Distributed Crawler Application the Site object properties

The properties are custom algorithm-based behavior values used as key-value pairs. Usage can be subdivided on dedicated modules that are implements specific processing or target business logic. Most of them are implemented as dedicated independent modules that are used as session-based tasks instances.

The CrawlerTask module

Name	Description

HTTP_HEADERS	HTTP headers passed to fetcher. If exist fetcher use that headers. Example of using that properties:
	"properties": {
	"HTTP_HEADERS": "<local_file_path>"
	}
	local_file_path – the “User-Agent” header values plain-text file reference in format: file://<full_path_file_name>
	Comments are supported as lines started with hash character “#”.
	one value per line. If property not defined - headers read from header file. Default headers file name get from config file with the name "headers_file". In example: headers_file=../ini/crawler-task_headers.txt
	or
	"properties": {
	"HTTP_HEADERS": {"User-Agent":["my-bot","Geko-bot",…]}
	}
	or
	"properties": {
	"HTTP_HEADERS": {"User-Agent":"<local_file_path>"}
	}
	or
	"properties": {
	"HTTP_HEADERS": {"User-Agent":"My bot"}
	}
	and any recombination of formats defined above.
	
	The rotated HTTP request header fields on the basis of local per site fields lists defined by the site’s properties.
	The target purposes is to provide the possibility to specify the list of a HTTP request header values, for example - the "User-Agent" as a record of site’s properties with the name "HTTP_HEADERS" as dict of lists where key on top level will be the header name and the value will be a list of values of that header.
	When crawler need to made the headers set for concrete HTTP request it need to get them all from a site’s properties record and to get frequencies list of usage of each value for each field from local file stored in the directory by scheme:
	<PATH_CONFIG>/<REQUEST_DOMAIN/>[Site_Id].json
	where the PATH_CONFIG - it is static path component configured in the crawler's main config file “header_file_dir”, for example - "/tmp/dc_crawler/headers/";
	and the REQUEST_DOMAIN - domain name or IP address string of the current request URL.
	[Site_Id] is a site id that formed json storage file name.
	When crawler gets the frequencies of usage of values for some header field - it sorts them and select with lower frequency. Then use its value to build headers, increment frequency of that value, serialize frequencies data and store them back in to the file.

	Functional not completely approved with tests!

HTTP_COOKIE	Format property:
	{"<re_expr>":{"stage": <stage_number>, "cookie":"<cookie_string>"}}
	
	<re_expr> - regular expression content domain name
	<stage_number> - numeric bitmask,stage for apply already saved cookies (mandatory parameter)
	<cookie_string> - default value of cookie string (always used if exists parameter).
	
	Example:
	{"nytimes.com":{"stage":1, "cookie":"RMID=007f010026e957ee843f0147;"}}
	OR
	{"nytimes.com":{"stage":1}}
	
	stage_number support next values:
	1 - regular
	2 - redirect
	3 - robots
	4 - RSS
	
	HTTP cookies passed to fetcher. If exist fetcher use that headers. Example of using that properties:
	"properties": {
	"HTTP_COOKIE": "{\"nytimes.com\":{\"stage\":4, \"cookie\":\"12345\"}}"
	}
	If not exists cookies read from cookie file. Default cookie file name get from config file with the name "cookie_file". In example:
	headers_file=../ini/crawler-task_cookie.txt

STORE_HTTP_REQUEST	store http request (positive value) or not
STORE_HTTP_HEADERS	store http headers (positive value) or not
HTTP_PROXY_HOST	Proxy host
HTTP_PROXY_PORT	Proxy port

MIME_TYPE_STORE_ON_DISK	List of mime types allowed to store on disk
	Allowed values:
	Empty string “” – don’t store on disk,
	“*” – store any content on disk
	Example:
	"properties": {
	" MIME_TYPE_STORE_ON_DISK ": "*"
	}

HTTP_AUTH_NAME	Authentication name
HTTP_AUTH_PWD	Authentication password
HTTP_POST_FORM **obsolete**	HTTP POST form
EXTERNAL_URL	External fetcher URL, format: <some_url>%URL% The %URL% macro will be substituted with effective target URL value before fetcher call. For example:

	http://mysite.com/wrapper.php?url=%URL%
HTML_RECOVER	Use tidy lib to recover the source HTML content or not. Value “1” – means always recover. Value “2” – means recover if the regular DOM parser failed cause wrong HTML structure. In case of another value – no recovering at all.
MIME_TYPE_AUTO_DETECT	Auto Detect MIME content property format: [{"url_expression":"regular_expression", "mode":mode_number, "url_types":[list_of_url_types], "url_parent":[list_of_parent_type], "content_types":[list_of_content_types]}, …] regular_expression – regular expression to match the URL. If omitted – any url assumed as matched. mode_number: 0 – always; 1 – if not defined in HTTP header response, 2 – if match listed in content_types, 3 – if not matched listed in content_types. If omitted – 0 assumed. list_of_url_types – list of value of url.type field to apply only. If omitted – apply to any type. list_of_parent_type – list of parent type: 0 – only for parent urls, 1 – only for none parent. If omitted – apply for any type. list_of_content_types – list of MIME content types names, for example: “text/html”, “text/xml”, etc… Optional only if referenced in mode_number. Examples: ["mode":1] Apply check and detect for any url in any case if content-type not defined in the HHTP response header. [{"url_expression":"(.*)mydomain.com(.*)"] Apply check and detect always for URLs matched mydomain.com domain. ["url_parent":[0]] Apply check and detect always for parent URLs [{"url_expression":"(.*)mydomain.com(.*)", "mode":3, "url_types":[1], "url_parent":[0], "content_types":["text/xml","application/rss+xml"]}] Apply check and detect for domain mydomain.com, only for url.type=1, only for parent URLs if defined by HTTP response header content type is not "text/xml" or "application/rss+xml".

PROCESSOR_NAME	Don’t process content
	Allowed values:
	"NONE" – Don’t process content
	“FEED_PARSER” – apply “FEED_PARSER” processor
	“RSS” – apply “RSS” processor
	Example:
	"properties":
	[
	{
	"siteId": "rss_feed",
	"name": "PROCESSOR_NAME",
	"value": "FEED_PARSER"
	}
	]

STORE_COOKIES	Store url’s cookies on disk (positive value) or not
AUTO_REMOVE_RESOURCES	If not set or empty auto remove is off.
AUTO_REMOVE_ORDER	SQL “ORDER” criterions for “AUTO REMOVE” operation

	Example: "AUTO_REMOVE_ORDER": "ContentType ASC, Crawled ASC, TagsCount ASC, CDate ASC"
AUTO_REMOVE_WHERE	SQL “WHERE” criterions for “AUTO REMOVE” operation Example: "AUTO_REMOVE_WHERE": "ParentMd5<>\"\" AND Status IN (4,7) AND DATE_ADD(UDate, INTERVAL %RecrawlPeriod% MINUTE)<NOW()"
AUTO_REMOVE_WHERE_ACTIVE	The condition criterion to calculate the number of active URLs for the site. By default if not defined – the: “NOT (`Status`=4 AND `Crawled`=0 AND `Processed`=0)” used. This means enumerate all URLs that are not set as processed on another host in multi-host installation.

HTTP_REDIRECTS_MAX	Max HTTP redirects (hops count)
	Example:
	“properties”:
	[
	{
	"siteId": "c241444dcd1b03bf04549448830c8942",
	"name": "HTTP_REDIRECTS_MAX",
	"value": 5
	}
	
	]

HTML_REDIRECTS_MAX	Max HTML redirects (hops count)
	Example:
	“properties”:
	[
	{
	"siteId": "c241444dcd1b03bf04549448830c8942",
	"name": "HTML_REDIRECTS_MAX",
	"value": 5
	}
	
	]

COLLECT_URLS_XPATH_LIST	List of XPath to collect urls from page. Format: {"sets":{"url_re":["xpath1", "xpath2"], …}, "mode": joing_mode, "date_format":""} "sets" – key-value dict() of regular expression applied to the URL and list of xpathes, used if matched. "url_re" – regular expression value used to test URL, empty string value matched any URL. "xpath1" – an xpath expression. "mode" – optional, defines how the set of xpath will be joined with default. If omitted – overwrite mode used. joing_mode – mode of joining, 0 – overwrite, 1 – append. "date_format" – format for the date macro same as for filters (DATE, SHORTYEAR, YEAR, MONTH, DAY, HOUR, MINUTE, SECOND). If not set, default value is '%Y-%m-%d %H:%M:%S' Example1: {"sets":{"":["//rss/channel", "//channel/link"]} } To use defined list of xpathes for any URL and overwrite default list of xpath expressions. Example2:

	{"sets":{".*domain.com/index1/.*":["//img/@src"]}, "mode": 1} To add xpath to collect URLs of images to default set of xpathes if URL matching with regular expression.
RECRAWL_NO_ROOT_URLS	0 – Don't recrawl urls except root urls

COLLECT_POST_DATA	Rule to collect POST data from urls collected from resource (not working if url.State=1 – not collect urls)
	Example of using that properties:
	"properties": {
	"COLLECT_POST_DATA ": "1"
	}
	If not set or not exist – not collect POST data from collected urls
	OBSOLETE!

URL_TEMPLATE_REGULAR	Defines a template for the not realtime crawling batch type. The template used to substitute the original URL and use a resulted URL in farther processing. This substitution done before HTTP request and can be used to submit some request to the external web server or service to process it instead to submit it regular way. For example, if the value of this property is: http://external.com?url=%URL% and the processing URL is: http://mypage.com the resulted URL will looks like: http://external.com?url=http://mypage.com if the URL_TEMPLATE_REGULAR_URLENCODE is not set or value is not positive. In another case it will looks like: http://external.com?url=http%3A%2F%2Fmypage.com and the substituted value will be urlencoded.
URL_TEMPLATE_REALTIME	The same as the URL_TEMPLATE_REGULAR but for the realtime batch crawler type.
URL_TEMPLATE_REGULAR_URLENCODE	Defines is the substituted URL value for the URL_TEMPLATE_REGULAR will be urlencoded or not. Positive value means encode.
URL_TEMPLATE_REALTIME_URLENCODE	The same as URL_TEMPLATE_REGULAR_URLENCODE but for realtime batch crawler type.
RECRAWL_URL_AGE_EXPRESSION	The SQL expression to evaluate is the URL was in activity during the re-crawl period for this site. Used in crawling cycle and defines does URL state and dates will be updated if RecrawlPeriod>0 and URL is not an RSS (URL.Type<>4). If the evaluated value >0 – this treated as that URL is fresh and will not be updated. If this property is not define the default expression used: (DATE_ADD(UDate, INTERVAL %RECRAWL_PERIOD% MINUTE)-NOW())
RECRAWL_URL_UPDATE_STATUS	The integer value that defines the URL.Status value that will be used to update in case of freshness check described in RECRAWL_URL_AGE_EXPRESSION returns not positive value and URL status will be updated. If this property not set the default value is 1 (NEW) that will lead to completely re-crawl this URL. If the value is "-1" this means that the Status will not be updated at all.
RECRAWL_URL_UPDATE_TCDATE	The SQL expression to update the URL.TcDate the same condition as defined for the RECRAWL_URL_UPDATE_STATUS. If the value is not defined the default expression id NOW(). If the value defined as empty string "" – the URL.TcDate will not be updated at all.
RECRAWL_URL_UPDATE_CDATE	The SQL expression to update the URL.TcDate the same condition as defined for the RECRAWL_URL_UPDATE_STATUS. If the value is not defined the default behavior is not to update the URL.CDate. If the value defined as empty string "" – the URL.CDate will not be updated at all.
RECRAWL_URL_UPDATE_UDATE	The same as defined for the RECRAWL_URL_UPDATE_TCDATE.
RECRAWL_URL_UPDATE	Manages a behavior of the crawler during a URL collect operation. Format: {"url_re":{"new":insert_new, "fields":{"Status":value, "UDate":value, ...}}(positive value) or not url_re - regular expression for URL match check. If a URL matched - dictionary of fields used. Dictionary of fields: insert_new - positive value means insert if not exists; "fields" - dictionary fields names in DB schema namespace. If value is null - it will not be used in insert or update and save it's default value.

	Also the SQL expression values need to be supported for datetime fields like UDate, for example: "NOW()" can be used.
ROBOTS_MODE	Explicitly enables or disables support of the robots.txt functionality. If positive or not defined it is enabled.
ROBOTS_COLLECT	The same as ROBOTS_MODE for collect URLs operation. Effective usage with depth=2.
ROBOTS_CACHE	Use local FS cache per domain name per site the same way as it is done for headers rotation. Reset of the cache make in first implementation based on re-crawl event (when crawling a root URL).
DYNAMIC_FETCH_ONLY_FOR_ROOT_URL	
REFERER_SELF_URL	Sets mode of the referrer field value: 0 – do not add referrer 1 – self URL (default) 2 – self's URLS domain name 3- parent URL
URLS_SCHEMA	The crawler the URLs schema automated variations json in format: {"type":1, "parameters":{...}, "file_path": "/tmp/urlvarifile.json", "mode":0, "max_items":1, "delimiter":";","format":"plain-text","url_encode":1,"reset_on_change_items":1, "batch_insert": 1} Parameters value depends on type: 0 - Disabled {} 1 - Predefined sets of parameter's names and values to vary: {"param1":["value1", "value2", "value3"],...} 2 - Predefined types of parameter's values: incremental integer; {"param1": {"min":0, "max":100, "step":1},...} 3 - Predefined types of parameter's values: random integer; {"param1":{"min":0, "max":100000},...} 4 - Predefined types of parameter's values: random string; {"param1":{"min":0, "max":8, "chars":0,"case":0},...} chars: 0 - any alphanumeric latin1, 1 - only hexadecimal case: 0 - lower, 1 - upper. The "max_items" - limits maximum items in the list of a parameter's values to be processed in one crawler session for that batch item. At next start next items in the list of parameter's values will be processed that based on frequencies for each value. The "mode" defines how to behave with the batch items: 0 – one item and URL with replace existing macro; 1 – several items: replace macro in the existing batch item and add new items up to "max_items" limit. The "file_path" – file (extension is .json) use for load "parameters" list from file system The "format" - type of format input data from "file_path". Supported values: "json" and "plain-text" The "delimiter" - delimiter value using in case of "plain-text" value in "format" field. The "url_encode" - flag for optionally support a urlencoding of a values of parameters as a part of destination URL creation. If positive then encode any parameter value. The defined positive value of the "reset_on_change_items": 0 - lead to reset all frequencies to zero value if list of values changed (by number or value); 1 – lead to reset all frequencies always; 2 – not reset, 3 – set frequencies value only for new by minimal value of present. (NOT IMPEMENTED) The "batch_insert" defines does new item will be inserted in to this batch or not. Default value or if this field omitted is 0. If value is zero – no one new item inserted in to this batch and it will finished as empty. If value is 1 – all new items inserted in to this batch. If value is 2 – only first new item inserted in to this batch.

	Mandatory parameters: `type`, `mode`, `max_items` and `parameters`. Note: in crawler-task.ini exist parameter “url_schema_data_dir=” in section “CrawlerTask”. Use this path will be created automatically file to store stat data to save rotation time marks and frequencies. File name make use this rule: <url_schema_data_dir>/url_schema_data_<SiteId>.json

USER_PROXY	Set the proxy configuration for rotated proxy settings, the value is json in format:
	"{\"source\": 0, \"file_path\": \"file11.json\", \"proxies\":
	{\"toxic.com:9000\" : {\"host\": \"toxic.com:9000\", \"domains\":[\"www.latimes.com\"],
	\"priority\": 44, \"limits\": null},
	\"proxic.com:9000\" : {\"host\": \"proxic.com:9000\", \"domains\":[\"*\"],
	\"priority\": 11, \"limits\": null},
	\"nosic.com:9000\" : {\"host\": \"nosic.com:9000\", \"domains\":[\"www.latimes.com\"],
	\"priority\": 1, \"limits\": null}}}"
	
	in most common case if you want to use one proxy for any domain json will be looks like this:
	
	{
	"source": 0,
	"file_path": "/home/hce/proxy.json",
	"proxies": {
	"85.17.141.35:80": {
	"host": "85.17.141.35:80",
	"domains": ["*"],
	"priority": 11,
	"limits": null
	}
	}
	}
	
	where file can be empty json
	
	host - used proxy, matches with host
	
	domains - in this case means 'any', but can be like
	
	"domains": ["www.domain.com"],
	
	note, that proxies can be a list like:
	
	"proxies": {
	"121.41.161.110:80": {
	"host": "121.41.161.110:80",
	"domains": ["www.indiegogo.com"],
	"priority": 44,
	"limits": null
	},

	"124.206.133.227:80": {
	"host": "124.206.133.227:80",
	"domains": ["*"],
	"priority": 11,
	"limits": null
	}
	
	and all this json must be a string, i.e. must be decoded so whole property must be like this:
	
	Sample 1:
	
	"USER_PROXY": "{\"source\":
	0,\"file_path\":\"\/home\/hce\/proxy.json\",\"proxies\":{\"84.23.107.195:8080\":{\"host\":\"84.23.107.195:8080\",\"domains\":
	[\"*\"],\"priority\":11,\"limits\":null}}}"
	
	Sample 2:
	
	"USER_PROXY":
	"{\"tries_count\":4,\"source\":0,\"proxies\":{\"84.23.107.195:8080\":{\"host\":\"84.23.107.195:8080\",\"domains\":[\"www.indiegogo.com\"],\"priority
	44,\"limits\": null}}}",
	
	Status value for update for USER_PROXY:
	fields names:
	"status_update_empty_proxy_list"
	"status_update_no_available_proxy"
	"status_update_tries_limit"
	fields value - the status values (1 - NEWS, 2 - SELECTED_FOR_CRAWLING, etc...)
	Example:
	{"tries_count":2,"source":1,"proxies":{},"file_path":
	"/tmp/dc_crawler/","status_update_empty_proxy_list":4,"status_update_no_available_proxy":4,"status_update_tries_limit":4}
	
	The raw content check with RE pattern string, rotate proxy and increment faults counter if not passed:
	raw_content_check – settings for raw content RE pattern check;
	raw_content_check content:
	patterns – list of regular expressions patterns;
	rotate – if positive – rotate proxy to next one and repeat request; Optional, default value 1.
	faults – increment proxy's faults counter of on this value (zero means to leave not incremented). Optional, default value 1.
	
	source – type of source proxies list (0 – in site property, 1- from DB)
	file_path – path for file for save frequencies of used proxies
	Example:
	{"tries_count":2,"source":1,"proxies":{},"file_path": "/tmp/dc_crawler/", "raw_content_check":{"patterns":[".*Pardon.*"," .*suspicious\sactivity.*"],
	"rotate":1, "faults":1}}

FETCHER_MACRO	Set the list of JavaScript macro that will be executed by the dynamic fetcher sequentially in direct order. Simple format: [\"MacroCode1\", \"MacroCode2\",\"MacroCodeN\"]

	Extended format: {"name":"macro sets name", "sets":[{"name":"set name", "items":["macro item 1","macro item 2",...], "repeat":3, "delay":2}, ...], "result_type":result_type, "result_content_type":"result_content_type", "result_fetcher_type":result_fetcher_type} "name" – string name; "sets" – list of sets of macro, each set can to have name, macro items (list of macro strings), repeat and delay fields; "repeat" – number of tries to execute a macro set; "delay" – delay between tries to execute of a macro set, sec; "result_type" – 0 (default) the named array of series fields and values of scraped data per detected item, for example [{"title":"title_value", "link":"link_value", ...}]; – 1 the links strings list/array, for example ["http://url1.com", "http://url2.com", ...]; 2 – string value that will be treated by a crawler as a content buffer, for example as HTML or XML; 3 – auto, will set results type depending on structure returned from macro: list of dicts, list of strings or string. "result_content_type" – a MIME content-type name, for example "text/html", "text/plain", "text/xml" and so on; depends on a content type the returned content buffer will be passed to a proper post-fetching and scraping processing. "result_fetcher_type" – a fetcher type that need to be used to crawl result item in case of "result_type" is 1. Macro will be executed if no fatal error happened before. The execution acts before a delay of wait on page became ready (HTTP response timeout in request for the dynamic fetcher). All items will be executed in sequential direct order. If a macro returns result and it is a valid json – it is accumulated in list and returned instead of page content with the proper MIME content-type "text/json". **macro item** – string contains one of this values: - has a numeric characters only, for example "5" – this means that it is not a code of a macro, but timeout in seconds that will lead to sleep() before next macro item will be executed; - starts with "http://" or "https://" - this means that it is a URL to download a body of a macro JS code from external server; - starts with file:// - this means that it is a FS path to load a body of a macro JS code from local file. - has a numeric values delimited with a colons characters in format: <sleep_delay>:<tries_number>:<JS_code_check_data_ready> for example: 5:20:return window.DATA_READY; this means that macro code will be executed and if is not returned Boolean True, will sleep on 5 seconds and will to try this up to 20 times or until True will be returned; - another textual value treated as a JS code and executed directly.
HTTP_FREQ_LIMITS	The possibility to limit the maximum number of requests from host configurable for Project/Site or domains names list based on local file frequency data storage. Make a crawling timeout for HTTP request on the basis of local per site fields lists defined by the sites_properties record. The target purposes is to provide the possibility to specify the list of a definitions for domain names and maximum frequencies as a record of site_properties with the name "HTTP_FREQ_LIMITS" as dict of lists where key on top level will be the domain name regular expression and value is a set of options like "max_freq", "max_delay", etc. When crawler need to made the concrete HTTP request it need to check is frequency is okay. To do this check it stores and reads a local file in json format. A fields from local file stored in the directory by scheme: <PATH_CONFIG>/<REQUEST_DOMAIN/>[Site_Id].json where the PATH_CONFIG - it is static path component configured in the crawler's main config file, for example - "/tmp/dc_crawler/frequencies/"; and the REQUEST_DOMAIN - domain name or IP address string of the current request URL. [Site_Id] is a site id, that formed json storage file name.

	When crawler get the frequencies of usage and compares it with limit value "max_freq" - a delay calculated. If delay is succeeds a delay's limit "max_delay" and "max_delay">0 - it used before request. If not, the request skipped and batch item need to be updated to NEW state to be fetched for next another crawling try. Sample: {"127.0.0.1" : {"max_freq": 20, "max_delay": 10, "randomized":1}, "PATH_CONFIG": "/tmp/"}
CONTENT_TYPE_MAP	Defines mapping of content types from original returned from site to another that will be stored. For example: "CONTENT_TYPE_MAP": "{\"text/html\": \"text/xml\"}"
CONNECTION_TIMEOUT	Set the dedicated timeout for the requests fetcher (static fetcher) for the TCP socket connects operation. If not set, default value is 500 ms.
HOST_ALIVE_CHECK	Check is a URL's host alive, including the domain name resolving and TCP socket connect. Format: {"method":0, "domain_name_resolve":1, "connect_resolve":1, "connection_timeout":0.5}
HOST_ALIVE_CHECK_PROXY	The same as HOST_ALIVE_CHECK but check proxy if rotated before use.
HTTP_REDIRECT_LINK	Numeric value: 0 – nothing to do 1 – if set, the "link" tag value replaced with source URL field from URL object. 2 - if set the link URL detected by scraper (any kind) and set as "link" tag value will be replaced with the URL from the "Location" HTTP header if it was returned in response during a crawling. 3 – create "redirect_url" tag value will be replaced with the URL from the "Location" HTTP header if it was returned in response during a crawling. 4 - create "redirect_url" tag value will be replaced from URL object if the URL cannot be extracted from the "Location" HTTP header.
FETCHER_TYPE	Changes the fetcher type from default to specify by RE expression for the URL string. Format simple: {"url_re_pattern":fetcher_type, ...} url_re_pattern - regular expression to check on URL fetcher_type - 1 static, 2 dynamic format full: {"url_re_pattern":{"fetcher_type":fetcher_type, "url_gen":url_generation, "url_type":[0,1,…]}, ...} url_generation: 0 – only root, 1 – only not root, 2 – any; url_type: corresponds with URL.type field values [0,1,…] not implemented
LOGGER	Defines custom logging per project file name for criticl section of the crawler application, format: {"suffix":"%PROJECT_ID%", "dir":"/tmp/%PROJECT_ID%"} suffix – file name suffix dir – directory name/path (Not supported yet)
RSS_FEED_ZERO_ITEM	???
HTTP_CODE_STATUS_UPDATE	Defines the status value to update for HTTP codes. Format: {"200": 4, "403": 1, "503": 1} If a code not listed the default Status value is 4 (crawled).
URLS_FIELDS_INIT	Add the support of the pre-defined sets of URL object fields initialization depends on condition. The project property "URLS_FIELDS_INIT" structure: {code} {"field_name_to_init1": {"conditions":["condition_expression1", ...], "value":"value_expression"}, ...}

	"field_name_to_init" – a field name in new URL object to init "value" – an initialization value "conditions" – a list of conditions to check and apply initialization value if condition matched. If condition is not matched – a value will be taken from parent URL object. Format of a condition list item string: 1) [parent.]field_name = value 2) [parent.]field_name == value 3) [parent.]field_name <> value 4) [parent.]field_name != value 5) [parent.]field_name is empty 6) [parent.]field_name match regular_expression 7) [parent.]field_name search regular_expression “parent.” – prefix used to refer on parent URL object field, if omitted – the current/default URL object’s field used. “field_name” – name of a field of the URL object. “value” – some initialization value, numeric or string, depends on field. “=” – operation to compare on equal. “match" - operation to execute RE match. “search" - operation to execute RE search. *Note, that three parts of condition need to be separated with space. For example: {"type": {"conditions":["parent.contentType = \"text/html\"", "parent.url match \"*.site\\.com/news\""], "value":"0"}} Defines the set field “type” with value “0” in case of the parent URL field “contentType” equal with value “text/html” and parent URL field “url” matched with regular expression “*.site\.com/news”
SQL_EXPRESSION_FIELDS_UPDATE_CRAWLER	An SQL-expression based update of a values of a Site or URL objects before final update, format: [{"object_name":{"field_name":{"sql_expression":value_type}}}] "object_name" – name of an object to affect: “Site” or “URL” "field_name" – field name in DB schema names space for the Site and URL objects. "sql_expression" – the SQL expression that will be evaluated on the MySQL engine in space of the host’s DC’s DB and with support of a macro names of correspondent object_name fields in upper case. value_type – the type of a value that will be used to update, 0 – result of an SQL expression calculation customized to integer, 1 – customized to string, 2 – customized to DateTime string ISO 8601.. Examples: [{"URL":{"Status":{"IF(%ERRORMASK% & 128 > 0, 1, %STATUS%)":0}}}] To set the URL.Status field value as 1 (NEW) in case of the URL.ErrorMask bit 7 is ON and leave untouched if not.
ALLOWED_CTYPES	String content allowed ctypes used ‘,’ as delimiters.
PROTOCOLS	Allowed protocols list in json string format
DETECT_MODIFIED	{"compare":0, "algorithm":1, "behavior":1, "mode":0}
HTTP_REDIRECT_RESOLVER	A resolver configuration to resolve redirects for collected URLs before insertion in to the storage DB. {"METHOD":"method_name", "URL":["url_item", …], "MAX":max_redirects, "TYPES":types_list} also support extended form of property: {"METHOD":"method_name", "URL":["url_item", {"URL":["url_item", ...],"METHOD":"method_name"}, …], "MAX":max_redirects, "TYPES":types_list}

	method_name – HEAD, SGET, DGET; (SGET – method GET by static fetcher, DGET – method GET by dynamic fetcher) url_item – regular expression for URL match; max_redirects – maximum redirects, 10 for default types_list – redirect types, [301,302,303,304] by default Example: {"METHOD":"HEAD", "URL":["http://test.com/.*"]} or {"METHOD":"HEAD", "URL":["http.*", {"URL":[".*theaustralian.*"], "METHOD":"SGET"}]}
REPLACE	Format: [{"WHERE":"value", "WHAT":"value", "WITH":"value", "CONDITION":"value"} ,...] keys: WHERE - name of a content, can be "RAW" at least; WHAT - regular expression defines what to replace; WITH - string to replace with; CONDITION - optional, the SQL expression to be executed with urls_<site_id> table; Sample site property: [{"WHERE":"RAW", "WHAT":"<!--(.|\n)*?-->", "WITH":" ", "CONDITION":"ContentType='text/html'"}] Sample for demo-form: "REPLACE": "[{\"WHERE\":\"RAW\", \"WHAT\":\"<!--(.|\\n)*?-->\", \"WITH\":\" \", \"CONDITION\":\"\"}]",

The ProcessorTask module

Name	Description

PROCESS_CTYPES	List of MIME content types supported by proceoosr.
	Example:
	“properties”:
	[
	{
	"siteId": "2f105d68146db820c23aa3fc6010888d",
	"name": "PROCESS_CTYPES",
	"value": "text/html,text/xml,application/rss+xml"
	}

	]

TIMEZONE
DEPRECATED
OBSOLETE
	Site's timezone
	Example:
	"properties":
	[
	{
	"siteId": "2f105d68146db820c23aa3fc6010888d",
	"name": "TIMEZONE",
	"value": "JST"
	}
	]

REFINE_TAGS	List of tags to be refined after extracting
PROCESSOR_NAME	Processor's name Allowed types: "" – default processor "STORE" – store on disk "FEED_PARSER" – feed parser processor "RSS" = rss processor
CONTENT_HASH	Sets the way to calculate unique content hash stored in the dc_urls.<SITE_ID>_urls.ContentURLMd5 field. Calculations made using processed result's tags. Example: "properties": ["CONTENT_HASH": "{ 'algorithm': 1, 'tags': 'title,description,link,pubdate' }",] Allowed values of 'algorithm': 1 – calculate use MD5 algorithm, 2 - The preparation is split the content with set of standard space delimiters like regular expressions \S, and all none alphanumeric characters, sort split set in alphabetical order, remove duplicates, join without delimiter in one string. 3 - The preparation is the same as for the algorithm 2 but all items in set convert to their stems with the snowball stemmer (https://pypi.python.org/pypi/snowballstemmer) before sort and duplicates delete. 4 - The preparation makes the soundex string from each word 'tags' – a csv list of tag names using for calculate
PROCESSOR_PROPERTIES	TODO: description need to be updated for common cases of template-based and news type of the scraping. Used to set the processing algorithm and modules, for example for the real-time request to return RAW HTML buffer base64-encoded. For example for the "NEWS" scraping it is: "PROCESSOR_PROPERTIES": "{\"algorithm\":{\"algorithm_name\":\"user_name_algorithm\"},\"modules\":{\"user_name_algorithm\": [\"ScrapyExtractor\",\"GooseExtractor\",\"NewspaperExtractor\"]}}" The name "SCRAPER_TEXT_REDUCER" can be use to set up the text reducer procedure patterns set in format:

['\n', '\r\n', '\t', ' ', '
', ' ', ' ']
This set used by default if not defined.
Support macro %ATTRIBUTES%. Sample: ["a":"a %ATRIBUTES%"]

The name "SCRAPER_TEXT_REDUCER_MASK" it is integer bit mask can be use for setting index element from "SCRAPER_TEXT_REDUCER".
Default value used if property not set: 65535

The name "SCRAPER_KEEP_ATTRIBUTES" can be use for save attributes in output response. It has format:
{"IMG":["src","title","alt"]}
The name "CLOSE_VOID" can be used to set mode for add '/' to not closed tags.
Support values: 0,1,2
0-remove '/'. (Default behavior)
1-always add '/'
2 – auto mode. Add only if '/' was found in raw document.

The name "EXTRACTOR_NEWSPAPER_MAX_EXECUTION" can be used to set up max allowed timeout for 'Newspaper extractor'
The name "EXTRACTOR_GOOSE_MAX_EXECUTION" can be used to set up max allowed timeout for 'Goose extractor'
If property is not defined use default timeout value 20 seconds.
Sample usage:
"PROCESSOR_PROPERTIES": "{\"algorithm\":{\"algorithm_name\":\"user_name_algorithm\"},\"modules\":{\"user_name_algorithm\":
[\"ScrapyExtractor\",\"GooseExtractor\",\"NewspaperExtractor\"]}, \"EXTRACTOR_NEWSPAPER_MAX_EXECUTION\":30, \"EXTRACTOR_GOOSE_MAX_EXECUTION\":30},
\"CLOSE_VOID\":2"

The name "SCRAPER_TEXT_MARKUP" can be used to as the innerText method to save the markup like tags P,H,TD,LABEL and so on. It optionally and configure the set of the tags names and marker patterns that will be inserted in to the text in places where that tags was closed.

Format example:
{"P": "\n", "H": "\n", "TD": " ", "TR": "\n", "BR": "\n", "LI": "\n", "LABEL": " "}
Sample usage:
"PROCESSOR_PROPERTIES":
"{\"algorithm\":{\"algorithm_name\":\"user_name_algorithm\"},\"modules\":{\"user_name_algorithm\":[\"ScrapyExtractor\",\"GooseExtractor\",\"NewspaperExtractor\"]},\"SCRAPER_DOWNLOAD_IMAGES\":1,\"SCRAPER_TEXT_MARKUP\":{\"DIV\":\"\\n\",\"P\":\"\\n\",\"H\":\"\\n\",\"TR\":\"\\n\"}}"

The name "SCRAPER_LANG_DETECT" can be used to optionally the language detection for set of the tags.

Format example:

{"prefix":"lang_","suffix":"_lang","tags":["title","content_encoded"]}}

"prefix" - prefix for generate new tag name use base name from "tags"

"suffix" - suffix for generate new tag name use base name from "tags"

"tags" – list of base names of tags. Also support values "*" or ["*"] for use all tags and values "&" or ["&"] for detect summary lang from all tags.

"map" - dictionary of rules for mapping languages.

"size" - limit size of the text buffer for detecting of the languges.

	Sample usage: "PROCESSOR_PROPERTIES": "{\"algorithm\":{\"algorithm_name\":\"user_name_algorithm\"},\"modules\":{\"user_name_algorithm\":[\"ScrapyExtractor\",\"GooseExtractor\",\"NewspaperExtractor\"]},\"SCRAPER_DOWNLOAD_IMAGES\":1,\"SCRAPER_TEXT_MARKUP\":{\"DIV\":\"\\n\",\"P\":\"\\n\",\"H\":\"\\n\",\"TR\":\"\\n\"},\"SCRAPER_LANG_DETECT\":{\"prefix\":\"lang_\",\"tags\":[\"title\",\"content_encoded\"]}}" or "PROCESSOR_PROPERTIES": "{\"algorithm\":{\"algorithm_name\":\"user_name_algorithm\"},\"modules\":{\"user_name_algorithm\":[\"ScrapyExtractor\",\"GooseExtractor\",\"NewspaperExtractor\"]},\"SCRAPER_DOWNLOAD_IMAGES\":1,\"SCRAPER_TEXT_MARKUP\":{\"DIV\":\"\\n\",\"P\":\"\\n\",\"H\":\"\\n\",\"TR\":\"\\n\"},\"SCRAPER_LANG_DETECT\":{\"suffix\":\"_lang\",\"tags\":[\"title\",\"content_encoded\"]}}" or "PROCESSOR_PROPERTIES": "{\"algorithm\":{\"algorithm_name\":\"user_name_algorithm\"},\"modules\":{\"user_name_algorithm\":[\"ScrapyExtractor\",\"GooseExtractor\",\"NewspaperExtractor\"]},\"SCRAPER_DOWNLOAD_IMAGES\":1,\"SCRAPER_TEXT_MARKUP\":{\"DIV\":\"\\n\",\"P\":\"\\n\",\"H\":\"\\n\",\"TR\":\"\\n\"},\"SCRAPER_LANG_DETECT\":{\"tags\":[\"content_encoded\",\"title\",\"description\"]}}", or "PROCESSOR_PROPERTIES": "{\"algorithm\":{\"algorithm_name\":\"user_name_algorithm\"},\"modules\":{\"user_name_algorithm\":[\"ScrapyExtractor\",\"GooseExtractor\",\"NewspaperExtractor\"]},\"SCRAPER_DOWNLOAD_IMAGES\":1,\"SCRAPER_TEXT_MARKUP\":{\"DIV\":\"\\n\",\"P\":\"\\n\",\"H\":\"\\n\",\"TR\":\"\\n\"},\"SCRAPER_LANG_DETECT\":{\"tags\":[\"content_encoded\",\"title\",\"description\"],\"size\":100}}", **Metrics**: The name “CONTENT_METRICS” used to provide json definition in format: {“Name”: Value, …} metrics names: “TAGS_NUMBER” - Number of extracted tags in processed content. “TAGS_NUMBER_PERCENT” - Percent of extracted tags among all tags. “WORDS_NUMBER” - Words number in processed content. “WORDS_NUMBER_PERCENT” - Percent of good words among all words in processed content. “CONTENT_SIZE” – Total number of characters in processed content. “CONTENT_SIZE_PERCENT” - Percent of good characters (alphanumeric only) among all content's characters. * Metrics values can be “null” or numeric pre-calculated. * Metrics stored in the processed content if the template has a “%metrics%” macro definition. * Metrics can be used for real-time API request processing if the fetcherType is set as 7 (auto). In this case, the additional field - “FINALIZER_METRICS” need to be set as json string with structure: {“TYPE”: 0, “CONTENT_METRICS”: [{“NAME” : “TAGS_NUMBER”, “LIMIT_MAX”: 2, “LIMIT_MIN”:2}]}

	where “TYPE” values are: 0 – simple, 1 – OR operation among all contents in CONTENT_METRICS, 2 – AND operation among all contents in CONTENT_METRICS.

template	Project’s/Site’s template:
	Example:
	
	For the “News scraping” (the same as on demo page#1):
	{
	"templates":[
	{
	"output_format":{
	"name":"json",
	"header":"[\n",
	"items_header":"",
	"item":"{\n\"pubdate\":\"%pubdate%\",\n\"title\":\"%title%\",\n\"media\":\"%media%\",\n\"author\":\"%author%\",
	\n\"dc_date\":\"%dc_date%\",\n\"link\":\"%link%\",\n\"keywords\":\"%keywords%\",\n\"content_encoded\":\"%content_encoded%\",
	\n\"html_lang\":\"%html_lang%\"\n}\n",
	"items_footer":"",
	"footer":"]\n"
	},
	"tags":{
	"pubdate":{
	"default":""
	},
	"title":{
	"default":""
	},
	"media":{
	"default":""
	},
	"author":{
	"default":""
	},
	"dc_date":{
	"default":""
	},
	"link":{
	"default":""
	},
	"keywords":{
	"default":""
	},
	"content_encoded":{
	"default":""
	},
	"html_lang":{

	"default":"" } }, "priority":100, "mandatory":1, "is_filled":0 }], "select":"first_nonempty" } For the TEMPLATE scraping tags names of response format definitions macro: %TAG_NAME% %TAG_NAME%_extractor %TAG_NAME%_xpath %crawler_time% %scraper_time% %errors_mask% Optional template's fields: "condition" - on the same level as template's fields like: "priority", "mandatory", "state", etc... The value of this new field: {"type":type_value, "pattern":"pattern_value", "field":"URL_object_field_name"} type_value values: 0 - field from URL object with name in the field "field" match with the "pattern" field value used as regular expression pattern_value: the RE pattern expression URL_object_field_name: Any URL's field supported, numeric values casting to str() before usage in regular expression comparison. For example: {"type":0, "pattern":"*.mysite\.com*.", "field":"url"} NOT IMPLEMENTED MUST TO BE INCLUDED IN THIS DOCUMENT: https://jira.ioix.com.ua/browse/DC-1469
URL_CHAIN	Definition of the key-value pairs of the URL chains patterns. For each pair the detection of the URL by the crawler collect URL process and download the page content performed in the same crawling iteration and after that raw content processing done the template scraping algorithm concatenates content of that tag(s) in one for initial resource. Format: {"url_pattern":"...","delimiter":"...","tags_name":["..."]} The functional sense is to provide the possibility to join all parts of some tag from several pages.
SCRAPER_DOWNL	Defines does the scraper's algorithms downloads images as a part of the document scraping if positive. If omitted or is not positive (default) images not downloaded and only

OAD_IMAGES	inline images can be detected.
SCRAPER_TEXT_REDUCER	Optionally the reducing of the \n, \n\r and \t sequences. The site property named "SCRAPER_TEXT_REDUCER" in format of the list of patterns to reduce: ["\n", "\r\n", "\t"] this default set need to be used. Need to be inserted in to the level: PROCESSOR_PROPERTIES = {"algorithm":{"algorithm_name":"user_name_algorithm"},"modules":{"user_name_algorithm": ["ScrapyExtractor","GooseExtractor","NewspaperExtractor"]},"SCRAPER_DOWNLOAD_IMAGES":"1", " SCRAPER_TEXT_REDUCER ":["\n", "\r\n", "\t"]}
SCRAPER_LANG_DETECT	Sets optionally the language detection for set of the tags, for example: PROCESSOR_PROPERTIES ={"SCRAPER_LANG_DETECT":{"prefix":"lang_","suffix":"_lang","tags":["title"]}} If the “tags” field value is “*” or [“*”] all tags in result will be language detected. If the “tags” field value is “&” or [“&”] summary lang for all tags in result will be language detected. Example: PROCESSOR_PROPERTIES ={"SCRAPER_LANG_DETECT":{"prefix":"lang_","tags":["content_encoded"`]}} Also support mapping languages and truncate text buffer by size use options “map” and “size” Example: PROCESSOR_PROPERTIES = {"SCRAPER_LANG_DETECT":{"suffix":"_lang", "tags":["content_encoded"], "maps":{"en":["fr", "es", "*"], "ja":["ja-123", "zh", "za"], "ru":["ru", "uk"], "pl":["pl"], "de":["de"]}, "size":100}}
SCRAPER_TAG_ITEMS_DELIMITER	The value is string used as the delimiter for tag value items list. By default the comma “,” if not defined.
SCRAPER_TAG_ITEMS_INNER_DELIMITER	The value is string used as the sub items (inner) delimiter for tag value items list. By default the comma “,” if not defined.
FETCH_RAW_CONTENT	If this property is defined and value is positive, the response for real-time API request will contains a raw content.
TEMPLATE_SOURCE	Defines a custom source for the template in format: [{"name":"template_name", "source":"source_name", "request":"request_string", "post":"post_buffer", "schedule":{"type":0, "at":"", "step":0, "file":"/tmp/template_source.dat"}}, ...] "template_name" - template name string "source_name" and "request": file - local file, the "request" field value is file name; http - http request, the "request" field value is HTTP URL used to make request, possible with the post_buffer value; "schedule"."type":

	0 - if this URL is root, parentMD5=""; 1 - each URL; 2 - once if NOW() > value of "at" date string in ISO "Y-m-d H:i" format; 3 - periodic, the same as 2, but recalculate the value by add the "step" in minutes; (need to discuss) "schedule"."file" - the file used to store some values like timestamps, iterations, etc... It is optional. For example: "TEMPLATE_SOURCE": "[{\"name\":\"ttA\", \"source\":\"http\", \"request\":\"http:\/\/fields-extractor.snatz.com\/extract_template_by_url\", \"post\":\"{\\\"url\\\": \\\"http://www.overstock.com/Jewelry-Watches/Mens-Jewelry/2356/cat.html?TID=TN:JW:MJewelry\\\"}\", \"schedule\":{\"type\":0, \"at\":\"\", \"step\":0}}]" "TEMPLATE_SOURCE": "[{\"name\":\"ttA\", \"source\":\"file", \"request\":\"..\/localfile.json\", \"post\":\"\", \"schedule\":{\"type\":0, \"at\":\"\", \"step\":0}}]"
SCRAPER_TAGS_VALIDATION Not implemented	Defines a behavior and some special settings for scraping modules: Structure: {"SCRAPING_MODULE_NAME":{"TAG":"TAG_NAME", "METHOD":"METHOD_NAME", "OPTIONS":{<OPTIONS_DICT>}}, …} SCRAPING_MODULE_NAME – “Scrapy”, “Goose”, “Newspaper” or * for any. TAG_NAME – name of a tag to validate, for example “media” METHOD – validation method, for example: “HTTP” OPTIONS – options dict. for a concrete method, for example: {"HTTP_REQUEST":"HEAD", "CONTENT_TYPES":["img/jpeg","img/gif","img/png"],"ACTION":""} ACTION – an action performed if validation faults (optional). Possible value: "TRUNCATE" (default) – delete tag or mark as not scrapped; "EMPTY" – make a tag value empty; "DEFAULT" – make a value from a key with a "DEFAULT" value. Example of a settings for any scraper of a NEWS scraping to validate a media tag value by the HTTP HEAD request and to delete tag definition if not valid content type in HTTP response headers returned: {"*":{"TAG":"media", "METHOD":"HTTP", "OPTIONS":{"HTTP_REQUEST":"HEAD", "CONTENT_TYPES":["img/jpeg","img/gif","img/png"]}}}
URL_NORMALIZE_MASK	Defines options bit mask for normalization and canonicalization of URLs for accumulation process. Value is bit set with bits meanings: 0 – not used normalization 1 – skip 'www' prefics 2 – use validator

	4 – apply main algorithm of normalization By default if is not set or empty value is 4
URL_NORMALIZE_MASK_PROCESSOR	Similar to 'URL_NORMALIZE_MASK' but apply only for processor.
URL_NORMALIZE	Url normalize execution. Struture: {"mask": <mask_value>,"replace":[{ <re_pattern>:<repl_value>}]} <mask_value> - normalization mask integer value. This value accord to 'URL_NORMALIZE_MASK' value and replace it if was set. <re_pattern> - regular expression pattern for search in url string. <repl_value> - string value for replace in url string if pattern has entry. "mask" and "replace" are optional options. Sample filled property: {"mask":0,"replace":[{"\?ref=rss":""}]} Sample input url: http://www.asahi.com/articles/ASK95352MK95OHGB001.html?ref=rss Sample output url: http://www.asahi.com/articles/ASK95352MK95OHGB001.html
EXTRACTOR_USER_AGENT	Set custom HTTP header "User-Agent" string for the extractors modules.
EXTRACTOR_GOOSE_MAX_EXECUTION	Max execution time limit for a Goose extractor process, default if not set defined with TIME_EXECUTION_LIMIT
EXTRACTOR_NEWSPAPER_MAX_EXECUTION	Max execution time limit for a Newspaper extractor process, default if not set defined with TIME_EXECUTION_LIMIT
EXTRACTOR_CUSTOM_MAX_EXECUTION	Max execution time limit for a Custom extractor process, default if not set defined with TIME_EXECUTION_LIMIT
SQL_EXPRESSION_FIELDS_UPDATE_PROCESSOR	The same as a SQL_EXPRESSION_FIELDS_UPDATE_CRAWLER property, but applied at the processor task module.
DEFAULT_METRIC	???

TEXT_STATS	???
SOCIAL_RATE	Used for social network processing by dynamic fetcher with macro execution. https://twitter.com/search?q=%QUERY_STRING%&src=typd###window.IFRAME_SFIELD='source_url';'; Target url: <query_string>###window.IFRAME_SFIELD=<query_field>;' <query_string> - query string from site property <query_field> - query field from site property request file name read from config , support macro %PID% and %SUFFIX%. If is empty, default combined: %SUFFIX%+%PID%+'.txt', %PID% - current process ID %SUFFIX% - suffix for temporary file name (set from config file) Property has view: {"social_list": <social_list>, "timeout": <timeout_value>, "debug": <debug_value>, "sentiment": <sentiment_value>, "unique": <unique_value>, "lang":<lang_value> , "retries":<retries_value>, "retries_delay":<retries_delay_value>, "retries_type":<retries_type_value>,"interval":<interval_value>, "retries_use_proxy":<retries_use_proxy_value>, "user_proxy":<user_proxy_value>} <social_list> - social networks list (dictionary). Mandatory parameter. <timeout_value> - timeout value wait result from macro execution. <debug_value> - work in debug mode and output full list social metrics tags if value more than 0. (now, it use only for output format response). <sentiment_value> - sentiment mode value. Support values: 0 - accumulated posts per article, it's default behavior; 1 - posts one by one <unique_value> - flag of unique source field for macro execution (check unique – if value more than 0) <lang_value> - language name as string for replace macro %LANG% in parameter 'cmd' from config file. If not set will be use 'en' (English) as default value. Options for support the retrying for social module dynamic data fetching: <retries_value> - number of retries dynamic data fetching <retries_delay_value> - delay in seconds before retries

	<retries_type_value - bit mask, bit #1 - fetcher renderer timeout case identified by substring "Timed out receiving message from renderer" <interval_value> - increment of interval timeout value <retries_use_proxy_value> - list of indexes retries use proxy settings. Sample of 'retries_use_proxy': [0,1,2,3] - indexes of tries started from 0 (zero), if you will use [1,2,3] – first try must be without proxy. In case use [] (empty list) – proxy will be use for all tries. <user_proxy_value> - json content property similarly to site property USER_PROXY Parameter <social_list> has structure: {"<social_network_name>":[<parameter1>, <parameter2>, <parameter3>, <parameter4>]} <parameter1> - start page as string <parameter2> - query string <parameter3> - macro code as dictionary <parameter4> - additional parameters as dictionary. Now, supported: 'proxy' and 'fields'. 'proxy' must be string host:port (dev.hce-project.com:3129) 'fields' must be list of fields names. Support names: 'likes","reposts","posts","sentiment","subLikes","pw',"authors" Sample of apply 'social_list': "social_list":{"fb":["https://www.facebook.com","window.IFRAME_QUERY_URL=\"https://www.facebook.com/search/top/?q=%25QUERY_STRING%25\";window.IFRAME_CSCROLL_COUNT=100;window.IFRAME_MAX_TIME=350;window.IFRAME_SFIELD='title';",{"name":"tests","sets":[{"name":"set1","items":["1","%MACRO_DATA%","http://127.0.0.1/social.js","!5:76:return window.IFRAME_DATA_READY;","return window.MACRO_COLLECT;"],"repeat":1,"delay":0}],"result_type":0,"result_content_type":"text/json"},{"fields":["likes","reposts","posts","sentim

	ent","subLikes","pw"]}],"tw":["https://www.twitter.com","window.IFRAME_QUERY_URL=\"https://twitter.com/search?f=tweets&vertical=default&q=%25QUERY_STRING%25&src=typd\",window.IFRAME_CSCROLL_COUNT=100;window.IFRAME_MAX_TIME=350;window.IFRAME_SFIELD='source_url';",{"name":"tests","sets":[{"name":"set1","items":["1","%MACRO_DATA%","http://127.0.0.1/social.js","!5:76:return window.IFRAME_DATA_READY;","return window.MACRO_COLLECT;"],"repeat":1,"delay":0}],"result_type":0,"result_content_type":"text/json"},{"fields":["likes","reposts","posts","sentiment","subLikes","pw"]}]} OR "social_list":{ "fb":["https://www.facebook.com","window.IFRAME_QUERY_URL=\"https://www.facebook.com/search/top/?q=%25QUERY_STRING%25\";window.IFRAME_CSCROLL_COUNT=100;window.IFRAME_MAX_TIME=350;window.IFRAME_SFIELD='title';",{"name":"tests", "sets":[{"name":"set1", "items":["1", "%MACRO_DATA%", "http://127.0.0.1/social.js", "!5:76:return window.IFRAME_DATA_READY;", "return window.MACRO_COLLECT;"], "repeat":1, "delay":0}], "result_type":0, "result_content_type":"text/json"}, {“proxy”:“dev.hce-project.com:3129”,“fields”:[“posts”,”likes”,”reposts”]}],"tw":["https://www.twitter.com","window.IFRAME_QUERY_URL=\"https://twitter.com/search?f=tweets&vertical=default&q=%25QUERY_STRING%25&src=typd\",window.IFRAME_CSCROLL_COUNT=100;window.IFRAME_MAX_TIME=350;window.IFRAME_SFIELD='source_url';",{"name":"tests", "sets":[{"name":"set1", "items":["1", "%MACRO_DATA%", "http://127.0.0.1/social.js", "!5:76:return window.IFRAME_DATA_READY;", "return window.MACRO_COLLECT;"], "repeat":1, "delay":0}], "result_type":0, "result_content_type":"text/json"}]} Macro %KEYWORDS_FILE% replaces in macro body to <request_file> use rule from config file (%SUFFIX% + %PID%) Macro %KEYWORDS_URL% make full url use virtual host and replaces in macro body to <request_file> use rule from config file (%SUFFIX% + %PID%) %KEYWORDS_FILE% and %KEYWORDS_URL% use if in config set macro_data_type=0 Macro %MACRO_DATA% replaces in macro body initialize macro set (use only if in config set macro_data_type=1) Sample property for Demo form: "SOCIAL_RATE": "{ \"timeout\":60, \"social_list\":{\"tw\":[\"https:\/\/twitter.com\/search?f=tweets&vertical=default&q=%25QUERY_STRING%25&src=typd\",\"window.IFRAME_SFIELD='source_url';\",{\"name\":\"tests\", \"sets\":[{\"name\":\"set1\", \"items\":[\"1\", \"file:///%KEYWORDS_FILE%\",

	\"http:\/\/127.0.0.1\/social.js\", \"50\", \"return window.MACRO_COLLECT;\"], \"repeat\":1, \"delay\":0}], \"result_type\":0, \"result_content_type\":\"text\/json\"}]} }", OR "SOCIAL_RATE": "{\" debug\":1, \" sentiment\":1, \"timeout\":60, \"social_list\":{\"tw\":[\"https:\/\/twitter.com\/search?f=tweets&vertical=default&q=%25QUERY_STRING%25&src=typd\",\"window.IFRAME_SFIELD='source_url';\",{\"name\":\"tests\", \"sets\":[{\"name\":\"set1\", \"items\":[\"1\", \"%KEYWORDS_URL%\", \"http:\/\/127.0.0.1\/social.js\", \"50\", \"return window.MACRO_COLLECT;\"], \"repeat\":1, \"delay\":0}], \"result_type\":0, \"result_content_type\":\"text\/json\"}],\"fb\":[\"https:\/\/www.facebook.com\/search?f=tweets&vertical=default&q=%25QUERY_STRING%25&src=typd\",\"window.IFRAME_SFIELD='source_url';\",{\"name\":\"tests\", \"sets\":[{\"name\":\"set1\", \"items\":[\"1\", \"%KEYWORDS_URL%\", \"http:\/\/127.0.0.1\/social.js\", \"50\", \"return window.MACRO_COLLECT;\"], \"repeat\":1, \"delay\":0}], \"result_type\":0, \"result_content_type\":\"text\/json\"}]} }", OR "SOCIAL_RATE": "{\"lang\":\"en\", \"sentiment\":1, \"debug\":1, \"timeout\":400, \"social_list\":{\"tw\":[\"https:\/\/twitter.com\/search?f=tweets&vertical=default&q=%25QUERY_STRING%25&src=typd\",\"window.IFRAME_CSCROLL_COUNT=100;window.IFRAME_MAX_TIME=350;window.IFRAME_SFIELD='source_url';\",{\"name\":\"tests\", \"sets\":[{\"name\":\"set1\", \"items\":[\"1\", \"%MACRO_DATA%\", \"http:\/\/127.0.0.1\/social.js\", \"!5:76:return window.IFRAME_DATA_READY;\", \"return window.MACRO_COLLECT;\"], \"repeat\":1, \"delay\":0}], \"result_type\":0, \"result_content_type\":\"text\/json\"}]} }", OR "SOCIAL_RATE": "{\"lang\":\"en\", \"sentiment\":1, \"debug\":1, \"timeout\":400, \"social_list\":{\"fb\":[\"https:\/\/www.facebook.com\",\"window.IFRAME_QUERY_URL=\\\"https:\/\/www.facebook.com\/search\/top\/?q=%25QUERY_STRING%25\\\";window.IFRAME_CSCROLL_COUNT=10;window.IFRAME_MAX_TIME=190;window.IFRAME_SFIELD='title';\",{\"name\":\"tests\", \"sets\":[{\"name\":\"set1\", \"items\":[\"1\", \"%MACRO_DATA%\", \"http:\/\/127.0.0.1\/social.js\", \"!5:76:return window.IFRAME_DATA_READY;\", \"return window.MACRO_COLLECT;\"], \"repeat\":1, \"delay\":0}], \"result_type\":0, \"result_content_type\":\"text\/json\"}]} }", OR "SOCIAL_RATE": "{\"retries\":3, \"retries_delay\":5, \"retries_type\":1, \"interval\":10,\"retries_use_proxy\":[1,2,3],\"lang\":\"en\", \"sentiment\":1, \"debug\":1, \"timeout\":400, \"social_list\":{\"fb\":[\"https:\/\/www.facebook.com\",\"window.IFRAME_QUERY_URL=\\\"https:\/\/www.facebook.com\/search\/top\/?q=%25QUERY_STRING%25\\\";window.IFRAME_CSCROLL_COUNT=100;window.IFRAME_MAX_TIME=350;window.IFRAME_SFIELD='title';\",{\"name\":\"tests\", \"sets\":[{\"name\":\"set1\", \"items\":[\"1\", \"%MACRO_DATA%\", \"http:\/\/127.0.0.1\/social.js\", \"!5:76:return

	window.IFRAME_DATA_READY;\", \"return window.MACRO_COLLECT;\"], \"repeat\":1, \"delay\":0}], \"result_type\":0, \"result_content_type\":\"text\/json\"}],\"tw\":[\"https:\/\/www.twitter.com\",\"window.IFRAME_QUERY_URL=\\\"https:\/\/twitter.com\/search?f=tweets&vertical=default&q=%25QUERY_STRING%25&src=typd\\\",window.IFRAME_CSCROLL_COUNT=100;window.IFRAME_MAX_TIME=350;window.IFRAME_SFIELD='source_url';\",{\"name\":\"tests\", \"sets\":[{\"name\":\"set1\", \"items\":[\"1\", \"%MACRO_DATA%\", \"http:\/\/127.0.0.1\/social.js\", \"!5:76:return window.IFRAME_DATA_READY;\", \"return window.MACRO_COLLECT;\"], \"repeat\":1, \"delay\":0}], \"result_type\":0, \"result_content_type\":\"text\/json\"}]} }", Note: Used only on post-processing stage.
LINK_RESOLVE	{"method":{"retrip.jp/external-link":"GET"}} “method” - dictionary of methods will be apply accord to regular expression rule (key – pattern, value – method name) Note: Used only on post-processing stage.

Service inside

Name	Description
RECRAWL_PERIOD_MODE	0 – no auto tune during the re-crawl period; 1 – auto tune during re-crawl period based on number of URLs in NEW (1) status.
RECRAWL_PERIOD_MIN	Min possible value of RecrawlPeriod for auto tune procedure.
RECRAWL_PERIOD_MAX	Max possible value of RecrawlPeriod for auto tune procedure
RECRAWL_PERIOD_STEP	Step value for increment or decrement during the auto tune procedure.
MODES_FLAG	Defines bit flags for modes of tasks where is site will be selected. Bits: 0 – crawling (1D) 1 – processing (2D) 2 – purging (not implemented) (4D) 3 – re-crawling (8D) 4 – aging (16D) If bit is OFF – site will not be selected for that mode of tasks. Default value if is not set treated as (1+2+4+8+16) (i.e. the site will be selected for all tasks). Zero value will disable all modes of tasks.
AGING_URL_TTL	TTL of URL resource in the system, used in condition for delete resource automates way. If not set or NULL – default value from AgingBatchURLCriterion configuration variable used defined in dc-daemon.ini. Minutes, default is one day or 1440.
AGING_URL_CRITERION	String used as alternate criterion for provided for URLs selection for the URLAge object instance. The default value if this property not defined for site is defined in the dc-daemon.ini by the AgingBatchURLsCriterion option.
RECRAWL_DELETE	If dc-daemon.ini option DefaultRecrawDeleteOld is positive and for this site is positive – enables delete all resources crawled before re-crawling.
RECRAWL_DELETE_WHERE	The criterion to select resources to delete from previous crawling while re-crawl procedure.

PROCESS_WHERE_URLS	The SQL query condition to select URLs from the `dc_urls`.`urls_SITE_ID` table to process them in the process batch. The %SITE_ID% value macros supported. The default value defined by the BatchDefaultWhereURLs parameter of the dc-daemon.ini.
PROCESS_WHERE_SITES	The same as PROCESS_WHERE_URLS, but for the site select SQL query. The default value defined by the BatchDefaultWhereSites parameter of the dc-daemon.ini.
PDATE_SOURCES_MASK	It is a bits set. Each bit corresponds to sequentially usage of a source to set the PDate value to update it in the DB. Bits: 0 – Extracted from URL name (not implemented) 1 - URL object the "pdate" field (supposed was got from the RSS feed). 2 - URL object "Date" field (supposed was got from the web server's HTTP response header). 3 - URL object "lastModified" field (supposed was got from the web server's HTTP response header). – URL object "lastModified2" field applied on scraper (supposed was got from the web server's HTTP response header). (not implemented) 4 - Normalization procedure after the scraping, supposes the tag dc_date for the NEWS or TEMPLATE scraping. 5 - Normalization procedure after the scraping, supposes the "pubdate" tag for the NEWS or TEMPLATE scraping. 6 - Current date (SQL NOW()). Apply only if extracted pubdate NULL. 7 – Custom SQL expression defined in the property PDATE_SOURCES_EXPRESSION The priority corresponds with the bit order reversed, less bit signs highest priority. If the value defined with hi priority source low priority sources will not use. If the property is not defined the mask value is all bit are set - 255D by default. If the value 0D – don't apply any steps.
PDATE_SOURCES_EXPRESSION	The SQL expression involved in case of the bit 7 is set for the PDATE_SOURCES_MASK value.
PDATE_SOURCES_MASK_OVERWRITE	Bit set mask for PDate value update. Use with mask PDATE_SOURCES_MASK. Bit value ON - means that value from source need to overwrite current value only in case of current value is not set before (None/Null). Bit value Off - means that value must be overwritten in any case. If the property is not defined the mask value is all bit are set - 192D by default.
PDATE_TIMEZONES	Defines a json to configure time zones translation with URL pattern of regular expression. Format of json: [{"pattern": ".*bbc.com\/dir1\/.*", "zone_to":"+2", "condition": 0}, ...] pattern - regular expression, if matched with the URL string, definition used; zone_to - string defines zone to transform to. zone_from - string defines zone to transform from. Optional, if omitted - default local used. condition - defines additional condition for transformation action: 0 - always perform transformation; 1 - only if source date has timezone defined;

	2 - only if source date has no timezone defined; 3 - only if the source date timezone equal with zone_from value; 4 - only if the source date timezone is not equal with zone_from value; For 'zone_from' allowed to special: "[{\"pattern\": \".*\", \"zone_to\":\"+2\", \"zone_from\":\"LOCAL\", \"condition\": 0}]"
PDATE_TIME	Defines a json to configure time values overwrite with URL pattern of regular expression and SQL expression to evaluate time value. Format of json: [{"pattern": "url_re_expression", "value":"sql_expression"}, ...] "re_expression" – a regular expression for a URL match check. "sql_expression" – a SQL expression that calculates a formatted time value string or timestamp un UNIX epoch format. Expression can to resolve all URL object's fields based macro like a %URL%, %CDATE%, %UDATE%, %SIZE% and so on. Example: [{"pattern": ".*bbc.com\/dir1\/.*", "value":"IF(TIME(%PDATE%)='00:00:00', TIME(NOW()), TIME(%PDATE%))"}, ...] To set a time value from now if it is not detected as a part of a PDate value and leave untouched in opposite case. * A time overwrite will be applied after time detection and all time zones and another PDate transformations and translations as a last step PDate value calculation.
PDATE_DAY_MONTH_ORDER	Defines a json to configure a day and month date's compounds with URL pattern of regular expression. Format of json: [{"pattern": "RE_PATTERN_VALUE", "order": order_value} , ...] pattern - regular expression, if matched with the URL string, definition used; order - number, 0 - means day follows month, 1 - means month follows day. Example: [{"pattern": ".*bbc.com.*", "order": 0}] Means that any PDate detected from a "bbc.com" domain will be treated and fixed as day follows month sequence.
MEDIA_LIMITS	For improve the media (image) detection post validation for any scraper to limit media format characteristics and remove if not matched. Any limit field is optional.

	Format:
	{
	"img": {
	"Content-Type": <content_types_list>,
	"min_width": <min_width_value>,
	"min_height": <min_height_value>,
	"max_width": <max_width_value>,
	"max_height": <max_height_value>,
	"min_ratio": <min_ratio_value>,
	"max_ratio": <max_ratio_value>,
	"min_colors": <min_colors_value>
	}
	}
	"img" – name of dictionary of media limits for 'image' type media
	Variables used for set image limits:
	<content_types_list> - list names of allowed Content-Types,
	<min_width_value> - min allowed width value as numeric,
	<min_height_value> - min allowed height value as numeric,
	<max_width_value> - max allowed width value as numeric,
	<max_height_value> - max allowed height value as numeric,
	<min_ratio_value> - min allowed ratio value as numeric with floating point,
	<max_ratio_value> - max allowed ratio value as numeric with floating point,
	<min_colors_value> - min allowed colors count value as numeric
	
	Example:
	{"img":{"Content-Type":["jpeg","gif","png"], "min_width":30, "min_height":30, "max_width":3000, "max_height":3000, "min_ratio":0.25, "max_ratio":0.75, "min_colors":32}}

The DB-task module

Name	Description
STATS_FREQ_ENABLED	Accumulating frequencies statistics for site. 0 – means disabled and no data will be accumulated. 1 – means enabled.
STATS_LOG_ENABLED	Accumulating log statistics for site. 0 – means disabled and no data will be accumulated. 1 – means enabled.
COUNTER_CRIT_RESOURCES	Criterion for SQL query to calculate counter of sites.Resources
COUNTER_CRIT_CONTENTS	Criterion for SQL query to calculate counter of sites.Contents
COUNTER_CRIT_CLURLS	Criterion for SQL query to calculate counter of sites.CollectedURLs
COUNTER_CRIT_NURLS	Criterion for SQL query to calculate counter of sites.NewURLS
COUNTER_CRIT_DURLS	Criterion for SQL query to calculate counter of sites.DeletedURLs

COUNTER_CRIT_CRURLS	Criterion for SQL query to calculate counter of sites.CrawledURLs
COUNTER_CRIT_PURLS	Criterion for SQL query to calculate counter of sites.ProcessedURLs

Temporarily not documented properties

Name	Possible description
SCRAPING_TYPE_NAME	"Manage properties" tab in project view. Check the button "Set scrapping options". This is fast way to check the *Scraping options* for the demo form