CATCH – USER MANUAL

CRIF

Together to the next level

CATCH – User Manual

INDEX

1. INTRODUCTION

1.1. Purpose

The present manual outlines the functionalities and the key interactions with CATCH solution, providing both technical and conceptual information.

There are essentially two different kinds of interaction with CATCH solution: Batch and APIs. The two interactions are quite different, in terms of what they are meant to achieve and how they are carried out. For this reason are treated in two separate chapters which gives the details to implement the interactions components to CATCH and describes the different functionalities available.

The document is structured as follow:

- Chapter 2 provides a conceptual and high level overview of CATCH and of the main functionalities;

- Chapter 3 focuses on the Batch processes;

- Chapter 4 explains the online functionalities;

- Chapter 5 provides the list of information about the Entity (Bank/Group) that are needed to subscribe the service

1.2. Who is this manual for?

This manual is meant to be used by the staff of the Financial Institutes as the:

- technical staff (BATCH DEV) responsible for designing and developing the software components for the batch communication, i.e. for submitting data to CATCH, for performing the Batch enquiry and for managing the response Batch file;

- technical staff (API DEV) responsible for developing the component for the REST/JSON API calls;

- business analyst (BA) responsible for collecting the information on CATCH and for gathering the requirements needed to subscribe to CATCH and utilize its services.

1.3. How to Read the Manual

The following table lists all the CATCH available methods and provides details on the interaction that support it. Details on each interaction are provided in the specific chapters of this manual.

CATCH is a modular product that can be activated in two different modalities:

- FULL mode – where all the functionalities are available

- Categorization-only (CAT-only) mode – where the product is used just to categorize Bank's portfolio transactions and all KPIs calculation and retrieval are not enabled. This mode is out of scope for a Group (for more details about the Group entity see the 2.7 chapter), for this reason, every specific function related to this type of entity is disabled too.

Interaction	Process/Method	BATCH	Synchronous API	Asynchronous API	Available in mode
Submission	Periodical data submission (account, transactions and subject data) - LAT	✔			FULL and CAT-only
	Single transaction submission (account transaction and subjectNo) - PAT		✔		FULL and CAT-only
KPI Enquiry	Get Portfolio KPI for <PackageID> - GKP	✔		✔	FULL
	Get Portfolio KPI for <PackageID> and <FromDate> - GKU	✔		✔	FULL
	Get KPI for <PackageID> and <SubjectNo> - GKS		✔		FULL
	Request KPI for a PackageID and for a given list of Subjects - GKL	✔			FULL
Delete	Delete Subjects' Accounts and related transactions from the DB - DEA	✔			FULL and CAT-only
	Delete all transactions submitted by one or more specific LAT submission RequestId, from the DB - DER	✔			FULL and CAT-only
	Delete a list of transactions from the DB - DET	✔			FULL and CAT-only
Categorized Transactions	GetCategorizedTransactions from <StartDate> to <EndDate> or by <latRequestId> - GTD	✔		✔	FULL and CAT-only
	Get Full CategorizedTransactions from <StartDate> to <EndDate> or by <latRequestId> - GTF	✔		✔	FULL and CAT-only
Group Links	Submission of the link between the Group and the Bank subjectNo - GRL	✔			FULL
Status check	Check Sysout data of elaborations		✔		FULL and CAT-only
	Check Queue Status for Asynchronous elaborations		✔		FULL and CAT-only

1.4.   Definition of Terms

Term	Description
General	
User – Financial Institution	The terms are used interchangeably to mean the bank or financial institution that performs the data submission
Submission	The process of submitting/updating account data, transaction data and subject data to CATCH DB on a regular basis
SFTP File Transfer Protocol	Standard secured network protocol used for the file transfer.
Subject	The terms is used to define the Financial Institution customer and account holder
Account	Financial Institution account (considering savings account, credit card account, payment account)
Transaction	Action performed by an account holder to move money in or out from the account
Category	Used to define the saving/spending transaction type
Consumer	A natural person acting outside his trade, business or profession; Consumer is a Subject
Business	Company, form of corporate body or corporation; Business is a Subject
Submission file structural classification	
Record	It is the single structured data item of the submission file. Each row of the submission file is a *record* and every record in the submission file has the same structure (i.e. contains the same data fields)
Data Fields	Each record contains a set of data fields, each containing a different piece of information
Data Field Classification	
Mandatory Field	This field must always be populated every time the report is reported. Failure to do so will result in an error being generated.
Optional Field	This field should be populated if applicable.

2. OVERVIEW

CATCH is the CRIF solution intended to support Financial Institutions to unlock the potential of their own Client Portfolio data. The solution facilitates the creditworthiness evaluation and catches the relationship development opportunities throughout the usage of bank account transaction data.

CATCH analyses the accounts and related transactions data, summarizing the most relevant savings and spending behaviours of the user's customer base and calculates:

- Risk KPIs, to monitor customer credit worthiness

- Cross-selling and Up-selling KPIs, to analyse behaviours and preferences to run appropriate commercial actions

The KPIs are *standard* and *ready-to-use*, when returned by existing calculation models already tested by CRIF and therefore ready to be implemented, and *custom*, when obtained through the development and activation of customized models tailored on the specific necessities of each single institute. The models development and activation is institute-based, therefore every Institute can apply its own calculation models, independently from the others.

CATCH saves the account and the related transactions data on the proprietary database, allowing the calculation of always updated Subjects KPI.

Further details on the high-level architecture of the solution are provided in the next sections of this chapter, while the succeeding chapters outline the interaction and utilization processes of the system.

2.1.   Structure and main components

This section provides a high-level description of the solution, outlining the main components and their functions.

The main objective of CATCH is to return the KPI obtained via the elaboration of the client portfolio data provided by the Financial Institute. Because KPIs calculation leverages on the capability of the solution to categorize each single transaction, CATCH can also provide the list of categorized transactions (unique output available in CAT-only mode).

The data is contributed with a defined frequency and saved on the database dedicated to each single institute. The bank accounts data with related transactions are the minimum data set required for the KPIs calculation and each contributed record should contain the unique identifier of the subject. When KPIs calculation requires it, CATCH allows also the contribution of additional subject-related data (e.g. socio-demo data or specific indicators of the financial institute) via a personalized file, whose structure should be defined during the institute configuration phase.

Before saving the data in the DB, formal checks are executed on the data and the transactions are categorized into different "expenses categories". In fact, CATCH comes integrated with CRIF categorizer, an engine developed by CRIF which uses a complex mix of Machine Learning algorithms and experts rule to categorize account movements and furnish an "expenses category" by analysing the transaction description text and other fields like date and time of the transaction. Additionally CATCH handles the possibility to switch off the categorisation step allowing the import of external categories assigned and contributed by the Financial Institute itself or simply managing not categorized transactions.

Eventually, by enquiring the system, the institute triggers the KPIs calculation. The enquiry mode and the type of enquiry determine the KPIs which are computed and the list of subjects for which KPIs are retrieved.

The client interacts with CATCH in two distinct moments: the input to the system and consequent alimentation of the DB and the output enquiry.
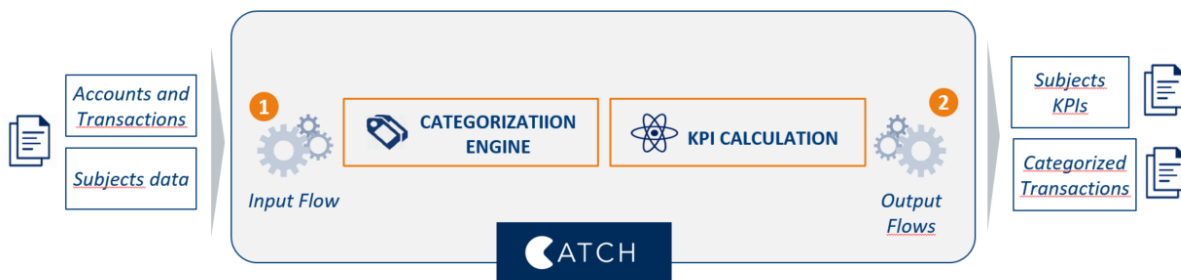
- **Data Submission (Input)**: process where data are received, elaborated, checked, categorized and saved in DB

- **Data Integration (Output)**:

  o Enquiry, elaboration and restitution of subject/account KPIs (not available in CAT-only mode)

  o Enquiry, extraction and restitution of categorized transactions

2.2. Submission Flow (Input)

The submission of the institute data to CATCH is done via BATCH channel. File containing accounts/credit cards data and relative transactions must have a definite structure which is a combination of mandatory and non-mandatory fields. The two files structure is designed to be easily manageable by the financial institutes using the solution, maintaining at the same time a certain rigid structure necessary to guarantee the consistency of data set used to elaborate KPIs. Besides account/credit card data and transactions, a third file, containing the optional subject data, can be submitted. The layout and the content of the file can be customized at institute level and they fulfil the

informative requirements needed to compute the specific KPIs required by the client institute. The periodicity with which the subject file is submitted can also be customized.
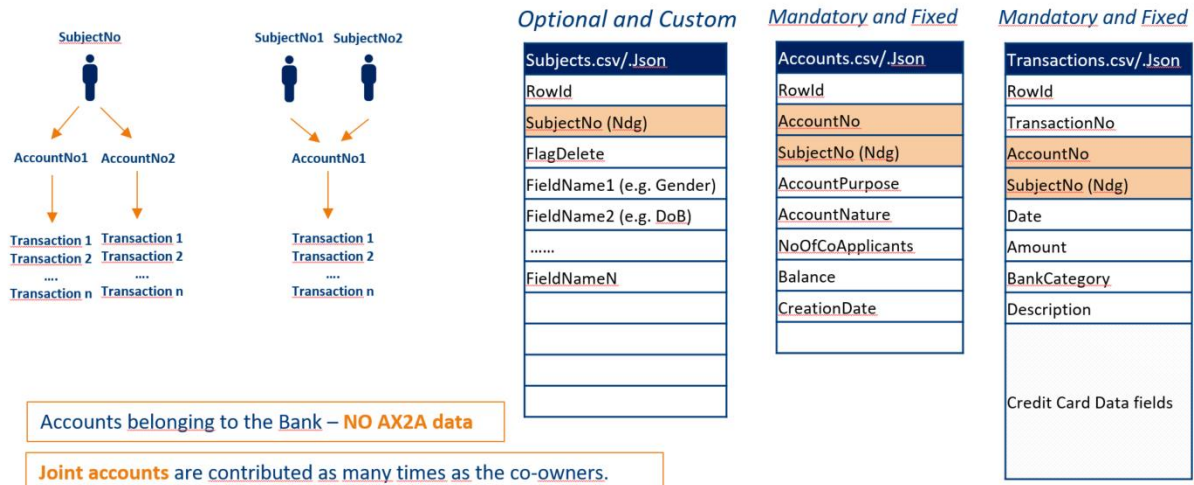
Once the files are received, CATCH executes a series of control procedures aiming at ensuring the quality and validity of the data before saving them in DB. The transaction categorization is executed during this phase so that each transaction is associated to the corresponding category of income/expense.

As a final step of the alimentation procedure, the data is saved in DB: due to the presence of unique identifiers of Subject, Account and Transaction, which represent the unique key of the entity, the data submitted are inserted or updated in DB. The outcome of the loading step and any error which may be raised during various steps of the alimentation procedure are traced and retrieved by two output files: the SysOut (summary file) and the error file.

As well as the BATCH submission procedure, CATCH exposed an API for the online submission of a single transaction. The API (**Post Transaction API**) allows for the synchronous submission of a transaction with relative account data for a single subject. Before saving the data in DB, CATCH executes the same control and categorization procedures of the BATCH submission.

2.2.1.   Categorization

Categorizing the transactions is functional to the calculation of the KPI, as it allows the calculation models to analyse the expenses in separate categories, to determine thresholds and trends, and therefore to generate the KPI.

CATCH supports four different Categorization approaches:

- **Transactions categorization using CRIF Categorization Engine**. CRIF Categorization engine is a component developed by CRIF to categorize transactions and return a category of expenditure/saving through the analysis of the text description and other meta data. The engine differentiates by consumer type (consumer

and business) and transaction type (credit or debit), ensuring a high-level accuracy in the recognition of the categories. The taxonomy of CRIF Categorization Engine has 120+ categories for the consumer segment and as much for the business segment. If the Categorization engine is not able to categorize a transaction (e.g. when the description is composed just by special characters or numbers), an undefined category (NO_DE) is given in output. The CATCH system enables the possibility of setting a specific confidence threshold under which all the categories received in output from the Categorization engine (included the NO_DE) can be replaced by an undefined category (ZZ_98 or ZZ_99 based on the amount). This feature allows the possibility to group a set of data with a low confidence under the same category, for analytics purposes. This confidence threshold parameter, called Categorizer threshold, is set at entity level during the configuration phase (see chapter 5).

- **Transactions categorization done by the Financial Institute**. With this approach the solution supports

  o **Category mapping**, module to map the categories assigned by the Financial Institute and submitted with the transaction data with the CRIF categories. This approach consents the usage of CRIF taxonomy for the KPI calculation

  o Employing **Institute Categories as they are assigned by the Institute** and submitted with the transaction data. With this approach, the calculation models should be developed using the Institute taxonomy

- **No categorization at all.** With this approach, transactions are submitted by the Institute, without any pre-assigned category, CATCH during the upload phase does not assign any category too, and the KPIs will be calculated on raw transactional data.

The categorization approach is defined during configuration phase and includes the setup of specific parameters at institute level.

2.3.   Submission rules

CATCH calculates metrics and indicators on the data that receives in input from its users in terms of Subjects portfolio and historical depth of transactional data. The CATCH user is responsible for the accuracy and reliability of the data submitted and shall verify the information transmitted according to this manual.

In order to build in CATCH a proper depth history of transactional data, the Financial Institute should

- **Address an initial data loading phase** – during this phase, for all Subjects who the Institute will want to monitor in CATCH, all the current accounts should be loaded and for each current account, all transactions

with a proper historical depth should be sent. Proper historical depth of transactions to load initially is defined by the calculation requirements of KPIs that the Institute will activate.

- **Ongoing batch submission** – periodically, the Institute need to submit in CATCH the new transactions for all the Subjects and accounts provided during Initial data loading. With ongoing batch submission, new subjects and current accounts can be added. Maintaining CATCH always updated with the last transactions enables the possibility to retrieve KPIs calculated on most updated data. Ongoing loading periodicity is based on:

    o   Requirements of the Institute, in term of KPIs periodicity update

    o   Portfolio dimension

- **Instant updating** – on top of ongoing batch submission, the Institute has the possibility to update CATCH with a new transaction when it happens, in order to enable a near real time fresh and updated KPIs calculation on the Subjects.
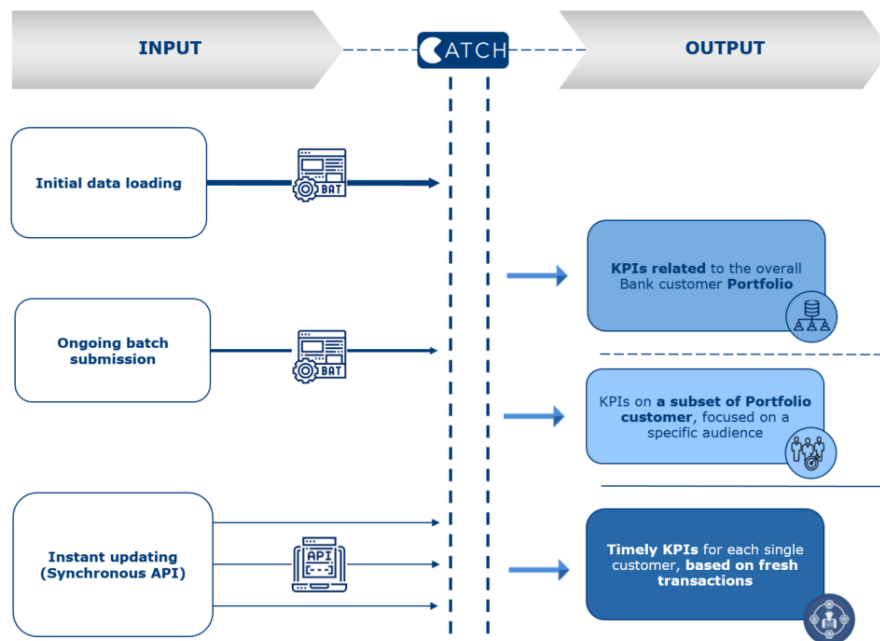
2.4.   Enquiry Flows (Output)

2.4.1.   KPIs calculation and retrieval (not available in CAT-only mode)

The enquiry procedure aims at the Subject KPIs retrieval. CATCH exposes different modalities of enquiry which allow for the execution of different requests both for different KPI packages and for different subsets of the institute Customers Portfolio. These modalities ease the usability of KPIs, making it possible for different groups of users

within the same institute, to manage the use and the retrieval of KPIs, and allows to compute KPIs always up-to-date.

The KPIs calculus models are organized in packages: one package can contain more than a model and allows the combined execution of different models interconnected. The configuration of KPIs packages is executed for each institute. The system manages both *standard* models, for KPIs that can be used by different financial institutes, and *custom* models, for KPIs developed for a specific institute.

More in details, CATCH exposes the following enquiry methods:

- the JSON/REST asynchronous API, triggers the batch process that performs the KPIs calculation and returns the file with the KPIs for each subject

- the JSON/REST synchronous API, returns the subject's KPIs directly in the API response

- the BATCH mode, calculates and returns the KPIs only for the list of subjects provided with an input file

Going through details, the list of methods exposed are the following:

- **PostPortfolioKPI for <PackageID**>, API, which triggers KPIs computation and BATCH process for the retrieval of the output file. The KPI output file contains the KPI declared in input (PackageID) for ALL the portfolio subjects contributed by the institute.

- **PostPortfolioKPI for <PackageID> and <FromDate>**, API, which triggers the KPI computation and the BATCH process for the asynchronous retrieval of KPIs. The output file contains the KPIs for the package declared in input (PackageID) for the portfolio subjects which have at least a transaction whose date equal or greater to the input date (FromDate)

- **Batch mode,** to extract KPIs of a certain package for a list of subject submitted via the BATCH input file

- **PostKPI for <PackageID> and <SubjectNo>,** synchronous API which triggers the KPIs computation and their retrieval via synchronous response. The response, given via API, contains all the KPIs of the input package (PackageID) for the input subject.

The KPI calculation occurs at the time of the enquiry and, except the synchronous API, the list of KPI is returned with/via file.

The KPI *packages* are created and assigned at institute configuration level. Further details are provided in the next sections.

2.4.2. Extraction Of Categorized Transactions

As mentioned above, CATCH saves in DB the transactions with their associated category. CATCH DB contains therefore all the transactions already categorized, and exposes the following API to extract them:

- **PostTransactions belonging to a <latRequestID> or a range <DateFrom> and <DateTo>,** API which triggers the BATCH process which allows to extract the categorized transactions submitted by a specific latRequestId or whose dates are within the date range specified in input (DateFrom- DateTo). The BATCH process retrieves in asynchronous modality the list of transactions in an output file. For each transaction, only the identification code is retrieved.

- **PostTransactions belonging to a <latRequestID> or a range <DateFrom> and <DateTo> in full mode,** API which triggers the BATCH process which allows to extract the categorized transactions submitted by a specific latRequestId or whose dates are within the date range specified in input (DateFrom- DateTo). The BATCH process retrieves in asynchronous modality the list of transactions in an output file. For each transaction, the detail of each record field stored in DB is retrieved.

2.5. Delete Process

CATCH manages the possibility to delete specific set of data submitted by the user.

This flow allows the deletion of specific data from the Institute DB, it is possible to choose between three different types of delete:

- DEA - through the submission of a file containing the list of Accounts that must be removed

- DER - through the submission of a file containing the list of LAT submission RequestId, whose transaction must be removed

- DET - through the submission of a file containing the list of Transactions that must be removed.

Each process has a specific and unique data structure and set of key that enable the identification, extraction and deletion from the Institute DB, of the corresponding information, *see CATCH – Data Structure (sheet DELACCOUNT_Input, DELTRANSACTION_Input, DELREQUESTID_Input)*.

2.6. Configuration and Web Admin Interfaces

CATCH has a modular and scalable architecture, which allows to manage the several institute in a unique installation, maintaining data and services segregated.

CATCH admin interface (managed directly by CRIF, in SaaS mode) provides a set of components to manage common functionalities as the LOGs, the Audit Manager and the Elaborations Monitoring section. Besides, it also exposes the

shared configuration components as the KPI calculation models library, the module used for the creation and maintenance of KPI models packages and finally, the Institutes configuration module.

During the onboarding process, the Financial Institute name, its unique identification code and all the relevant parameters, which define the profile of the institute, are stored in the system.   Example of these parameters are the following:

- Categorization Type (e.g. CRIF Categorization Engine, Mapping etc.)

- Definition of the *Subject Record*, used to submit additional Subject information (when applicable)

- Historical depth in terms of days Data stored

- Definition of the Packages used for the KPI calculation

2.7.   Group of Institutes (not available in CAT-only mode)

The concept of Group of Institutes is used to enquire the system, with the data (accounts/transactions/subject data) that a specific Subject has in different Banks that belong to a Group, and get the KPIs.

The KPI calculation engine receives all the data of the different banks, which belong to the same group, producing an output that should be more complete because of the enriched input. A Group of institutes is an Entity type, with a configuration that is similar to the Bank, the main characteristic is that a Group is composed by different Banks, associated with a link.

To create a Group of institutes in CATCH it is necessary to create the link between the group and the banks in the WEB Admin site. A group of institutes is made by at least 2 Banks, but a Bank can be part of just one Group; being part of a group of institutes is not mandatory into CATCH system.

Each Group, using the Group Link Submission process, inserts in the DB the link between their own SubjectNo and the Bank SubjectNo, creating a unique identifier.

The data submission (Accounts/Transactions and Subject data) to the system is in charge of each single Bank, also if they belong to a Group; some functions like the Batch Submission of Accounts/Transactions or the enquiry for getting Categorized Transactions are not available for Groups.

The functions available to a Group are the following:

1.   The API that are used to return KPI

2.   The Batch KPI enquiry process

3. The Group Link Submission

Each Bank that belongs to a group will still be able to enquire the system using the subset of data belonging to the single Institute DB.

As for a Bank, a Group needs an active subscription to make an enquiry; it is possible that a Bank and a Group have different active packages, both will be able to use only the related ones.

3. BATCH PROCESSES

This chapter provides the functional and technical details for all the Batch Interactions between the User and CATCH. The first Section (Section 3.1) gives all the technical specification of the batch process, such as the file naming convention, while the succeeding sections treat the Submission (Section 3.2), the Enquiry (Section 3.3) and the Delete (Section 3.4) process separately providing detailed information on main phases, files involved, record structure.

3.1. Technical Specification

Batch Processing needs a standard folder structure to make CATCH work properly; the Bank/Group file system folder needs to be divided in two main parts the INPUT Folder and the OUTPUT folder.

In case of Bank activated in FULL mode, the INPUT folder is structured in three subfolders DEL, GKL, LAT; where the .zip related to the specific processes should be uploaded to make the system run properly. On the other hand, a Bank activated with the CAT-only mode will be able to view just the LAT and DEL subfolders.

Each subfolder is linked one to one to a specific process, the .zip folders containing the data needed for the BATCH processes to be launched are uploaded there.

In the OUTPUT folder of the FULL mode Bank, besides the DEL/GKL/LAT subfolders there are also all the Enquiry folders such as GKP, GKU, GTD and GTF, where the output .zip folders produced by the various elaborations, are stored. The CAT-only banks will find the DEL/LAT subfolders, just like in the INPUT, in addition to the GTD and GTF ones; KPI-enquiry subfolders are not visible to these banks.

If the Entity is a Group, in the INPUT folder the subfolders would be only GKL and GRL because the data management (as for the submission or the deletion) is still a Bank responsibility. The OUTPUT folder will contain also the GKP and GKU. In case the Bank/Group needs to send the contribution to CATCH from a different machine than the machine, hosting the folders described above this can be achieved by means of a SFTP file transfer, which is established between the User and CATCH. This SFTP file transfer is secured by standard encryption technology. Dedicated FTP folders per process type are made available to the institute, where the files are uploaded and downloaded.

Once the User uploads the files, the data is retrieved and processed by CATCH. It is possible for the user to upload more than one file in the INPUT folder, but the system will always proceed with the execution of one process at a time, starting from the file with the oldest upload timestamp.

The data processing depends on the activity (e.g. Submission, Enquiry) and always implies the data validation of the supplied file records against data quality rules, in order to ensure that the data quality standards are met prior to the information being loaded onto the database or further used. At the end of file processing, the system generates errors and warnings whenever the data quality rules are not met. The errors are collected and returned in the form of error files that can be downloaded.

The list of possible Process Type Identifiers and local folder names is provided in the file *CATCH – Data Structure (sheets Processes and IOFiles)*.

3.1.1. File Naming

The naming convention adopted for CSV files and for the zipped folders that contain them is detailed in the file *CATCH – Data Structure (sheet IOFiles)*.

Regardless of the process type, the zipped folder name always contains:

- **[ENTITY ID]** → the Identifier code that uniquely identifies the Institute. The code is defined during the on boarding.

- **[FILE CREATION DATE]** → timestamp of the creation of the zipped folder. The time stamp has the format *YYYYMMDDHHmmss* and must be equal or less then current date. The following table provides the details on the date format used for [FILE CREATION DATE] naming convention:

Format specifier	Description
YYYY	Four digits year
MM	Two digits month - from 01 to 12
DD	Two digits day - from 01 to 31
HH	Two digits hour, using a 24-hour clock from 00 to 23
mm	Two digits minute, from 00 to 59
ss	Two digits second, from 00 through 59

The files exchanged with the zipped folder are always CSV pipe ("|") separated.

It is important to highlight that all the pipes ("|") present in the submission files, that are not used as separators, must be deleted before the submission, or the process will raise an error and the data will not be loaded in the DB.

The pipe must be used exclusively as a separator in the files to let the processes succeed. The number of pipes in the submission file must be invariable; this means that a pipe for every field of the *Data Structure* must be written in the file row, even if one or more non mandatory fields are not filled.

Every input file must have at least one row and no headers, otherwise an error will be raised.

3.1.2.   Errors and SysOut

Every Batch Process produces always a zipped folder as response. This folder contains process-specific files, i.e. files generated only in case of specific processes such as the Enquiry, and standard files, returned by every Batch Process.

CATCH returns two standard files:

Error File

This file lists all the errors raised during the different batch processes steps. The file collects the error across all the submitted files:

- If no errors are raised during the elaboration, the file is returned indicating *No Errors*

- Each Error is associated to a specific record

- Just in case of failure during the Pre-check step, the file is not created and the Errors are written in the SysOut

CATCH raises warnings and errors:

- **Warnings**, mark the record and return it in the Error File; yet the record remains valid and updates the CATCH DB.

- **Errors**, mark the record and return it in the Error File; the record is discarded and not processed

The file *CATCH – Data Structure* provides the Error File record layout for each Process Type, while the most significant fields are described hereafter.

SysOut File

The SysOut file contains the overall result of the batch process in terms of metrics on the records processed (e.g. number of records successfully processed, total number of generated errors, etc). The SysOut file is always returned in the batch processes, without exceptions.

The file *CATCH – Data Structure* provides the SysOut File record layout.

3.1.3. RowId

RowId is a field present in the majority of CATCH data structures. This field is a technical key used to univocally identify every record of the file. The User must fill the RowId with a key which is unique for each record in the file. CATCH does not store the RowId for checks in next submissions, so the field is considered only in the context of each submission. A very simple way to fill this field could be using the corresponding line number of each record in the file.

3.2. Submission (LAT)

The Submission is the process by which relevant data is contributed, checked and loaded in the CATCH Database, during initial data loading or during ongoing periodic update.

Once the User supplies the files, the data is retrieved and processed by CATCH. The data processing validates each record provided in the file against the data quality rules in order to ensure that the data quality standards are met prior to the information being loaded onto the database.

The following diagram shows the end to end submission process at glance:
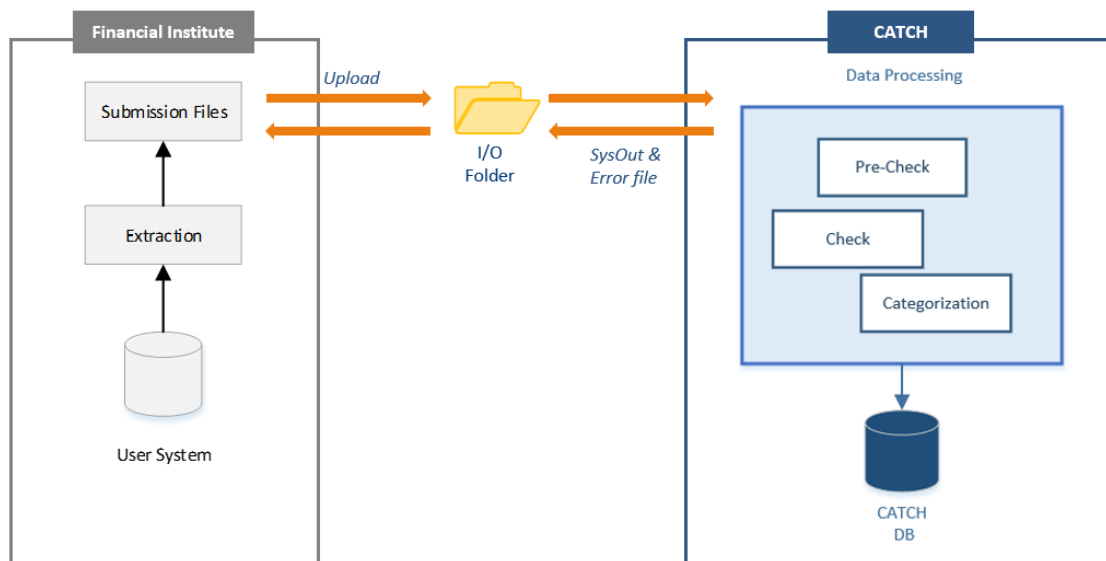
Once the Institute loads the zipped folder onto the LAT Input folder, CATCH runs a series of validation checks prior to the information being loaded onto CATCH database:

1. **Pre-check** – this stage determines whether the structure of the file is compliant with the expected file format specifications. In case one of the checks fails, the flow is stopped, nothing is loaded in CATCH DB and the SysOut File is created.

2. **Validation and Loading –** this stage performs a number of more in depth checks (i.e. formal, logical and consistency checks) for each record of the submission files. Errors in this step cause that single record affected is discarded and does not update the DB.

3. **SysOut and Error File Creation** – this stage creates two files: the SysOut file, with summary indicators, and the Error file, with the list of errors encountered during the previous stages.

The diagram below represents the high-level overview of the entire submission file processing and validation:
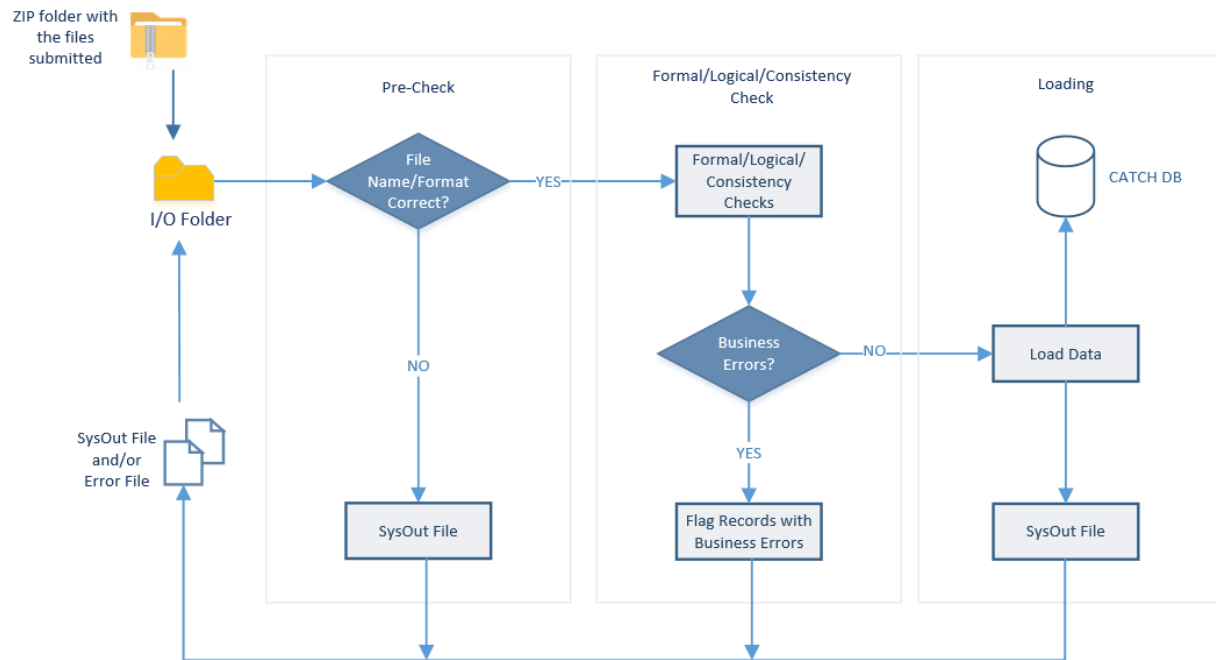
3.2.1.   Submission Files

As seen in the previous section, there are three types of files used for the data submission:

- Accounts.csv, with the institute accounts/credit cards data

- Transactions.csv, with the transactions data details of the accounts/cards contributed

- Subjects.csv, with custom fields defined during the on boarding and configuration phase of each Institute
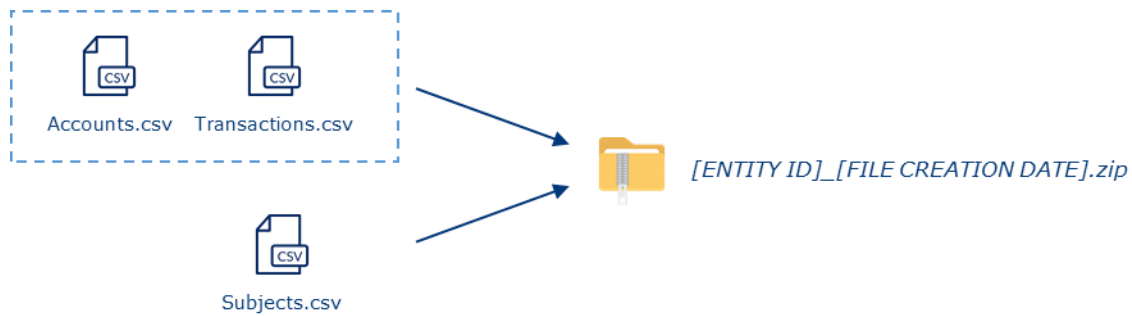
The submission of the Accounts.csv and Transactions.csv is mandatory and performed through two standard files with defined layouts.

The submission of the Subject.csv is optional and the file data layout is custom for each institute. If needed, this file must be added to the submission zip folder with the Accounts.csv and Transactions.csv. It is not possible to submit files with a different name or structure; otherwise, an error will be raised.

All the details on the files layout and data are discussed in Section 3.2.3

3.2.2. Process Specification

The Initial data loading submission is the process to feed the system, with the proper historical data depth. Depending on portfolio dimension, during the onboarding phase, will be defined the initial data loading elapsed time and if and how splitting the historical data into input submission data structure files.

The Regular Submission is the standard process periodically used to submit the data to CATCH database. The submission frequency could vary in accordance to the user requirements of and data volume and it is defined at the time of the onboarding. The more frequently the Institute needs to retrieve fresh and updated categories and KPIs on the Portfolio, the more frequently the institute should update the system with the raw data. It is recommended to users to schedule a periodic submission process for the Accounts.csv and the Trasactions.csv. When needed, submission frequency of Subjects.csv can be different and lower that submission frequency of Accounts.csv and the Trasactions.csv.

3.2.3. Input Files – Record Layout

As introduced in the previous chapter, the submission of accounts and related transactions data is mandatory and it is executed through the two files: Accounts.csv and Transactions.csv. Additional Subjects' data are submitted, prior custom configuration, with the ad hoc Subjects.csv file.

Every file contains only one record type that reflects the type of information that is being provided with that particular file. Hence, every record carries one single nature of 'atomic' set of information: account, transactions or subject information.

Defined rules must be applied when populating the record fields to be supplied. Each Record Type has a well-defined record layout as clarified later in this section.

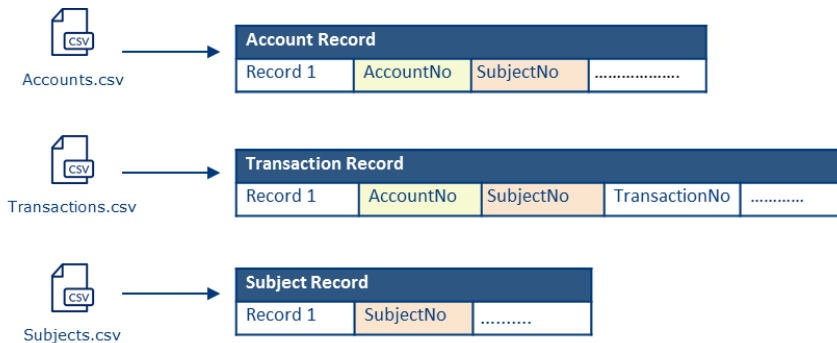### 3.2.3.1. Information on record keys and joint accounts management

The information submitted in the files (accounts, transactions and subjects) should be extracted from the Institute DB grouping them by each subject. Specifically, in case of an account that belong to more subjects (co-holders), accounts and transactions file should contain the same account and transaction information replicated for all co-holder subjects. Consequence of this is that each of the records of the accounts and transactions have keys which are composed by more fields and one of them is the SubjectNo which is the identifier of the subject.

In order to extract data in the correct way and prepare the files:

1. The Institute must identify the subjects portfolio who wants to load into CATCH. As a result, the list of accounts and related transactions should be retrieved and referred to each subject (by SubjectNo).

2. The data are then submitted using three separate files, each containing one record type with its corresponding unique key:

File Name	Record	Key	Description
Accounts.csv	Account	*AccountNo + SubjectNo*	key used by the Institute for the Account. It is mandatory, and must be unique.
Transactions.csv	Transactions	*AccountNo + TransactionNo + SubjectNo*	key used by the Institute for the Transaction. It is mandatory, and must be unique.
Subjects.csv	Subject	*SubjectNo*	key used by the Institute for the Subject. It is mandatory, and must be unique.

In particular the following schema describes the high-level structure of the submission with the three record types:

where:

SubjectNo
Unique identifier assigned by the Financial Institute to the subject. Normally the Institute has a unique customer code in its internal system, which can be used as SubjectNo. The SubjectNo allows the linkage in CATCH database of all the accounts related to the same subject

AccountNo
Unique identifier assigned by the Financial Institute to the account. Each account reported must have a unique identifier assigned. The AccountNo must be unique at Institute level, meaning that the Financial Institute must maintain this number in its own system in order to submit the updates on the account periodically. The AccountNo allows the linkage in CATCH database of all the transactions related to the same accounts

TransactionNo
Unique identifier assigned by the Financial Institute to the transaction. Each transaction reported must have a unique identifier assigned. The TransactionNo must be unique at account level, meaning that there could not be two transactions with the same TransactionNo for the same account. Nonetheless, in case of joint accounts, the same transaction (i.e. a record with the same AccountNo and TransactionNo) could refer to two or more subjects.

In case of an Account with two or more Co-applicants, the same AccountNo could occur more than once in the same LAT and the number Of Co-applicants parameter in the submission file will be filled. The Number of Co-applicant must be lower than the sum of Account records having the same AccountNo, considering both the input Records and the records already present in DB. The following schema shows the contribution of the joint Account01, which is shared between the two holders: Subject_A and Subject_B:

Accounts.csv		
Account01	Subject_A	....
Account01	Subject_B	....

Transactions.csv			
Account01	Subject_A	Transaction001	....
Account01	Subject_A	Transaction002	....
Account01	Subject_B	Transaction001	....
Account01	Subject_B	Transaction002	....

3.2.3.2. Account record

The *Accounts.csv* file contains the list of account records available to be loaded in CATCH database. The Account records contains all details of the given account or card account. The following information are highlighted:

- The order in which the different records appear does not matter as long as the structure is correct

- The records combines mandatory and optional elements, meaning that a record is considered valid only when all the mandatory fields are filled

- The account must be included in the submission, even if no change has occurred during the time interval between two periodic submission

- Closed (or expired) Accounts and Cards must not be submitted or must not be submitted any more

- If an error is found on a specific Account and the record is not loaded, all the related transactions are discarded too

This section clarifies the meaning of the most significant record fields, while the file *CATCH – Data Structure* describes the full list of Account record data fields.

AccountPurpose
Identifies the type of account holder. CATCH differentiates between: • Consumer Account, registered in the name of a private individual. Payment and account management are in the name of that person • Business Account, registered in the name of a company, with payments and account management being in that's company's name Sole Traders should fall into the Consumer category to have a better and more precise transaction categorization.

CATCH leverages on the assumption that a single subject could have several accounts with different AccountPurposes. CATCH checks the alignment between the AccountPurpose in input, during a submission, and the one already stored in the Database.

CreationDate

The creation date is the date on which the account has been made active and it is opened in the Financial Institute system. The creation date must be lower or equal to the file creation date of the submission (see paragraph 3.1.1).

AccountBalance

The Account Balance is the amount of money present in the account at the submission Reference Date. The Balance is the sum of all the entries booked to the account (debit and credit) at the point of time in which the data are extracted to perform the submission. Hence the Account Balance reflects the Amount of all the Transactions contained in the Transaction.csv file within the same submission flow

AccountNature

The Account Nature is the type of Account and can be between Credit Card or Account.

If an account is already in the DB, when the data are updated the nature must be consistent with the one that is already uploaded.

3.2.3.3. Transaction Record

The *Transactions.csv* file contains all the list of transaction records available to be loaded in CATCH database. The Transaction record contain all the details of the given transaction. The following Information are highlighted;

- The order in which the different records appear does not matter as long as the structure is correct

- The records combines mandatory and optional elements, meaning that a record is considered valid only when all the mandatory fields are filled

- The file contains always the list of transaction records that have not yet been submitted to CATCH, therefore all the transactions occurred between the last submission and the current submission. Transactions already submitted should not be sent

- Each transaction must be linked to an account of the Accounts.csv file, otherwise a Warning is raised and the specific row is discarded

This section clarifies the meaning of the most significant record fields, while the file *CATCH – Data Structure* describes the full list of Transaction record data fields.

BankCategory
The Bank Category is used when the financial institute runs its own process to assign the saving/spending category to the transaction. CATCH supports the submission of the category that has been assigned at Institute level and, when available, of the corresponding Category Confidence and Category Model.

The transaction record contains a set of optional information that should be filled just in case of Credit Cards; in particular the Card Acceptor information are about the merchant involved in the Credit Card transaction and the Merchant Category Code is the merchant category of the Card Acceptor.

When one of the credit card related data is filled, all the credit card parameters must be filled; otherwise the partial data will not be uploaded. In case the account Nature is Account and one or more card related data is loaded, all the card parameters are deleted; only the mandatory data is loaded, to maintain the data consistency.

3.2.3.4.   Subject Record

The *Subject.csv* file contains the list of subject records available to be loaded in CATCH database. The Subjects.csv is optional, meaning that CATCH users decide whether or not utilize this file to submit additional information on their subject portfolio. The *Subject.csv* is required when the CATCH user needs additional information – other than accounts and related transactions data – to be considered for Subject KPI calculation.

For this reason, the Subject record is defined at Institute level during the onboarding phase and every Financial Institute has its own Subject record in accordance to the fields that want to submit.

The following Information are highlighted:

- The order in which the different records appear does not matter as long as the structure is correct

- The records combines mandatory and optional elements, meaning that a record is considered valid only when all the mandatory fields are filled

- Even when configured, the submission of the *Subject.csv* is not mandatory and could be performed less frequently than the contribution of the Accounts.csv and Transactions.csv file

- Every submission overwrites the existing information of the corresponding Subject matched by SubjectNo, replacing the old record previously loaded. For example if a Subject.csv is submitted with the optional attribute DateOfBirth, and after a while, another Subject.csv is loaded without it, this attribute is erased from the DB.

- The Institute can delete Subject records submitted in the past setting the FlagDelete to "1". In this way Subject information will not be used for KPI calculation.

The file *CATCH – Data Structure* provides details on the subject record data fields that are in the record regardless of the specific Institute customization. Meaning and description on these fields are also provided hereafter.

FlagDelete
Flag used to determine whether the record (in case of matching with DB) should be removed or simply replaced. If the Flag is "1", the system checks the presence of an existing record with the same SubjectNo and remove it. In contrast, when the flag is "0", the system checks the presence of an existing record with the same SubjectNo in the DB and either update it, if already present, or insert it.

3.3. KPI Enquiry (GKL, GKP, GKU) (not available for CAT-only mode)

The KPIs Enquiry is the process by which the KPIs per Subjects are requested, calculated and received by the User.

The KPIs are obtained through standard or custom calculation algorithms and/or models that are activated at institute level, thus every institute can apply its own calculation algorithms. The calculation algorithms are then organized into Packages. Hence, the Package is a mandatory input parameter for any performed enquiry and should be declared at the moment of the *Request*. In this manual, terms model and algorithm are considered as synonyms.
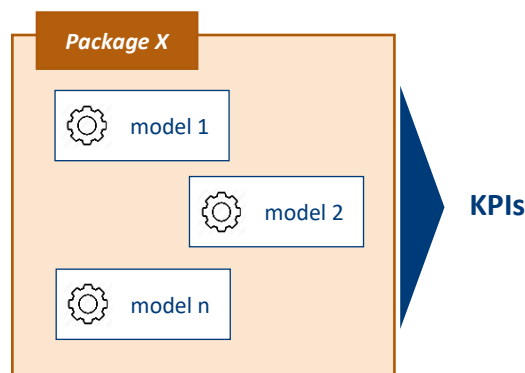

The Batch enquiry retrieves the KPIs for a Package and for a specific set of Subjects from the User's portfolio. The set of Subjects can be specified by submitting an input file with the list of Subjects identifiers (*SubjectNo*) for which the KPIs should be calculated (batch process GKL). The Batch process is the same in the three different enquiries, with one main difference; the GKL is a complete BATCH process, both   the Request and the Response are via file, while in the GKP and GKU the BATCH process is triggered by an Asynchronous API. This means that The GKL will need an Input file, whether the GKP and GKU will be started with an HTTP Request; but all the three processes will produce the response files. Regardless of the list of Subjects, the enquiry triggers a standard flow for the KPIs calculation and computes the KPIs always on the last snapshot of contributed data. KPIs are always calculated at request-time and are not saved and kept in CATCH database.

3.3.1. Process Specification

CATCH supports three different batch process (GKL, GKP, GKU) that returns the KPIs per subject. The processes differs only on the list of subjects for which the KPIs should be calculated:

- GKL → the list of subjects is provided with an input file submitted by the user;

- GKP → the list consists on the complete subjects portfolio of the user and it is triggered via Asynchronous API Enquiry (details in Section 4.3.1)

- GKU → the list consists on subjects whose account was updated after a defined date and it is triggered via Asynchronous API Enquiry (details in section Section 4.3.1)

The list of subjects identifiers (*SubjectNo*) are used as a key to retrieve the related data (accounts, transactions and subjects data when available) from the User DB and perform the KPIs calculation. The diagram below represents the high-level overview of the GKL process:
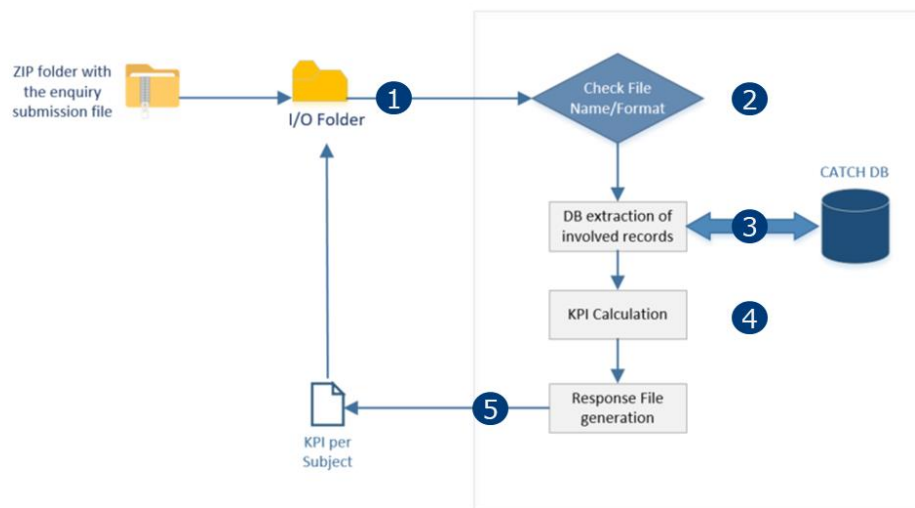
1. **Enquiry Input**: submission of the enquiry file with the list of *SubjectNo, calling one specific PackageID*. The same step might be triggered by an asynchronous API Enquiry

2. **Check Step**: checks on the input data

3. **Records DB extraction**: process to extract the required records (account, transaction and subject records) from the User database

4. **KPI Calculation**: module to calculate the KPI for the declared PackageID

5. **Response File**: process to generate the response file and trigger the batch response flow

3.3.2.   KPIs Output Files – Record Layout

CATCH produces the response file containing the KPIs per Subject and sends it back via zipped folder together with the SysOut and the Error file. For details on the structure of these files and the input file refer to document *CATCH – Data Structure (sheets GKL_KPI, KPI_Output, KPI_Error and Sysout)*

3.3.2.1.   Request File Record Layout

As introduced in the previous section, only the GKL Batch enquiry is executed through the submission of a single file, the *KPI.csv*, which contains the list of unique subject identifiers for whom the KPI should be calculated.

The file contains only one record type, with only the SubjectNo data field. The SubjectNo is the unique identifier assigned by the Financial Institute to the subject and used for the data submission.

Details on the record structure are provided in the file *CATCH – Data Structure (sheet GKL_KPI)*.

3.3.2.2.   Response File Record Layout

The *[ENTITY ID]_KPI_OUTPUT.csv* contains the list of KPI calculated per Subject. The file contains only one record type, which combines both mandatory and optional fields. The fields returned depend on the Package stated in the enquiry.

This section clarifies the meaning of the most significant record fields, while the file *CATCH – Data Structure (sheet KPI_Output)* describes the full record structure.

| SubjectNo |
| --- |
| Unique identifier of the Subject for which the KPIs are calculated. The SubjectNo is used for the Data Submission and allows the linkage in CATCH database with all the subject-related data employed for the KPI calculation |
| **AccountNo** |
| Unique identifier assigned by the Financial Institute to the account. Each account reported must have a unique identifier assigned. The AccountNo is returned only for account-based KPIs, thus for those KPIs used to provide information on the specific Subject Account. |
| **PackageID** |
| Identifier of the Package for which the KPI are calculated. The Package is declared in the enquiry request. |

3.4. Delete (DEA-DER-DET)

The Deletion is the batch process by which the Institute can remove previously submitted data from CATCH Data Base. The flow allows deleting specific data from the Institute DB, through the submission of a file containing the list of information that must be removed. There are three types of deletion; DEA- deletion of accounts, DER- deletion of RequestId related transactions, DET- deletion of a list of transactions.

The three processes diverge from each other on the type of data that are deleted, the structure and the naming convention of the INPUT file. It is possible to use a specific deletion type at a time. Once the User supplies the file, the data is retrieved and processed by CATCH, which uses different keys that depend on the specific process:

- the list of unique Subject and Account as keys to identify, extract and delete from the Institute DB the corresponding Accounts and the related set of connected transactions

- the list of unique Subject, Account and Transaction as keys to identify, extract and delete from the Institute DB the corresponding Transactions

- the list of unique RequestId related to the LAT submission to identify, extract and delete from the Institute DB all the Transaction submitted with that process.

The following diagram show the Delete Flow at glance:
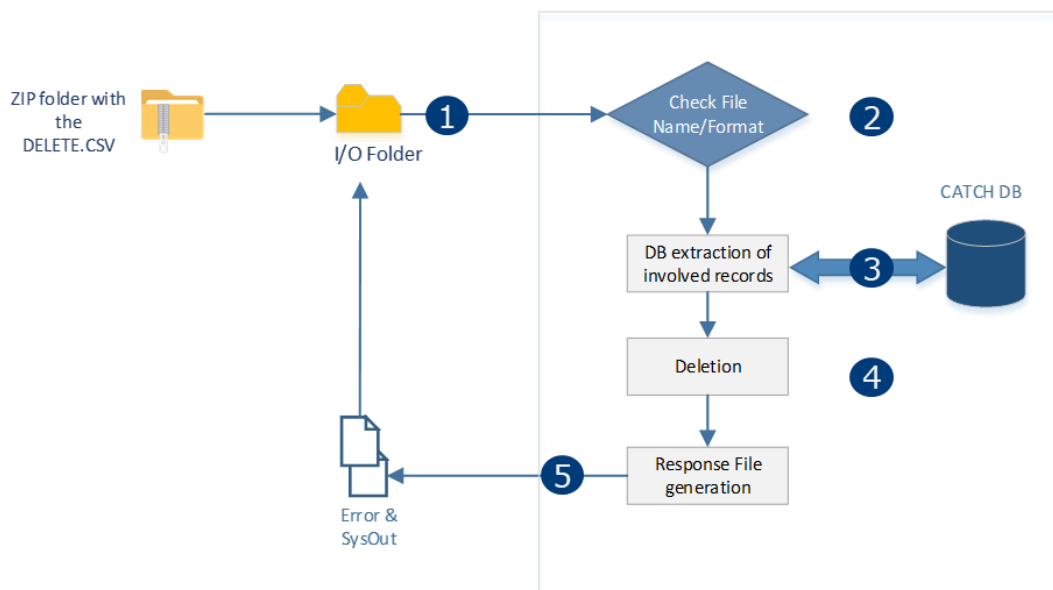
1. **Batch Submission**: submission of the file with the list of *AccountNo, TransactionNo or RequestId* to be deleted

2. **File Check**: checks on the enquiry file and record structure;

3. **Records DB extraction**: process to extract the required records (Account, Transaction or RequestId related data) from the User database

4. **Deletion**: Deletes the records from the DB

5. **Response File**: process to generate the response file and trigger the batch response flow

3.4.1. Request File Record Layout

The DELETE.CSV file contains the list of data that have to be removed, with the specific schema, depending on the deletion type.

During the DEA, CATCH deletes the Accounts and all the Transactions related to them, the DER cancels all the Transactions that were submitted with the specific RequestId listed in the .csv file, and the DET erases all the transactions in the file.

The file contains only one record type, with mandatory fields. The file record structure is provided in the file *CATCH – Data Structure (sheet DELACCOUNT_Input, DELTRANSACTION_Input, DELREQUESTID_Input)*.

3.4.2. Response File Record Layout

The three Deletion processes return only the standard SysOut and Error files *see CATCH – Data Structure (sheets DEL_Errors and Sysout)*. Details in section 3.1.2.

3.5. Categorized Transaction Retrieval (GTD, GTF)

The Request Categorized Transactions process allows to retrieve the list of transactions from the User database with the transaction category. When the Categorization option is active, CATCH assigns the category during the submission process before loading the records into the DB. The category assigned depends on the categorization approach defined at the time of the Institute Onboarding. Refer to Section 2 for details on the categorization approaches supported.

This batch process is not exposed per-se, but is triggered by the Asynchronous APIs that allows to extract the transactions; these can be retrieved using two different filters, a defined interval of dates or the request id of the LAT that submitted the data.

Details on Section 4.4; two APIs and two retrieval options are provided

- A light retrieval mode (GTD) – that provides a thin output where for each transaction only the transaction code is provided to identify the transaction

- A full option (GTF) – that provides a detailed output where for each transaction, all attributes and also some account data are provided

3.5.1.   Response File Record Layout

The [BANK ID]_TRANSACTIONS_OUTPUT.CSV and [BANK ID]_TRANSACTIONS_FULL_OUTPUT.csv contain the list of transactions with the Category and the Category Model, respectively, for option 'not full' and 'full'. The files contains only one record type, details on the record structure are provided in the file *Data Structure CATCH (sheets GTD_Categories, GTF_Categories and Sysout).*

3.6.   Group Link Submission (not available for CAT-only mode)

This is the submission needed to create a link between the SubjectNo used by the group and the one used by each Bank that belongs to it. It is important to note that not every Bank of the Group must have information about the same subject (it may occur that just some of them have the information).

Once the user supplies the file, the data is retrieved and processed by CATCH, which uses the list of unique Group SubjectID, Bank Code Id and Bank subject id as key to identify the Logical Unit.

From a high-level perspective, the following steps compose the workflow:

1. **Pre-Check** - this stage verifies that the overall structure of the file is compliant with the expected format, raising an error otherwise

2. **Delete** - the process deletes all the links that belong to the same Group. It is important to keep in mind this behaviour; each time a new file is updated, all the previous links are erased and replaced by the new ones

3. **Loading** - executes the logical, formal and input/DB checks on the LU in input, and raises errors for all the records, which are part of the LU, if something is wrong; loads in the DB the data that have successfully passed the Check, storing in temporary tables the error information about those records that failed one or more checks

4. **Error File Creation** - Error file is created ,zipped and moved to the output folder

3.6.1.   Request File Record Layout

This process needs only the LINKS.csv file to be executed. The LINKS.csv file contains the list of Subjects Identifiers used by the group, and the related one, used by each bank.

The Bank SubjectNo should be the same value sent in the SubmissionAccountsInput for each Bank of the Group that have the same subject. The file contains only one record type, with mandatory fields. The file record structure is provided in the file *CATCH – Data Structure (sheet GRL_Input)*.

3.6.2. Response File Record Layout

The Link submission process returns the standard Sysout file and the Error file, whose structure is detailed in the *CATCH-Data Structure (sheet GRL_Errors and Sysout)*.

4. ONLINE API

This chapter provides the functional and technical details for all the Online Interactions between the User and CATCH. The first Section gives all the technical reference specification of the online process and list the available API methods, which are then treated separately in the succeeding sections.

It should be highlighted that Users must be authenticated and authorized to access the exposed API and the allowed resources.

4.1. Technical Specification

CATCH exposes both Asynchronous and Synchronous APIs methods. The APIs specification are available upon request on CRIF API Developer Portal (https://developer.crif.com/). To access the Portal the User must receive a set of authentication credentials which are created during the on boarding phase. The API developer portal provides the description of the API methods exposed along with the API Swagger and all the support documents and use cases.

CATCH exposes both Asynchronous and Synchronous JSON/REST API methods:

- The online Asynchronous API generates the query used for the data extraction needed to feed the process itself. First the API returns an online status confirmation that the request is taken in charge and inserts the request in a queue. Afterwards the corresponding batch process reads the queue, extracts the required set of data and produces the CSV response files returned via Batch Channel

- Synchronous API methods, where the API response is returned directly in the online API response

4.1.1. Details on Asynchronous API flow

The two sequence diagrams that follows illustrate the concept of Asynchronous API.

The Asynchronous API add the process request to the CATCH Queue. The request contains the details of the query to extract the input data for the specific process.
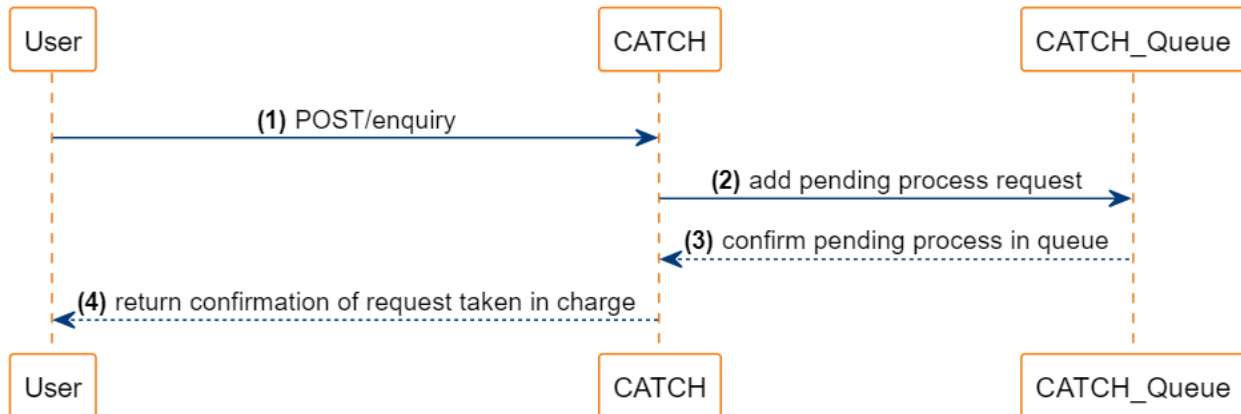
The CATCH internal scheduler launches the application that reads the request from the queue, extracts the input data needed to run the query and executes the specific flow producing the output file.
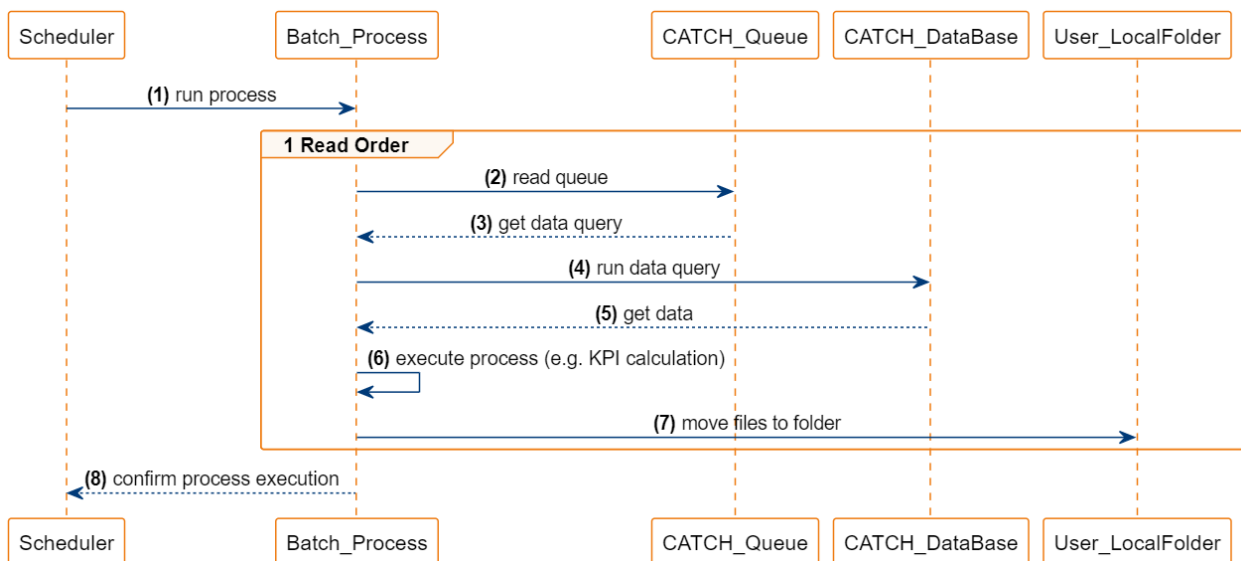
Into CATCH System, we can find the following Asynchronous processes:

- KPI Enquiry GKP - GetPortfolioKPI for <PackageID> (detailed in chapter 4.3.1) (not available in CAT-only mode)

- KPI Enquiry GKU – GetPortfolioKPI for <PackageID> and <FromDate> (detailed in chapter 4.3.1) (not available in CAT-only mode)

- Categorized transaction enquiry GTD – GetCategorizedTransactions from <StartDate> to <EndDate> or by <latRequestId> (detailed in chapter 4.4)

- Full Categorized transaction enquiry GTF - GetCategorizedTransactions from <StartDate> to <EndDate> or by <latRequestId> with detailed output (detailed in chapter 4.4)

If a process of the same Bank and of the same type is already waiting in the queue of processes, an error is raised.

In the Synchronous API flow, CATCH reads the request, gets the input data needed to run the query and executes the specific flow producing the output.

Into CATCH System, we can find the following Synchronous processes:

- Single Transaction Submission PAT (detailed in chapter 4.2)

- KPI Enquiry GKS - GetKPI for <PackageID> and <SubjectNo> (detailed in chapter 4.3.2) (not available in CAT-only mode)

- Check Sysout data of elaborations (detailed in chapter 4.5.1)

- Check Queue Status for Asynchronous elaborations (detailed in chapter 4.5.2)

4.2.   Single Transaction Submission via API

As already mentioned, a financial Institute can submit data to the Catch DB via BATCH, but if a Bank needs to submit just a single transaction, it is possible to load it with the PAT - Synchronous API flow.

This API allows to insert a single transaction following the structure depicted in the Swagger, the transaction submitted must refer to the SubjectNo indicated in the input.

The main steps that compose the Process are the following:

- Validation - This stage performs a number of formal, logical and consistency checks for the input data. These checks are performed to verify that the mandatory fields have been submitted correctly. If an error is raised the process is stopped and the entire record is discarded.

- Loading data - This stage performs the acquisition of the record, just in case no error is raised in the previous step.

The Acquisition is executed as for the Batch Submission, with the difference that in this case we have a single transaction and a single Account and not a set of them.

All the responses returned from this process follow the HTTP standard, and each time there are more than one error found, an array of them is returned.

It is important to highlight that different check steps, ordered in a priority queue, compose the validation process. This means that if an error is raised in the first set of checks and the submission is stopped, the response will contain only the errors found and it will not be possible to know if others could be raised during the following steps.

During the loading step the system checks if there is one or more pipes ("|") in the transaction description or in the transaction, account and subject Id fields, in that case this special character is automatically converted into a blank. This is necessary to avoid problematics with the data management of the batch processes, which use the pipe ("|") only as a separator.

All the details about the Request and the possible responses, for the API Submission, are available in the Swagger.

4.3.   KPIs Enquiry (not available in CAT-only mode)

As mentioned already in the Batch Processes chapter, the KPIs Enquiries are the batch processes by which the KPIs per Subjects are requested, calculated and received by the User. These processes can be triggered by Asynchronous API Enquiries described below.

KPI enquiry of a single subject is possible also using the Synchronous API Enquiry.

The KPIs are obtained through standard or custom calculation models organized into Packages. Hence, the Package is a mandatory input parameter for any performed enquiry and should be declared at the moment of the *Request*. Details on the Exposed methods are provided in the following sections.

4.3.1.   Asynchronous API Enquiry (GKP, GKU)

The Asynchronous API Enquiry retrieves KPIs for a significant number of subjects of the Users' portfolio. The output file contains the KPIs for a specific Package that is declared in the API request.

The diagram below represents the high-level overview of the entire Asynchronous API Enquiry process:
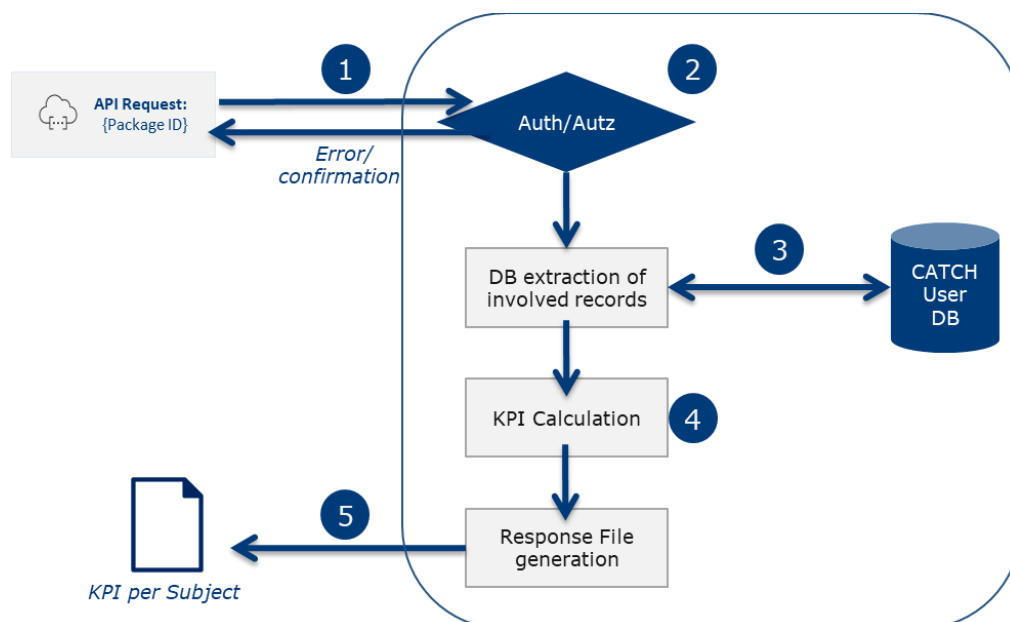
1. **API Request**: submission of the API request (with *PackageID* declaration)

2. **Authentication and Authorization**: User Identity and permissions check

3. **Records DB extraction**: process to extracts the required records (account, transaction and subject records) from the User database

4. **KPI Calculation**: module to calculate the KPI for the declared PackageID

5. **Response**: process to generate the response file and trigger the batch response flow

The list of Subjects for which the KPI calculation is performed depends on the API request submitted. Refer to Section 3.3 for details on the batch channel set up, while the enquiry process is described in Section 3.3. The response files and on their record structure corresponds to those of the Batch Enquiry, which are provided in Section 3.3.2.

The following table provides the NEOS Enquiry Asynchronous API methods exposed:

GKP - GetPortfolioKPI for <PackageID>
The method triggers the KPI calculation for the declared Package (PackageID). The KPIs are calculated for all the subjects of the User portfolio
GKU - GetPortfolioKPI for <PackageID> and <FromDate>
The method triggers the KPIs for the declared package (PackageID). The KPI are calculated for all the subjects that have at least one transaction whose transaction date (TransactionDate) is equal or subsequent to the date provided in input (FromDate).

The APIs specification for the request are available in the Swagger.

4.3.2.   Synchronous API Enquiry (GKS)

The Synchronous API Enquiry requests the KPIs for one single subject of the Users' portfolio. This API triggers the KPIs calculation for the package (*PackageID*) and subject (*SubjectNo*) declared in the request and returns them directly in the API synchronous response.
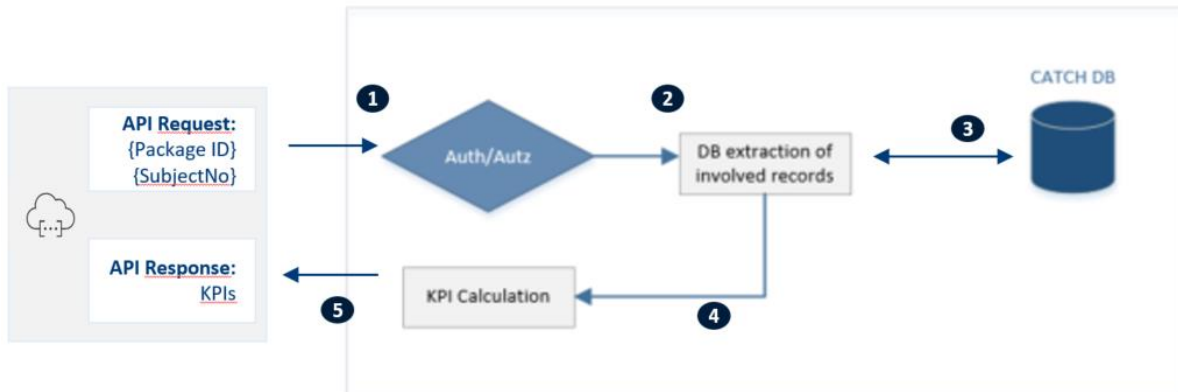
The diagram below represents the high-level overview of the entire Synchronous API Enquiry process:

1. **API Request**: submission of the API request, with *PackageID* and *SubjectNo*

2. **Authentication and Authorization**: User Identity and permissions check **Records DB extraction**: process to extract the required records for the SubjectNo (account, transaction and subject records) from the User database

3. **KPI Calculation**: module to calculate the KPI for the declared PackageID

4. **Response File**: the KPI per subject is generated and returned via API response

CATCH exposes one synchronous API method:

GKS - GetKPI for <PackageID> and <SubjectNo>
The method triggers the KPI calculation for the declared Package (*PackageID*) and Subject (SubjectNo)

The API specification are available in the Swagger that gives insight into how the API responds to parameters and options, providing all the specification for the API implementation.

4.4. Categorized Transactions Enquiry (GTD, GTF)

As mentioned, the Categorized Transactions Enquiry allows to retrieve the list of transactions from the User database with the corresponding transaction category.

CATCH exposes two Asynchronous API Enquiries which trigger the extractions, from the User DB, of the list of categorized transactions. The user can choose one between the following filters:

*   Transactions whose *Date* is between the date interval provided in the API request

    o   In In case Start Date or End Date is filled both date fields must be filled. End Date must be greater or equal than Start Date.

- Transactions submitted with a specific LAT, and whose latRequestId (the name of the submitted file) is the one in the API request.

  - In case RequestId is filled, the system checks that at least one LAT elaboration with this RequestId in DB exists and is already completed.

The two filters are alternatives; it is not possible to use both in the same request.

The Asynchronous APIs are the following:

GTD - GetCategorizedTransactions from <StartDate> to <EndDate> or by <latRequestId>
The method triggers the retrieval of the Transactions with the corresponding category whose *date* is between the <StartDate> and <EndDate> or whose <latRequestId> is the one sent via API. The list is returned via CSV file.
GTF - GetCategorizedTransactions from <StartDate> to <EndDate> or by <latRequestId>, with a detailed output
The method triggers the retrieval of the Transactions with the corresponding category whose *date* is between the <StartDate> and <EndDate> or whose <latRequestId> is the one sent via API. The list is returned via CSV file where the full detailed information about transactions is returned.

Refer to Section 3.3 for details on the batch channel set up, while the categorized transactions enquiry process is described in Section 3.5.

The API specification for the request are available in the Swagger.

4.5.   Status check enquiry

CATCH environment enables the possibility to use two different APIs to check the status of a specific elaboration and a set of data, usually shown in the Sysout csv. These APIs can be very useful to check the status, or a specific info about an elaboration submitted by the bank, without the necessity to open the output .zip folder.

The APIs are:

- POST/sysouts/execution/status – the result is a set of data related to the output of a specific elaboration

- GET/enquiries/{requestId} – the result is the status of a request in the queue

The first API can be used on every CATCH process and the output is very similar to the sysout file, while the second one checks specifically the status in the queue of elaborations, because of that, only the Asynchronous API requests can be checked.

4.5.1 Check Sysout data of elaborations

This method returns a set of data, related to the elaboration and filtered by requestId. If there is a set of different executions with the same requestId (this may happen only when an execution fails and is not set "abandon") only the data related to the last one executed, is shown by the API.

In the output, the user will find the following info:

- RequestId

- Start, End, Elapsed

- Status

- Success records

- Records With Errors

- Precheck Error data, if the elaboration failed during the precheck validation step

The system checks if the requestId is in the DB and if the request body is correctly filled, otherwise an error will be shown. All the details about the Request, Responses and the possible errors are available in the Swagger.

4.5.2 Check Queue Status for Asynchronous elaborations

This method might be called to know the status of a submitted request in the queue of executions. The API requires in input only the requestId and the output is the status in the queue.

This API checks the status of a request in the enquiry queue, so it is possible to check only on the asynchronous elaborations (GTD-GTF-GKP-GKU).

The system checks if the requestId is in the DB and if the request is correctly filled, otherwise an error will be shown.

All the details about the Request, Responses and the possible errors are available in the Swagger.

5. USER CONFIGURATION

CATCH uniquely identifies the Financial Institute through the EntityID, which is assigned during the set up phase together with all the relevant parameters and settings that determine the User interactions with CATCH.

As briefly introduced in the Overview chapter, CATCH enables a number of customized functionalities as the categorization approach, the custom subject Record and the Package for the KPI calculation. These functionalities are defined during the User assessment phase taking into consideration the User requirements.

The following table lists the parameters to be configured during the user set up that reflect the User customization defined on the assessment and the functionalities employed. It should be highlighted that certain parameters are defined only for specific user requirements.

Parameter	Description
Entity Identifier	Financial Institute unique identifier (official code that identifies the specific Financial Institute in CATCH). This identifier is used in all the interactions between the User and CATCH (e.g. submission, enquiry etc.)
Entity Type	Type of Financial Institute. CATCH could manage single Financial Institutes (Banks) and Groups of Financial Institutes (Banking Groups), allowing specific interactions to both of them
Retention Days	Period of time (expressed in number of days) for which the data submitted is kept stored in the User database of CATCH. At the end of every LAT & DEL, CATCH runs the automatic deletion of old accounts and transactions data in compliance with the Retention Days
Log Retention Days	Period of time (expressed in number of days) for which Logs are kept stored in CATCH for trouble shooting purposes
Package Time Intervals	Max historical depth of the account and transaction data applied for the KPI Calculation Models. CATCH does not utilize for the KPI calculation accounts and transactions in accordance with the Package Time Interval. Normally the Package Time Interval is less or equal than the Retention Days
Categorizer Call	Parameter used to define the categorization mode applied by the Financial Institute
Categorizer Model Source Code	Financial Institute' model unique identifier, used for the call to the CRIF categorization engine.
Mapping Table of Categories	When the transactions are categorized by applying a custom mapping between the bank-categories and the CRIF categories, the mapping table is loaded into CATCH system.
Subject Record	Definition of the record structure used to submit additional Subjects' information to be used for the KPI calculation. The number of elements, their type and length is custom for every institute.

Message Storage	Parameter used to define if the API's input and output messages should be stored in the log services table.
Default Culture	Default Culture chosen by the specific Financial Institute and inherited by all the accounts and transaction submitted in the Database.
Default Currency	Default Currency chosen by the specific Financial Institute and inherited by all the accounts and transaction submitted in the Database.
Categorizer Threshold	During the categorization phase, the system will replace every category of categorized transaction with "Confidence" < "Categorizer Threshold". This Parameter allows the reclassification of a set of categorized transactions with a low confidence; this means that these transactions will be saved with an Undefined category set by the system (ZZ_98 or ZZ_99 based on the amount sign). If this parameter is set to 0 the system will not replace any category. In the CATCH environment, this is the only way to receive in output the undefined Categorization engine category NO_DE.

In addition, during the on boarding, the Financial Institute defines the *packages* used for the KPI calculation. Packages could be made of *standard* CRIF calculation models or *custom* models developed specifically for the Institute.

APPENDIX

APPENDIX I – NEOS Suite Package (not relevant for CAT-only mode)

The NEOS Suite is the ready-to-use model specifically designed for the credit risk assessment and the evaluation of consumption habits. The Suite available for CATCH comprehends:

- NEOS VIEW score, the CRIF score based on the information obtained from the payment account movements

- NEOS LINK KPIs set, which summarizes the expenditures and saving behavior of the subject in Key Performance Indicators (KPI). The KPIs differ between the Consumer and Business Segment.

The NEOS Suite is based on the analysis of payment account transactions processed via the CRIF proprietary categorization tool.

NEOS Suite elaborates only payment accounts and related transactions, while Credit Card accounts data are not considered. This means that NEOS Suite discards all the accounts and the related transactions with the *AccountNature* = 3 (credit cards)

APPENDIX II – NEOS Wide Suite Package (not relevant for CAT-only mode)

The NEOS Wide Suite is another ready-to-use package specifically designed for the credit risk assessment. The Suite available for CATCH comprehends the NEOS Wide Score model.

NEOS Wide Package is a combination of the NEOS View score and the Credit Bureau score, this means that the Neos Wide Score is a level 2 model, because a previous calculation is needed (NEOS View). The user should be able to interrogate the Credit Bureau system, retrieve the Credit Bureau Score, submit this information in the LAT and finally call CATCH to enquire the NEOS Wide Package that will return the NEOS Wide.

It is important to underline that this is an Italian applicable model because of its structure.

To calculate the NEOS Wide Score the institute needs to submit three specific parameters in the Subject.csv file:

- The Score type

- The Credit Bureau Subject Type

- The Credit Bureau Score

CRIF

Together to the next level

CRIF is a global company specializing in credit reporting and business information systems, as well as credit solutions and outsourcing services. Established in 1988 in Bologna (Italy), CRIF has an international presence, operating over four continents (Europe, America, Africa and Asia).

Via M. Fantin, 1-3 | 40131 Bologna, Italy

Tel. +39 051 4176111

Fax +39 051 4176010

Email info@crif.com

www.crif.com