRepository, starts a MC session. Call this before any other MC operation.***

> ***It initiates specified MCRepository and sets it as current/selected.***

Command arguments:

ID	Name	Type	Default	Comment
01	*repo*	*String*	*word*	*Name of repository to start working with*
02	*log_msg*	*String*		*A message to be written into repository log after it is initiated, e.g. adding new MCWebSites, or running Data Mining procedure XXXX*
03	*debug*	*Boolean*	*False*	*If true, it will print out short report on content of the repository (if any)Values: True,False*

Example :

```
mcm.Open repo="word";log_msg="";debug=False;
```

016 Plugin: nlp

Class *imbNLP.Project.Plugin.imbNLPProjectPlugin*

```
imbWBI.ConsoleTool Console.imbNLPProjectPlugin
```

> ***This is imbACE advanced console plugin for imbNLPProjectPlugin***

1 nlp.HarvestODP

> *Harvests the Open Directory Projects*

> *It will scan for subdirectories and extract outer links*

Command arguments:

ID	Name	Type	Default	Comment
01	*start*	*String*	*https://dmoztools.net/Business/*	*Starting URL*
02	*steps*	*Int32*	*3*	*How deep the harvester is allowed to go*

Example :

```
nlp.HarvestODP start="https://dmoztools.net/Business/";steps=3;
```

2 nlp.LoadWebKB

> *Loads WebKB web site datasets*

> *It will load WebKB 7Sectors dataset*

Command arguments:

ID	Name	Type	Default	Comment
01	*path*	*String*	*G:_DOKTORAT_MAIN\SM03_Datasets\7sectors*	*Path to 7Secotrs dataset*

Example :

```
nlp.LoadWebKB path="G:\_DOKTORAT_MAIN\SM03_Datasets\7sectors";
```

017 Plugin: exp

Class *imbWBI.IndustryTermModel.consolePlugin.becExperimentPlugin*

```
imbWBI.ConsoleTool Console.becExperimentPlugin
```

> *Plugin for BEC Experiment configuration and creation*

1 exp.AddFeatureDimension

> *Defines new feature dimension*

> *It creates new feature dimension and optionally clears existing*

Command arguments:

ID	Name	Type	Default	Comment
01	*type*	*FeatureVectorDimensionType*	*similarityFunction*	*Dimension typeValues: directTermWeight,similarityFunction,topicWeight*
02	*function*	*String*	*CosineSimilarityFunction*	*Name of the function, if it is required by dimension type (e.g. similarity with class dimensions)*
03	*clearExisting*	*Boolean*	*False*	*Removes any existing feature dimension from the experiment settingsValues: True,False*

Example :

```
exp.AddFeatureDimension type=similarityFunction;function="CosineSimilarityFunction";clearExisting=False;
```

2 exp.Dataset

> *Configures dataset to be used for the experiment*

> *Set path, filter and selection options*

Command arguments:

ID	Name	Type	Default	Comment
01	*path*	*String*	*G:\imbWBI\datasets\7sectors*	*Diskdrive path, pointing to the root folder of the dataset (WebKB format)*
02	*pageLimit*	*Int32*	*1*	*Minimum number of pages that a document set (website) must have in order to be accepted for the experiment*
03	*filterEmpty*	*Boolean*	*True*	*Filters out empty documents from the datasetValues: True,False*

Example :

```
exp.Dataset path="G:\imbWBI\datasets\7sectors";pageLimit=1;filterEmpty=True;
```

3 exp.FeatureFilter

> *Configures feature selection filter*

> *It will set feature count limit and function*

Command arguments:

ID	Name	Type	Default	Comment
01	*function*	*String*	*IDFElement*	*Name of the global function that will rank the features*
02	*limit*	*Int32*	*4000*	*Number of features to be adopted*
03	*TDP*	*TDPFactor*	*chi*	*TDP factor to be applied (when used with Collection basedGlobal element)Values: none,idf,idf_prob,chi,ig,gr,or,rf*
04	*IDFc*	*IDFComputation*	*logPlus*	*Inverse Document Frequency computation variationValues: logPlus,modified,DF*

Example :

```
exp.FeatureFilter function="IDFElement";limit=4000;TDP=chi;IDFc=logPlus;
```

4 exp.GlobalIGMWeight

> *Configures a global function based on Gravity moment*

> *What it will do?*

Command arguments:

ID	Name	Type	Default	Comment
01	*l*	*Double*	*7*	*Lambda factor of IGM*
02	*weight*	*Double*	*1*	*Weigth associated with the function*
03	*removeExisting*	*Boolean*	*False*	*If any existing global factor should be removedValues: True,False*

Example :

```
exp.GlobalIGMWeight l=7;weight=1;removeExisting=False;
```

5 exp.GlobalTDPWeight

> *Configures a global function based on Term Discrimination Power*

> *It will add specified global factor, optionally if will remove any existing global factors*

Command arguments:

ID	Name	Type	Default	Comment

ID	Name	Type	Default	Comment
01	*factor*	*TDPFactor*	*chi*	*What factor should be addedValues: none,idf,idf_prob,chi,ig,gr,or,rf*
02	*weight*	*Double*	*1*	*Weigth associated with the function*
03	*removeExisting*	*Boolean*	*False*	*If any existing global factor should be removedValues: True,False*

Example :

```
exp.GlobalTDPWeight factor=chi;weight=1;removeExisting=False;
```

6 exp.GlobalWeight

> *Configures a global function in the feature weighting model - supports: ICF, ICSd, IDF, IGM, mIDF*

> *It will add specified global factor, optionally if will remove any existing global factors*

Command arguments:

ID	Name	Type	Default	Comment
01	*function*	*String*	*IDFElement*	*Name of function elemenet*
02	*weight*	*Double*	*1*	*Weigth associated with the function*
03	*IDF*	*IDFComputation*	*logPlus*	*How IDF should be computedValues: logPlus,modified,DF*
04	*removeExisting*	*Boolean*	*False*	*If any existing global factor should be removedValues: True,False*

Example :

```
exp.GlobalWeight function="IDFElement";weight=1;IDF=logPlus;removeExisting=False;
```

7 exp.kNN

> *Sets multi-class k-nearest neighbours classifier*

> *Removes any existing classifier and sets k-NN with specified settings*

Command arguments:

ID	Name	Type	Default	Comment
01	*distance*	*DistanceFunctionType*	*SquareEuclidean*	*Distance function to be used with k-NN classifierValues: SquareEuclidean,Euclidean,Cosine,Jaccard,Hamming,Dice*
02	*k*	*Int32*	*5*	*K parameter - number of neighbours to vote for class membership*

Example :

```
exp.kNN distance=SquareEuclidean;k=5;
```

8 exp.Load

> *Loads an experiment setup from the file*

> *It will try to load the specified file*

Command arguments:

ID	Name	Type	Default	Comment
01	*filename*	*String*	*bec_exp*	*name for the setup*

Example :

```
exp.Load filename="bec_exp";
```

9 exp.LocalWeight

> *Configures local function in the feature weighting model*

> *It will set computation and normalization options for feature weighting*

Command arguments:

ID	Name	Type	Default	Comment
01	*computation*	*TFComputation*	*normal*	*--Values: normal,squareRooted,glasgow,modifiedTF*
02	*normalization*	*TFNormalization*	*divisionByMaxTF*	*--Values: divisionByMaxTF,squareRootOfSquareSum*

Example :

```
exp.LocalWeight computation=normal;normalization=divisionByMaxTF;
```

10 exp.mSVM

> *Sets multi-class Support Vector Machine classifier*

> *Removes any existing classifier and sets mSVM with specified settings*

Command arguments:

ID	Name	Type	Default	Comment
01	*loss*	*Loss*	*L2*	*Loss function to setValues: L1,L2*
02	*model*	*mSVMModels*	*linear*	*Model to be used with SVM classifierValues: linear,gaussian*

Example :

```
exp.mSVM loss=L2;model=linear;
```

11 exp.PageFilter

> *Ranks and selects top-n documents from a document set*

> *It will set DocumentFilter function of EntityPlaneMethod*

Command arguments:

ID	Name	Type	Default	Comment
01	*function*	*String*	*DocumentEntropyFunction*	*Name of filter function class*
02	*limit*	*Int32*	*-2*	*number of top n pages to select, -2 will leave existing settings*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
exp.PageFilter function="DocumentEntropyFunction";limit=-2;debug=True;
```

12 exp.ParallelExecution

> *Sets allowed number of parallel threads*

Command arguments:

ID	Name	Type	Default	Comment
01	*n*	*Int32*	*5*	*Number of parallel threads*

Example :

```
exp.ParallelExecution n=5;
```

13 exp.RenderInstruction

> *Instructs HTML to text extraction engine (EntityPlaneMethod) to produce text from xpath*

> *It will add specified instruction to the rendering instruction set, and optionally remove all existing instructions before it.*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*::BODYTEXT::*	*Instruction name, it is human-readable descriptive name or special instructin name like ::BODYTEXT::*
02	*xpath*	*String*		*XPath associated with the instruction, selects nodes to be rendered into text*
03	*weight*	*Double*	*1*	*Weight factor of the instruction, i.e. number of times the content should be repeated (boosting TF)*
04	*remove*	*Boolean*	*False*	*If true it will remove any existing instruction in the setValues: True,False*

Example :

```
exp.RenderInstruction name="::BODYTEXT::";xpath="";weight=1;remove=False;
```

14 exp.Save

> *Saves the experiment setup*

> *Destination folder is command console workspace folder*

Command arguments:

ID	Name	Type	Default	Comment
01	*filename*	*String*	*bec_exp*	*name for the setup*

Example :

```
exp.Save filename="bec_exp";
```

15 exp.SignatureSuffix

> *Custom appendix to configuration signature*

> *Sets the custom signature suffix, to be added at the end of signature used for experiment run name*

Command arguments:

ID	Name	Type	Default	Comment
01	*suffix*	*String*	*word*	*textual suffix*

Example :

```
exp.SignatureSuffix suffix="word";
```

16 exp.Validation

> *Configures k-fold cross validation or single-fold calidation*

> *Updates active instance of the configuration*

Command arguments:

ID	Name	Type	Default	Comment
01	*K*	*Int32*	*1*	*Number of folds, if 1 (or 0) it will go into single-fold mode*
02	*TestFolds*	*Int32*	*1*	*Number of folds to be used as test folds, usually 1*
03	*Randomize*	*Boolean*	*True*	*Shell content of the folds be randomizedValues: True,False*
04	*LimitExecution*	*Int32*	*-1*	*When above 0, only specified number of folds will be executed*

Example :

```
exp.Validation K=1;TestFolds=1;Randomize=True;LimitExecution=-1;
```

018 Plugin: wds

Class *imbWEM.Core.consolePlugin.webDatasetPlugin*

```
imbWBI.ConsoleTool Console.webDatasetPlugin
```

> ***This is imbACE advanced console plugin for webDatasetPlugin***

1 wds.ExtractDomainList

> ***Extracting sample list for crawl from existing data set***

> ***It will load dataset specified and extract domain list from it***

Command arguments:

ID	Name	Type	Default	Comment
01	*DataSet*	*String*	*word*	*Path to dataset*
02	*Output*	*String*		*Path where domain list should be saved*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
wds.ExtractDomainList DataSet="word";Output="";debug=True;
```

2 wds.ExtractURLsFromDataset

> ***Extracts all crawled urls from the dataset***

> ***It creates single txt file with list of all URLs crawled by the dataset***

Command arguments:

ID	Name	Type	Default	Comment
01	*runName*	*String*	*word*	*Name of the report folder*
02	*datasetPath*	*String*		*Path to dataset - when other than currently loaded should be reported about*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
wds.ExtractURLsFromDataset runName="word";datasetPath="";debug=True;
```

3 wds.GetDomains

> ***It will execute subset compilation and set result as active sample list***

> ***It will query domains from the dataset source, using subset compilation specified***

Command arguments:

ID	Name	Type	Default	Comment
01	*subsetCompilation*	*String*	*ODPBusinessDistantTopics*	*name of the subset compilation to activate*
02	*saveFile*	*Boolean*	*True*	*if true it will save result to the sample list fileValues: True,False*

ID	Name	Type	Default	Comment
03	*construct*	*Boolean*	*True*	*if true it will prepare output WebDocumentCategory directory to store crawled contentValues: True,False*
04	*limit*	*Int32*	*-1*	*Upper limit for crawl size*

Example :

```
wds.GetDomains subsetCompilation="ODPBusinessDistantTopics";saveFile=True;construct=True;limit=-1;
```

4 wds.InitDatasets

> *Performs initiation of the mail dataset sources*

> *It will connect and check state of WebKB and ODP datasources*

Example :

```
wds.InitDatasets
```

5 wds.LoadDomainCategory

> *Loads WebDomainCategory tree from specified path*

> *It will search the specified path and load hierarchical domain list*

Command arguments:

ID	Name	Type	Default	Comment
01	*path*	*String*	*word*	*WebDomainCategory root folder to load from*

Example :

```
wds.LoadDomainCategory path="word";
```

6 wds.Test

> *It will run several diagnostic procedures*

> *What it will do?*

Example :

```
wds.Test
```

019 Plugin: itm

Class *imbWBI.IndustryTermModel.consolePlugin.itmPlugin*

```
imbWBI.ConsoleTool Console.itmPlugin
```

> *This is imbACE advanced console plugin for itmPlugin*

1 itm.CloneFVEOnSTX

> *Ensures consistant settings accross FVE models in the experiment*

> *It will take first FVE model, and make clones with different Semantic Expansion setting - other FVEs will be removed.*

Command arguments:

ID	Name	Type	Default	Comment

ID	Name	Type	Default	Comment
01	*name*	*String*	***	*Name of the experiment to process, or * to use currently selected*
02	*start*	*Int32*	*3*	*Starting value for Stx*
03	*end*	*Int32*	*8*	*Ending value for Stx*
04	*code*	*String*		*3 or 4 letter code indicating how the settings are different then in other experiments,* means sufix will be created automatically according to other modifications*
05	*kFold*	*Int32*	*-1*	*number of k-folds, when specified it overrides settings of the experiment*
06	*rnd*	*Int32*	*0*	*Overrides sample randomization, 0:no change, 1:randomize true, -1:randomize false*
07	*rename*	*String*		*Renames the experiment, if * the name is set automatically*
08	*mode*	*String*	*stx*	*Clone modes: stx, lpf*

Example :

```
itm.CloneFVEOnSTX name="*";start=3;end=8;code="";kFold=-1;rnd=0;rename="";mode="stx";
```

2 itm.Compose

> *Utilize composite template system to create an experiment instance*

> *What it load component specified and set the experiment as current*

Command arguments:

ID	Name	Type	Default	Comment
01	*shell*	*String*	*default*	*Name of experiment schell definition from composite templates sub folder of the project*
02	*classifiers*	*String*	*default*	*Name of classifier set XML file from composite template sub folder of the project*
03	*fve*	*String*	*default*	*Name of the Feature Vector Extractor XML file from composite template sub folder of the project*

Example :

```
itm.Compose shell="default";classifiers="default";fve="default";
```

3 itm.CrawlScript

> *Creates script to perform (re)build MCRepository crawl*

> *Creates script that prepares crawler, crawl job and performs the crawl*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*inIndustry*	*What component of the model to crawl*
02	*clearRepo*	*Boolean*	*False*	*If true it will clear existing MC RepoValues: True,False*
03	*debug*	*Boolean*	*True*	*If true it will show some extra information during the processValues: True,False*
04	*autorun*	*Boolean*	*True*	*If true it will autorun created script without asking userValues: True,False*

Example :

```
itm.CrawlScript name="inIndustry";clearRepo=False;debug=True;autorun=True;
```

4 itm.Diagnostic

> *Evaluates current Industry Term Model project's state*

> *It will check in which phase the current project is, and if it is ready for application*

Command arguments:

ID	Name	Type	Default	Comment
01	*verbose*	*Boolean*	*True*	*If true it will produce more in-detail report on the state of the current projectValues: True,False*

Example :

```
itm.Diagnostic verbose=True;
```

5 itm.DiagnosticMode

> *Sets configuration into system-check mode where minimum computation load is run, in single thread, just to enable debugging and system health*

> *It will: set to single thread execution, one web page per web site is loaded, k is set to 1 (k-fold validation), uses cached lexic resource.*

Command arguments:

ID	Name	Type	Default	Comment
01	*toggle*	*Int32*	*1*	*if value 0, nothing changed, if its -1 it will turn off the diagnostic mode, if its 1 it will turn on the diagnostic mode*

Example :

```
itm.DiagnosticMode toggle=1;
```

6 itm.Experiment

> *Executes currently selected experiment*

> *It will create experiment execution context and run*

Command arguments:

ID	Name	Type	Default	Comment
01	*classReporting*	*Boolean*	*True*	*If true it will create additional report on each industry / categoryValues: True,False*
02	*caseReporting*	*Boolean*	*True*	*If true it will create additional report for each processed caseValues: True,False*
03	*resultReporting*	*Boolean*	*True*	*if true it will create additional report for each validation caseValues: True,False*
04	*makeGeneralReport*	*Boolean*	*True*	*if true it will create General reportsValues: True,False*
05	*sufix*	*String*		*Adds specified sufix to experiment report folder name*

Example :

```
itm.Experiment classReporting=True;caseReporting=True;resultReporting=True;makeGeneralReport=True;sufix="";
```

7 itm.ExperimentMatrix

> *Automatically creates 3x3 matrix of experiments variating TW, TC and RX options*

> *Creates experiment matrix using LPF, DFC, IDFOn specified*

Command arguments:

ID	Name	Type	Default	Comment
01	*LPF*	*Int32*	*2*	*Low pass frequecy--*
02	*DFC*	*Double*	*5*	*Document Frequency Correction*
03	*IDFOn*	*Boolean*	*True*	*IDF on / offValues: True,False*
04	*fve*	*String*	*CSSRM*	*FVE name*
05	*shell*	*String*	*k1*	*Shell name*
06	*stxStart*	*Int32*	*2*	*Stx start*
07	*stxEnd*	*Int32*	*6*	*Stx end*

ID	Name	Type	Default	Comment
08	*StrictPOS*	*Boolean*	*False*	*Strict POS PolicyValues: True,False*
09	*Report*	*Boolean*	*False*	*Should generate general report on classesValues: True,False*

Example :

```
itm.ExperimentMatrix LPF=2;DFC=5;IDFOn=True;fve="CSSRM";shell="k1";stxStart=2;stxEnd=6;StrictPOS=False;Report=False;
```

8 itm.ExperimentRange

> *What is purpose of this?*

> *What it will do?*

Command arguments:

ID	Name	Type	Default	Comment
01	*TW*	*String*	*std*	*Term Weight Flag for HTML Tags*
02	*TC*	*String*	*std*	*Term Category*
03	*RX*	*String*	*div*	*Redux option*
04	*DFC*	*String*	*1.5,2.0,3.5,5.0,7.5,10*	*DFC values*
05	*LPF*	*String*	*2*	*LPF values*
06	*IDFOn*	*Boolean*	*True*	*IDF turn on and offValues: True,False*
07	*fve*	*String*	*CSSRM*	*FVE name*
08	*shell*	*String*	*k1*	*Shell name*
09	*stxStart*	*Int32*	*2*	*Stx start*
10	*stxEnd*	*Int32*	*6*	*Stx end*
11	*StrictPOS*	*Boolean*	*False*	*Strict POS PolicyValues: True,False*
12	*Report*	*Boolean*	*False*	*Should generate general report on classesValues: True,False*
13	*rnd*	*Int32*	*0*	*Overrides sample randomization, 0:no change, 1:randomize true, -1:randomize false, more than 1 will repeat the same model specified number of times*
14	*exName*	*String*	***	*Experiment name*
15	*classifier*	*String*	*default*	*Name of classifier compound*

Example :

```
itm.ExperimentRange TW="std";TC="std";RX="div";DFC="1.5,2.0,3.5,5.0,7.5,10";LPF="2";IDFOn=True;fve="CSSRM";shell="k1";stxStart=2;stxEnd=6;StrictPOS=False;Report=False;r
```

9 itm.GetExperiment

> *Provides new experiment or loads existing*

> *It will load or create new experiment and set is as the active one, ready for execution*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*Exp01*	*name of the experiment*
02	*makeDefault*	*Boolean*	*True*	*if true it will create experiment setup with default settings (if experiment not found)Values: True,False*
03	*debug*	*Boolean*	*True*	*if true it will provide more verbose outputValues: True,False*

Example :

```
itm.GetExperiment name="Exp01";makeDefault=True;debug=True;
```

10 itm.ImportExperiment

> *Allows reproduction of earlier experiment by loading experimentSetup.xml from report folder.*

> *It will load the experiment from specified address and set it as current experiment in the console, optionally it can run the experiment immediatly.*

Command arguments:

ID	Name	Type	Default	Comment
01	*path*	*String*	*reports\STAGE05*	*Path pointing to the folder with experiment report. You don't need to specify exact file name, just directory path. The path is relative to application installation folder*
02	*execute*	*Boolean*	*True*	*If true it will automatically execute loaded experimentValues: True,False*
03	*silent*	*Boolean*	*False*	*If true, it will apply default answers if otherwise user input was required. This is intended for scripted/automatic executionValues: True,False*
04	*suffix*	*String*	*_redone*	*Report suffix used for report folder name (it will also filter out targeted experiments, in case you are pointing at* $reports$ *folder - to avoid endless execution loop)*
05	*onlyWithError*	*Boolean*	*False*	*Execute only if experiment had errorsValues: True,False*

Example :

```
itm.ImportExperiment path="reports\STAGE05";execute=True;silent=False;suffix="_redone";onlyWithError=False;
```

11 itm.Modify

> *Adjusting experiment settings*

> *Modifies current experiment*

Command arguments:

ID	Name	Type	Default	Comment
01	*TW*	*String*	*std*	*Term Weight Flag for HTML Tags*
02	*TC*	*String*	*std*	*Term Category*
03	*RX*	*String*	*div*	*Redux option*
04	*LPF*	*Int32*	*2*	*Term Weight Flag for HTML Tags*
05	*DFC*	*Double*	*1.1*	*DF Correction*
06	*IDFOn*	*Boolean*	*True*	*IDF turn on and offValues: True,False*
07	*StrictPOS*	*Boolean*	*False*	*Strict POS PolicyValues: True,False*

Example :

```
itm.Modify TW="std";TC="std";RX="div";LPF=2;DFC=1.1;IDFOn=True;StrictPOS=False;
```

12 itm.Open

> *Sets the current Industry Term Model projects*

> *It will try to load the project or to create new under specified name*

Command arguments:

ID	Name	Type	Default	Comment

ID	Name	Type	Default	Comment
01	*name*	*String*	*itm01*	*Name of project to load*

Example :

```
itm.Open name="itm01";
```

13 itm.RepoOptimization

> *Removes unused content from repositories*

> *It will use current project settings to filter repo entries for pages on language of preference and to delete irrelevant content*

Example :

```
itm.RepoOptimization
```

14 itm.Save

> *Saves the currently selected Industry Term Model project*

> *It will call for save of the currently selected Industry Term Model project*

Example :

```
itm.Save
```

15 itm.Setup

> *Performs basic setup procedure, should be called after new project created*

> *It will ask user to insert or confirm each property in configuration modules: crawlSetup, nlpRepoProcessSetup, etc...*

Command arguments:

ID	Name	Type	Default	Comment
01	*verbose*	*Boolean*	*True*	*Provides additional hints and explanations during the setup processValues: True,False*

Example :

```
itm.Setup verbose=True;
```

16 itm.SingleFoldTest

> *Trains the classifier and performs classification over complete industry classes, used for debug purposes*

> *It will use complete sample of the class to train classifier, later it will run classification on the same class*

Command arguments:

ID	Name	Type	Default	Comment
01	*prefix*	*String*		*Test name prefix*
02	*model*	*itmModelEnum*	*chunkTFIDF*	*Model to use for training and classificationValues: none,wordTFIDF,chunkTFIDF,wordTFIDF_complex,semantic*
03	*report*	*Boolean*	*False*	*if true it will create extensive debug informationValues: True,False*
04	*debug*	*Boolean*	*True*	*if true it will create extensive debug informationValues: True,False*

Example :

```
itm.SingleFoldTest prefix="";model=chunkTFIDF;report=False;debug=True;
```

17 itm.Summary

> *Scans report folders to produce summary report on all experiments*

> *Scans for experimentSetup.xml files and collects all FVE data into single table*

Command arguments:

ID	Name	Type	Default	Comment
01	*reportName*	*String*	*Summary*	*Prefix to insert in name*
02	*subfolder*	*String*		*Subfolder within report directory*

Example :

```
itm.Summary reportName="Summary";subfolder="";
```

18 itm.Test

> *Used for development, to run quick test*

> *on what ever*

Command arguments:

ID	Name	Type	Default	Comment
01	*word*	*String*	*word*	*--*
02	*steps*	*Int32*	*5*	*--*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
itm.Test word="word";steps=5;debug=True;
```

020 Plugin: wem

Class *imbWEM.Core.consolePlugin.crawlJobPlugin*

```
imbWBI.ConsoleTool Console.crawlJobPlugin
```

> *This is imbACE advanced console plugin for crawlJobPlugin*

1 wem.Crawler

> *Defines new instance of the specified crawler. LT_t defines link take per iteration, I_max is iteration limit, PL_max defines max. page loads, PS_c is count of selected pages at end.*

> *New crawler is attached to the AnalyticJobRecord and set as current on the state level*

Command arguments:

ID	Name	Type	Default	Comment
01	*classname*	*String*	*SM_LTS*	*Name of the crawler class*
02	*LT_t*	*Int32*	*1*	*Load take - number of parallel loads*
03	*I_max*	*Int32*	*100*	*Iteration number limit*
04	*PL_max*	*Int32*	*50*	*Page Loads limit*
05	*instanceNameSufix*	*String*		*Crawler name sufix*
06	*primLanguage*	*basicLanguageEnum*	*serbian*	*Primary languageValues: serbian,italian,english,german,russian,slovenian,unknown,serbianCyr,french,catalan,danish*

ID	Name	Type	Default	Comment
07	*secLanguage*	*basicLanguageEnum*	*english*	*Secondary languageValues: serbian,italian,english,german,russian,slovenian,unknown,serbianCyr,french,catalan,danish*

Example :

```
wem.Crawler classname="SM_LTS";LT_t=1;I_max=100;PL_max=50;instanceNameSufix="";primLanguage=serbian;secLanguage=english;
```

2 wem.CrawlJobEngineSettings [CJES]

> *Crawl Job Engine controls the parallel execution of the Crawl Job.*
> *Tdl_max defines max. minutes per one domain level crawl, Tll_max per single link load and TC_max defines number of parallel domain loads.*

> *This command sets the most important parameters of the Crawl Job execution. For Tdl_max and Tll_max value -1 means limit is off, for TC_max value -1 means auto management.*

Command arguments:

ID	Name	Type	Default	Comment
01	*TC_max*	*Int32*	*8*	*Maximum number of parallel DLC executing in the same moment*
02	*Tdl_max*	*Int32*	*50*	*Maximum minutes allowed for single DLC to run*
03	*Tll_max*	*Int32*	*20*	*Maximum minutes of single iteration allowed for a DLC before its termination*
04	*Tcjl_max*	*Int32*	*100*	*Maximum minutes for the complete Crawl Job execution*

Example :

```
wem.CrawlJobEngineSettings TC_max=8;Tdl_max=50;Tll_max=20;Tcjl_max=100;
```

3 wem.Job

> *AnaliticJob declares one experimental run, this is the first command to call in scripts with experiment definitions*

> *Creates new instance of ActivityJog and assigns it to the current state.*

Command arguments:

ID	Name	Type	Default	Comment
01	*jobName*	*String*	*job*	*Name of the Job to define*
02	*jobDesc*	*String*		*Description for the job*
03	*defaultStage*	*Boolean*	*True*	*If true it will prepare default crawler stage to execute crawl inValues: True,False*
04	*stampPrefix*	*String*		*Prefix at timestamp*
05	*stampCount*	*Int32*	*1*	*Stamp version count*

Example :

```
wem.Job jobName="job";jobDesc="";defaultStage=True;stampPrefix="";stampCount=1;
```

4 wem.OpenSession

> *Selects and preloads local index and Experiment session information. useJobSettings option will ignore other params and use Job definition*

> *It will set report output information and create or load local index*

Alias*: OpenSession*

Command arguments:

ID	Name	Type	Default	Comment
01	*experimentSession*	*String*		

ID	Name	Type	Default	Comment
02	*IndexID*	*String*		
03	*useJobSettings*	*Boolean*	*False*	*Values: True,False*
04	*crawlFolderNameTemplate*	*String*	*	

Example :

```
wem.OpenSession experimentSession="";IndexID="";useJobSettings=False;crawlFolderNameTemplate="*";
```

5 wem.Plugin

> *Allows additional execution customization by crawling plugin*

> *It will create instance of specified plug in and set it into proper collection*

Command arguments:

ID	Name	Type	Default	Comment
01	*plugin_classname*	*String*	*	*Proper name of the crawling plugin class*

Example :

```
wem.Plugin plugin_classname="*";
```

6 wem.Run [R]

> *Runs the current crawl job*

> *Starts crawl execution*

Example :

```
wem.Run
```

7 wem.SampleFile

> *Imports sample from text file*

> *Loads the file and adds domain urls from it into context's sample list*

Command arguments:

ID	Name	Type	Default	Comment
01	*path*	*String*	*	*path to file with samples, if * it will open dialog to select the file*
02	*inWorkspace*	*Boolean*	*True*	*if true, the file path is interpreted as relative to console workspaceValues: True,False*
03	*sampleName*	*String*		*Name of the sample list, if empty it will not change current sample list name*
04	*replace*	*Boolean*	*False*	*if set to true it will replace any existing samples in the listValues: True,False*
05	*skip*	*Int32*	*0*	*Number of entries to skip, from the imported file*
06	*limit*	*Int32*	*-1*	*If set above 0, it limits the total number of domains imported*
07	*debug*	*Boolean*	*True*	*if true it will report on link preprocessingValues: True,False*

Example :

```
wem.SampleFile path="*";inWorkspace=True;sampleName="";replace=False;skip=0;limit=-1;debug=True;
```

021 Plugin: bec

Class *imbWBI.IndustryTermModel.consolePlugin.becPlugin*

```
imbWBI.ConsoleTool Console.becPlugin
```

> *This is imbACE advanced console plugin for becPlugin*

1 bec.CompareDataSets

> *What is purpose of this?*

> *What it will do?*

Command arguments:

ID	Name	Type	Default	Comment
01	*datasetA*	*String*		*--*
02	*datasetB*	*String*		*--*
03	*runName*	*String*	*DataSets*	*--*
04	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
bec.CompareDataSets datasetA="";datasetB="";runName="DataSets";debug=True;
```

2 bec.CrawlScript

> *Creates script to perform (re)build MCRepository crawl*

> *Creates script that prepares crawler, crawl job and performs the crawl*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*ODPBusinessDistantTopics*	*What component of the model to crawl*
02	*clearRepo*	*Boolean*	*False*	*If true it will clear existing MC RepoValues: True,False*
03	*debug*	*Boolean*	*True*	*If true it will show some extra information during the processValues: True,False*
04	*autorun*	*Boolean*	*True*	*If true it will autorun created script without asking userValues: True,False*

Example :

```
bec.CrawlScript name="ODPBusinessDistantTopics";clearRepo=False;debug=True;autorun=True;
```

3 bec.DataSetGraphs

> *Constructs web site graphs for pointed dataset*

> *It will load dataset from the path and construct web site graphs*

Command arguments:

ID	Name	Type	Default	Comment
01	*runName*	*String*	*word*	*report directory name*
02	*datasetPath*	*String*		*path containing dataset to load*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
bec.DataSetGraphs runName="word";datasetPath="";debug=True;
```

4 bec.DataSetReport

> *Provides detailed statistics on dataset, when current BEC configuration is applied*

> *What it will do?*

Command arguments:

ID	Name	Type	Default	Comment
01	*runName*	*String*	*word*	*Name of the report folder*
02	*datasetPath*	*String*		*Path to dataset - when other than currently loaded should be reported about*
03	*reconstruct*	*Boolean*	*True*	*If true, it will perform recornstructive analysis of all urls in the datasetValues: True,False*

Example :

```
bec.DataSetReport runName="word";datasetPath="";reconstruct=True;
```

5 bec.ExperimentGroup

> *Sets active experiment group*

> *Creates subfolder in report directory*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*BEC*	*Name for the group*
02	*description*	*String*		*Optional description for the group*
03	*script*	*String*		*Script associated with this group*

Example :

```
bec.ExperimentGroup name="BEC";description="";script="";
```

6 bec.GroupDescription

> *Sets description for current experiment group*

Command arguments:

ID	Name	Type	Default	Comment
01	*description*	*String*		*Description for the experiment group*

Example :

```
bec.GroupDescription description="";
```

7 bec.HarvestDomainList

> *Runs crawler to harvest domain list categories currently loaded*

> *It will generate and run crawler script for each category*

Command arguments:

ID	Name	Type	Default	Comment
01	*path*	*String*	*word*	*--*
02	*steps*	*Int32*	*5*	*--*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
bec.HarvestDomainList path="word";steps=5;debug=True;
```

8 bec.ImportDomains

> *Import currently selected sample lists into imbIndustryTermProject category*

> *Takes categories from webDataset*

Command arguments:

ID	Name	Type	Default	Comment
01	*word*	*String*	*word*	*--*
02	*steps*	*Int32*	*5*	*--*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
bec.ImportDomains word="word";steps=5;debug=True;
```

9 bec.Init

> *Initialize project*

> *Loads the project or creates new*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*bec01*	*project name*

Example :

```
bec.Init name="bec01";
```

10 bec.OverviewReport

> *Provides overview spreadsheet report with cross-fold average F1 metrics for all experiments ran since console boot*

> *Generates an overview report for active experiment group*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*word*	*filename without extension for the overview report*

Example :

```
bec.OverviewReport name="word";
```

11 bec.RunGroupScript

> *Runs the script associated with experiment group*

Command arguments:

ID	Name	Type	Default	Comment
01	*overview*	*Boolean*	*True*	*if overview report should be createdValues: True,False*

Example :

```
bec.RunGroupScript overview=True;
```

12 bec.Save

> *Saves project*

> *What it will do?*

Example :

```
bec.Save
```

13 bec.ToolkitTest

> *What is purpose of this?*

> *What it will do?*

Command arguments:

ID	Name	Type	Default	Comment
01	*runName*	*String*	***	*Name of the experiment log*
02	*runComment*	*String*	*BEC experiment run*	*Comment on the test run*
03	*silentDatasetLoad*	*Boolean*	*True*	*Prevents printout during dataset loadValues: True,False*

Example :

```
bec.ToolkitTest runName="*";runComment="BEC experiment run";silentDatasetLoad=True;
```

022 Plugin: mcm

Class *imbMiningContext.imbMCManager*

```
imbWBI.ConsoleTool Console.MC Manager Plugin
```

> *Manager plugin takes care about MC repositories, loads, creates and provides the instances.*

1 mcm.Close

> *Saves and unselects the currently selected MC Repository*

> *It will save all muttable variables of the repository and its content*

Command arguments:

ID	Name	Type	Default	Comment
01	*log_msg*	*String*		*Log message to write into repository log just before unselecting it, and after saving it*
02	*doReport*	*Boolean*	*True*	*if true, it will create report files for complete content of the currently selected repository: spreadsheets, folder content readmes etc.Values: True,False*
03	*debug*	*Boolean*	*False*	*If true, it will print out short report on content of the repository (if any)Values: True,False*

Example :

```
mcm.Close log_msg="";doReport=True;debug=False;
```

2 mcm.Open

> *Opens or creates new MCRepository, starts a MC session. Call this before any other MC operation.*

> *It initiates specified MCRepository and sets it as current/selected.*

Command arguments:

ID	Name	Type	Default	Comment
01	*repo*	*String*	*word*	*Name of repository to start working with*
02	*log_msg*	*String*		*A message to be written into repository log after it is initiated, e.g. adding new MCWebSites, or running Data Mining procedure XXXX*
03	*debug*	*Boolean*	*False*	*If true, it will print out short report on content of the repository (if any)Values: True,False*

Example :

```
mcm.Open repo="word";log_msg="";debug=False;
```

023 Plugin: nlp

Class *imbNLP.Project.Plugin.imbNLPProjectPlugin*

```
imbWBI.ConsoleTool Console.imbNLPProjectPlugin
```

> *This is imbACE advanced console plugin for imbNLPProjectPlugin*

1 nlp.HarvestODP

> *Harvests the Open Directory Projects*

> *It will scan for subdirectories and extract outer links*

Command arguments:

ID	Name	Type	Default	Comment
01	*start*	*String*	*https://dmoztools.net/Business/*	*Starting URL*
02	*steps*	*Int32*	*3*	*How deep the harvester is allowed to go*

Example :

```
nlp.HarvestODP start="https://dmoztools.net/Business/";steps=3;
```

2 nlp.LoadWebKB

> *Loads WebKB web site datasets*

> *It will load WebKB 7Sectors dataset*

Command arguments:

ID	Name	Type	Default	Comment
01	*path*	*String*	*G:_DOKTORAT_MAIN\SM03_Datasets\7sectors*	*Path to 7Secotrs dataset*

Example :

```
nlp.LoadWebKB path="G:\_DOKTORAT_MAIN\SM03_Datasets\7sectors";
```

024 Plugin: exp

Class *imbWBI.IndustryTermModel.consolePlugin.becExperimentPlugin*

```
imbWBI.ConsoleTool Console.becExperimentPlugin
```

> *Plugin for BEC Experiment configuration and creation*

1 exp.AddFeatureDimension

> *Defines new feature dimension*

> *It creates new feature dimension and optionally clears existing*

Command arguments:

ID	Name	Type	Default	Comment
01	*type*	*FeatureVectorDimensionType*	*similarityFunction*	*Dimension typeValues: directTermWeight,similarityFunction,topicWeight*
02	*function*	*String*	*CosineSimilarityFunction*	*Name of the function, if it is required by dimension type (e.g. similarity with class dimensions)*
03	*clearExisting*	*Boolean*	*False*	*Removes any existing feature dimension from the experiment settingsValues: True,False*

Example :

```
exp.AddFeatureDimension type=similarityFunction;function="CosineSimilarityFunction";clearExisting=False;
```

2 exp.Dataset

> *Configures dataset to be used for the experiment*

> *Set path, filter and selection options*

Command arguments:

ID	Name	Type	Default	Comment
01	*path*	*String*	*G:\imbWBI\datasets\7sectors*	*Diskdrive path, pointing to the root folder of the dataset (WebKB format)*
02	*pageLimit*	*Int32*	*1*	*Minimum number of pages that a document set (website) must have in order to be accepted for the experiment*
03	*filterEmpty*	*Boolean*	*True*	*Filters out empty documents from the datasetValues: True,False*

Example :

```
exp.Dataset path="G:\imbWBI\datasets\7sectors";pageLimit=1;filterEmpty=True;
```

3 exp.FeatureFilter

> *Configures feature selection filter*

> *It will set feature count limit and function*

Command arguments:

ID	Name	Type	Default	Comment
01	*function*	*String*	*IDFElement*	*Name of the global function that will rank the features*
02	*limit*	*Int32*	*4000*	*Number of features to be adopted*
03	*TDP*	*TDPFactor*	*chi*	*TDP factor to be applied (when used with Collection basedGlobal element)Values: none,idf,idf_prob,chi,ig,gr,or,rf*
04	*IDFc*	*IDFComputation*	*logPlus*	*Inverse Document Frequency computation variationValues: logPlus,modified,DF*

Example :

```
exp.FeatureFilter function="IDFElement";limit=4000;TDP=chi;IDFc=logPlus;
```

4 exp.GlobalIGMWeight

> *Configures a global function based on Gravity moment*

> *What it will do?*

Command arguments:

ID	Name	Type	Default	Comment
01	*l*	*Double*	*7*	*Lambda factor of IGM*
02	*weight*	*Double*	*1*	*Weigth associated with the function*
03	*removeExisting*	*Boolean*	*False*	*If any existing global factor should be removedValues: True,False*

Example :

```
exp.GlobalIGMWeight l=7;weight=1;removeExisting=False;
```

5 exp.GlobalTDPWeight

> *Configures a global function based on Term Discrimination Power*

> *It will add specified global factor, optionally if will remove any existing global factors*

Command arguments:

ID	Name	Type	Default	Comment
01	*factor*	*TDPFactor*	*chi*	*What factor should be addedValues: none,idf,idf_prob,chi,ig,gr,or,rf*
02	*weight*	*Double*	*1*	*Weigth associated with the function*
03	*removeExisting*	*Boolean*	*False*	*If any existing global factor should be removedValues: True,False*

Example :

```
exp.GlobalTDPWeight factor=chi;weight=1;removeExisting=False;
```

6 exp.GlobalWeight

> *Configures a global function in the feature weighting model - supports: ICF, ICSd, IDF, IGM, mIDF*

> *It will add specified global factor, optionally if will remove any existing global factors*

Command arguments:

ID	Name	Type	Default	Comment
01	*function*	*String*	*IDFElement*	*Name of function elemenet*
02	*weight*	*Double*	*1*	*Weigth associated with the function*
03	*IDF*	*IDFComputation*	*logPlus*	*How IDF should be computedValues: logPlus,modified,DF*
04	*removeExisting*	*Boolean*	*False*	*If any existing global factor should be removedValues: True,False*

Example :

```
exp.GlobalWeight function="IDFElement";weight=1;IDF=logPlus;removeExisting=False;
```

7 exp.kNN

> *Sets multi-class k-nearest neighbours classifier*

> *Removes any existing classifier and sets k-NN with specified settings*

Command arguments:

ID	Name	Type	Default	Comment
01	*distance*	*DistanceFunctionType*	*SquareEuclidean*	*Distance function to be used with k-NN classifierValues: SquareEuclidean,Euclidean,Cosine,Jaccard,Hamming,Dice*
02	*k*	*Int32*	*5*	*K parameter - number of neighbours to vote for class membership*

Example :

```
exp.kNN distance=SquareEuclidean;k=5;
```

8 exp.Load

> *Loads an experiment setup from the file*

> *It will try to load the specified file*

Command arguments:

ID	Name	Type	Default	Comment
01	*filename*	*String*	*bec_exp*	*name for the setup*

Example :

```
exp.Load filename="bec_exp";
```

9 exp.LocalWeight

> *Configures local function in the feature weighting model*

> *It will set computation and normalization options for feature weighting*

Command arguments:

ID	Name	Type	Default	Comment
01	*computation*	*TFComputation*	*normal*	*--Values: normal,squareRooted,glasgow,modifiedTF*
02	*normalization*	*TFNormalization*	*divisionByMaxTF*	*--Values: divisionByMaxTF,squareRootOfSquareSum*

Example :

```
exp.LocalWeight computation=normal;normalization=divisionByMaxTF;
```

10 exp.mSVM

> *Sets multi-class Support Vector Machine classifier*

> *Removes any existing classifier and sets mSVM with specified settings*

Command arguments:

ID	Name	Type	Default	Comment
01	*loss*	*Loss*	*L2*	*Loss function to setValues: L1,L2*
02	*model*	*mSVMModels*	*linear*	*Model to be used with SVM classifierValues: linear,gaussian*

Example :

```
exp.mSVM loss=L2;model=linear;
```

11 exp.PageFilter

> *Ranks and selects top-n documents from a document set*

> *It will set DocumentFilter function of EntityPlaneMethod*

Command arguments:

ID	Name	Type	Default	Comment
01	*function*	*String*	*DocumentEntropyFunction*	*Name of filter function class*
02	*limit*	*Int32*	*-2*	*number of top n pages to select, -2 will leave existing settings*

ID	Name	Type	Default	Comment
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
exp.PageFilter function="DocumentEntropyFunction";limit=-2;debug=True;
```

12 exp.ParallelExecution

> *Sets allowed number of parallel threads*

Command arguments:

ID	Name	Type	Default	Comment
01	*n*	*Int32*	*5*	*Number of parallel threads*

Example :

```
exp.ParallelExecution n=5;
```

13 exp.RenderInstruction

> *Instructs HTML to text extraction engine (EntityPlaneMethod) to produce text from xpath*

> *It will add specified instruction to the rendering instruction set, and optionally remove all existing instructions before it.*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*::BODYTEXT::*	*Instruction name, it is human-readable descriptive name or special instructin name like ::BODYTEXT::*
02	*xpath*	*String*		*XPath associated with the instruction, selects nodes to be rendered into text*
03	*weight*	*Double*	*1*	*Weight factor of the instruction, i.e. number of times the content should be repeated (boosting TF)*
04	*remove*	*Boolean*	*False*	*If true it will remove any existing instruction in the setValues: True,False*

Example :

```
exp.RenderInstruction name="::BODYTEXT::";xpath="";weight=1;remove=False;
```

14 exp.Save

> *Saves the experiment setup*

> *Destination folder is command console workspace folder*

Command arguments:

ID	Name	Type	Default	Comment
01	*filename*	*String*	*bec_exp*	*name for the setup*

Example :

```
exp.Save filename="bec_exp";
```

15 exp.SignatureSuffix

> *Custom appendix to configuration signature*

> *Sets the custom signature suffix, to be added at the end of signature used for experiment run name*

Command arguments:

ID	Name	Type	Default	Comment
01	*suffix*	*String*	*word*	*textual suffix*

Example :

```
exp.SignatureSuffix suffix="word";
```

16 exp.Validation

> *Configures k-fold cross validation or single-fold calidation*

> *Updates active instance of the configuration*

Command arguments:

ID	Name	Type	Default	Comment
01	*K*	*Int32*	*1*	*Number of folds, if 1 (or 0) it will go into single-fold mode*
02	*TestFolds*	*Int32*	*1*	*Number of folds to be used as test folds, usually 1*
03	*Randomize*	*Boolean*	*True*	*Shell content of the folds be randomizedValues: True,False*
04	*LimitExecution*	*Int32*	*-1*	*When above 0, only specified number of folds will be executed*

Example :

```
exp.Validation K=1;TestFolds=1;Randomize=True;LimitExecution=-1;
```

025 Plugin: wds

Class *imbWEM.Core.consolePlugin.webDatasetPlugin*

```
imbWBI.ConsoleTool Console.webDatasetPlugin
```

> *This is imbACE advanced console plugin for webDatasetPlugin*

1 wds.ExtractDomainList

> *Extracting sample list for crawl from existing data set*

> *It will load dataset specified and extract domain list from it*

Command arguments:

ID	Name	Type	Default	Comment
01	*DataSet*	*String*	*word*	*Path to dataset*
02	*Output*	*String*		*Path where domain list should be saved*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
wds.ExtractDomainList DataSet="word";Output="";debug=True;
```

2 wds.ExtractURLsFromDataset

> *Extracts all crawled urls from the dataset*

> *It creates single txt file with list of all URLs crawled by the dataset*

Command arguments:

ID	Name	Type	Default	Comment

ID	Name	Type	Default	Comment
01	*runName*	*String*	*word*	*Name of the report folder*
02	*datasetPath*	*String*		*Path to dataset - when other than currently loaded should be reported about*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
wds.ExtractURLsFromDataset runName="word";datasetPath="";debug=True;
```

3 wds.GetDomains

> *It will execute subset compilation and set result as active sample list*

> *It will query domains from the dataset source, using subset compilation specified*

Command arguments:

ID	Name	Type	Default	Comment
01	*subsetCompilation*	*String*	*ODPBusinessDistantTopics*	*name of the subset compilation to activate*
02	*saveFile*	*Boolean*	*True*	*if true it will save result to the sample list fileValues: True,False*
03	*construct*	*Boolean*	*True*	*if true it will prepare output WebDocumentCategory directory to store crawled contentValues: True,False*
04	*limit*	*Int32*	*-1*	*Upper limit for crawl size*

Example :

```
wds.GetDomains subsetCompilation="ODPBusinessDistantTopics";saveFile=True;construct=True;limit=-1;
```

4 wds.InitDatasets

> *Performs initiation of the mail dataset sources*

> *It will connect and check state of WebKB and ODP datasources*

Example :

```
wds.InitDatasets
```

5 wds.LoadDomainCategory

> *Loads WebDomainCategory tree from specified path*

> *It will search the specified path and load hierarchical domain list*

Command arguments:

ID	Name	Type	Default	Comment
01	*path*	*String*	*word*	*WebDomainCategory root folder to load from*

Example :

```
wds.LoadDomainCategory path="word";
```

6 wds.Test

> *It will run several diagnostic procedures*

> *What it will do?*

Example :

```
wds.Test
```

026 Plugin: itm

Class *imbWBI.IndustryTermModel.consolePlugin.itmPlugin*

```
imbWBI.ConsoleTool Console.itmPlugin
```

> *This is imbACE advanced console plugin for itmPlugin*

1 itm.CloneFVEOnSTX

> *Ensures consistant settings accross FVE models in the experiment*

> *It will take first FVE model, and make clones with different Semantic Expansion setting - other FVEs will be removed.*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	***	*Name of the experiment to process, or * to use currently selected*
02	*start*	*Int32*	*3*	*Starting value for Stx*
03	*end*	*Int32*	*8*	*Ending value for Stx*
04	*code*	*String*		*3 or 4 letter code indicating how the settings are different then in other experiments,* means sufix will be created automatically according to other modifications*
05	*kFold*	*Int32*	*-1*	*number of k-folds, when specified it overrides settings of the experiment*
06	*rnd*	*Int32*	*0*	*Overrides sample randomization, 0:no change, 1:randomize true, -1:randomize false*
07	*rename*	*String*		*Renames the experiment, if * the name is set automatically*
08	*mode*	*String*	*stx*	*Clone modes: stx, lpf*

Example :

```
itm.CloneFVEOnSTX name="*";start=3;end=8;code="";kFold=-1;rnd=0;rename="";mode="stx";
```

2 itm.Compose

> *Utilize composite template system to create an experiment instance*

> *What it load component specified and set the experiment as current*

Command arguments:

ID	Name	Type	Default	Comment
01	*shell*	*String*	*default*	*Name of experiment schell definition from composite templates sub folder of the project*
02	*classifiers*	*String*	*default*	*Name of classifier set XML file from composite template sub folder of the project*
03	*fve*	*String*	*default*	*Name of the Feature Vector Extractor XML file from composite template sub folder of the project*

Example :

```
itm.Compose shell="default";classifiers="default";fve="default";
```

3 itm.CrawlScript

> *Creates script to perform (re)build MCRepository crawl*

> *Creates script that prepares crawler, crawl job and performs the crawl*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*inIndustry*	*What component of the model to crawl*
02	*clearRepo*	*Boolean*	*False*	*If true it will clear existing MC RepoValues: True,False*
03	*debug*	*Boolean*	*True*	*If true it will show some extra information during the processValues: True,False*
04	*autorun*	*Boolean*	*True*	*If true it will autorun created script without asking userValues: True,False*

Example :

```
itm.CrawlScript name="inIndustry";clearRepo=False;debug=True;autorun=True;
```

4 itm.Diagnostic

> *Evaluates current Industry Term Model project's state*

> *It will check in which phase the current project is, and if it is ready for application*

Command arguments:

ID	Name	Type	Default	Comment
01	*verbose*	*Boolean*	*True*	*If true it will produce more in-detail report on the state of the current projectValues: True,False*

Example :

```
itm.Diagnostic verbose=True;
```

5 itm.DiagnosticMode

> *Sets configuration into system-check mode where minimum computation load is run, in single thread, just to enable debugging and system health*

> *It will: set to single thread execution, one web page per web site is loaded, k is set to 1 (k-fold validation), uses cached lexic resource.*

Command arguments:

ID	Name	Type	Default	Comment
01	*toggle*	*Int32*	*1*	*if value 0, nothing changed, if its -1 it will turn off the diagnostic mode, if its 1 it will turn on the diagnostic mode*

Example :

```
itm.DiagnosticMode toggle=1;
```

6 itm.Experiment

> *Executes currently selected experiment*

> *It will create experiment execution context and run*

Command arguments:

ID	Name	Type	Default	Comment
01	*classReporting*	*Boolean*	*True*	*If true it will create additional report on each industry / categoryValues: True,False*
02	*caseReporting*	*Boolean*	*True*	*If true it will create additional report for each processed caseValues: True,False*
03	*resultReporting*	*Boolean*	*True*	*if true it will create additional report for each validation caseValues: True,False*
04	*makeGeneralReport*	*Boolean*	*True*	*if true it will create General reportsValues: True,False*
05	*sufix*	*String*		*Adds specified sufix to experiment report folder name*

Example :

```
itm.Experiment classReporting=True;caseReporting=True;resultReporting=True;makeGeneralReport=True;sufix="";
```

7 itm.ExperimentMatrix

> *Automatically creates 3x3 matrix of experiments variating TW, TC and RX options*

> *Creates experiment matrix using LPF, DFC, IDFOn specified*

Command arguments:

ID	Name	Type	Default	Comment
01	*LPF*	*Int32*	*2*	*Low pass frequecy--*
02	*DFC*	*Double*	*5*	*Document Frequency Correction*
03	*IDFOn*	*Boolean*	*True*	*IDF on / offValues: True,False*
04	*fve*	*String*	*CSSRM*	*FVE name*
05	*shell*	*String*	*k1*	*Shell name*
06	*stxStart*	*Int32*	*2*	*Stx start*
07	*stxEnd*	*Int32*	*6*	*Stx end*
08	*StrictPOS*	*Boolean*	*False*	*Strict POS PolicyValues: True,False*
09	*Report*	*Boolean*	*False*	*Should generate general report on classesValues: True,False*

Example :

```
itm.ExperimentMatrix LPF=2;DFC=5;IDFOn=True;fve="CSSRM";shell="k1";stxStart=2;stxEnd=6;StrictPOS=False;Report=False;
```

8 itm.ExperimentRange

> *What is purpose of this?*

> *What it will do?*

Command arguments:

ID	Name	Type	Default	Comment
01	*TW*	*String*	*std*	*Term Weight Flag for HTML Tags*
02	*TC*	*String*	*std*	*Term Category*
03	*RX*	*String*	*div*	*Redux option*
04	*DFC*	*String*	*1.5,2.0,3.5,5.0,7.5,10*	*DFC values*
05	*LPF*	*String*	*2*	*LPF values*
06	*IDFOn*	*Boolean*	*True*	*IDF turn on and offValues: True,False*
07	*fve*	*String*	*CSSRM*	*FVE name*
08	*shell*	*String*	*k1*	*Shell name*
09	*stxStart*	*Int32*	*2*	*Stx start*
10	*stxEnd*	*Int32*	*6*	*Stx end*
11	*StrictPOS*	*Boolean*	*False*	*Strict POS PolicyValues: True,False*
12	*Report*	*Boolean*	*False*	*Should generate general report on classesValues: True,False*
13	*rnd*	*Int32*	*0*	*Overrides sample randomization, 0:no change, 1:randomize true, -1:randomize false, more than 1 will repeat the same model specified number of times*
14	*exName*	*String*	***	*Experiment name*
15	*classifier*	*String*	*default*	*Name of classifier compound*

Example :

```
itm.ExperimentRange TW="std";TC="std";RX="div";DFC="1.5,2.0,3.5,5.0,7.5,10";LPF="2";IDFOn=True;fve="CSSRM";shell="k1";stxStart=2;stxEnd=6;StrictPOS=False;Report=False;
```

9 itm.GetExperiment

> *Provides new experiment or loads existing*

> *It will load or create new experiment and set is as the active one, ready for execution*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*Exp01*	*name of the experiment*
02	*makeDefault*	*Boolean*	*True*	*if true it will create experiment setup with default settings (if experiment not found)Values: True,False*
03	*debug*	*Boolean*	*True*	*if true it will provide more verbose outputValues: True,False*

Example :

```
itm.GetExperiment name="Exp01";makeDefault=True;debug=True;
```

10 itm.ImportExperiment

> *Allows reproduction of earlier experiment by loading experimentSetup.xml from report folder.*

> *It will load the experiment from specified address and set it as current experiment in the console, optionally it can run the experiment immediatly.*

Command arguments:

ID	Name	Type	Default	Comment
01	*path*	*String*	*reports\STAGE05*	*Path pointing to the folder with experiment report. You don't need to specify exact file name, just directory path. The path is relative to application installation folder*
02	*execute*	*Boolean*	*True*	*If true it will automatically execute loaded experimentValues: True,False*
03	*silent*	*Boolean*	*False*	*If true, it will apply default answers if otherwise user input was required. This is intended for scripted/automatic executionValues: True,False*
04	*suffix*	*String*	*_redone*	*Report suffix used for report folder name (it will also filter out targeted experiments, in case you are pointing at* $reports$ *folder - to avoid endless execution loop)*
05	*onlyWithError*	*Boolean*	*False*	*Execute only if experiment had errorsValues: True,False*

Example :

```
itm.ImportExperiment path="reports\STAGE05";execute=True;silent=False;suffix="_redone";onlyWithError=False;
```

11 itm.Modify

> *Adjusting experiment settings*

> *Modifies current experiment*

Command arguments:

ID	Name	Type	Default	Comment
01	*TW*	*String*	*std*	*Term Weight Flag for HTML Tags*
02	*TC*	*String*	*std*	*Term Category*
03	*RX*	*String*	*div*	*Redux option*
04	*LPF*	*Int32*	*2*	*Term Weight Flag for HTML Tags*

ID	Name	Type	Default	Comment
05	*DFC*	*Double*	*1.1*	*DF Correction*
06	*IDFOn*	*Boolean*	*True*	*IDF turn on and offValues: True,False*
07	*StrictPOS*	*Boolean*	*False*	*Strict POS PolicyValues: True,False*

Example :

```
itm.Modify TW="std";TC="std";RX="div";LPF=2;DFC=1.1;IDFOn=True;StrictPOS=False;
```

12 itm.Open

> *Sets the current Industry Term Model projects*

> *It will try to load the project or to create new under specified name*

Command arguments:

ID	Name	Type	Default	Comment
01	*name*	*String*	*itm01*	*Name of project to load*

Example :

```
itm.Open name="itm01";
```

13 itm.RepoOptimization

> *Removes unused content from repositories*

> *It will use current project settings to filter repo entries for pages on language of preference and to delete irrelevant content*

Example :

```
itm.RepoOptimization
```

14 itm.Save

> *Saves the currently selected Industry Term Model project*

> *It will call for save of the currently selected Industry Term Model project*

Example :

```
itm.Save
```

15 itm.Setup

> *Performs basic setup procedure, should be called after new project created*

> *It will ask user to insert or confirm each property in configuration modules: crawlSetup, nlpRepoProcessSetup, etc...*

Command arguments:

ID	Name	Type	Default	Comment
01	*verbose*	*Boolean*	*True*	*Provides additional hints and explanations during the setup processValues: True,False*

Example :

```
itm.Setup verbose=True;
```

16 itm.SingleFoldTest

> *Trains the classifier and performs classification over complete industry classes, used for debug purposes*

> *It will use complete sample of the class to train classifier, later it will run classification on the same class*

Command arguments:

ID	Name	Type	Default	Comment
01	*prefix*	*String*		*Test name prefix*
02	*model*	*itmModelEnum*	*chunkTFIDF*	*Model to use for training and classificationValues: none,wordTFIDF,chunkTFIDF,wordTFIDF_complex,semantic*
03	*report*	*Boolean*	*False*	*if true it will create extensive debug informationValues: True,False*
04	*debug*	*Boolean*	*True*	*if true it will create extensive debug informationValues: True,False*

Example :

```
itm.SingleFoldTest prefix="";model=chunkTFIDF;report=False;debug=True;
```

17 itm.Summary

> *Scans report folders to produce summary report on all experiments*

> *Scans for experimentSetup.xml files and collects all FVE data into single table*

Command arguments:

ID	Name	Type	Default	Comment
01	*reportName*	*String*	*Summary*	*Prefix to insert in name*
02	*subfolder*	*String*		*Subfolder within report directory*

Example :

```
itm.Summary reportName="Summary";subfolder="";
```

18 itm.Test

> *Used for development, to run quick test*

> *on what ever*

Command arguments:

ID	Name	Type	Default	Comment
01	*word*	*String*	*word*	*--*
02	*steps*	*Int32*	*5*	*--*
03	*debug*	*Boolean*	*True*	*--Values: True,False*

Example :

```
itm.Test word="word";steps=5;debug=True;
```