

Exceptions Management and Automated Processing

An Exception Software White Paper

Author: Exception Software Ltd.
Date: 4th April 2007
Copyright: 2007 © Exception Software Limited. All rights reserved.

This document contains Confidential, Proprietary and Trade Secret Information (“Confidential Information”) of Exception Software Ltd. and may not be copied, distributed, duplicated, or otherwise reproduced in any manner without the prior written consent of Exception Software Ltd..

While every attempt has been made to ensure that the information in this document is accurate and complete, some typographical errors or technical inaccuracies may exist. Exception Software Ltd. does not accept responsibility for any kind of loss resulting from the use of information contained in this document. The information contained in this document is subject to change without notice.

The incorporation of the product attributes discussed in these materials into any release or upgrade of any Exception Software Ltd. software product—as well as the timing of any such release or upgrade—is at the sole discretion of Exception Software Ltd..

This edition published March 2007

Table of Contents

1	Corporate Information Flows	4
1.1	Background	4
1.2	Automated Processes	4
1.3	Handling Exceptions	5
1.4	Breaks in Automated Processes	5
1.5	Data Management	6
1.6	Compliance	6
2	Exception Handling in Data Warehousing	7
2.1	Automation through ETL	7
2.2	The process	7
2.3	Manual intervention	8
3	Exception Handling in Straight Through Processing	9
3.1	What is Straight Through Processing?	9
3.2	The problem with STP	9
3.3	Exception Requirements for STP	9
4	Exception handling in Data Quality projects	11
4.1	The requirements for tools in DQM initiatives	11
4.2	The need for Exception Handling tools	11
4.3	Streamlining a consistent data correction processes	12
4.4	Exception data are handled within the required time frames	12
4.5	Data manipulation and correction is logged and audited	13
5	Exception Handling & Master Data Management	14
5.1	What is Master Data?	14
5.2	What is MDM?	14
5.3	Issues	14
5.4	Statistics	15
5.5	MDM products	15
5.6	The missing link – Exception Handling	15
6	Exception Handling & Basel II Compliance	16
6.1	What is Basel II?	16
6.2	Status	16
6.3	Data Requirements for Basel II	17
7	Traditional Approaches	18
7.1	Data Quality Initiatives	18
7.2	Automatic Data Cleansing	18
7.3	Fixes 95% of Data	18
7.4	Manual Intervention	18
8	Exception Software	21
8.1	Correct	21
8.2	Approve	21
8.3	Escalate	22
8.4	Audit	22
8.5	The solution to Exception Data Management	22
8.6	Components Architecture	24

1 Corporate Information Flows

1.1 Background

The advent of computers has heralded a new era, revolutionising human processes and offering ever increasing possibilities of automations of tasks requiring a high element of repetitiveness or complexity. Broadly speaking the information technology revolution has allowed computers to replace humans in a whole range of activities, reducing the time taken to process any discrete operation and guaranteeing much higher levels of accuracy.

1.2 Automated Processes

Broken down to their lowest common denominator computing systems apply a pre-defined set of instructions (the software) to process input data, in order to provide a deterministic set of output operations. The quality of the output from a computer system is a function of the quality of the programs as well as the inputs provided to such programs.

1.2.1 Inputs

Inputs methods to software programs have evolved considerably over the course of computing history. Initial inputs on punch cards have rapidly evolved to provide evolved interfaces allowing humans and other automated systems to interface with the software programs. In order to perform the required processing it is imperative that the input data sets satisfy the requirement of the system: lack of information or dubious quality of input typically result in either rejection of the input or unsatisfactory and erroneous processing within the software program.

1.2.2 Input Validation

Given that the supply of input data sets consistent with the expected input model is a key enabler of correct software functioning, a number of different solutions have been adopted to ensure that processing is isolated from incomplete or incorrect input being provided - typically software applications provide a level of input validation which aims to reject the supply of input data inconsistent with the software model. Such records are typically saved to exception files, which can be either discarded or subjected to subsequent "cleansing" processing, either manual or automated.

Input validation and control is a science in its own right, and the level of validation of data input is highly dependent on the type of data as well as the input method. Data provided by automated systems (e.g. a point of sale application or a railway track sensor) will tend to be highly consistent with the predefined format and minimal validation at record level will be necessary, while a CRM application, requiring input from service personnel in a call centre will typically force little validation on the input, owing to the requirement for quick manual input, resulting in a poor level of data quality being introduced.

1.2.3 Input validation in the Web applications era

The last generation of applications, based on web front ends, has allowed companies to enlist consumers (B2C) or other organisations employees (B2B) amongst the users of such applications. While this paradigm shift has allowed for consistent cost savings in sales and support departments, it also has a direct impact on the quality of the input and of the data captured by such applications.

1.2.4 Processing Exceptions caused by faulty logic

Having accounted for potential erroneous inputs to software programs, poor design of the logic of software applications can lead to Exceptions being generated, resulting in transactions not being processed as desired. Most software programs embed sophisticated error handling routines to manage unexpected conditions, enabling the application to handle such condition; it is typical for applications to raise errors and generate log files or reports that would enable users to rectify the conditions that generated the exception. Often the users would have to audit the data and make the appropriate corrections to enable the application to re-process it.

1.3 Handling Exceptions

The aim of organisations implementing IT Systems is to provide increasingly higher levels of automation, replacing labour intensive operation with, effectively, mechanical substitutes for human activity. While full substitution of human intervention in a business process is possible, it comes at a cost, as implementing a computer based system capable of handling all possible combinations of inputs as well as un-expected conditions both in the input and in the processing is difficult and will typically require considerable time, both in design and implementation. As a result most organisations opt for the choice of implementing a system that will cater for the large majority of inputs and accept that some "exception" conditions will be met which will require user input or supervision.

1.4 Breaks in Automated Processes

1.4.1 Causes

Exceptions to automated processes may be the result of a number of causes - typically the input to the system is incomplete or inconsistent with the expected format or range of acceptable values. This may be the result of poor user (or indeed automated system) input, lack of input validation, missing data, or even failures in the software; failures in the software may be the result of poor or weak design (bugs) or of changes to the business requirement that have not been implemented in the systems.

1.4.2 Effects

Breakdowns of automated processes, even when only partial and affecting only an extremely small proportion of the overall workload, have serious consequences in terms of delays and costs: manual

processing to implement the end to end process is several orders of magnitude more costly than the automated computer driven implementation. And incommensurably slower, which typically results in additional costs and lost opportunities, reducing efficiency and competitiveness.

1.5 Data Management

Exceptions to automated processing pose a few problems to organisations as any automated process that fails, whether event or data driven, will require human intervention in order for it to be completed - this intervention comes at a cost, as companies have to employ operatives to deal with these types of problems (when exceptions are common enough) or ask staff in support roles to deal with such exceptions. These exceptions most probably have been raised by line of business applications of which the support staff has little or no knowledge, making the handling of such cases both difficult and expensive.

Furthermore, and increasingly more important, exceptions are typically dealt with by introducing ad-hoc workarounds, which over time enormously complicate the end-to-end process, burying business rules in tools such as Excel or, even worse, in individual employees knowledge.

With each manual intervention to manipulate business process data, the company is introducing the risk of errors, losing visibility of the process and of the data itself.

1.6 Compliance

Most industries have been subjects to increasing regulatory requirements for transparency, quality and rigorous adherence to defined processes for the usage, manipulation and reporting of data; typical examples of such regulations include Sarbanes-Oxley, Basel II, and various local Anti Money Laundering regulations. The aim of the various regulations is to ensure that the data reported to the market or used internally for management and strategic reporting is of high quality and accurate. Listed companies subject to Sarbanes-Oxley or financial institutions needing to adhere to Basel II regulations have invested large budgets into building automated processes to ensure the flow of data within their organisations is consistent with the demands of these regulations.

Unfortunately, as in most implementations of automated processes, it still is the case that exceptions to automated data flows are generated for human consumption; in order to deal with exceptions most organisations need to build new, bespoke applications to track the movement, correction, and management of these exceptions, capturing and storing auditing information that will sustain the scrutiny of regulatory bodies.

2 Exception Handling in Data Warehousing

2.1 Automation through ETL

ETL is the acronym for “Extract, Transform and Load” technologies. These are complex pieces of software whose primary usage is in the automation of bulk movement of data between source and target systems at the database/file level, enriching and altering the data as it is moved. Historically ETL tools have been used to ease the development and maintenance of data movements within data warehousing projects.

Developers use a highly productive graphical interface to create the mappings of data between source and target, and define reusable components which alter and clean the data.

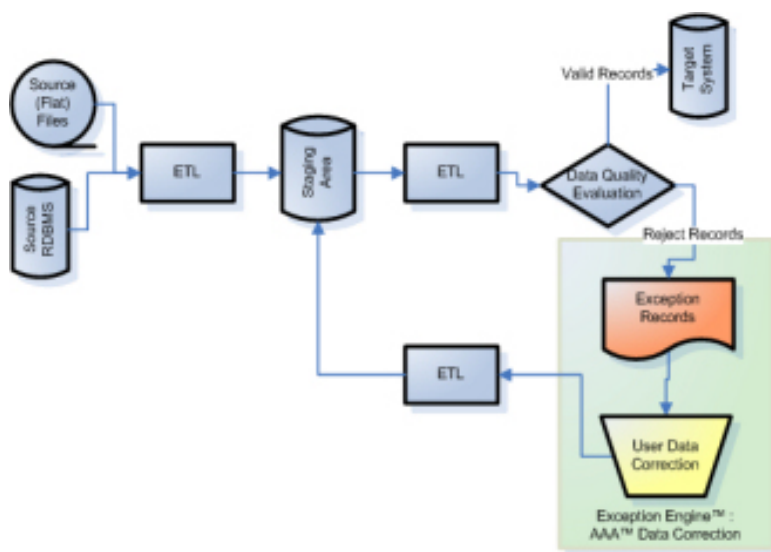
All of this occurs irrespective of the underlying data platforms and operating systems meaning the same developers do not have to have skills in many types of application, but can concentrate on the process of data movement instead.

2.2 The process

Typically ETL tools will have a scheduling facility which determines when data is extracted from the sources and under what conditions.

Workflows or data transformation flows are then executed which take this data and pass it through a series of transformation routines to, for example, alter the format, lookup the information in a master table, apply a calculation or aggregate values together, prior to loading the cleansed data into the target system.

The diagram here shows a typical data flow process for an ETL tool being used within a large enterprise.



2.3 Manual intervention

The rules which govern the cleansing of data as it is transformed will rarely be able to catch all of the different types of errors and exceptions which the source data can throw at them. In these cases it is important to ensure that information is not lost even though the target cannot be loaded with dirty data.

Consequently sets of exceptions records are created, generally in flat file format. These are then manually altered by business users who understand the data and then resubmitted into the process for loading to the target. In the diagram above this process is represented by the shaded area. Unfortunately there are a number of drawbacks within this model:

- There is no method by which the exceptions are automatically routed to users meaning that many errors are not corrected until far too late, or are lost in the system completely
- There is no way of auditing and logging the changes which have been made as in most cases this is performed within Excel or Notepad
- No security models exist to ensure only those people with the correct access can either correct or approve changes
- There is no integration back into the ETL environment to minimize the latency and share metadata.

3 Exception Handling in Straight Through Processing

3.1 What is Straight Through Processing?

Straight Through Processing (STP) is the end to end automation of the trading process within and between both buy and sell side institutions, from the first capture of an order through to final settlement.

Achieving the goal of Straight Through Processing is certainly very attractive to financial institutions engaged in multi-party settlement and transaction processing as it promises to minimise settlement risks while also reducing operating costs through automation from the institutions' core processes.

To meet the requirements raised by this goal a number of solutions have been provided by the IT industry to remove manual interventions in the flow of information related to capital market, payment transactions or other asset classes.

Typical solutions have included anything from packaged trading application providing transaction lifecycle management capabilities to bespoke point to point integration implementations sustained by large investments in EAI or other integration technologies.

3.2 The problem with STP

But while business process interoperability (BPI) initiatives aim to deliver Straight Through Processing to the financial markets, some industry analysts believe that STP is not an achievable goal in the sense that firms are unlikely to find the cost/benefit to reach 100% automation.

For example, in April and May of 2003, analysts at GartnerG2, and the Securities Industry Association (SIA) conducted a joint survey by phone and e-mail in which they surveyed 184 financial services providers in 21 countries worldwide about their STP initiatives.

The survey revealed that although two-thirds of respondents have launched at least one STP initiative or are planning one by year-end 2003, manual processes are still prevalent. Forty-two percent of all transactions continue to be paper-based and almost 40% of firms manually enter data at least twice for the same transaction.

3.3 Exception Requirements for STP

Whenever full automation of transaction processing cannot be achieved, manual remediation processes need to be put in place. It is essential to provide financial organisations with the tools to handle exceptions to transactions in a managed environment. As your organisation is aiming to meet the stricter demands of Basel II to display effective controls and management processes for the handling of pre or post settlement

exception resolution, it will benefit from the deployment of a controlled environment to deliver benefits in a number of areas:

Fully Audited exception data management – ensuring all data alterations to any data exception, generated at any point in the transaction lifecycle, are comprehensively logged, offering full visibility of transaction data flows for internal reporting and compliance management.

Easy integration within any STP flow – Exception data generated at any point in the transaction lifecycle can be captured and displayed in context sensitive data editing screens for users to edit. Exceptions to matching of trading or settlements can be viewed and edited, and an approval workflow ensures the correct authorisation level is applied before transactions are released into the STP flow.

Easy deployment, minimal training – the new environment should ideally include a web based application, delivered to the desktop via a browser interface. Minimal user training should be necessary to access the correction functionality and provide the highest level of context sensitive help to users, are dynamically generated by the application according to the type of exception data being processed.

4 Exception handling in Data Quality projects

4.1 The requirements for tools in DQM initiatives

The rapid change in B2B and B2C channels of the last decade, fundamentally driven by the web applications revolution, has created opportunities as well as challenges: the volume and fragmentation of data between systems, applications, internal and external suppliers has increased exponentially. One casualty of this wave of change has been the quality of data.

While historically the merging of systems was a long process, requiring months if not years of planning and development, the increased global competition and accelerating speed of change has driven organisations to rationalise and automate much of these programmes through the deployment of technologies such as Data and Information Integration as well as Data Quality.

The former set of technologies addresses the issue of consolidating systems, by either physically moving data or by providing logical views of the extended data sets enabling organisations to merge the data.

The latter address the issues of standardising the formats of typically inconsistent data sets, removing duplicates, and generally ensuring that the data is fit for purpose.

Both Integration and Data Quality initiatives are resourced through the deployment of automation tools, which allow organisations to efficiently design the business rules which will merge and cleanse (although not necessarily in that order!) the data.

4.2 The need for Exception Handling tools

The automation rules are typically designed to process automatically most of the data within the systems, but it is quite usual that a small percentage of data will be rejected as “exceptions” from the automated processes, possibly owing to the quality of records being below an acceptable level, or possibly because the automated systems have not been updated to cater for the constant changes in data feeds formats or system usage patterns.

In order to deal with such Exceptions it is common for organisations to build bespoke manual processes, requiring users to access the exception data, manually correct it and finally feed it back into the automated system.

It is very common for large organisations to have implemented several such manual data correction processes, both in line of business operations as well as in back office functions. The proliferation of these

ad-hoc procedures is not necessarily problematic, as it is often a temporary fix put in place while the automated processes are updated, but it should not prevent organisations from achieving the following goals:

- Streamline and implement consistent manual data correction processes
- Exception data are handled within the required time frames
- Data manipulation and correction is logged and audited to ensure regulatory compliance, when required

4.3 Streamlining a consistent data correction processes

It is of great benefit for organisations to plan for the handling of the Exceptions and to put in place procedures and tools that will cater for this recurring set of requirements. Exceptions will be generated by many applications, with different styles of records being produced, varying data formats and specific editing rules, but the general requirements will be the same for all exception handling needs:

- The exception data will need to be collected – the data or information integration technologies deployed will greatly simplify this task.
- The users will need to be able to view the data – a set of specific tools will need to be created to allow the users to view the data in its context.
- Corrections to the data will need to be applied – users can correct data more efficiently if the semantics of the data are known to them, therefore an exception handling application should allow for context sensitive help to guide users through the process of correcting individual data attributes.
- Validation rules should be exposed – basic data validation rules should be exposed to the users - for example, a missing product code should be filled in from a list of known valid values – and imposed on the data before it is released.

4.4 Exception data are handled within the required time frames

Another critical attribute of data often is its timeliness. If the availability of transaction data within the required time frame is essential for the company operations it follows that exceptions to those processes or information flows need to be handled in a timely (and speedy) fashion – ultimately exceptions may represent sales transactions not being processed, customers not being contacted or products not being sold.

In order to ensure that Exceptions are handled within the appropriate time frames, the exception handling application needs to ensure that:

- Exception data collection is consistent with timeframe requirements – the application should collect exception data whenever required, in batch or real-time mode.
- Exceptions should be displayed to users with the relevant processing time frame indicators

- An escalation workflow is available, so that data that is not processed within the expected timeframe is highlighted.

4.5 Data manipulation and correction is logged and audited

Depending on the type of data being processed, visibility and tracking of the changes that were applied may be not only relevant, but mandatory. The flow of financial transactions amongst systems, be them operational or analytical, is subject to an ever increasing scrutiny by a range of regulations, from Sarbanes-Oxley to Patriot Act, from Basel II to Solvency 2, etc.

Similar regulations affect data generated and managed by organisations in the pharmaceutical as well as government sectors. When no external regulatory bodies demand close scrutiny on data handling, internal initiatives to adhere to high processing standards in order to gain ISO certifications will generally require that changes to data are logged and auditable.

An Exception handling application should ensure that:

- A copy of the original data set is held unchanged
- All changes to the data are stored, for each data record and attribute
- All user activity is recorded
- A set of reports can be produced at any time, showing which changes were applied by whom at what time.
- Exports of data back into the automated process are logged, including a full copy of the released data set.

5 Exception Handling & Master Data Management

The area of master data management (MDM) has become a hot-topic in the last 12 months. Many vendors are proposing a technology solution to the problem and in most cases redefining the scope of MDM to fit those products. Unfortunately MDM does mean different things to different companies, and an approach including not just technology but process and people is required for success.

5.1 What is Master Data?

Master data can loosely be defined as that fairly static reference data which is used across many areas of a business. It can include physical items such as products or customers, as well as business information such as the rules definitions for profit or risk.

5.2 What is MDM?

Management of this master data involves ensuring that the same information is available to all consumers of the data in its most up-to-date format. As such this data must be propagated, synchronized, shared and consolidated within all application touch-points in and outside the enterprise.

Achieving this is not just a case of having a standard central repository of 'clean' master data and expecting all applications to access it. Wherever data can be updated or added there must be standard data quality rules in place, and as soon as data has been altered this must be synchronized with all other applications which also store that data locally.

5.3 Issues

The issues are widespread and a short list is described below:

Distributed data – the same information often appears in many places e.g. products list can be in a warehouse application, invoicing system and data warehouse

Unknown formats – master data information can be held and stored in different formats e.g. the same part code might be P1234, P-1234 and the number 1234 as the key identifier in different applications

Poor data quality – manual entry of this data invariably leads to mistakes or misspellings due to human error.

Real-time access – batch update of the data is often too slow for LOB applications

Unstructured information – often information comes in non-structured formats such as pdf

No process or ownership – with data spread across the enterprise and different areas of responsibility assigned by application or LOB, there is rarely a custodian or owner of the data quality and integration requirements of the infrastructure

5.4 Statistics

The complexities of these solutions are such that several analysts have performed a number of studies to understand the exact size of the problem:

TDWI – the cost to US companies for poor quality data within their enterprise is \$616bn per year.

MetaGroup – 80% of all companies have NO centralized data strategy

TowerGroup – following a survey of 100 companies, they found the following:

- on average there are 11 different definitions for the same product held in multiple databases
- one company had 23 different definitions for profit
- the average number of incorrect invoices is 30%
- a component manufacturer had 20,000 products in their database, when they only sell 5,000.

5.5 MDM products

Many large vendors have brought components of their existing technologies to market to solve this large problem of data stewardship. This includes the sectors of:

- Business Intelligence and Data Warehousing – the building, reporting and management of master data tables
- Data Quality – the de-duplication, matching and cleansing of data in multiple places
- Data Integration – real-time extraction transformation and loading of cross-platform master data
- Application – embedded master data functionality in ERP and CRM systems.

5.6 The missing link – Exception Handling

In reality, elements of all of these are required for a truly comprehensive implementation. Whilst all of them cater for the automation of the most common elements of a centralized data management strategy, the area of manual intervention for the correction and auditing of exception data is not handled.

Automated exception handling fills this gap with a business user interface to exception data coming from any source application, data integration tool, data quality product or business intelligence vendor. It provides functionality to rapidly allocate, correct and redistribute erroneous data through manual input, whilst providing full workflow management, auditing and validation of error records.

6 Exception Handling & Basel II Compliance

6.1 What is Basel II?

Basel II is the second Basel Accord and represents recommendations by bank supervisors and central bankers from the 13 countries making up the Basel Committee on Banking Supervision (BCBS) to revise the international standards for measuring the adequacy of a bank's capital. It was created to promote greater consistency in the way banks and banking regulators approach risk management across national borders.

There are 3 Pillars of Basel II:

- Minimum capital requirements - the ability to capture and record risk
- Supervisory review - the use of risk information in running the business
- Market discipline – the statutory reporting of a company's risk position

6.2 Status

Implementation of Basel II is in 2 waves. Wave 1 covered pillars 1 and 3 of the accord stating that financial institutions must comply with the 'minimum requirements' by 1st January 2007 for collecting and reporting financial risk. This was not easy:

"Complying with Basel II is more difficult than most bankers thought."

-- Paul Cartwright, Accenture Financial Services Operating Group

Many institutions either missed this deadline, or implemented a much simpler risk solution than they had intended. In nearly all cases the technology requirements were underestimated and the processes and people were not adequately prepared.

Wave 2 covers pillar 2 and enforces the banks to prove that they are managing their business appropriately through a close loop implementation of the risk position they have from wave 1. This deadline is 1st January 2008. It is likely that the same problems will apply and that much manual processing will be still be in place long into that year before an automated architecture is deployed effectively.

"Not only are larger volumes of data needed more than most banks currently capture, but it needs to be of significantly higher quality, with greater consistency, auditability, and transparency than before. These changes also demand greater access, sharing and reconciliation of information... To meet this need, banks today require more integrated data management solutions... Many banks see this as the largest implementation challenge they face in their Basel programmes."

-- "Reality Check on Basel II," The Banker, Accenture, Mercer Oliver Wyman and SAP

6.3 Data Requirements for Basel II

The main components of data management which are necessary for a successful implementation of a Basel II programme can be summarized as:

Access: Access to all relevant data sources, internal and external.

Quality: Completeness, accuracy and appropriateness of data.

Reconciliation of data across risk management, reporting and accounting systems.

Auditability: Transparent processes with audit trail on data.

Delivery: Deliver data when, where, and how needed.

Availability/Scalability: High availability of data and ability to process large volumes of data.

Typical data issues challenging banks in their efforts to implement their compliance programs include:

Unreliable aggregate exposure data. Duplication of customer records, errors in customer's group hierarchies and incomplete data on loan maturity dates raise regulatory concerns on the determination of aggregate exposure amounts – a key requirement for the corporate asset class.

Disorganised collateral records. Deficiencies in collateral identification and linkages between collateral codes and exposure amounts pose questions on reliability of LGD estimation and risk mitigation calculations in both retail and corporate books

Incomplete loss data records. Ill-defined business processes and incomplete data recording materially affect LGD calculation and estimation.

Inaccuracies in rating and scoring data history. Major gaps in this data raise challenges to validation of banks' PD estimation processes.

7 Traditional Approaches

Given that most IT systems generate exceptions, in varying degrees of quantity and quality, companies have adopted a range of techniques to reduce the impact on their business of managing such cases.

7.1 Data Quality Initiatives

Data Quality initiatives have been put in place to prevent exceptions occurring or automate the processing of such occurrences. Preventative measures vary from applying tighter controls and validations on data input, to applying strict data quality and enhancement rules when moving data between applications or within a reporting (regulatory or management) information chain.

7.2 Automatic Data Cleansing

A number of products on the markets have addressed the requirement to automatically "cleanse" data as it is entered into a system or as it flows between systems, examples of these include products offered by companies such as pure-play Data Quality vendors such as Harte-Hanks(Trillium), or BI and Data Integration specialists such as Business Objects (First Logic), Informatica (Similarity Systems), IBM (Vality) or SAS (DataFlux).

7.3 Fixes 95% of Data

Unfortunately, even with a large investment in both software acquisition and implementation costs, automated Data Quality tools still generate a generous amount of exception records, potentially recording the reasons for data rejection in additional data attributes, which it is hoped will allow operators to properly assess, investigate and fix the causes of such exceptions. It is estimated that Data Quality tools will enable the automatic cleansing (correction) of up to 98% of records. So what happens to the remaining 2%?

The remaining 2% of records present a set of problems: they cannot be discarded, as they may represent high net worth transactions or highly business relevant information assets; losing such data may be too costly an option to consider. And correcting them manually may introduce additional errors and lack of auditing and visibility in the end-to-end process.

7.4 Manual Intervention

Most organisations still implement manual workarounds to deal with exceptions to the automated processes; most typically departments will appoint a number of data managers whose task is to correct the errors in the data, so that it can either be pushed through the automated system or it can be directly loaded into the target application.

7.4.1 Existing Processes

Typically users will receive files of exceptions, possibly in Excel format, or collect the erroneous data from the application database using typical desktop tools, such as Excel or Access; more sophisticated users may query the database using dedicated Business Intelligence tools (Business Objects, Cognos, or similar), while the more technically inclined will also use database specific tools, such as SQLPlus for Oracle, SQL Analyzer for SQL Server, etc.

Once the data has been extracted to the data industry entry tool standard, Excel, the data manager will "eyeball" the problem, and proceed to correct the erroneous records and fields. This is to the untrained eye paramount to a black art. The rules that dictated what are the acceptable values within fields, and the possible combinations among them, may be defined in some rulebook, most likely they're applied from memory or learnt from trial and error experience by the data manager.

In fact manual intervention in exception handling suffers from a number of problems:

- 1) The exception handling processes will be ad-hoc, with many parts of a large organisation supporting similar, but not necessarily consistent implementations. Duplications represent an unnecessary cost to organisations.
- 2) Manual processes are typically poorly documented and even when they are well documented adherence to the protocol is difficult to enforce and audit. Lack of documentation is a hidden cost of manual processes.
- 3) Changes to data are poorly or completely not logged, breaking any auditing chain that may be required on the end-to-end data flow. Manual processes may break regulatory compliance requirements.
- 4) Un-assisted and un-audited data correction leads to further errors being introduced.
- 5) When further errors are introduced by the manual data correction it is difficult to identify the individuals and circumstances responsible - management information is completely unavailable.

7.4.2 Ad-hoc

Typically exceptions are not foreseen, planned events; as such the system provides no interface for handling such cases; exception data is made available to operators through files or spreadsheets. The content and format varies from application to application. Manual error handling is an ad-hoc process, implemented as a last minute fix.

7.4.3 Difficult to document

Documentation related to the errors, and their context is typically lacking; the process itself for dealing with the exceptions is poorly documented and understood within the organisations. Typically individuals will develop patterns to deal with specific types of exceptions. Remove the individual and the exception will no longer be handled.

7.4.4 Breaks auditing chain

Whenever the correction of data happens using bespoke applications built on top of tools such as Excel or Access the business is losing visibility and control - the organisation can no longer have confidence that the data has been handled throughout its lifecycle, and answer questions such as: where is this record coming from, who's changed it and when?

8 Exception Software

Exception Software is the provider of the next generation exception data management facilities. Its Exception Engine software enables organisations to quickly implement a framework for the correction of data, enabling organisations to achieve Assisted, Assured and Audited (AAA™) data quality edits.

Exception Engine can be deployed within any project to close any data audit loopholes, be it within a financial regulatory reporting warehouse or part of a transactional straight through process, be it implemented within the context of an ETL or EAI architecture.

The primary objective guiding the architecting and development of Exception Engine was to provide organisations with a tool based approach to the manual handling of data which could be implemented extremely quickly, requiring minimal training of end users and administrators alike.

Exception Software key idea when conceiving the product was that the tools should provide advanced functionality to aid users in the correction of data, keeping a thorough audit trail of all data manipulation operations, while at the same time retaining the ease of deployment and adoption of Excel.

Most companies have spent considerable time and effort automating their operations through the deployment of information systems and application software and yet the requirement to enable users to edit data outside the confines of such applications is still real and pervasive.

Exception Engine provides a turnkey solution to manage the correction of any data addressing your four key requirements:

8.1 Correct

Organisations no longer have to struggle to achieve the objectives of Efficient processing of manual input, with consistent high Quality data outputs. The adoption of Exception Engine, encapsulating the AAA exception management principles, will deliver these benefits through:

Error Highlighting: error values are colour coded according to predefined error classifications

Assisted Correction: values lists may be predefined to ensure only valid corrections are made

Context Sensitive Help: guides the users suggesting which corrections are appropriate

8.2 Approve

Exception Engine provides an approval workflow, ensuring that records are released from the application only by users with the appropriate level of authorisation. A sophisticated role based security model ensures

that users perform only the operations congruent with their role, while record definitions dictate which rights users have on specific fields:

Users Roles: provide predefined rights to Editors, Reviewers and Data Managers

Data Rights: on each record type, define whether users can view or edit specific attributes

Approval: only users with the appropriate rights can approve records and release them from Exception Engine

8.3 Escalate

Exception Engine offers benefits to data or line of business managers, as it provides full data correction workflows ensuring data is corrected within the required timeframes and to required standards.

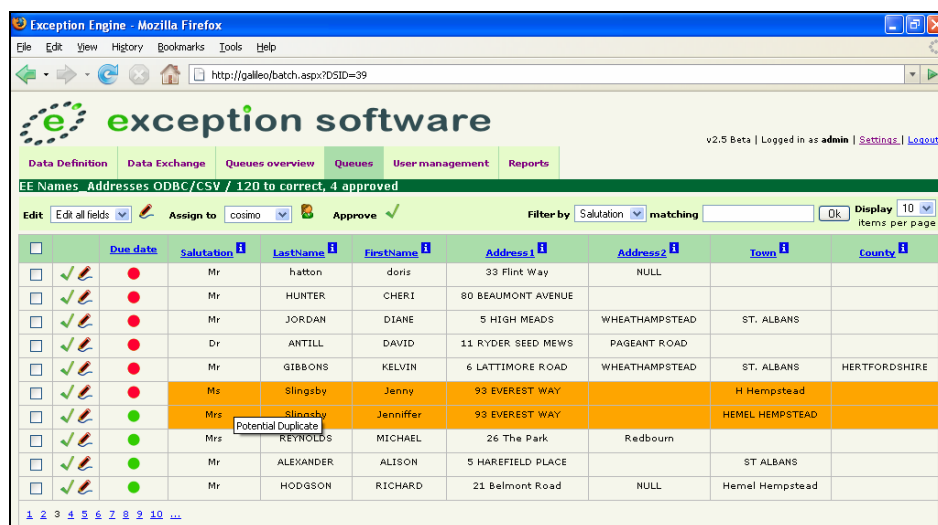
Each batch of records loaded into Exception Engine is assigned its due date; records are tracked within the system and an escalation workflow is initiated if they are not released within the predefined time frame.

8.4 Audit

Most importantly Exception Engine records all data manipulation operations, providing full auditing to enable organisations to meet their regulatory requirements or internal data governance rules. Data manipulation reports are provided by Exception Engine and the full audit trail can be exported or integrated within the organisation's Business Intelligence or Metadata Analysis preferred solution.

8.5 The solution to Exception Data Management

Quite simply, Exception Engine™ provides organizations with an out-of-the-box solution to the problem of dealing with the exceptions occurring in any data driven process. Exception Engine™ is a web-based application which enables users to apply the necessary manual corrections to the exceptions in order for the data to then be successfully processed by the automated flow.



	Due date	Salutation	LastName	FirstName	Address1	Address2	Town	County
<input type="checkbox"/>	✓	Mr	hatton	doris	33 Flint Way	NULL		
<input type="checkbox"/>	✓	Mr	HUNTER	CHERI	80 BEALMONT AVENUE			
<input type="checkbox"/>	✓	Mr	JORDAN	DIANE	5 HIGH MEADS	WHEATHAMPSTEAD	ST. ALBANS	
<input type="checkbox"/>	✓	Dr	ANTILL	DAVID	11 RYDER SEED MEWS	PAGEANT ROAD		
<input type="checkbox"/>	✓	Mr	GIBBONS	KELVIN	6 LATTIMORE ROAD	WHEATHAMPSTEAD	ST. ALBANS	HERTFORDSHIRE
<input type="checkbox"/>	✓	Mrs	Slingsby	Jenny	93 EVEREST WAY		H Hempstead	
<input type="checkbox"/>	✓	Mrs	Slingsby	Jennifer	93 EVEREST WAY		HEMEL HEMPSTEAD	
<input type="checkbox"/>	✓	Mrs	Potential Duplicate REYNOLDS	MICHAEL	26 The Park	Redbourn		
<input type="checkbox"/>	✓	Mr	ALEXANDER	ALISON	5 HAREFIELD PLACE		ST ALBANS	
<input type="checkbox"/>	✓	Mr	HODGSON	RICHARD	21 Belmont Road	NULL	Hemel Hempstead	

8.5.1 Exception Engine™ Benefits:

- Easily deployed web application
- Securely handles any data
- Drives user efficiency
- Audits and logs all data edits

The packaged application delivers a high level of flexibility by allowing any data set to be processed. At the same time, a high level of control and transparency of user activity is retained allowing all operations to be fully audited.

Input data for correction can be collected from a number of sources, including flat files or relational databases - an import screen allows administrators to define the data formats, as well as which elements of the data can be edited by which users.

The result is **AAA** exception data management:

Assisted Correction - Helpful hints at record and field level allow users to understand the context within which they are working, enabling even inexperienced staff to quickly validate and correct exception records - valid values drop-down boxes allow them to choose the appropriate content for specific fields, while a read-only view of the exception record ensures all available information is displayed to aid the correction.

Approved Correction - A set of pre-defined validation rules imposed on individual fields or across fields within the record ensures that records can be released by data editors as "corrected" only when all validation rules have been positively met.

Furthermore, all record edits are subjected to an approval workflow, ensuring that only fully validated records are reintroduced into the automated process. Edits performed by an Editor role require approval by an Auditor user in order to be released by the application.

Audited Correction - Only users with the appropriate rights and roles are allowed to view or edit records. All user edits are audited, down to individual field level - record and field level audit trails can then be used for management and regulatory reporting, either through the application's built-in reports or through export/exposure of logs to external Business Intelligence tools.

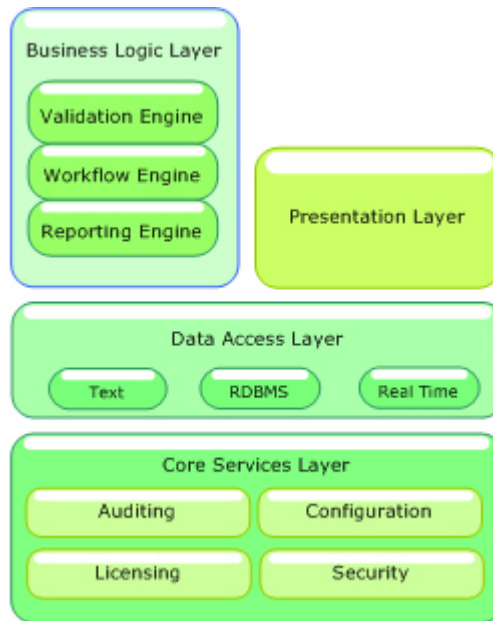
Data exceptions are presented to users within work queues and a data editing screen allows data editors with appropriate privileges to view and amend data elements, providing context sensitive hints and guidelines. Users with approval privileges are able to sign-off the corrected data.

Corrected records are then re-inserted into the automated process, be it within a data-warehouse loading cycle, a data cleansing batch process or any other line-of-business workflow.

All user activity is logged and a full audit trail of changes to data is available to the relevant users and can easily be made available to any automated Straight Through Processing application.

8.6 Components Architecture

The application is built around the core principles of SOA and component driven architecture and provides users with a zero-footprint, browser based interface.



The architecture diagram on the left depicts the main components of the architecture.

The web interface delivers Content to the users through a set of highly intuitive screens, which can be easily customised in look and feel to fit the company's corporate identity through a set of editable theme elements. The browser connection to Exception Engine™ is encrypted using the SSL encryption standard, ensuring that data access is secure, enabling deployment both within the company's secure zone and also to remote locations on extranets.

8.6.1 Data Edit Screens

Users with appropriate privileges can view the list of Queues with data items that need to be processed. Queues provide users with visual clues as to how many items need to be processed and convey vital information regarding the time criticality of the work. Upon opening a Queue, users can view and edit data records. The original record is displayed and changes to it can be applied; contexts sensitive help offers users information regarding the expected formats and values of data. Validation rules built by administrators ensure that the corrected record satisfies the business rules imposed on the data. Users with the appropriate

sign-off privileges can release corrected batches of data from Exception Engine, while users with basic editing rights can release batches into the approval workflow.

8.6.2 Management Screens

The management screen enables administrators to import the definitions for Datasets. Dataset definitions can be thought of as the general specification for a type of records - in programming terms Datasets can be thought of as "Classes", while the records should be thought of as "Objects". The definition of a Dataset can be derived from an XML file, or entered directly within the Exception Engine's administration page. The record definition used as a basis for the specification of the Dataset can be imported from a flat file (delimited or fixed width text) or directly from a database table. The administration screen then allows the administrator user to define parameters such as whether certain fields should be read-only or editable, visible or not, whether they should be validated and what the validation rules should be. Once the administrator has defined the type of data collection mechanism, i.e. whether the input data should be read from file or database, the application is ready to provide data to the users for correction!

8.6.3 Role Based Security

One of the most important aspects in the implementation of a data correction process is that of the security of the data. Exception Engine™ enforces security through a number of layers: all data is stored centrally on the server and accessed by the users exclusively through the application's data editing screens, delivered to the desktop within a secure browser connection. Authenticated users can view and edit data belonging to the specific datasets for which they have been granted access rights. Role based security ensures that only users with Reviewer role are allowed to release corrected records from Exception Engine™ back into the automated process.

8.6.4 Data Auditing

All data manipulation activities including edit, sign-off and release of data from Exception Engine™ are logged by the application. A set of built in reports allows users with the appropriate privileges to view the history of any batch, including all individual record attribute changes. This vital information can be exported directly for further analysis; the reporting model can also easily be extended through the company's own management information system.