

White Paper

# Data Matching with IQ Office

COMMERCIAL IN CONFIDENCE

Version 4

Marketing

© 2019 Intech Solutions Pty Ltd. All rights reserved. Intech makes no warranties, expressed or implied, in this summary. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

# CONTENTS

Glossary.....	4
Overview .....	6
About Intech Solutions .....	6
Intech's IQ products.....	6
Third-party relationships .....	6
Data matching .....	7
Applications .....	7
The case for Data matching.....	7
Examples of use .....	8
What is Data matching?.....	9
What should a Data matching product offer?.....	9
Probabilistic matching versus Deterministic matching models.....	10
Why use <i>IQ Office</i> ? .....	11
Functional description of Data matching.....	12
IQ Standardiser .....	12
IQ Matcher .....	13
Technical features for Data matching .....	13
Key characteristics .....	13
Overview of the Matching process .....	14
Data matching flow process .....	16
Standardising .....	17
Matching .....	18
Indexing.....	21
Project encapsulation and automation.....	21
Technical architecture .....	23
How <i>IQ Office</i> works.....	23
IQ Standardiser .....	24
IQ Matcher .....	25
Appendix .....	28
IQ Rapid Address .....	28
IQ Profiler .....	28

## FIGURES

Figure 1: Model of entity records .....	15
Figure 2: Data flows between <i>IQ Office</i> and users .....	16
Figure 3: Generic standardising data flows in <i>IQ Office</i> .....	17
Figure 4: Screen shot of <i>IQ Matcher</i> – De dupe .....	19
Figure 5: Screen shot of <i>IQ Matcher</i> – OrgBatchMatch.....	20
Figure 6: <i>IQ Office</i> data flow.....	23
Figure 7: Screen shot of <i>IQ Matcher</i> – Database connectivity .....	26

## TABLES

Table 1: Data element example .....	18
Table 2: Comparison types and tolerances .....	20

## GLOSSARY

These abbreviations, acronyms, initialisms and special terms appear in this document:

Term	Definition
Aggregation	Calculation performed to group related records based on desirable attributes
AMAS	Address Matching Approval System (Australia Post)
API	Application programming interface
ARF	Address reference file
AS	Australian Standard
C	A programming language
Candidate Match	A matching score that lies between the Upper and Lower Cut offs
CCD or CD	Census collection district
CLR	Common language runtime, also known as the Virtual Execution System (VES)
COBOL	A programming language
COM / DCOM	Component object model / distributed COM
CRM	Customer relationship management
CSV	Comma separated value
Cut-off	See Lower Cut-off and Upper Cut-off
Data steward	A more experienced and possibly trained operator or administrator
DB	Data base
DTS	Data Transformation services
Entity	Person, organisation or thing that is of interest
Geocoding	The process of matching geographic data to reference data, for instance supplying latitude and longitude for a street address
GIF	Geographic information file
G-NAF	Geo-coded national address file
Grammar	A configurable rule-set for <i>IQ Office</i>
GUI	Graphical user interface
IP	Internet protocol
Integration	Smooth inter-operation between <i>IQ Office</i> components and with customer applications
ISO	International Standards Organisation
JCL	Job control language
JNI	Java Native Interface
Lower Cut-off	The minimum matching score needed to consider the comparison a Candidate Match

Term	Definition
Match type	See Candidate Match, Non-Match and True Match
Metaphone	An algorithm that generates a code representing the sound of a string of characters
Non-Match	A matching score that falls below the Lower Cut-off
NYSIS	New York State Identification and Intelligence System phonetic code
NZS	New Zealand Standard
OLE	Object linking and embedding
PAF	Postal address file
Parse	To apply 'grammar' rules to incoming data to make it readable in the next step in the process
PL-SQL	Procedural language extension to SQL
PSMA	Public Sector Mapping Agencies Australia Limited
RDBMS	Relational data base management system
RESTful	Representational state transfer – an alternate to SOAP – a web service
SaaS	Software as a service
SOAP	Simple Object Access Protocol – a web service
Soundex	An algorithm that generates a code representing the sound of a string of characters
SQL	Structured Query Language
Standardise	To bring up to a standard
TCP IP	Transmission Control Protocol / Internet Protocol
T-SQL	Transact SQL
True Match	A matching score that falls above the Upper Cut off
UI	User interface
Upper Cut-off	The minimum matching score needed to consider the comparison a True Match
VAR	Value-added reseller

## OVERVIEW

This document contains information relating to the use of *IQ Office* for Data Matching as a business practice for improving effectiveness and efficiency, whether you are in the public or private sector. It includes overview information, a functional perspective of the proposed solution and a brief description of the technical architecture. Finally there are brief descriptions of the other components of the *IQ Office* product.

## ABOUT INTECH SOLUTIONS

Intech Solutions (Intech) helps you to improve the quality of your data by providing a broad range of products and services that identify and manage information quality issues.

Intech is an Australian-owned proprietary limited company headquartered in Sydney, New South Wales. Intech prides itself on the quality of its solutions and its many years' experience providing state-of-the-art products and services to its customers.

Intech has been delivering information quality solutions since its founding in 1996. Intech now has over 130 customers including Australasian government agencies (local, state and commonwealth), utilities, telecommunication, not-for-profit, automotive and large corporates. Users and managers in these, all rate themselves as extremely satisfied customers who maintain excellent ongoing, and in many cases, long-term, mutually beneficial relationships with Intech.

## INTECH'S IQ PRODUCTS

Intech's products are the result of years of dedicated software development via a process of customer-driven desire to increase functionality, effectiveness and efficiency. This development method has ensured that the products developed are superior in design providing the highest levels of relevance, functionality, performance, robustness, accuracy and longevity. The software products incorporate state-of-the-art technologies and algorithms published by tertiary institutions, and are continuously re-assessed and upgraded in response to customer demands for increased functionality.

Intech's software is compatible with many operating systems including Windows, Unix-based and AS/400, with such integration paths as Component object model / distributed COM (COM / DCOM - Windows specific), Object linking and embedding database (OLE DB) connectors, database stored procedures, Java, C Library, IP Sockets, and web services such as Simple Object Access Protocol (SOAP).

## THIRD-PARTY RELATIONSHIPS

Intech Solutions has held numerous mutually beneficial relationships with such third parties as Customer relationship management (CRM) suppliers, Hewlett Packard, IBM, Microsoft Corporation and Oracle.

The reference data used by *IQ Office* is sourced directly from their relevant suppliers. Intech Solutions is:

- an Australia Post Level-2 Address Matching Approval System (AMAS®) Licensee - Intech Solutions software has been approved under Australia Post's AMAS® program for Batch and Rapid functionality on all major platforms
- a Full Access PSMA Geo-coded National Address File (G-NAF) Value Added Reseller (VAR)
- a corporate licensee for distribution of New Zealand Post's Geo Postal Address File (PAF)
- a NZ Post's SendRight™ Certification partner and holds rights to use the Geocoded Postal Address File
- a supplier of *IQ Office* capabilities to, and receives address and geocoding data from Critchlow Ltd, for its Critchlow National Address Register (CNAR) of New Zealand physical addresses and geocoding data.

# DATA MATCHING

This section of the document contains information about data matching in five sections:

- Applications
- What is Data matching?
- Functional description of Data matching
- Data matching flow process
- Technical Architecture

‘Applications’ describes typical usages and makes the case for Data matching. The next sub-section defines Data matching and discusses the merits, and describes the characteristics of, *IQ Office*. The ‘Functional description’ provides information on how Data matching is achieved, its challenges and common pitfalls as well as best practice methods for Data matching; it is aimed at the purchase decision maker. ‘Data matching flow process’ section provides a more detailed explanation for potential users. ‘Technical architecture’ provides detailed technical information on how Data matching is performed with *IQ Office* software, and is here to assist the developer.

## APPLICATIONS

Probabilistic data matching emerged in the 1960s as a methodology for improving data quality. Data matching (also known as record linkage) can be done entirely without the aid of a computer, but the primary reasons computers are often used for data matching are to reduce or eliminate manual review and to make results more easily reproducible. Computer matching has the advantages of allowing central supervision of processing, better quality control, speed, consistency, and better reproducibility of results.

Data matching quickly became important to administration, management and marketing personnel in government agencies and the private and Not-for-Profit sectors as they sought to improve service delivery, minimise operating costs and optimise marketing returns.

### The case for Data matching

All data about an entity, whether it be a customer, an organisation, a product or a service, has value for the purpose of matching entity records. However, records in relational databases are often kept in unstructured or semi-structured form, and many commercial applications will force their own structures on the data. These kinds of structuring can result in records that, in their current form, are not fit for any other purpose. Additionally, there may be no ‘native’ mechanism to assist in preventing the creation of duplicate records or the identification of existing similar records. The purpose for which the records will be used, eg, corresponding with customers, must be considered so that appropriate mechanisms can be put in place to facilitate this use.

Data matching gives you an additional level of confidence compared with having only separate instances of the entity record, eg, by identifying duplication within a single data source or related records across multiple sources, thus enhancing the effectiveness of your decision making, marketing and service delivery. Your goals may include optimal use of your matched data for:

- fine-tuning your market segmentation
- identifying customer preferences and activities across multiple touch points and potentially predicting future consumer behaviours – comparing, buying, using, repairing, discarding, bad-mouthing
- targeting communication with your customers
- upselling or cross-selling to your customers
- local community and geo-demographic analysis to develop better policy

- unique entity identification that can be focused on clearly defined persons of interest
- identification of activities and relationships of persons of interest that may assist fraud detection.

More recently, emergency services operators (ESOs), health-care providers, law enforcement officers, customs and immigration authorities have embraced Data matching for the effective fulfilment of their respective roles. Today, Data matching is regarded as an essential part of a robust data quality regimen.

### Examples of use

The benefits of Data matching can touch almost every aspect of an organisation's relationship with its customers; for instance:

- A community service provider, acting as a first response coordinator, offering 'front line' assistance in times of natural disaster may need to consolidate records of affected people with those of relatives in order to reunite them at some future time. Data matching can aid in the comparison and ultimate association of the records of affected people and their relatives. Additionally, the service provider may have a duty to coordinate with emergency services to ensure that the right people receive the right assistance at the right time. Data matching can help here, too, to ensure that of a number of 'John Smiths', the right 'John Smith' receives assistance and that time and resources are not wasted attending to the wrong 'John Smith'.
- A member of parliament wants to contact all members of the electorate who have not been in contact with the member over the last three years, in order to target a re-election push more efficiently than a whole-electorate mail-box drop. A matching of data records between the electoral roll for that member's electorate and the member's incoming correspondence files would be a good start for identifying the gaps, and hence be a good start for a targeted mail-out.
- A marketing manager wants to sharpen up the department's market segmentation and consumer behaviour analytics to focus on attracting specific, prospective buyers to a new product. Data matching can be used to bring together all instances of a particular customer across the organisation's data holdings so that the analysis performed can be applied to that single representation of the customer rather than to each separate instance in turn. Further to this, the identification of a single representation of the customer can facilitate improved communication with the customer and the ability to upsell or cross-sell because the marketing manager has greater certainty about products already owned or services already in use.
- An airline marketing manager wants to reward the more frequent flyers as well as 'high value' customers. Data matching allows all instances of a customer to be compared and a single representation of that customer to be identified and assigned a unique frequent flyer number. An analysis of ticket sales \$\$ data records against frequent flyer numbers will then allow a quick Pareto analysis (the 80/20 rule), ie, which 20% of flyers consume 80% of revenue-producing seat-km and what the total 'spend' is for each of those flyers.
- A construction industry supervisory body holds information about its members, being both employers and employees. To provide assistance to and manage complaints made by those members, customer service operators at the supervisory body must be able to identify an employer or employee confidently using limited information. Data matching can assist with the correct identification of the employer or employee by comparing information received over the phone, in hand written forms or via the internet with employer or employee information already held by the supervisory body.
- A government department responsible for co-ordinating school student and other concession-based travel must be able to determine whether an applicant already exists among their records. People applying for concession cards may attempt to defraud the department by applying for multiple cards using names of other family members or applying for cards to which they are not entitled. Data matching can assist by comparing the record of an applicant with records already held by the department. Similarly, the record of an applicant can be compared to a pre-qualified list to determine eligibility for the requested concession. This task also benefits from location awareness that can be provided using geographic coding functionality such as that found in *IQ Office*.
- A council of industry stakeholders may decide to establish a central repository of customer information that will allow all stakeholders to access and interrogate the information to identify instances of fraud. The ability to identify a customer of interest and determine with which stakeholders that customer has done business,



is invaluable in combating fraud. The first, critical step is to use data matching to compare many instances of 'John Smith' within the central repository to determine which instances represent the same customer of interest. Using that information, the council can trace the interactions with stakeholders and assess the legitimacy of those interactions.

- The ability to track the progress of students as they move through the various stages of the education system, eg, primary, secondary and tertiary, is vital when creating policy, planning services and reviewing the students' progress to determine whether the policy has been successful. Data matching can be used to ensure that numerous instances of a student can be brought together and a unique and de-personalised identifier assigned to facilitate the tracking of that student. Data matching can also be used to ensure that any potential new instances of that student are compared to existing instances to avoid the creation of duplicates.
- A government agency is concerned about transportation of people across national borders, eg, visa applications, or illegal immigration. For example, data matching can be used to ensure that applicants for new visas can be compared to records of existing visa holders. This can assist in identifying instances of fraud such as one person applying for multiple visas, potentially on behalf of others who might seek to sell them or use them illegally using the initial applicant's identity rather than their own. Data matching would also allow comparison between visa applicants and persons of interest in an attempt to identify 'undesirables' trying to enter the country.

## WHAT IS DATA MATCHING?

Data matching is the process of comparing entity records within a data set or across multiple sets of data. The purpose is to discover entity records that are likely to represent the same entity because of similarities between specific data elements in the entity records being compared.

### What should a Data matching product offer?

The act of matching entity records is also known in the literature as 'record linkage'. Thus, 'matching' and 'linking' are often used as synonyms. In this paper we will use 'matching' exclusively.

A data-matching product should enable the transformation of data of any type, eg, name, address, company name, email, phone number, product description, ABN, to pre-defined standards; this is the process of parsing and standardising. Ideally, it should cleanse 'messy data' to produce well-structured, high-quality information optimised for downstream data matching and analytics.

A data-matching product should enhance the data, eg, address, ABN, within each entity record by converting 'dirty data' into a standard form and potentially render it more accurate by comparing it with an authoritative source, thus validating the data. Similarly, by comparison with a name reference file, it should flag similarities between Pat, Patrick, Paddy and Rick such that each could contribute positively to a match between two entity records containing any of those names.

Standardised and validated data could then be used to achieve a more reliable match between two or more entity records.

The solution should be highly configurable and extensible to meet user-specific requirements.

Data matching should compare data within a single data source or integrate data from multiple sources, offer probabilistic or 'fuzzy' matching and suggest a grouping of records that can then be analysed to garner insights.

Data matching should enable:

- extraction of information from key data
- tagging the data with value-added metadata
- indexing the information

- applying intelligent algorithms to discover relationships between data entities
- aggregation for analytics.

## Probabilistic matching versus Deterministic matching models

### *Deterministic matching*

In deterministic (rules driven) matching, a match would be made when a sufficient number of data elements agree between specific attributes or data fields of two records. In the simplest and most restrictive case, all data elements must agree. More flexible rules can allow some pre-defined subset of a record's attributes to 'determine' a match, eg, 'match on at least three of five attributes' or 'match on tax file number, gender, and two of birth year, birth month and first initial'.

### Limitations

Major limitations of deterministic matching are that:

- each attribute is considered to be of equal weighting; in practice, attributes differ in the amount of information they contain about an individual; for example, with deterministic matching, a name attribute would carry the same weight as an address attribute
- real data often contains missing or incorrect data elements, with some attributes coded more reliably than others, ie, data finding and entry issues
- a single miscoded data element within one attribute can prevent a match emerging, even if the evidence for a match, based on other attributes, is perfect
- agreement or disagreement on one attribute provides a fixed 'score'; in practice, different data elements within an attribute have different impacts on a match. For example, when considering the data element 'State' (NSW, Vic, ACT ...), a match is more likely if both records are from NT, as opposed to both records being from NSW. This is because there is a higher probability that two records are from NSW by chance as opposed to NT (a less populated state). The same principle applies for popular versus rare names, and data elements in other attributes. Deterministic matching is, therefore, typically agnostic to the differential weighting of attributes.
- it is impossible to resolve ties that occur when one record matches with two (or more) other records on the same number of attributes without applying an arbitrary preference to the attributes.

These can increase the need for manual resolution.

### *Probabilistic matching*

A solution to the limitations of deterministic matching is probabilistic matching. Here we are not only concerned with how many attributes match, but also which ones, and the values of their data elements. A match on three strong attributes would be accepted over a match on three weaker ones, whereas deterministic matching would have resulted in a tie.

The strength of an attribute is most commonly measured by calculating the amount of information conveyed by the values of the data elements within the attribute. Attributes with many potential data element values, such as birth date or month, usually contain more information than ones with few, such as gender. It is much less likely, for example, that two records selected at random will have the same birthday than the same gender. A match on date of birth, then, is considered stronger evidence for a match than a one on gender, because of the much higher probability of the match on gender being closer to pure chance.

Depending on the type of comparison, probabilistic weights can be either non-specific or value-specific.

- General (non-specific) weights should be based on the agreement / disagreement with a specific data element. For example, using general weights, agreement on birth date may be given a weight of +3.5, while disagreement may be given a weight of -2.7.

## Data matching with IQ Office

- Value-specific weights would be based on the agreement of specific data elements of the attribute being compared. For example, if comparing name initials using value-specific weights, a match of the initial B can receive a different (lower) weight from a match of the initial Z. This is because 'Z' occurs less often as the initial letter in common Australian names.
- In general, rarer agreements should carry higher weights.

Multiple records matching equally are much less likely in probabilistic matching because records would have to agree and disagree on exactly the same attributes and their data elements, not just the same quantity of attributes.

### Why use *IQ Office*?

If you are in a complex, high-volume and cross-platform business environment, *IQ Office* provides a solution for your Information Quality issues. It can include data profiling, standardisation, record matching, de-duplication, geographic coding and address validation functionality, all in an integrated set of components. In other words, *IQ Office* is a comprehensive solution.

*IQ Office* is available in three editions – Address Reference File (ARF), Data Transformation services (DTS) and Enterprise. Only *IQ Office* Enterprise edition contains *IQ Matcher*.

All *IQ Office* components have these characteristics:

- Mature components – not only is *IQ Office* a complete solution, but also each component represents a superior solution in its own right, with the ability to be deployed independently to meet your specific business requirements (such as address validation, geographic coding or unique entity identification).
- Data capture is more accurate – *IQ Office* helps ensure that data captured by users of organisational systems or that already exists within organisational systems is of the highest accuracy. Depending on the edition of *IQ Office* you license from Intech, this is achieved by:
  - validating postal addresses
  - geocoding of address data to pinpoint location
  - confirming that an e-mail address or a phone number is potentially valid
  - suggesting a person's gender based on name
  - potentially identifying multiple instances of a customer within your data holdings.

*IQ Office* does this without requiring you to deviate from your normal data-entry routines, further enhancing the likelihood that data will be captured accurately and quickly.

- Data is more consistent – *IQ Office* helps ensure that data captured by users of organisational systems or that already exists within organisational systems is consistent, complete and, where the data can be validated against a source of truth, also accurate. The ability to ensure that an organisation's data meets minimum criteria, conforms to standards and is ultimately 'fit-for-purpose' is absolutely vital for maintaining efficient and effective customer communications, analysis and reporting.
- Data is 'Fit-for-Purpose' – *IQ Office* helps managers ensure that data captured by or held in transactional and reporting systems is 'fit-for-purpose' and of sufficiently high quality to meet organisational requirements, goals and obligations. This provides a reliable source of information for use in transactional and reporting systems and for customer communication, presentation and other downstream purposes.
- Informed decision making – Decisions can only ever be as good as the underlying data, making data quality a top priority for all organisations. *IQ Office* assists you to gain a higher level of confidence in:
  - the accuracy of the data held by your organisation
  - the analysis and reporting that you perform using that data.
- Native platform support – for each supported operating system. *IQ Office* supports client / server platforms in their native environments (ie, without the need for one platform to emulate another). This includes 32-bit and 64-bit client / server operating systems such as Windows and Unix-based.

- Seamless integration— *IQ Office* integrates seamlessly into a wide range of technical architectures including many operating systems (Windows and Unix-based) with such interfaces as COM / DCOM (Windows only), C, Java, IP Sockets and web services such as Simple Object Access Protocol (SOAP). Sample integration code is available for all of these development environments, and many others.  
*IQ Office* supports connectivity to common database systems, eg, Oracle, Structured Query Language (SQL) and Sybase. *IQ Office* can access the database via Object linking and embedding (OLE) database (DB) connectors including via stored procedures. Additionally *IQ Office* can be integrated into databases such as Oracle and MS SQL for direct access via user-defined extensions such as external procedures and Common language runtime (CLR).
- Faster implementation— the use of existing ‘tried and tested’ software components, with open interfaces, reduces risk or uncertainty regarding implementation timeframe, increases maintainability and durability, and as well enables additional systems to connect to the functionality rapidly.
- Functionally ‘open configuration’ – *IQ Office* allows incorporation of site-specific configuration and rules that let you incorporate your existing data matching knowledge.
- Open data layer - *IQ Office* adopts (connects to / configures to) your data format; it does not ‘force’ data to its own format. This eliminates the:
  - need for software customisation
  - risks associated with transferring site-specific data into different formats
  - need for user training.
- Multi-threading - batch processes can also be configured to make use of parallel processing capabilities of the host environment.
- Scalability—allows an unlimited number of threads (client connections) optimally and simultaneously to handle end-user connections to the data processing engines.
- Stability and reliability - are proven to be exceptional even under heavy load.

The remainder of this document focusses on Data Matching. For overview descriptions of the other software components of Intech’s *IQ Office*, see the Appendix.

## FUNCTIONAL DESCRIPTION OF DATA MATCHING

For Data Matching, Intech’s *IQ Office* product in its various editions, offers these components:

- IQ Standardiser
- IQ Matcher.

These two components of *IQ Office* provide you with an uncompromised solution to your requirement to match entity records. Matching can proceed:

- interactively - a real-time search scenario
- non-interactively - a batch processing scenario or in real-time where a user is not required to interact with the result of the data matching.

*IQ Standardiser* and *IQ Matcher* combine to provide an extensive set of interfaces suitable for batch deployment and real time integration. You can access these interfaces from within your own environment, enabling the *IQ Office* functionality to be used throughout your organisation. Here is a brief overview of these two components:

### IQ Standardiser

*IQ Standardiser* is a high performance data transformation engine designed to parse, validate, enhance and standardise (transform) data of any type into a structure or format that conforms to external (eg, AS, NZS, ISO, or industry-standard reference files) or your organisational standards. *IQ Standardiser* makes use of soft-coded logic files called grammars that define the transformation process. *IQ Standardiser* can, for example, generate phonetic codes such as Soundex, New York State Identification and Intelligence System (NYSIIS) phonetic code, Metaphone or

## Data matching with IQ Office

common alias name lists based on given names. Among other transformations, *IQ Standardiser* performs address standardisation and validation against an Address Reference File (ARF). *IQ Standardiser* accepts unstructured or semi-structured data and returns clean and distinct data elements (attributes and their values) suitable for matching.

*IQ Standardiser* is typically implemented in batch or real-time scenarios where a user is unavailable or not required to interact with the processing in order to influence the result.

### IQ Matcher

*IQ Matcher* is an accurate, high-performance, probabilistic record-matching engine that:

- matches (identifies similarities between) records, potentially across multiple data sources
- detects duplicate information within a data source.

With *IQ Matcher*, large volumes of records can be matched, even where the data elements in the attribute fields vary or contain disparate information (such as misspellings, abbreviations, entity record and element variations and other inconsistencies). To find matches, *IQ Matcher* applies an advanced statistical algorithm to determine the similarity between records and assigns a matching score. The matching score becomes the basis for linking records that meet or exceed nominated Cut-off levels; the matching score for a:

- 'True Match' exceeds the Upper Cut-off
- 'Candidate Match' lies between the Upper and Lower Cut-offs
- 'Non-match' falls below the Lower Cut-off.

These Cut-off levels can be adjusted to meet your requirements.

*IQ Matcher* can integrate enterprise-wide data, while also providing the means to aggregate data quickly and accurately across many data sources.

Thus, the data elements within two records can be compared field to field (within the matching engine); the software will produce elemental scores that it combines as a field score. The sum of these field scores gives the matching score, for you – or more likely, the system - to compare against your required acceptance specifications, ie, your Upper and Lower Cut-offs. In summary, each field comparison gets a field score. The total score of all the field comparisons gives a matching score that tells us how similar the two records are. This allows two records to be matched, even though not all their respective fields are exactly equal.

The example outlined in this document (see Standardising in the Data matching flow process section) provides an indicative solution. It is important to note that *IQ Office* supports a wide range of platforms, interfaces and architectures, and can be deployed in many technical and business environments, different from the example outlined.

### Technical features for Data matching

*IQ Office* has a technical architecture that ensures low maintenance costs and reliable performance today and in the future. Distinguishing technical features include:

- Performance– *IQ Office* is optimized for batch and real-time data lookups that take place at high speed. This includes data parsing, standardising and matching performance.
- Reliable and valid accuracy - By incorporating advanced statistical algorithms, *IQ Office* provides unrivalled accuracy in its ability to parse free-format data, and perform intelligent data matching.

### Key characteristics

- User friendly – *IQ Office* lets you enter the data in your 'standard' application data entry screens rather than requiring you to enter the data into a separate or third-party data-capture screen; this can eliminate training requirements and accelerate the uptake of *IQ Office* by experienced and novice users alike.

- Confidence in results – by including state-of-the-art statistical matching algorithms (based on processes similar to the way ordinary people compare information records), and enabling incorporation of your existing matching rules, you gain a high confidence in the levels of match rates achieved while minimising the risk of your accepting false matches.
- Reduced manual remediation - the statistical algorithm in *IQ Office* offers an accurate assessment of matching probability. This not only reduces false matches, but also decreases the need for manual remediation.
- Architecture for precision customisation - *IQ Office* holds all of its data-specific logic in open interfaces allowing you to tune to meet your purposes – simple or complex. Each component represents a mature solution in its own right, yet is open so that you can deploy and edit ('tune') independently to meet your specific business requirements, such as unique entity identification. This tuning includes:
  - Parsing and standardising logic you can tune to deal with any type of data
  - Validation reference data is open – you can configure *IQ Office* to validate against any set of reference data
  - Match to your own data to prevent and remediate duplication and to provide a comprehensive view of the entities within your data.

## Overview of the Matching process

This section describes the data matching process from a high-level functional perspective – that is, how is it done by specific components of *IQ Office*. The data matching process is described in more detail in the Technical architecture section of this document.

*IQ Office* data matching features include two key processes: Standardising → Matching.

Data standardisation is achieved using Intech's *IQ Standardiser* software component. *IQ Standardiser* will accept structured and unstructured data, and return clean and distinct attributes and data elements suitable for matching.

An entity record will contain data fields (representing attributes, eg, components such as names, addresses, phone numbers or other data) made up of data elements. For instance, a Person record may contain a Name attribute that includes these data elements:

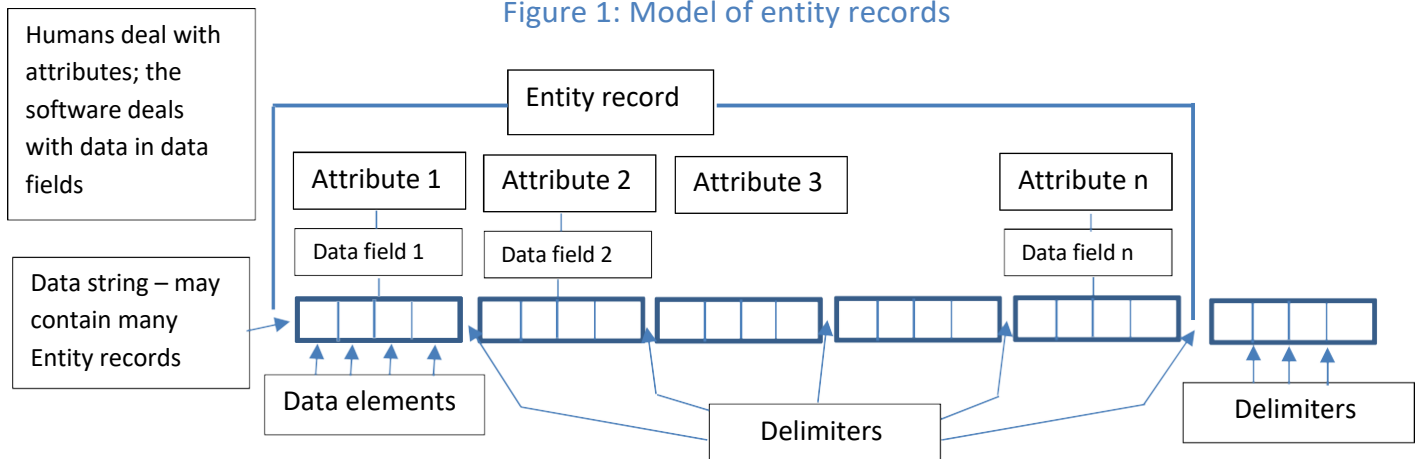
- a title or salutation
- a first given name or initial
- middle names or initials
- a surname or family name.

The title or salutation may include such element abbreviations as: Dr, Mr, Mrs, or Ms.

Further attributes of the person record may consist of an address, and a phone number.

Delimiters (space, comma, semi-colon, Tab) within a Data field can be different from those between Attributes.

Figure 1: Model of entity records



An entity record contains attributes, each of which contains data elements in specific data fields. Parsing each attribute identifies the data elements in an incoming entity record. The definition of field sizes (maximum number of characters) and delimiters (eg, space, comma, semi-colon, or Tab) occurs during standardising. Standardising is the process of transforming the parsed data into a structure that is common across all record data that is to be compared and, therefore, renders it suitable for data matching. Collectively, the processes of parsing and standardising are often referred to as standardising, the end result of which is standardisation.

Data matching is performed using Intech's *IQ Matcher* software component. In comparing two entity records, the software compares field to field; specifically, the comparison is of the data elements within each field. *IQ Matcher* uses a probabilistic data-matching algorithm that calculates the statistical likelihood of two entity records representing the same entity; this provides you with:

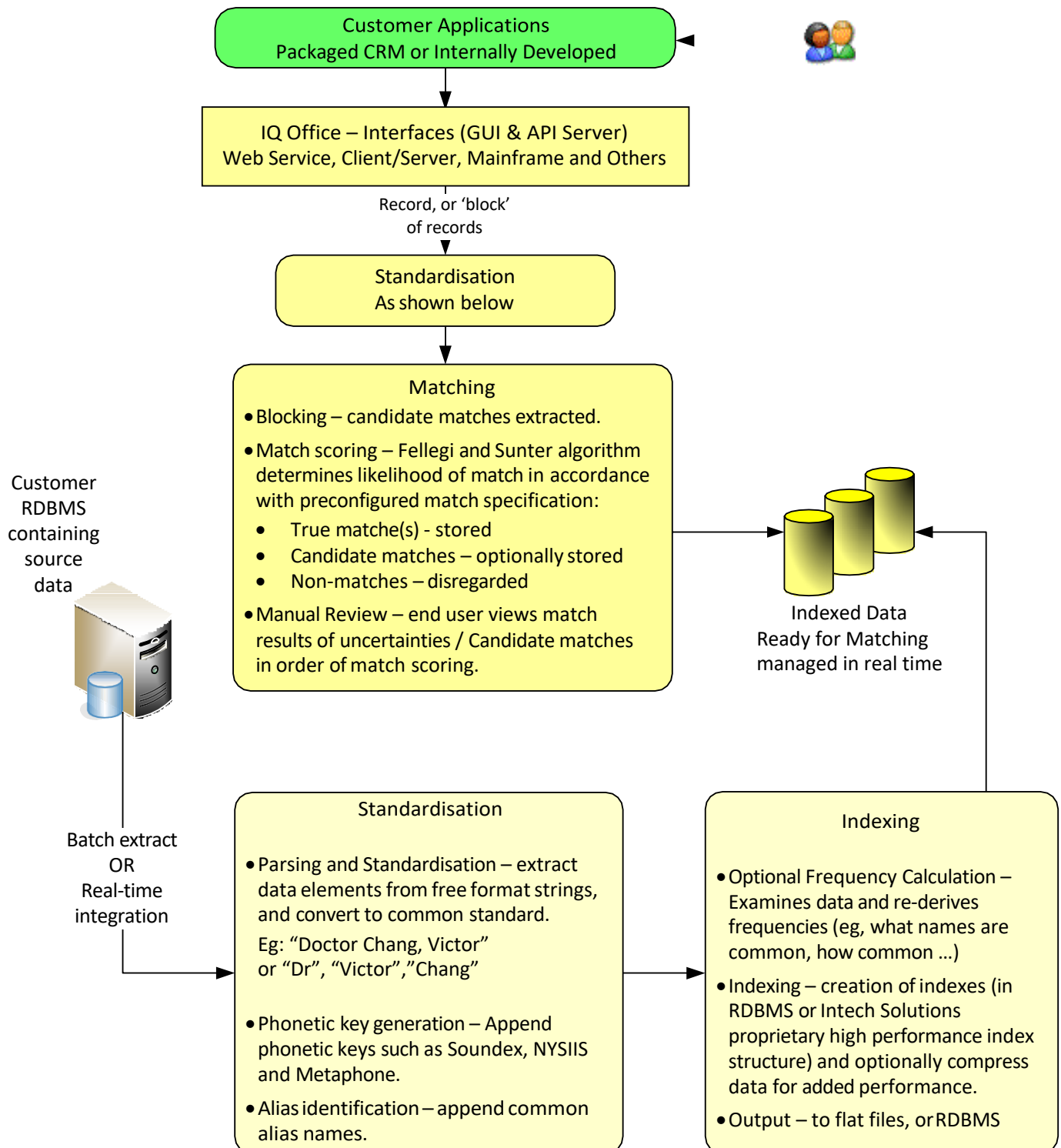
- exceptional confidence in the accuracy of the matching
- high throughput performance
- the ability to customise to meet your specific requirements.

The end result of comparison and calculation is the matching score.

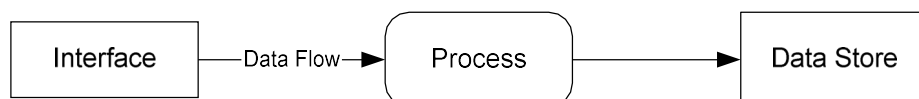


## DATA MATCHING FLOW PROCESS

Figure 2: Data flows between IQ Office and users



Key: The objects in Figure 2 and Figure 3 are:





NOTE – Alternative Option: Figure 2 shows the Indexed Data separately from the source data. As an alternative, *IQ Office* provides facilities to maintain these indices at the source of the data, via automated processes that can operate in real-time, providing enhanced matching that synchronises the data.

The remainder of this section describes the standardising, matching and indexing processes from a functional perspective, in greater detail.

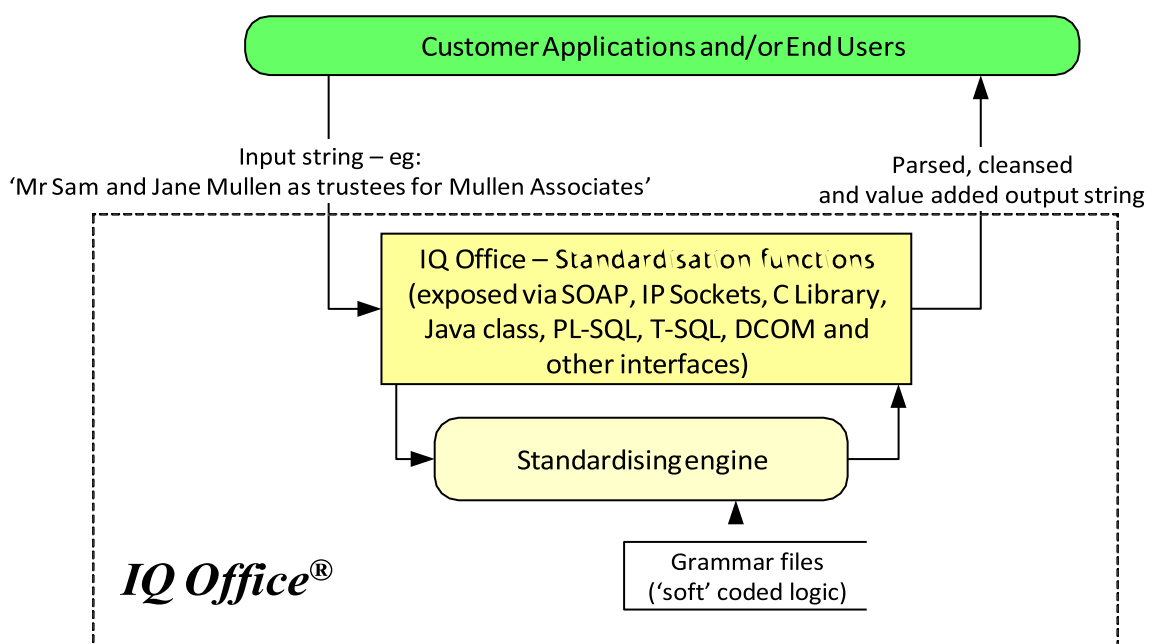
## Standardising

Standardising is the process of:

1. reading a data string, ie, an entity record
2. converting it to a known, standard format
3. identifying distinct data elements suitable for matching.

The standardising process can include name parsing and address validation, generation of phonetic operators (such as Soundex, New York State Identification and Intelligence System (NYSIIS), or Metaphone phonetic code), or conversion of data fields to conform to standards.

Figure 3: Generic standardising data flows in *IQ Office*



### ○ Customer Applications and / or End Users

Customer Applications include real-time data capture applications (such as CRM), middleware applications (such as a web service) or any other batch or real-time process that requires access to data parsing and standardising in preparation for matching and potentially, validation functionality.

### ○ *IQ Office* - Standardisation functions

The parsing and standardising functionality of *IQ Office* ensures that data extracted from all source systems can be passed to the data matching engine, *IQ Matcher*, in a consistent structure. This increases the likelihood of identifying commonalities between customer records or, conversely, confirming that the records being compared are not alike. This, in turn, can give you higher confidence in the matching that is performed.

### ○ Grammar files

These tell *IQ Office* how to parse and then standardise such data as phone numbers, company names, people's names, addresses and many other data types. These grammar files are completely configurable to standardise any type of data. Intech staff can add new grammar files or modify existing ones to meet your requirements.

For example:

- ▶ 'L 7, 35 Spring st, Bondi Junction NSW 2022'
- ▶ '7th fl 35 Spring st, Bondi Junction NSW 2022'
- ▶ 'Intech Solutions seventh floor, 35 Spring st, Bondi Junction New South Wales 2022'
- ▶ 'L 7, 35 Spring Bondi NSW 2022'

refer to exactly the same address. However, as simple string comparisons, they differ. *IQ Office* will parse the input string to break the address into its component data elements (tokenise the string) – level number, street number, street name, street type, and other components included in the input string. All of the example addresses (address attribute data elements) will be standardised to:

Table 1: Data element example

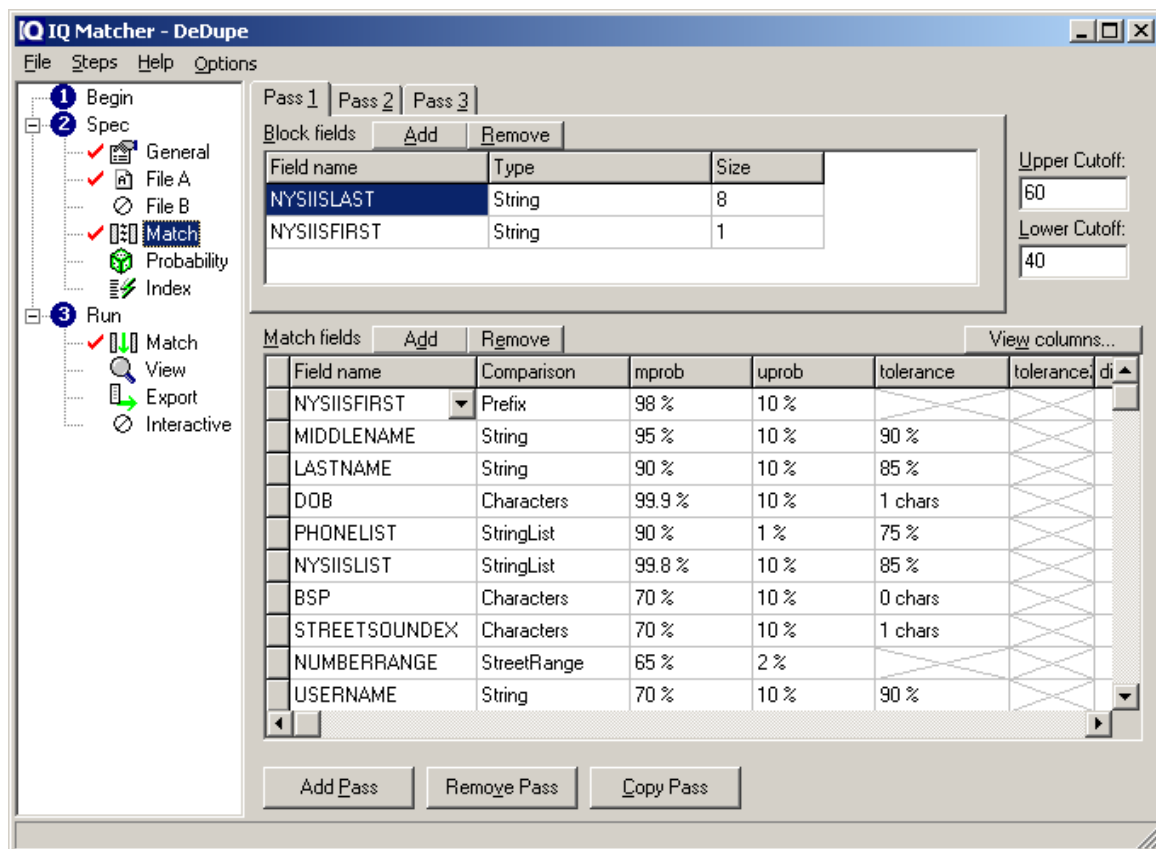
Data element name	Data element value	Data element name	Data element value
Level number	7	Street number	35
Street name	Spring	Street type	St
Locality	Bondi Junction	State	NSW
Postcode	2022		

## Matching

Once the data is in a known and standard format, the next stage is to identify records that may be close representatives of one another. *IQ Office* can identify matches through a unique statistical matching technique that works even when there is no common key. To find matches, *IQ Office* applies a statistical algorithm that compares any number of attributes and data elements within a record, using a scoring system of weights and penalties. For further information on the public domain algorithm *IQ Office* uses, see I. Fellegi and A. Sunter *A theory for record linkage*, Journal of the American Statistical Association.

In comparing two records, the software compares field to field. Specifically, the software compares the data elements within each field. The statistical algorithm automatically calculates a value of the elemental field comparison score (positive for agreement, negative for disagreement); thus, each field comparison gets a field score. The total score of all the field-to-field comparisons gives a matching score that tells us how similar the two records are. This allows two records to be matched, even though not all their respective fields are exactly equal.

Figure 4: Screen shot of IQ Matcher – De dupe



Different fields are compared in different ways; for example:

- Names will be compared for similarity, allowing for common typing errors, so that 'Johnson', 'Jonhson', and 'Jonson' will all be assessed as very similar, and 'Jackson' somewhat less similar.
- Dates and times will be compared allowing close dates to be considered as similar, so that 31 Jan and 1 Feb will be considered similar.

The value of each data element score given (positive for agreement, negative for disagreement) is calculated automatically by the statistical algorithm. These elemental scores are combined to form the field scores that in turn are summed to form the matching score.

At the end of the matching process, we will get a list of records that are similar, ie, matching scores tell how similar they are. The matching score becomes the basis for linking records that meet or exceed nominated Cut-off levels; the matching score for a:

- 'True Match' exceeds the Upper Cut-off
- 'Candidate Match' lies between the Upper and Lower Cut-offs
- 'Non-match' falls below the Lower Cut-off.

In addition to the probabilistic scoring, you can define rules that downgrade the match type into a different match bracket (Candidate Match or Non-Match).

There are more than 20 different standard comparison types, each allowing different types of tolerances to be defined.

For example:

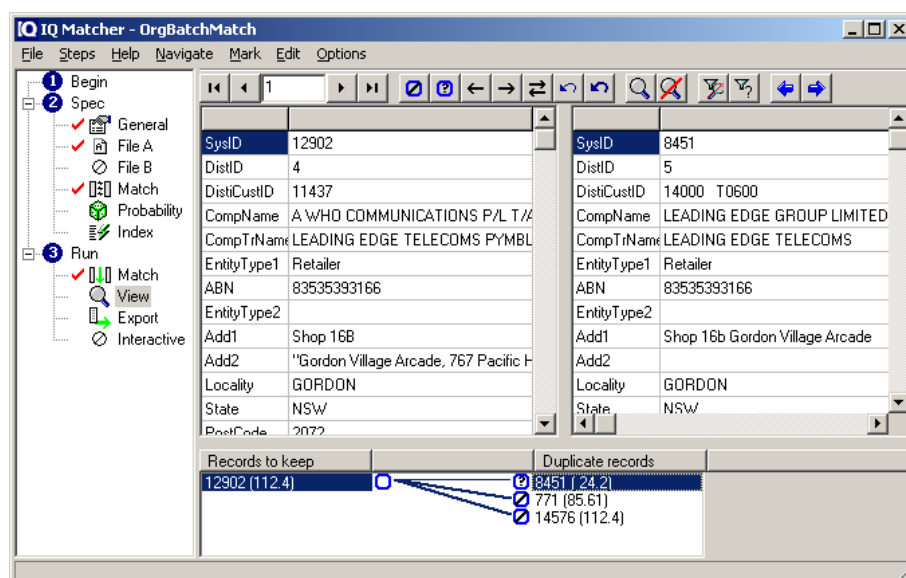
Table 2: Comparison types and tolerances

Comparison type	Definable tolerance	Example or description
String	Percent of the string that must agree.	Tolerance 80%: 'Johnson', 'Jonhson', and 'Jonson' partially agree (partial score)
Date	Number of days out	1/January/2016 - 30/December/2015 Tolerance of 1 → Partially Agree/part score Tolerance of 2 → Disagree/penalty
Character	Number of chars out	
Percent	Numeric percent	Tolerance 20%: 5 and 10 → Disagree/penalty Tolerance 20%: 50 and 55 → Agree Example use: data contains estimate of age
Integer	Absolute value	Tolerance 1 → Numeric values 5 and 7 disagree as difference greater than permitted tolerance of 1, 5 and 6 agree.
Street_Range		12-16 compared to 14 → agree 12-16 compared to 15 → disagree (because opposite sides of road)
Lists		Compares multiple words – good for alias names.
Many other...		

### Manual review

*IQ Office* has a manual review facility that can present an experienced operator / data steward with candidate matches, where an automated match did not achieve a matching score above the Upper Cut-off. *IQ Office* provides manual resolution interfaces natively within the batch GUI, and programmatically for integration into line-of-business applications. Figure 5 shows the manual review interface comparing two records.

Figure 5: Screen shot of *IQ Matcher – OrgBatchMatch*



## Indexing

In this context, Indexing is the process of creating and storing the relevant reference data, (including meta-data) and indices necessary to ensure optimal performance of the matching process. More specifically, *IQ Office* performs:

### *Frequency calculations*

In order to distinguish 'frequent agreements' that may happen by chance (hence attracting low match scores), from 'infrequent agreements' that are not likely to occur by chance, *IQ Office* can calculate, or you can provide it with, the frequency of each element of data. This can be done via an automated process that calculates these frequencies. This process does not need to run repeatedly. It is only necessary to re-calculate the frequencies when the data changes significantly in either volume or content.

### *Compress and index*

This process reads data and creates the compressed and indexed files in a format where the Matching Engine can access them optimally. This compressing and indexing process may be performed by one of:

- *IQ Office* internally – storing the data in flat files, with indices in a high-performance proprietary structure
- staged RDBMS – storing the data and supporting indices in an offline relational database
- production RDBMS – maintaining appropriate summarised data and indices on the production database. *IQ Office* provides RDBMS interfaces to allow these indices and derived data elements to be maintained in real time (via database triggers, and / or procedures).

Note: Indexing is an advantage of the solution in that it is not limited to predefined data structures or a requirement to have access to enterprise-wide data holding.

## Project encapsulation and automation

The above steps – parsing, standardising, indexing, de-duplication and matching - are encapsulated in a 'Project'. A project stores the information that specifies the data source, execution process and output destination.

Several projects can be created, each with different or shared tasks and configurations.

You can run projects manually, where you interact with the software and data, or you can automate projects by using a scheduling program.

### *Supported project types*

#### Matching 1 File - Groups

*Find duplicate records in one file.*

If several records are all similar to each other, they will be marked as a group of duplicates and will be provided with the same SetId in the output.

#### Matching 1 File – One-to-one

*Find duplicate records in one file.*

Each record can be marked as a duplicate of at most one other record on a 'first in, first out' basis.

If three records are similar, the first two will be marked as duplicates of each other.

If, however, four records are similar, they will be marked as matching pairs. For example, record A matches record B, and record C matches record D.

#### Matching 2 Files – One-to-one

*Find matching records between two files.*

Each record in a file can be matched to at most one record in the other file.

## Matching 2 Files – Many-To-One

*Find matching records between two files.*

Each record in file-A can be matched to at most one record in file-B. However, many records in file-A can be matched to the same record in file-B. This type of matching is also called classifying, where file-B is the reference file against which file-A records are classified.

## Matching 2-Files – Many-To-One Unique Best

'Best' is that match achieving the highest matching score.

This is the same as the above type (Many-To-One) with these differences:

- For each file-A record, 'Many-To-One' will find all possible candidate matches in File-B. If more than one is found, it will select one, but you will be presented with all of them, allowing this selection to be modified. 'Unique Best' will only return the best match.
- If a file-A record matches equally well with two file-B records, then 'Many-To-One' will mark one as a match, and the other as an alternative candidate (a second). 'Unique Best' will mark neither as a match.

## Interactively match records against file (One-By-One)

Use this match type if record pairs are going to be compared one at a time such as through a real-time API.

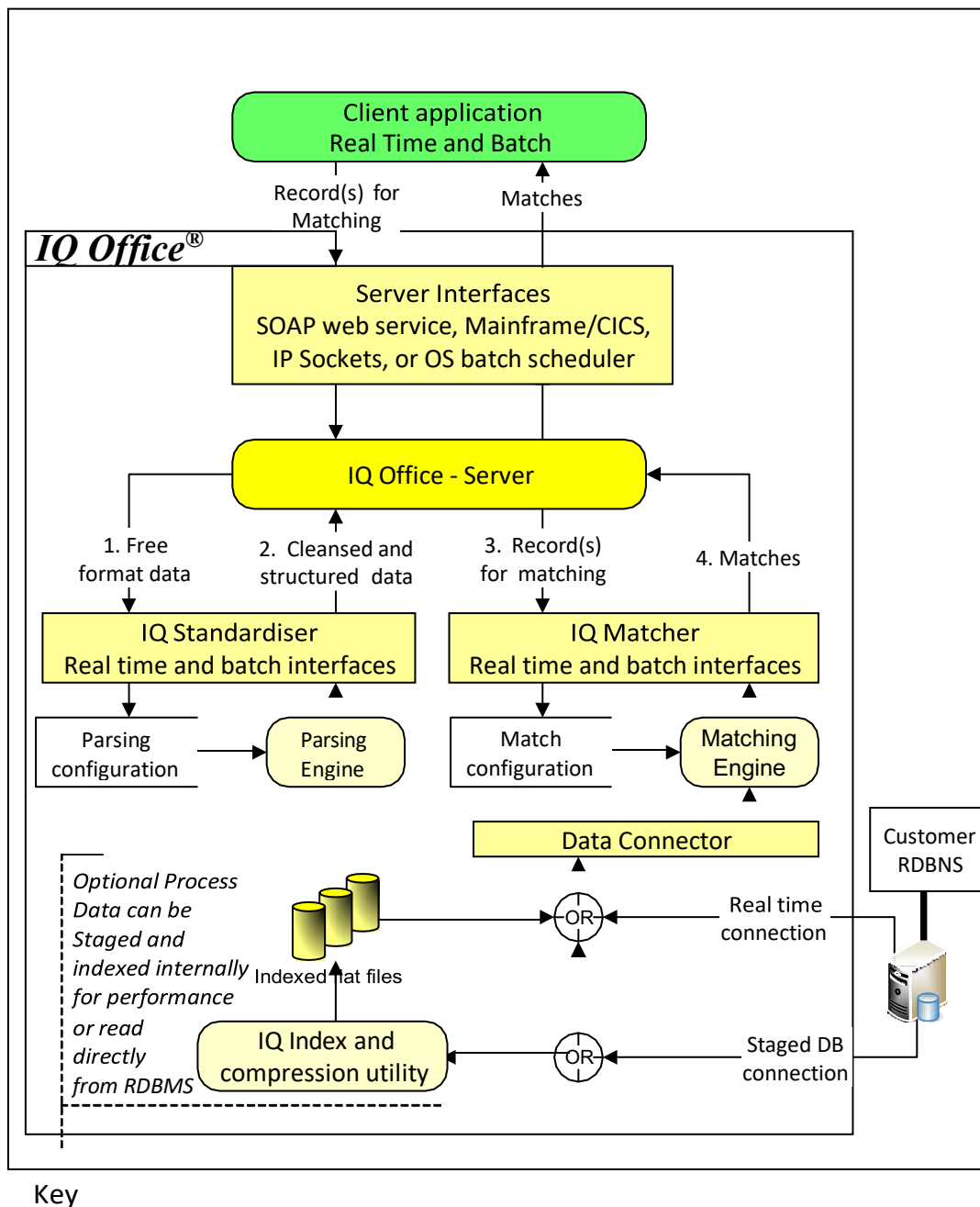
## Interactively match records against file Unique Best (One-By-One Unique Best)

This is the same as the above type (One-By-One) with these differences:

- For each File-A record, 'One-By-One' will find all possible candidate matches in file B. If more than one candidate is found above the Upper Cut-off, it will select one, but you will be presented with all of them, allowing you to interact with this selection. 'Unique Best' will only find the best match.
- If a file-A record matches equally well with two file-B records, then 'One-By-One' will mark one as a match, and the other as an alternative candidate (a second). 'Unique Best' will mark neither as a match.

## TECHNICAL ARCHITECTURE

Figure 6: IQ Office data flow



### How IQ Office works

#### Client Application – Batch and Real time

##### Batch

Entity records are sent to the server for matching. This can happen in several different ways:

- *IQ Office* reads text files from source systems and then sends them for processing by the server.

## Data matching with IQ Office

- An *IQ Office* project gets data from a Relational data base management system (RDBMS) by reading a table, viewing or executing a stored procedure that returns a data string. For example, a view can select entity records that are new, unmatched or flagged for re-matching.

To trigger batch processes, *IQ Office* uses standard scheduling and / or command line execution shells provided by the operating system. Scheduling, coupled with *IQ Office*'s ability to execute via a command line, allows remote activation of batch processes.

### Manual review

*IQ Office* has a manual review facility that can present an experienced operator / data steward with all match results including Candidate matches, where an automated match did not achieve a matching score above the Upper Cut-off – ie, a True match. *IQ Office* provides manual resolution interfaces natively within the batch GUI, and programmatically for integration into line-of-business applications.

### Real time

As you enter an entity record, it is packaged and sent to the *IQ Office* server to perform a real-time matching of that entity record to the entity records stored in the server. A list of none, one or more True Matches is returned, sorted in order of matching score.

### Server Interfaces

*IQ Office* exposes its functionality via cross-platform services accessible across the network (subject to network security). These services (SOAP and IP Sockets) are standard, and are not dependent on any third-party software, such as web servers.

### IP socket server

This enables a process to connect to the server via standard Transmission Control Protocol / Internet Protocol (TCP IP) that are supported by new and older platforms alike. This interface can be accessed directly by a client machine (without the need for any custom software to reside on that machine), or via *IQ Office* client applications.

### Web service – SOAP

This allows loosely coupled processes to access the *IQ Office* functionality via standard web services.

### Java class

The functionality of *IQ Office* is exposed via Java (using the Java Native Interface (JNI)), enabling Java applications to call the functionality of *IQ Office* directly.

## IQ Standardiser

### Interface functionality

*IQ Standardiser* accepts data as free-format text and breaks the data string into its components to populate structured data fields. It does this by using the same logic that you would use when looking at a freely formatted string. For example, when you see the address string '4 Campbell Pde', you would identify that the street number is '4', the street name is 'Campbell' and the street type is 'Pde' (a recognised and accepted abbreviation for 'Parade'). You know this by understanding the grammatical rules you use when you write addresses. *IQ Standardiser* stores and uses these grammatical rules to parse, standardise and validate (transform) data in real time (for data entry systems) and batch modes. This process is called 'standardising'.

*IQ Standardiser* comprises Application Programming Interfaces (APIs) and User Interfaces (UIs).

The *IQ Standardiser API* is a collection of programmable interfaces for Windows and Unix-based operating systems. On client server environments, this includes a COM (Windows specific), Java, C Library, IP Sockets, and web services such as SOAP.



## Data matching with IQ Office

Sample code is available in these languages: Java, .Net, C/C++, COBOL, Delphi, PHP, PL-SQL, T-SQL, asp and VB.

The *IQ Standardiser UI* is a set of applications that allows you (or a batch script) to process data via a feature-rich interface. These applications include:

- Stan console – a cross-platform application for running batch processes from the command line
- Windows GUI – including *IQ Standardiser*
- test applications – source code including COBOL, C, Java, Delphi, VB, and .Net.

These user interfaces allow you - or Intech specialists - to configure, tune and test batch processes in a Windows GUI environment and then deploy either for execution on a different environment (for example, Unix-based operating system) or by an automated scheduler.

### *Grammar files*

*IQ Standardiser* stores grammatical rules for parsing and standardising in a text format called Grammar Files. These form the core of *IQ Standardiser's* capabilities. The software comes supplied with several standard grammar files. You can use the standard grammar files, or get Intech to edit them or create new grammar files. These may include transformations of phone numbers, company names, people's names, international addresses or any other type of data.

For instance, the name parsing Grammar contains instructions that the Parsing engine follows, to:

1. parse the input data string to identify names
2. match a name to a name reference file you choose
3. produce structured output.

### *Parsing engine*

The Parsing engine (also known as *Standardiser Runtime* or *StanRt* for short) is a process that executes the logic in a Grammar file. It may be deployed as a standalone process or in a client / server-like architecture where the server (logical or actual) runs in an environment native to each platform - a Windows Service for Windows, or a daemon under Unix-based operating systems.

The Parsing engine will:

- run all logic on the server
- optimally load a copy of the grammar files, ARF and GIF
- allow simultaneous access by many concurrent client applications (or end users).

This has the benefit of:

- maintaining optimal performance – even under heavy load
- reducing maintenance- by reducing the need to update client machines for data updates.

## **IQ Matcher**

### *Interface functionality*

*IQ Matcher* exposes the functionality of the Matching Engine.

*IQ Matcher* comprises Application Programming Interfaces (APIs) and User Interfaces (UIs).

The *IQ Matcher API* is a collection of programmable interfaces for Windows and Unix-based operating systems. On client server environments, this includes a COM (Windows specific), Java, C Library, IP Sockets, and web services such as SOAP.

Sample code is available in these languages: Java, .Net, C/C++, COBOL, Delphi, PHP, PL-SQL, T-SQL, asp and VB.

## Data matching with IQ Office

The *IQ Matcher UI* is a set of applications that allows you (or a batch script) to process data via a feature-rich interface. These applications include:

- Match console – a cross-platform application for running batch processes from the command line
- Windows GUI – *IQ Matcher* windows application
- test applications – source code including COBOL, C, Java, Delphi, VB, and .Net.

These user interfaces allow you to configure, tune and test batch processes in a Windows GUI environment and then deploy for execution on a different environment (for example, Unix-based operating system) or by an automated scheduler.

### Matching engine

The Matching engine is a process that performs the data matching. Entity records are fed in for it to match, and it matches these to data received from the Data Connector.

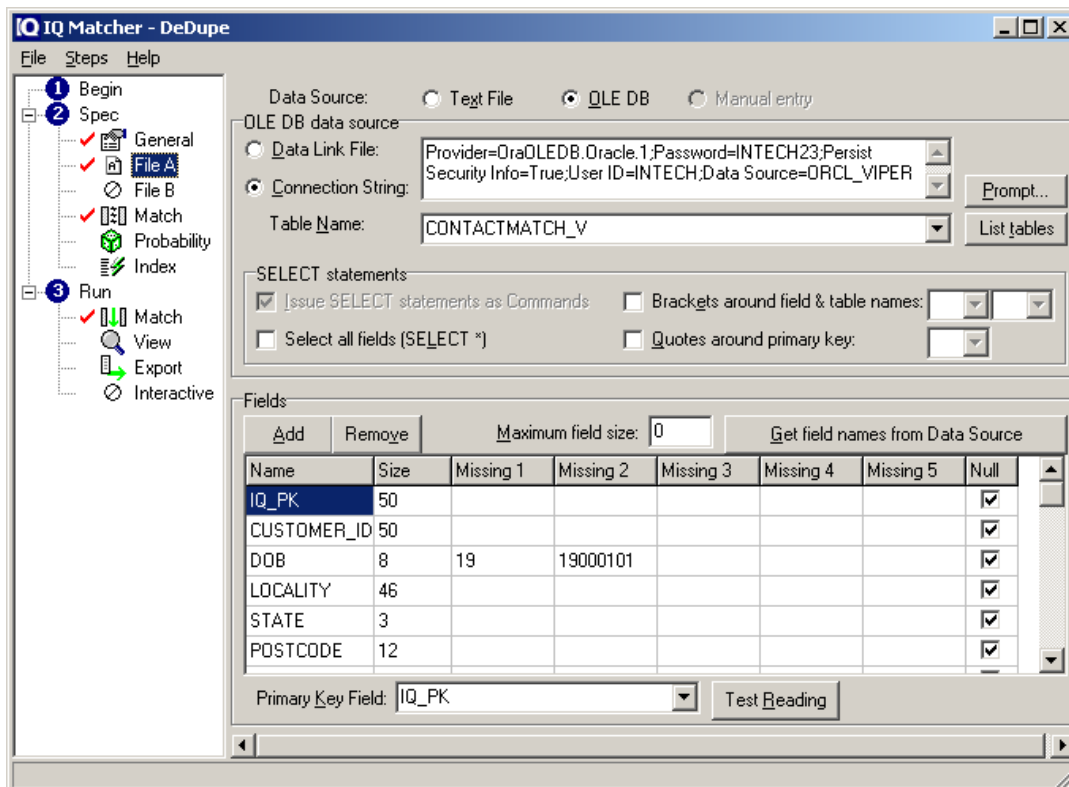
The Matching engine may be deployed as a standalone process or as a client / server application. It shares the same technical architecture and benefits as the Parsing engine.

When connecting to a RDBMS via OLE DB, security authentication credentials may be stored within *IQ Office* configuration set (as shown in Figure 7), or *IQ Office* may pass through the connected client's authentication credentials. This allows access security to be maintained via the operating system.

### Data connector

The data connector is an abstraction layer enabling *IQ Office* to support standard and proprietary data formats. The data connector can receive data by connecting to relational databases via OLE DB, or by accessing its internal data repository.

Figure 7: Screen shot of *IQ Matcher* – Database connectivity



The screenshot shows the 'IQ Matcher - DeDupe' application window. The left sidebar contains a tree view with steps: 1. Begin, 2. Spec (expanded), and 3. Run. Under 'Spec', there are sub-steps: General, File A, File B, Match, Probability, and Index. Under 'Run', there are sub-steps: Match, View, Export, and Interactive. The main window is titled 'Spec' and contains the following sections:

- Data Source:** Radio buttons for Text File, OLE DB (selected), and Manual entry.
- OLE DB data source:**
  - Data Link File:** A text box containing 'Provider=OraOLEDB.Oracle.1;Password=INTECH23;Persist Security Info=True;User ID=INTECH;Data Source=ORCL\_VIPER' and a 'Prompt...' button.
  - Connection String:** A text box containing the same connection string.
  - Table Name:** A dropdown menu showing 'CONTACTMATCH\_V' and a 'List tables' button.
- SELECT statements:**
  - ☒ Issue SELECT statements as Commands
  - ☐ Brackets around field & table names: [ ]
  - ☐ Select all fields (SELECT \*)
  - ☐ Quotes around primary key: ' '
- Fields:**
  - Buttons: Add, Remove, Maximum field size: 0, Get field names from Data Source.
  - Table with columns: Name, Size, Missing 1, Missing 2, Missing 3, Missing 4, Missing 5, Null, and a checkbox column.
- Primary Key Field:** A dropdown menu showing 'IQ\_PK' and a 'Test Reading' button.

Name	Size	Missing 1	Missing 2	Missing 3	Missing 4	Missing 5	Null	
IQ_PK	50							<input checked="" type="checkbox"/>
CUSTOMER_ID	50							<input checked="" type="checkbox"/>
DOB	8	19	19000101					<input checked="" type="checkbox"/>
LOCALITY	46							<input checked="" type="checkbox"/>
STATE	3							<input checked="" type="checkbox"/>
POSTCODE	12							<input checked="" type="checkbox"/>

### *Connecting to internal data repository*

The internal data repository is tuned for read performance, and provides the ultimate in throughput performance. This is because the data is stored and compressed in a read-only format that is optimized for read performance.

### *Connecting to data in RDBMS*

RDBMS connectivity provides greater flexibility in managing the data (because of the open nature of RDBMSs, and its ability to support incremental and / or real-time refresh methods).

Note: RDBMSs are built for transactional environments, compromising read performance in favour of a balance between read and write performance. While an RDBMS provides good read performance, it is slower than an internal data repository that is optimized for read performance. This is related to RDBMS technology rather than being specific to *IQ Office*.

## APPENDIX

Various editions of Intech's *IQ Office* include these additional components:

- IQ Rapid Address
- IQ Profiler

### IQ Rapid Address

*IQ Rapid Address* is a deterministic address-matching engine that matches address data to an ARF. In addition to address validation, it provides powerful and accurate geographic coding capabilities, ie, given an address, it can return a latitude and longitude accurate to within a few metres.

### IQ Profiler

*IQ Profiler* is a data discovery engine specifically designed for large volume datasets. Using *IQ Profiler*, you can examine large amounts of data and discover data structures, frequency distributions and various data anomalies, all via an easy-to-use interface. You can define complex rules and examine the data for compliance.