# Rego Data Processor Enhancements

**Install Instructions**

1. Xog in the Data Processor objects
   1. 01\_DataProcessorObjects.xml
   2. 02\_DataProcessorProcess.xml
2. Update GEL Scripts
   1. Copy the contents of Data Processor GEL.xml to retain the formatting
   2. Copy the contents of the Connection Manager GEL.xml to retain the formatting
   3. Validate and activate both processes
3. Install RegoTools.jar file
   1. For environments where documents are stored on the file system
      1. Create a folder in the Knowledge Store named “Data Processor”
      2. Upload the RegoTools.jar file to the folder
   2. For environments where documents are stored in the database
      1. Place the RegoTools.jar file on a shared network location. All servers in the cluster must have access to this location
      2. Update the “dataProcessorJarFilePath” gel variable in the Data Processor process to point to the full path of the jar file

**Future Enhancements**

* Connection Manager
  + Validate Clarity connection url against the properties file
* Notifications
  + Send individual emails instead of one email to all recipients
  + Modify the sender of an email
* Http Action
  + Specify connection status that represent success and failures
  + Specific a timeout for a connection and a response

**Built in Parameters**

* The following are built in parameters available in any run of the data process
* initiatorID – The internal ID from the cmn\_sec\_users table of the user that initiated the Data Processor.
* initiatorUserName – The user name of the user that initiated the Data Processor.
* initiatorEmailAddress – The email address of the user that initiated the Data Processor.
* gel\_processInstanceId – The ID of the Clarity process (BPM\_RUN\_PROCESSES.ID)
* gel\_objectInstanceId – The ID of the object that the data processor was initiated from.
* dataProcessID – The internal ID of the Data Process instance that is running
* dataActionID – The internal ID of the Data Action that is running. This value changes to the ID of each action as it begins to execute.
* clarityVersion – The version of the Clarity instance
* applicationURL – The URL of the Clarity instance. This is pulled from the properties.xml file Scheduler URL. If the Scheduler URL is not present it will take the URL from the first app service that it finds.
* schedulerUrl – The value of the schedulerURL from the properties.xml file
* entryUrl – The value of the entryUrl from the properties.xml file. Equates to the entry/sslentry URL from the service named “app”
* webServerInstanceUrls – <webServerInstanceID>EntryUrl
* excelInstanceCode – The external id of the instance from the Excel Data Manager object. Only available if the Data Processor was initiated via the Excel Data Manager.
* dataProcessorCode – The external id of the data process
* dataProcessorName – The name of the data process
* trailer\_row\_text – The text contained in the last row of a flat file if the “Last Row Is Trailer” option is selected
* jobRunID – The CMN\_SCH\_JOB\_RUNS.ID for the job that launched the process. If the job was not launched via a job the value will be -1.
* httpResponse – Populated with the string representation of any http response payload

**Version 5.1.3 – 9/27/23**

* Populated a new variable called httpResponse with the string representation of any http response payload. The “Set Parameter” action now includes the “Content Type” field to allow XML or JSON responses to be parsed to set a variable. If the Content Type is JSON or XML the “SQL Text” field can contain a list of variables and their expression. Key/Value separator is “~=~” and variable separator is “~|~”. They key will be the variable name and the value can contain a JSON path or XML xpath expression. If the Content Type field is not XML or JSON it will assume the value is a database query. This feature will be helpful with Rest where we need to get the ID of a record that was just interested (ie: timesheet interfaces)
* Added the custom object upgrade feature to the Apply Fixes method.
* Fixed bug with set parameter action where the prepared statement would get closed after the first execution
* Added support for nested REST imports

**Version 5.1.2 – 7/16/23**

* Fixed bugs with Fixed Width file imports
* Removed logging process functionality which was never fully implemented. Removed because the inactive attribute on the Data Processor object was not xog’ing out and causing an error when the data processor is run.

**Version 5.1.1 – 6/22/23**

* Added a Request Configuration “setCookieSpec” to web service calls to prevent the “Set-Cookie” warning in the bg log for AWS environments.

**Version 5.1.0 – 6/22/23**

* Built functionality for Clarity to xog in the Data Processor objects when the below queries fail. This would indicate that the jar file and the objects are out of sync. This will help minimize issues in GCP environments after upgrades are automatically pushed. If the queries fail the objects will be xoged in, and the query will be retried
  + Queries included:
    - DataProcessor: dataProcessSelect
    - DataProcess: dataParamsSelect
    - DataActionList: dataActionsSelect
    - DataStudioObject: dataFieldsSelect
* Added ability to persist the HTTP Response Code
* Removed thrown exception when a HTTP call does not return a payload.

**Version 5.0.4 – 6/20/23**

* Added support for Fixed Width file imports. This can be utilized with a File Import step by setting the delimiter to ~FW. Character index strings must be entered as the Field/File Position (ie: 0-11 or 11-21). Start Index is inclusive and End Index is exclusive.
* Removed Exception when Rest Import Paging is set to None and Total Number of records is not captured from the REST call.
* Added functionality to xog in data processor objects if the initialization queries cannot be executed. This functionality is only enabled to work in GCP environments where the RegoTools.jar has been uploaded to the customlib directory. The goal is to reduce exceptions when Broadcom auto-deploys updates to the jar file to existing customers.
* Removed Excel import validation. When the “Specific Tab” option is selected and no tab name is selected it will use the first tab in the workbook.

**Version 5.0.3 – 05/17/2023**

* Removed Data Connection user name requirement on a “No Auth” Connection
* Fixed bug to allow an Import via Rest to process an object or an array
* Small gel script updates to allow BC deployed jars to work

**Version 5.0.2 – 05/17/2023**

* OAuth modifications
* Updated logic when process is cancelled. Instead of making sure the process is still running it now kills the process instance if the instance is Aborted or Aborting. That way it will continue to run if there are result exceptions.
* Changed separators for OAuth connection so they can be parsed properly if there are equal signs in the parameter values.
  + Line Separator: ~|~
  + Key Value Separator: ~=~

**Version 5.0.1 – 05/12/2023**

* Rewrote underlying WS (Web Service Package)
* Added support for OAUTH http connections
  + Will not support interactive flows of any kind
  + Only support Client Credentials, Password Credentials, and Refresh Tokens
* Created new paging mechanism for “Import Data Rest”
  + Added variable “\_pageNum” to keep the page number
  + Variable “restIndex” – either the page number or start index (pageNum \* pageSize)
  + Added “Rest Paging Type” field
    - **None** – Works the same as the former “Disable Paging” option. Only one Rest call will be executed and the import will end
    - **Limit-Offset** – Works the same as paging has always worked. Requires and Offset field, Initial Offset, Page Field, and Page Size. A total record count will be captured from an Attribute, or Header based on the Rest Total Count field. Page indexes will be dynamically determined to pull the entire result set
    - **Pages** – Simply iterates through pages. This is used when an API returns the total number of pages and we simply have to iterate through each page. The Rest Offset field should be set the URL query parameter for the page number. The total number of pages must be captured by the Total Count fields.
    - **Next URL** – Use when the API returns a “next” field that contains the URL of the next page. The Rest Offset field should contain where the “next” attribute can be found.
  + Removed “Disable Paging” option
  + Created a limit to end the paging loop after 250 iterations
  + Added logic to override the max if a parameter is present named “restImportMaxPages”
* All web service actions (Import Rest, Xog, Http, Rest Local, Invoke Process, Http Bulk) now allow connection details to be access as parameters, only within the individual step using variables (\_httpHost, \_httpUser, \_httpPassword, \_httpBasicString, \_httpSessionID, \_httpToken, \_httpAuthUrl, \_httpClientID, \_httpClientSecret, \_httpClientAuthString, \_httpClientScope, \_httpRefreshToken)
* Combined Import Data, Import Data Attachment, Excel Data Import and Excel Attachment Import actions/sub-pages. This allows for easily switching between different types of imports, and eliminates redundant subpages which decreased the size of the xog files
  + The eliminated actions will be combined automatically into an Import Data File action
  + Files with a file extension of .xls or .xlsx will be imported as excel files
  + Excel imports can import bogus values into the File Name Prefix and File delimited fields as they are required. The values will be ignored by the import.
  + Attachment files will be processed if the File Prefix Option is “Attachment”
  + Deleted the corresponding sub-pages
  + Excel imports do not support encryption
* Combined Extract Data Flat File and Extract Data Excel steps
  + The actions will be combined automatically into an Extract Data Flat File action
  + Any extracts with a file extension of .xls or .xlsx will be produced as excel extracts
  + Excel extracts do not support encryption

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Parameter** | **None** | **Limit-Offset** | **Pages** | **Next URL** | | Offset Field | N/A | ((Page Number - 1) \* Page Size) | Page Number | Field name for the next URL | | Initial Offset | N/A | N/A | N/A | N/A | | Page Field | N/A | Field name for the page size | Field name for the page size | Field name for the page size | | Page Size | N/A | The page size | The page size | The page size | | Total Count Source | N/A | Location of the total # of records | Location of the total # of pages | Location of the total # of pages | | Total Count Field | N/A | Field name for total # of records | Field name for total # of pages | Field name for total # of pages | |  |  |  |  |

**Version 4.5.7 – 3/28/23**

* In response to json encoding introduced in 4.5.5 now require any fully formatted json variables to have a variable name ending in \_JSON}, just like xml does. This is used when building the json payload via a database query.

**Version 4.5.6 – 2/22/23**

* Expanded parsing of HTTP Response to include arrays. Original design assumed that an object would be returned. Allowed arrays to be returned from as an HTTP response.
* Fixed bug that improperly put the http response as the request when the response could not be parsed properly.
* Updated excel import to ignore any numeric formatting

**Version 4.5.5 – 2/22/23**

* Added method to json encode variable values that contain reserved characters (ie: double quotes, carriage return, etc)
* Added support for additional MSSQL data types in Data Action Query LONGVARCHAR and LONGNVARCHAR

**Version 4.5.4 – 2/16/23**

* Fixed bug for Import Data from Attachment/Excel Attachment. Bug introduced in 4.5.2 when expanding the import steps to be able to run within a loop. An overridden method was no longer being overridden leading to a null pointer exception when trying to find the attachment file.
* Added additional check for Notification steps that will replace comma separators with semi-colon separators.

**Version 4.5.3 – 2/11/23**

* Removed the requirement for an HTTP Patch, Post, or Put to not have a payload. Some Quickbase and Jira calls use Post to return data and do not require a payload

**Version 4.5.2 – 2/2/23**

* Added kill functionality to a REST API Import to accommodate incorrectly configured paging variables. If the page size was greater than the API could accommodate it would create an infinite loop by never reaching the number of records to equal the total result set.
* Added the ability to nest Import Actions under a data provider. This feature should be used very rarely. As a result I did not configure the sub-pages to include the parent action. It will have to be set from the list page instead.
* Created a new action type called Import Data – LDAP to allow importing data from Active Directory. The feature will be disabled by default as it will be rarely used. The sub-page will not be included in the default object xog, it will have to be configured manually. The action will operate the same as other import steps, and will use the properties.xml configuration for authentication and search features. Any attributes with multiple values will be combined into a separated list of values using ~|~ as the delimiter.
  + Search logic
    - If the ldapFilterOverride is not null it will be used as the query filter
    - The objectClass and groupDN will be added to the filter with an AND condition
    - Multiple groups can be used by separating them with a ~|~ delimiter. Multiple groups will use the OR condition
  + Hidden parameters will allow for overriding the properties.xml parameters
    - ldapUrlOverride – override the /directoryServer/url
    - ldapUserOverride – override the /directoryServer/adminUser
    - ldapPasswordOverride – override the /directoryServer/adminCredential
    - ldapRootContextOverride – override the /directoryServer/rootContext
    - ldapObjectClassOverride – override the /directoryServer/objectClass
    - ldapSearchFilterOverride – override the /directoryServer/searchFilter
    - ldapGroupDnOverride – override the /directoryServer/groupDN
    - ldapFilterOverride – Used to pass a custom ldap query. Overrides the entire Search Logic above
  + Log Level 2 will display the LDAP connection parameters but not the ldap password. If authentication errors are experienced a global parameter can be created called “displayLdapPassword” with the value of “unableToAuthToLdap” which will them output the password to the process console.

**Version 4.5.1 – 12/19/2022**

* Fixed bug with SFTP action where the connection was being closed after SFTP action. Removed the close method to allow the sftp connection to close only after all data actions are completed.
* Added logic to limit file selection criteria to all files in a directory when the “\*” is used. All files must now have the same file extension.

**Version 4.5.0 – 12/15/2022**

* Included new feature for database monitoring. The Data Processor and Data Action will now be set to the V$SESSION.CLIENT\_INFO and PG\_STAT\_ACTIVITY.APPLICATION\_NAME in the format “Data Processor: <data process external id>. Data Action <data action external id >”. This will allow for database monitoring and query killing for poorly performing queries. This behavior can be overridden by setting a gel variable named “disableDbMonitor” to “true”.
* Added variable replacement for File Path and File Name fields on the following actions
  + Encryption
  + Excel Extract
  + Extract File
  + Import Excel
  + Import File
  + Notification
  + SFTP
* Fixed bug in Excel Import to accommodate null values in a header column
* Introduced new method to get string value from an Excel field for an excel import
* Cleaned up SFTP action to use the same methods as super classes
* Modified HttpConn class to support GET and POST of documents. May include to the Data Processor in a future release if needed.

**Version 4.4.13 – 11/22/22**

* Fixed bug with Rest Import where paging is disabled. Null vales in the Rest Offset and Rest Page fields were throwing a Null Pointer Exception. Bug fix requiring not adding the Rest Offset and Rest Page fields to the variable map when paging is disabled. Also added a not-null check replace variable method to ensure the key is not null.

**Version 4.4.12 – 11/18/22**

* Account for environments where the schedulerURL is not populated in the properties file. Error was being thrown for file import steps as the DP was not able to get a xog session to create the Record Status lookup. Modified the HttpConn class to grab the applicationEntryURL if the schedulerURL is blank.

**Version 4.4.11 – 11/17/22**

* Added support for Tab delimited data extracts using “\t” as the delimiter.

**Version 4.4.10 – 11/07/22**

* Added support to cascade security permissions from the parent folder when creating a flat file extract and saving it to a Knowledge Store folder. This feature is not available for MSSQL due to lack of database sequences. Also added a cleanup method to remove records from CLB\_DMS\_FILES, CLB\_DMS\_VERSIONS, CLB\_DMS\_FILE\_STORE, and CMN\_SEC\_ASSGND\_OBJ\_PERM when an existing file is overwritten.
* Fixed bug in GEL script to issue proper error when docs are stored in the db and a full file path is not provided. If docs are stored in the db the jar files must be placed on the application server / sftp site.

**Version 4.4.9 – 11/01/22**

* Modified AES decryption to support BT and Expedia

**Version 4.4.8 – 10/26/22 – Base release for all AWS 16.0.3 Clients**

* Modified code to allow API Alias to be retained when staging table fields are modified by the integration
* Removed start condition from the Connection Manager Masking process. The process will not fire each time a connection instance is updated. This will make the process easier to update as only the gel script will need to change. The process is not used that much where this will be an issue.

**Version 4.4.7 – 10/19/22**

* Created a new field called “Number of Retries Allowed” to all web service actions to allow for HTTP retries. Field accepts an integer (max allowed is 5 retries). If the web service call fails, it will automatically try again up to the “Number of Retries Allowed”. If the max retries are reached the web service call will result in a failure and the error will be returned.
* Added variable replacement for File Name for data extracts and notification attachments. Allows for an extract to be generated with a specific file name, and then emailed as an attachment in a later step of the process.
* Internal enhancements to accommodate the new Rego Execute a Process job

**Version 4.4.6 – 9/22/22**

* Added vendor specific queries to get the job parameters to accommodate the change in the CMN\_SCH\_JOB\_RUN\_PARAMS.VALUE column to a CLOB.
* Added additional logging for jdbc field type mappings in db result sets
* Added additional variables to support varying on-prem configurations (schedulerUrl, entryUrl, webServerInstanceUrls). The variables can be smuggled into the Host field in the connection manager if you need to establish a connection against a specific app service.
* Replaced xslt method to accommodate CDATA tags

**Version 4.4.5 – 8/31/22**

* Added logic such that API Alias is retained for the Record Status field.
* Reduced the number of records from 500 to 50 for the Data Processor to check if the process instance is still running.

**Version 4.4.4 – 7/28/22**

* Added basic functionality for AES Decryption
  + Added a new Connection Type - AES Secret Key
  + Modified lookup: REGO\_ENCRYPTION\_KEY
* Added support for the new Rego Execute a Process job. This will allow for better concurrency handling as we will pass the job run ID to the process. The Job ID, Job Run ID, and Job Definition ID will be captured by the Data Processor
* Fixed bug in gel script when class is already loaded to initialize gel script

**Version 4.4.3 – 5/25/22**

* Reworked JumpToActions because controlled exceptions were not honoring the functionality
* Added a couple utility classes for Class Loading and getting GELVariables
* Made SFTPConn to an interface to support data migrations

**Version 4.4.2 – 4/29/22**

* Added the environment URL notifications
* Created a new feature solely to solve an issue with inbound resource leave integrations. We are not able to reset resource calendar dates to the base (ie: revoke a previous PTO date) via xog. The best we can do is set the date to a working day, but this requires us to set shifts, and leaves the date as a calendar exception. If that particular date is updated at a later date on the base calendar it will not cascade to the resource’s calendar. The new feature will reset the calendar to the base for a date range. This is a hidden feature that is expected to be used only in unique circumstances. This requires the PRVALUE, SQLCURVE, ODF\_CALENDAR, AND ODF\_NK\_CALENDAR to receive updates. These need to be accounted for in the Curve Update query. Parameters will be created with CalendarCurveField()+ "\_ODF and CalendarCurveField()+ "\_SQL. Also note, logic must be added to prevent this step from being executed if the resource is assigned to the base calendar. To prevent this it’s always a good idea to process reset days before applying new PTO days.

**Version 4.4.1 – 4/21/22**

* Added the ability to persist the HTTP Verb and HTTP Resource just like the Record Status, Record Details, Request and Response. Available on the Execute HTTP, Execute Rest – Local, and Execute HTTP - Bulk actions.
* Added an attribute on a Set Parameter action called “Parameter Scope”. The behavior of a Set Parameter action was to add a Global Parameter. This remains as the default. If “Result Set” is selected it will add the parameter to the result set instead. This will be used for REST API integrations when a POST is executed and a new ID is returned that will be referenced by subsequent steps within the result set. Setting the scope to “Result Set” will ensure that the value is not varied forward to the remaining values in the result set.
* Created a new field on each data process called “Concurrency Action”. Null values or “Continue Processing” options will continue with status quo. When “Stop Process” or “Throw Error” is selected the process will determine if another instance of that particular data process is currently running. If it is the process will either not process and write a warning to the logs, or throw an error and not execute the process. Any other data processes are eligible to continue processing.
* Changed file rename method to Files.move and explicitly define and close the FileOutputStream for excel extracts
* Replaced file delete method to Files.delete
* Reduced logging level for PrintOutput method on Web Service actions

**Version 4.4.0 – 4/05/22**

* Rewrote GEL script that launches data processor.
  + Due to configuration within GCP downloading files from the database will no longer be an option
  + For SaaS environments where documents are stored in the database the RegoTools.jar and any other jar files (ie: rego-pgp-0.1.2.jar or jsch-0.1.55.jar) must be placed on the SFTP site
  + For on-prem environments it is strongly recommended to place the jar file on a shared network location that all servers in the cluster can access

**Version 4.3.2 – 4/01/22**

* Fixed bug with Strip Nulls method
* Fixed bug to not associate process instances with the data processor instance if it’s an object-based process.

**Version 4.3.1 – 3/31/22**

* Fixed bug where to allow an http status of 204 to not return content from and HTTP action.

**Version 4.3.0 – 3/25/22**

* Added a new process level variable called “clarityVersion” with the version of the Clarity instance.
* Modified how null values are removed from json strings. Instead of removing the node entirely I replace any empty strings with null. This gives the ability to set fields to null through the rest API. Originally, they were removed entirely to prevent numeric values that were replaced with a null value to result in an empty string which the Clarity API does not like.
* Added logic to associate process instances to the data process instance that they belong to if only one process was executed. This will allow process instances to be displayed under the Processes tab of each data process and make them easier to identify within the Processes -> Initiated tab.
* Added support for Broadcom’s SendGrid email change for GCP clients
* Added a new field called Additional Email Addresses that is available for variable replacement field for additional email addresses. Will accept a semi-colon separated list of values
* Added “File Path” to the Send Notification action. A full file path can be passed and attached to the email. If the file is not found an error will be thrown. This required removing the Required value on the File Path field, but it’s enforced by any actions that use it.
* Fixed bug for record status counts of sql queries.
* Removed the CA Portal sub-page from the Data Action object

**Version 4.2.7 – 1/26/22**

* Fixed bug with log process spawning.

**Version 4.2.6 – 1/24/22**

* Added functionality to spawn a bogus process to log process console messages to. Behavior is controlled by the hidden “Create Logging Process” attribute on the Data Process object.
* Added variable replacement for the following Import Data fields
  + File Name Prefix
  + File Archive Directory

**Version 4.2.5 – 1/20/22**

* Fixed bug when importing data into an object with a lookup field. The lookup type of the field was being removed when data was imported.
* Removed the default value and allowed for a null value to be passed in the Escape Character for CSV File Read import. Noticed that this was causing legitimate “\” characters to be removed from the file. If no value is passed the default escape char will be ‘\0’ which tells CSVReader to not escape any characters.

**Version 4.2.4 – 1/12/22**

* Added a hidden field called “Go To On Exception” that will allow the process to jump to a later action if an exception is thrown. The process will still result in an error, displaying the last exception that was encountered.
* Added Execute If to File and Excel Extracts and SQL Queries
* Added a file signature check for a ZipFile. If the first type bytes of the file don’t equal ‘PK’ then an error will be thrown.

**Version 4.2.3 – 1/6/22**

* Reworked how parameters are captured from the job that launched the process
  + All job parameters are now available as parameters.
  + Captured the CMN\_SCH\_JOB\_RUNS.ID for the job that launched the process and stored in a variable called jobRunID.
  + This will allow for additional data processor jobs to be created to pass different parameters.
  + Useful when replacing the HCL data extractor which has the ability to pass parameters from a job
  + Added an exception when more than one job instance is returned by the query
* File Data Imports
  + Added a new option called “File Last Data Row”
  + File Imports will break the import after the row number is reached
  + Useful for reading specific header rows (ie: read the first row and break the import)
* DataActionImportFile
  + Reverted to an older method of creating a BufferedReader
  + Replaced Files.newBufferedReader with InputStreamRader/FileInputStream
  + Allows for a more flexible reading of UTF-8.
  + <https://stackoverflow.com/questions/34886312/different-results-reading-file-with-files-newbufferedreader-and-constructing-r>
* Added additional debugging for NamedPreparedStatement
  + Output the specific variable names that were not populated.
  + Debug level 3 will print out each variable and its value
* Changed excel extracts to a streaming workbook to allow large extracts to be generated
* Removed the file deletion from SFTP server when an extract file is uploaded to the Knowledge Store
* Added support for UTF\_16BE and UTF\_16LE
* Removed query timeout for data extracts

**Version 4.2.2 – 11/18/2021**

* Filed bug with Flat File import. “Delete Files” option was not being respected

**Version 4.2.1 – 11/11/2021**

* Added checksum functionality for flat file extracts

**Version 4.2.0**

* Corrected a bug when data is imported with errors. Corrected such that a commit will not occur unless the import error option is “Throw Error” or “Continue Processing”
* Added “Last Row is Trailer” option for flat file imports
  + If this option is selected the data import will assume the last row of the file is a trailer record.
  + The last row will not be read into the staging table
  + The text of the last row will be put into a variable named ${trailer\_row\_text}. The variable can be parsed by later steps within the process.
  + This should prevent error importing files with trailer rows
  + The feature will only support one trailer row at the end of the file.
* Added Checksum feature flat file imports
  + When a Checksum algorithm is selected the import step will look for an additional file with the same name and a file extension of the algorithm (ex: myfile.txt. sha256)
  + It will compute the file hash and compare it to the checksum file.
  + If the checksum file is not found, or if the hashes do not match the import will throw an error and not import the data

**Version 4.1.6**

* Updated Rest Imports to allow for entire Json arrays or objects to be read into clob fields. The objects can then be manipulated via db functions.

**Version 4.1.5**

* Added a new variable called ${restIndex} that can be referenced when importing data from a REST endpoint. The restIndex is a the product of the current page number and the rest page size. This can be used when fetching data using a POST in a query based API. The variable keeps the starting record index for each page.

**Version 4.1.4**

* Added ability to specify ${internal} in a connection host and have it replaced with the Scheduler URL from the properties.xml file
* Added a disabled CustomerResourceHandler connection type to get around on-prem instances with SSO. To use this option the Connection Name must end with “- BC CRH”
* Added ability to use a “/” as the Record Path for a rest import to start with the root path

**Version 4.1.3**

* Fixed bug in Import rest action where pipe delimited parameters were not being created properly.

**Version 4.1.2**

* Fixed bug in Import rest action where the debug message was not showing the full url.
* Replaced the clarity Base64 encode with the java.util Base 64 encoder. The Clarity version was adding a line feed at the end of the encoded string which Workday would not accept.

**Version 4.1.1**

* Added support for x-www-form-urlencoded data to HTTP Action.
* Modified Connection Manager gel to pull strings from clobs for Oracle

**Version 4.1.0**

* Reworked connection manager to use OOTB encryption. This allows the connection manager to be used by assets other than the data processor. Added a step to the data processor to convert any existing encrypted values to the OOTB encryption.
* Removed fields from Configuration Manager. All fields were strings and could not accommodate full key fields.
  + rego\_access\_token –field removed. Reused rego\_token which is a large string
  + rego\_access\_token\_e – field removed. Reused rego\_token\_e which is a large string
  + rego\_api\_key – field removed. Was not used.
  + rego\_api\_key\_e – field removed. Was not used.
  + rego\_key\_e – field removed. Was not used.
  + rego\_refresh\_token – field removed. Replaced with rego\_oauth\_refresh.
  + rego\_refresh\_token\_e – field removed. Replaced with rego\_oauth\_refresh\_e.

**Version 4.0.1**

* Added support for API Tokens with Bearer auth, or Header/Query Param authentication
* Added ability to specify a HTTP Verb and a json payload for an “Import Data – REST” action. This is needed to support GraphQL imports
* Added a new option for “Rest Total Count Source” named Exhaustion. This can be used for APIs that do not support paging in Headers or Attributes. The pagination loop will end after the number of records returned by the page is less than the page size. The values of the Rest Offset field and Rest Page Field are available as replacement variables in the JSON Payload.

**Version 4.0.0**

* Rewrote Local HTTP connections to operate using same methods as other HTTP actions. No longer using the CustomerResourceHandler will avoid bugs that have been uncovered with it.
* Added support for Token Authentication on HTTP actions

**Version 3.7.8**

* Added feature to RestConnLocal connections to execute the actions as the user defined in the Connection Manager.
* Catch Exception thrown by Clarity bug around DELETE requests. Clarity expects a response and masks a NullPointerException because no response is returned
* Added the ability to pass headers and parameters to the RestConnLocal
* Depreciated the Data Action for CA Portal by adding display condition to the sub-page and deactivating the Data Action lookup option

**Version 3.7.7**

* Add feature to trim whitespace from numeric, date, or Boolean values during an import. Whitespace will still be retained for strings.
* Added a feature that will kill the process if neither a Process ID or Process Group were passed to the data processor. This feature requires an additional parameter to be added to the gel script called “jobParamKill”. When this parameter is set to “true” and neither a Process ID or Process Group were passed to the gel script an exception will be thrown and the process will end with the message “The Process ID or Process Group parameter must be found when the GEL parameter 'jobParamKill' is set”

**Version 3.7.6**

* Added support for HTTP Headers for Import Data – Rest actions

**Version 3.7.5**

* Added ability to use a wildcard “\*” for the File Name Prefix to allow for all files in a directory to be processed. The following actions are impacted:
  + Import Data – Flat File
  + Import Data – Excel
  + File Encryption
  + SFTP

**Version 3.7.4**

* Added “Execute If” functionality to Set Param, Print Output action and all import actions
* Fixed a bug variable replacement method to ensure global variables are prioritized over those from a result set. This will allow result set variables to be updated by a Set Parameter action as a result set is iterated through

**Version 3.7.3**

* Added additional debug message for Curve Updates

**Version 3.7.2**

* Fixed MSSQL bug when consuming blobs from nvarchar and varchar(max)
* Fixed MSSQL bug replacing carriage return from CHAR(10) + CHAR(13) to CHAR(10) + CHAR(13)

**Version 3.7.1**

* Changed Curve Update and Segment Select fields to Clobs instead of VARCHARs.
* Closed inputstream on xml file loaded from jar file

**Version 3.6.3**

* Bug fix to process only the first or last file for SFTP actions

**Version 3.6.2**

* Bug fix to close database connection opened by the OOTB gel console logger

**Version 3.6.1**

* Bug fix to allow for persist object id to be a string or numeric value

**Version 3.6.0**

* Added ability to load data to sub-objects. To do so a parameter named ODF\_PARENT\_ID must exist

**Version 3.5.2**

* Added “Execute If” functionality to SQL Update, Curve Update, Next Auto-Number, Next ID actions
* Added “Tab Not Found” option to excel imports to control process flow if specified tab is not present in the workbook
* Added ability to load curves by FTE
* Added a new field called SQL Query (Optional) that can be used when processing file attachments to pass a query where the file attachment should be found. The query must contain a FILE\_VERSION\_ID and FILE\_NAME An attached file will override this option.

**Version 3.5.0**

* Updated logic for the Result Status field. Added functionality to not override a critical error with a lesser status
* Modified the file archive method to use Files.move instead of file.renameTo for a more descriptive message when the file is not moved
* Update to curve update functionality

**Version 3.4.2**

* Fixed a bug in the email signature of the Data Processor email. Was hard coded to ${entryURL} instead of the application url

**Version 3.4.1**

* Added an option called “Any Exception” to the “Notify When” options. This will allow a notification to be sent on any exceptions rather than a result or process exception.
* Added additional details when a child action fails. The name of the action(s) are now included with the error message
* Allow the Print Output to cause a process failure. Helpful for notifications
* Added a new field on the Print Output action to allow for a number of records to be attached. This is helpful on a summary level query for interface results

**Version 3.4.0**

* Added a new action called Execute HTTP – Bulk. Rather than passing a json payload a sql query is provided. A json array will dynamically created from the result set which will be used as the payload
* Removed the DataProcessorConfig.xml file. It is now packaged with the jar file and will not need to be uploaded to the Knowledge Store.
* **Upgrade Steps**: Xog in objects and processes, replace Jar file

**Version 3.3.5**

* Changed the behavior of prepared statements for Postgres. The default fetch size is now accepted for Postgres to prevent ResultSet cursors from being used which do not play nicely with transactions.

**Version 3.3.4**

* Fixed a bug related to how results are summarized when the “Execute If” parameter is used
* Added support for EDM table name parameter

**Version 3.3.3**

* Added a parameter called “File Reader” to the file import actions to allow either OpenCSV or a traditional Buffered Reader to allow the end user to control the reader. The value defaults to OpenCSV
* Added a parameter called “File Escape Character” to allow an escape character to be passed to OpenCSV when parsing a file. The value defaults to “\”.

**Version 3.3.2**

* Modified GEL Script to default to store docs in the File Index directory when docs are stored in the database. This will help with GCP migrations as we will just need to set the “useFilestoreFiles” to false.
* Added a new GEL parameter called “printRegoToolKitLoad” which defaults to false. This will suppress some of the gel logging when the RegoTools are loaded.

**Version 3.3.1**

* Added additional error information (line number) when a file import throws an error
* Modified GEL script to allow for more flexible downloads when documents are stored in the database
* Fixed bug to allow for tab delimited file imports
* Modified persist action to retain contents of Record Details field and add new values to the end of existing text
* Modified DataProcessorConfig.xml file to remove a duplicate query “ksFileVersionInsert”. This would only be used for data extracts that attach the extract to the Knowledge Store
* Made two changes to the “ksFileInsert” query to put quotes around the “name” and “language” column aliases for Postgres support.

**Version 3.3.0**

* Added support for SFTP Gets and Puts
* Modified file selection method for SFTP and File Import steps to better handle files without a file extension
* Added a new action called “File Encryption”. It will allow for existing files to be encrypted and decrypted with PGP. Note: for encrypted files being consumed, or extract files being encrypted it’s best to do it in real time using the Data Import / Data Extract steps respectively.

**Version 3.2.2**

* Fixed a bug where xslt was not being applied to xogs

**Version 3.1.8**

* Modified Notification step to send emails using the default sender email address from the properties.xml file instead of clarity@regoconsulting.com.

**Version 3.1.7**

* Fixed bug to allow HTTP requests to apply an XSTL for xml-based requests
* Added the Apply XSLT checkbox and XSLT Template fields

**Version 3.1.6**

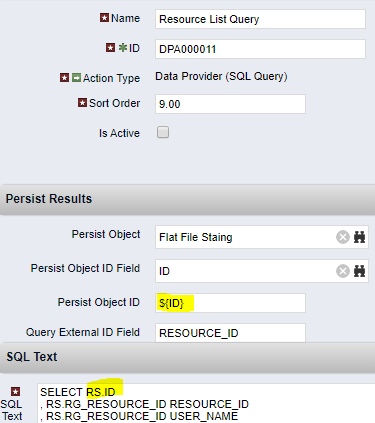
* Fixed bug to not require a payload for HTTP Get or Delete requests

**Version 3.1.5**

* Fixed bug for process invoking where a no payload was passed
* Built options within GEL parameters on where to find the Data Processor files
  + To use the files within the Knowledge Store directly set the ksOverride gel parameter to true
  + To copy the files from the Knowledge Store to the file store set the ksOverride to false.
    - This configuration offers the advantage of storing the files outside of the Knowledge Store so they can persist in the event of an environment refresh where the file store is not refreshed
    - If a new version of the files is desired the forceRefresh parameter can be used to create a fresh copy using the file from the Knowledge Store.
  + To specify a static location on the file system the dataProcessorJarFilePath, dataProcessorConfigFilePath, and the dataProcessorPgpJarFilePath can be set.

**Version 3.1.3**

* Increased flexibility for HTTP connection actions including ability to use soap and rest and xml or json.
* Added a new connection type for a HTTP connection with no authentication
* Ability to persist stand-alone HTTP results
* Modified Persist parameters on Data Provider actions to be consistent with other actions that persist results. Can now select a field to use in the persist query “Persist Object ID Field” and a variable for the ID of the object to save the record against “Persist Object ID”.



* Added a feature to the Print Output action called Break Condition. If “true” or “1” is passed as a value the process will break after the action is complete.
* Reworked GEL script to copy Data Processor files to the install directory. This provides the following advantages
  + Ability to version the files
  + Files can be retained even after environment refreshes where the filestore is not copied.
  + Jar files will not have to be downloaded each time the Data Processor runs when files are stored in the database.

**Version 3.0**

* PGP Encryption support
  + PGP decrypt import files
  + PGP encrypt extract files
* Added support for PostgreSQL
* Added support for OpenCSV in cases where the classes are found in the CLASSPATH. This will allow for more flexibility when reading flat files.
* Added a new action called Set Parameters (SQL Query). This action requires a query to be passed that returns only one record. The result set of the query will be added as process parameters.

**Version 2.1.4**

* Added support for attachment imports and checking if the file has been processed previously
* Fixed a bug related to how xog sessions are reestablished after a session timeout

**Version 2.1.3**

* Added the ability to specify column headers for file import in field mappings in place of the column number
* Deactivated all Excel based actions by default. These would need to be enabled in the Data Action Type lookup to use them.

**Version 2.1.2**

* Records imported through Data Import actions will be loaded using the Initiator ID as the Created By and Last Updated By
* Added for support of the Excel Data Manager

**Version 2.1.1**

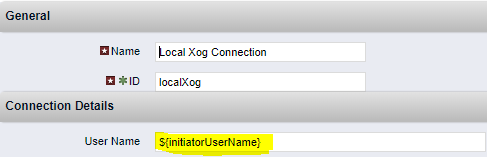
* Added a new option under file import actions called Archive Date Stamp Format. If the File Date Format is null, meaning that the file does not contain a date stamp, but an Archive Date Stamp Format is populated, and the File Archive Path is populated, a date stamp will be added to the file as it’s archived.

**Version 2.1**

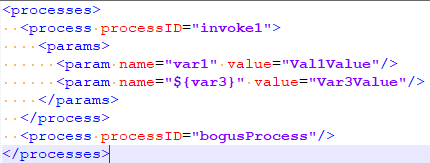
* Added an additional configuration option on Data Actions called “Email Initiator”. When checked, the user that initiated the process will be included in any notifications or emails.
* Created a new action “Update Curve” to allow writing directly to Curves (aka: blobs, aka: time scaled values). This action is disabled by default and should not be enabled without signoff.
  + The action must be nested under a Data Provider action which will supply the necessary data elements to update a blob.
  + A Curve Type must be selected as different types of curves behave differently.
  + In most cases a Calendar curve must be provided along with the curve that is being updated. Also, an availability curve may be required.
  + A sql statement must be passed instructing the data processor how to persist the curve back to the database.
  + Any loaded curve will create a variable with the field name followed by “\_SUM” (ie: ALLOCCURVE\_SUM) to hold the aggregate value of the curve. This is needed in some cases to update fields like the assignment Estimate Sum.
  + A sql statement must be passed to return the curve segments. The sql statement must include a segment start, finish, and value. Note, most curves require the finish date to be finish date +1 at midnight. The value must be passed in seconds, not hours. For allocation curves either percentages or seconds can be passed. The data processor will convert hours to percentages using the resources availability from their calendar.
  + For fiscal curves an Entity and Period Type must be passed as these are stored in fiscal curves.
* Added an option to the File Import actions to process the Last File only (in addition to the First File). The sort order is a generic alphanumeric sort based on the file name.
* Updated the action to Excel imports to use a read-only method of opening the workbook. This gets around an Apache POI bug the corrupts an excel file during the close method in Java 11.

**Version 2.0**

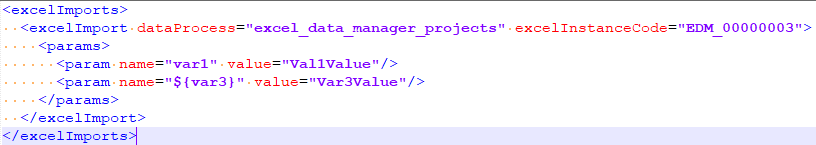
* New action (Extract Data – Flat File) for Data Extracts. A lite version of the Data Extractor with the ability to attach extract files to an existing folder within the Knowledge Store.
* New action (Extract Data – Excel) for Data Extracts. A lite version of the Data Extractor to extract file data into excel files, and an option to attach them to existing folder within the Knowledge Store.
* New action (Execute Rest – Clarity SSO) for Clarity Rest calls with an SSO/portal environment. This action uses the internal java classes to execute the Rest calls thereby circumventing the SSO auth/redirect. This action is only applicable for Clarity environments.
* New action (Get Next Auto-number) – This action will allow an object code, attribute code, and partition code (optional: defaults to NIKU.ROOT) to be passed to the next auto-number java method. It will return the next auto-number and store it in a global variable based on the “Variable Name” field. The object code, attribute code, and partition code fields can be populated via a variable.
* New action (Get Next DB ID) – This action will allow a table name, and a cache size (optional: defaults to 1) to be passed to the next id java method. It will return the next id for the table and store it in a global variable based on the “Variable Name” field. The table name and cache size fields can be populated via a variable.
* Added ability to nest SQL Updates under a parent action. This should be used sparingly, only in situations where a set-based update cannot be used. The update query can reference global parameters and/or parameters from a parent Data Provider action
* Modified parameterized lookups in the Data Action object so migration to higher environments is easier (mainly on persistence objects).
* Added a feature to allow for xml built in a sql Data Provider to be injected into an xml template. In most cases when variables are replaced they are xml encoded. However, any variable, either global or from a parent Data Provider that ends with “\_xml” will not be encoded. That will allow for xml to be built in a query, and injected into an xml template with a standard variable replacement.
* Added additional field support for CA Portal actions
  + Portal Additive App Instance List: Boolean. Defaults to “true”. Can be set to false by passing “false” or 0. This field is available for variable replacement.
  + Portal Default Password: String. If no value is passed it will not be included in the portal request. This field is available for variable replacement.
  + Portal Timezone: String: If no value is passed it will not be included in the portal request. This field is available for variable replacement.
  + Portal Send Welcome Email. Boolean. Defaults to “false”. Can be set to true by passing “true” or 1. This field is available for variable replacement.
  + Portal Force Password Reset: Boolean. Defaults to “false”. Can be set to true by passing “true” or 1. This field is available for variable replacement.
  + Portal Lockout: Boolean. Defaults to “false”. Can be set to true by passing “true” or 1. This field is available for variable replacement.
* Added three new global parameters to capture information on the process initiator
  + initiatorID – The ID from the cmn\_sec\_users table
  + initiatorUserName – The user name from the cmn\_sec\_users table. This parameter can be used in the Connection Manager User Name field to execute the Data Processor as the user that initiated it.



* + initiatorEmailAddress – The email address from the cmn\_sec\_users table
* Added support for passing in parameters (variables) when xog invoking the data processor. This required a change to the xml format for the xog invoke. Example below:



* Excel Data Manager Support
  + Added a new global parameter “excelInstanceCode” to support the Excel Data Manager
  + Added support for passing in parameters (variables) when xog invoking the data process. Example below:



**Version 1.8**

* Method to determine which version of the data processor. This will now print to the process console with each run of the processor.
* Added a new connection type for OAUTH support
  + An OAUTH connection can be setup in three different ways
    - By providing a current access token if you have one
    - By providing a current access token and a refresh token if you have one
    - By providing the authentication resource, user name, password, client ID, and client secret
  + If an access token is present the data processor will attempt to connect using the access token first. If the token is expired, it will try to use the refresh token to get a new access token
* Added password masking in the connection manager
  + The new process will execute when the access token, refresh token, client secret, or password fields change values
  + The value will be encrypted and copied to hidden fields
  + The original values will be replaced with “\*\*\*\*\*\*\*\*\*”
* Additional error handling options for data import actions
  + File Already Processed
    - Do not import – Skip File: The file data will not be imported, no errors will be thrown
    - Process Anyway: The file data will be imported, no errors will be thrown
    - Do not import – Stop Process: The file data will not be imported, no errors will be thrown, and the process will end. No subsequent data actions will be performed.
    - Throw Error: The file data will not be imported, an error will be thrown, and the process will end.
  + No Files Found
    - Continue Processing: No data will be imported, no errors will be thrown, and the process will continue.
    - Stop Process: No data will be imported, no errors will be thrown, and the process will end. No subsequent data actions will be performed.
    - Throw Error: No data will be imported, an error will be thrown, and the process will end.
  + Zero Imports (the import action resulted in zero records being imported to the staging table)
    - Continue Processing: No data will be imported, no errors will be thrown, and the process will continue.
    - Stop Process: No data will be imported, no errors will be thrown, and the process will end. No subsequent data actions will be performed.
    - Throw Error: No data will be imported, an error will be thrown, and the process will end.
  + Import Errors (data was imported but experienced errors with the data)
    - Stop Importing – Rollback: The process will end immediately after an error is encountered. Any records that have already been imported will be rolled back.
    - Import – Throw Error: The process will import all the records into the staging table and then thrown an error stopping the process. The imported records along with their detailed error messages can be viewed in the staging table.
    - Import – Continue Processing: The process will import all the records into to the staging table. An error will be thrown but the process will continue to process. The imported records along with their detailed error messages can be viewed in the staging table.

**Version 1.7**

* Added Data Parameters object to store parameters for the process
  + Parameters can be accessed in the actions in the form ${<param\_name>}
  + Parameters are supported for SQL Query or SQL Update actions (however this actions cannot be nested)
  + Every process will have the following parameters
    - ${gel\_processInstanceId} – The ID of the Clarity process (BPM\_RUN\_PROCESSES.ID)
    - ${dataProcessID} – The ID of the Data Process instance that is running
    - ${dataActionID} – The ID of the Data Action that is running
    - ${applicationURL} – The URL of the application running the process
* Added a new option (Rest Disable Paging) on Rest Data Import actions to disable paging. If this option is selected no paging logic will be executed. The data import will only make one Rest call to the source system and pull all available data. By default, this option is not selected, in which case all Rest paging parameters will be required. Note: supporting this option has resulted in the Rest paging parameters being not required on the Rest Data Import action page.
* Added a check before each step processes to ensure the Clarity process is still in a running status. If the process has been Cancelled the data processor will stop processing.

**Version 1.6**

* Renew xog session if it expires
* Notifications for individual steps
* Notification with a summary email after each process run
* New notification action type to generate emails
* Changed the Data Connection and Parent Action lookups to be external rather than internal id based
* Removed auto-number from the Connection Manager object so as not to overwrite connections when migrating
* New actions types for importing an attached flat file or excel file

**Version 1.5**

* Ability to invoke the Data Processor using xog invoke
  + If no payload is sent all active data processes will be invoked (just like running the Data Processor through the organizer)
  + An xml payload can be passed to the xog invoke to specify individual data process(es) that will be executed in order. The payload should take the following format:

<processes>

<process>processCode1</process>

<process>processCode2</process>

</processes>