



MDM for CRM - A Test Approach

Executive Summary

Every industry requires a cost-effective solution to manage customer information and ensure correctness of data. Applications for customer relationship management coupled with master data management (MDM) tools can be used to implement rules and processes for cleansing, standardizing, de-duplicating and enriching data. Such rules and processes are effective and can also handle exceptional situations while processing customer information within an MDM hub. Testing these implements helps to realize the objective. Data from external sources needs to be in the required structures and formats. Rules for de-duplication need to be effective in removing redundancy. The test scenarios include situations where MDM is to be customized to handle any exceptions where available customer information cannot be treated by existing rules and processes for cleansing, standardization de-duplication and enrichment.

Scope

Customer information such as identity attributes, demographics and communication attributes captured in CRM and other sources can be transmitted to the MDM hub for a round of processing in order to de-duplicate, cleanse, standardize and enrich it. To check if the MDM hub is configured as per requirements of the CRM integration, the data passing through the MDM hub needs to be tested at different stages of the purification process. MDM essentially involves processes for loading, changing and filtering of data based on rules that define the filters and dominance of one value over another.

Raw data loaded onto an Extract-Transform-Load (ETL) layer where transformation logic is defined to change the data into MDM-acceptable format. Data from ETL is then pushed into LANDING tables of MDM and are cleansed, supplemented

and standardized based on pre-defined rules. Non-conforming data is rejected -- this data can be sent back to the source or used after analysis and relevant design changes. Records failing integrity constraints inside the MDM hub are also rejected. Conforming customer data is passed into the MDM store where they are de-duplicated and enriched by merging with trusted values from other source records for the same customer. A copy of data loaded and processed within MDM is saved in "RAW" tables, "CROSS REFERENCE" tables and "MERGE" tables for future trace backs.

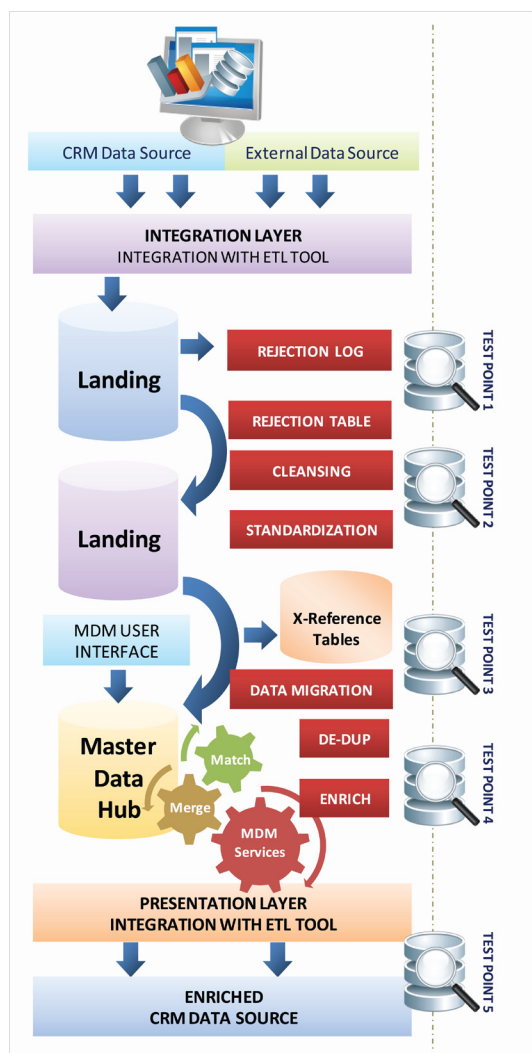
Testing the MDM configuration aims to verify:

- If the ETL layer is able to accept, change and fit incumbent data into placeholders and also handle exceptional situations like sudden design change of feeds.
- Data with higher precision is rejected and if the LANDING is designed to reject records without mandatory field values.
- During the cleansing process, data is checked for junk characters, formats that do not conform to data type conversion requirements, handling at check constraints and conditional filters and de-duplication of records.
- Standardization processes are deployed for proper type casting of data.
- Data is modified or supplemented wherever possible in order to correct the format.
- Data is modified to make it conform to defined standards.
- Referential integrity is maintained at the MDM store.

- Rules for de-duplication and data enrichment are effective.
- Data published by the MDM hub is structured as per requirement so it can be uploaded into the CRM database.
- All data processed by the MDM hub can be traced throughout for audit purposes.

Process

Inbound data to be processed by the MDM system arrives from different data sources both internal and external, including CRM sources. They may come as XML files, some as CSV or encrypted files, or even as fixed width flat files. What's more, values of an attribute may have different formats for different sources. As an example we can think of the attribute "Date of Birth" of customers. The format can be written as "31st January, 1968" for one source, whereas it can be "31-Jan-68" for another.



This heterogeneous population of attributes from "Customer Name" to "Customer's Privacy Preferences" needs to land within the MDM tables in the same type of structure and in pre-selected data formats. A very pertinent question is how the system will react if the structure of a text feed or XML gets changed before relevant modification of the placeholder in ETL or LANDING? It needs deep domain knowledge to test and verify that any conversion process deployed is fully effective and is designed to use a defined line of treatment for all types of exceptions. This conversion interface, which in most cases is an ETL layer, should be testable, even in absence of records that reflect real occurrence of exceptions.

After the integration layer, the data is pushed into the first landing zone in the MDM application. Usually, the data comes in text format only. So, like "Customer Name", all attributes -- "Date of Birth", "Identification Number", "Zip Code" - also arrive in text format rather than number. Data entering into LANDING may be from different countries. The column for "Postal Code" may be designed for a particular length considering a set of countries. However, if a new record comes in with a new value for country and a higher precision for the postal code, how will the system react?

Data Tracking

The system is to be tested at this point to verify that such exceptional values in the incoming records are automatically rejected. These can be used as input for future enhancements of data structures of MDM or to send feedback to the respective source system on the allowable countries and postal code formats. The LANDING zone can also be tested to reject records where mandatory data is missing, though such checks are mostly handled at the ETL layer, in case of a thin ETL, the LANDING zone is the next point where this verification can be done.

In the MDM store, processes and rules are configured to create a de-duplicated and enriched copy of the data. This is done by merging a set of matching records. Question is whether the data in such records are ready to take part in the match and merge process. Has the data been cleansed? How can we say that a solitary record that does not merge with any other is qualified to be in MDM hub? What are the areas we need to test to ensure this?

Data present at LANDING is in text format. Attributes such as telephone numbers and fax numbers in LANDING, may contain junk characters.

Is data available at Source?	Placeholder Created?			Treatment Area									
	ETL	MDM	CRM	ETL	LAND	STAGE	CLEANSING	STANDARDIZATION	LOAD	AUTOMERGE	MANUAL MERGE	PUBLISH	CRM
Available	Yes	Yes	Yes										X
Available	Yes	Yes	No									X	X
Available	Yes	No	Yes			X							X
Available	No	Yes	Yes	X	X								X
Available	Yes	No	No			X							X
Available	No	Yes	No	X	X								X
Available	No	No	Yes	X	X								X
Available	No	No	No	X	X								X
Not Available	Yes	Yes	Yes	X	X								X
Not Available	Yes	Yes	No	X	X								X
Not Available	Yes	No	Yes	X	X								X
Not Available	No	Yes	Yes			X							X
Not Available	Yes	No	No	X	X								X
Not Available	No	Yes	No			X							X
Not Available	No	No	Yes									X	X
Not Available	No	No	No										X

Attributes of customer, address, license and other entities are to be converted to relevant data types before sending to the MDM store. Testing is required to verify the treatment for such data if they are not in appropriate format for type casting - that the format wherever relevant is changed and otherwise rejected by the system. A monetary attribute of the customer at LANDING may contain the value -- "\$5004.9683", where the allowable length of text is sixteen. At the MDM store this must go with a currency data type rounded to nearest USD. Is the system equipped to change the above value to "5005" before type casting? Again, if the value comes in as "\$50.983", will the system reject the record and also handle child entities to ensure referential integrity?

Often two or more records arriving in the same batch may have customer details for the same customer number. The system must be checked to ensure that it is designed to keep the latest and reject the old record, thereby reducing redundancy.

Treatment of a field value inside the MDM may be based on the allowable range of values. Say, the "Last Updated Date" for the record may be absent or the "Account Opening Date" may be in future to the "Last Updated Date". Does the system cleanse such records or reject them?

A requirement can be such that only license records with the value for "Country" meaning "USA" are to be processed. Testing and extensive test data preparation is needed to verify filters and their functions to ensure requirements of the type.

In addition, treatment based on data validations by external tools can also be tested along with the above cleansing requirements.

After cleansing, data also needs to be standardized. Primarily, data type casting is required for many fields. Secondly, the value of an attribute may need to be in a pre-defined format based on the geography, any change in conventions, social norms, customer type or even custom requirements. As an example, upper casing of all text fields like customer names, address lines, may come as a custom requirement. Social norms may require all names to be suffixed by the word "san." Again, with geography there can be dependencies -- records with "Country" as "USA" - which will have a five-digit value for "Postal Code" whereas for "India" this will be six digits. If this is not taken care of in the ETL layer then it can be done now.

Now, after the data enters the MDM store, what happens if a customer's profession or name prefix is not among the values already listed in the system? How about address records entering without a parent customer record? Testing can verify the existence of proper treatment to handle such deviations. Thus, referential integrity is again ensured.

Source records entering the MDM hub must leave footprints on cross-reference tables. So, future data loads can first check if the same record was inserted before. And if so, can either update the old record with new values or insert the new record.

In the MDM hub, a match process is used to identify and group a set of records that match by a rule. These records are merged in order to de-duplicate. While merging, the most dependable value for an attribute is chosen from the set of records, thereby enriching the data. A plain scenario is where two address records flow in from two sources. Testing can determine if the trust setup helps point out the new address as expected or pushes the old address as correct information instead.

Suppose three records for a customer are coming from three sources -- banking, medical and insurance. They all have the same attributes. Now, for the value of the attribute "Gender", the record from medical source can get preference. Again, the value for the field "Is Insured?" from the insurance record needs to survive. How can you ensure that proper trust for an attribute is assigned to the proper source? In another scenario, a match rule is created to identify and match records with a similar customer name and address. There may be two records from two sources for the customer name "Mr. Charles Brown" with an address of

"11, Main Street, New Brunswick". Per the design, these two records will be merged by the system. In reality, there may be two persons with the same name, living in the same building. Only with MDM testing and extensive test data preparations, can one identify such design fallouts.

While merging multiple records, the winning value for an attribute can be one where the respective record source comes with the highest trust for this attribute. Let us look at a situation where two customer records are coming from source "A" and source "B" and the highest trust for the attribute "Age" is given to source "A". Thus while merging; the age is taken from the record of source "A". What if for a particular customer, the value from source "A" is "1" and from source "B" it is "41"? Such situations can be identified by MDM testing and help redesign or downgrade trust from case to case.

The de-duplicated, cleansed, standardized and enriched data is published as data files from MDM store to be loaded into CRM database. If the existing data for such customers in CRM is not purged before upload, reports and screens may show multiple instances of the same information. Testing is also required to ensure that data published by MDM has all the mandatory values required at CRM and also is in expected format.

A customer address in the CRM system can be "11, Main Street, New Brunswick, NJ 08520". After this is standardized at the MDM level it may come back as "11, Main Street, East Windsor, NJ 08520". Testing is required to ensure that before the correct address from the MDM hub is uploaded, the old data is purged or archived. Otherwise, users may see multiple contacts on screens and reports. Moreover, snail mails may still end up at New Brunswick instead of East Windsor.

Finally, an audit trail is required for such rule-based data processing at multiple stages. At any of these stages a record may fail and be rejected or may finally get merged into another record because of insufficient configuration of match rule. If the customer record gets rejected, it will not land up in the final set of customer records in the MDM data store. On the other hand it may get merged when it should have survived alone. Here, every detail, carried by this record may not find a place in the merger and thus is effectively rejected. Thus part of customer information does not find a way into the MDM hub. Information that never gets inside the MDM hub, will never be published to outbound applications. Customer information not available with business application will never be considered for business processing.

Thus a customer gets lost in stealth. One has to check that the system is well equipped to save the footprints of every record to trace back and rectify it. Before entering all records into the MDM hub, they need to be saved for cross reference so that they can be traced back to the source. This also prevents multiple inserts of the same record. These cross-reference tables can be tested to reconcile with the original raw data and the final gold copy of data in MDM that comes with a new identification number.

Another situation is mandatory data may or may not be available from all the sources whereas extra information may be available from some sources. Is the system designed to treat these variations? The MDM tables may be equipped to accept a new product code but not the CRM tables. Is there any way we can foresee this gap? The table in the next page shows where to check for the data rejection or acceptance in such situations.

Testing

As we have seen, MDM testing involves running many batch jobs for the purification process as well as to transfer data between intermediate tables. Running every individual batch job for every test case will incur a big latency with minimal achievement. So, all test cases acting on the same entity groups or aligned to the same functional path are grouped into a single CRM-MDM test set. All the test cases in a set are executed in unison from event to event in the MDM process flow. Enumerated below is a typical flow for testing such a set of test cases.

Report Log Generation

- Test data such as Customer Name, Customer House#, Street Address, Zip, Phone #, Email Address are created from the CRM screens in user mode and per the test objectives for the specific set of test cases. One needs to have login and user rights in the CRM application to prepare the data..
- This data is then exported and loaded into intermediate tables that are typically referred to as "Y" tables - structured as per individual CRM tables. So, if there are five CRM data tables then there will be five "Y" tables with the same structure.

- Data from Y tables are grouped and populated into the next set of intermediate tables, say the "Z" tables. The Z tables carry the structure of MDM LANDING tables; there should be one Z table for every individual LANDING table assigned to hold CRM data.

In the next step, data from the Z tables is pushed into the LANDING tables by batch jobs or the ETL tool; the exceptions are recorded in a rejection log.

Test Point 1

At this point observations are recorded on the following checks:

MDM rejects data with higher precision or whos NULL when a value is mandatory.

- The LANDING tables are the first to accept records comprised of fields having only text data type. The sample data set uploaded for testing was prepared with records having at least one value for each text field which surpasses the precision accepted by the corresponding table-column in the MDM data store. One record is prepared for each such case. Again for each field where a value is mandatory in the hub store, a test record is provided with a NULL value for the specific column. The tester checks the Reject Log generated at this point to see if all the above records are reflected therein. This is to ensure that the LANDING tables are configured as per the structure of final data tables in the MDM data store.

Data Quality in LANDING table

- The sample records that were successfully uploaded into LANDING tables are exported into spreadsheets or queried from the LANDING tables using SQL utilities such as TOAD. These records are then compared with their original values on the CRM screens. This is to ensure that no data gets inadvertently modified by the integration layer between CRM and MDM LANDING tables.

Standardization & Cleaning

- Batch Jobs, or commands, are next executed to push the data from LANDING tables into staging tables. During this process data that cannot be inserted is transmitted to rejection tables. The valid data is then standardized for Address using Trillium.

Test Point 2

At this point observations are recorded on the following checks:

MDM rejects records with special characters; non-conforming data types and field values that are out of bound.

- The staging tables are comprised of fields with data types as in the final tables of the MDM data store. Sample data set was prepared with records having data with embedded special characters; column values pertaining to data types other than what is in design for that column in the store and also values for number and date columns that should fail the check constraint as per the MDM configuration requirement. One record is prepared for each such field. The tester checks the rejection table for each staging table to check if all the above records are reflected therein. Correct values should only proceed to the next stage of MDM processing.

MDM converts data standard format.

- In the case of special data types such as currency or date, correct data that goes into staging can be formatted as per the requirement during the insert. The sample data inserted into staging is checked to see if such fields have been converted or formatted as per the requirement.

Standardization Tool plugged into MDM updates relevant data like Address Line or Zip Code with the most recent and correct value.

- In few cases, the information collected may not be accurate. Like a customer address fed into the CRM may have wrong street name; the last digit may have dropped off the house number; or maybe the zip code was not available.

Such information is not only incorrect but also incomplete. Standardization tools such as Trillium are plugged into MDM right before this stage to correct the address. For example say the test data was prepared with incorrect and incomplete address values for customers. Such records are then checked at staging tables to confirm if the standardization tool was able to correct these wrong or incomplete address values.

Data Migration to MDM Hub

- A data migration is next executed to load the records further into the core tables of the MDM hub. Here, while these records are usually given unique attributes which tie them to the hub, a copy of the old values is kept separate for future reference.

Test Point 3

At this point observations are recorded on the following checks:

Data Integrity in the core tables of the MDM hub.

- Data is now within the realm of core MDM services. Every entity whether master or child is saved in its respective table. Customer records are inserted in the table reflecting each customer's attributes; for example, the customer's address resides in the address table. So, first there is an entity hierarchy. The relationship between customer and his address or customer and his phone numbers can reside in a separate relationship table or the address table itself. Now, an instance of a child entity, such as address, cannot reside in the MDM hub without an instance of the master entity -- the customer. This defines the most important integrity within MDM -- the referential integrity. Secondly; an attribute of an entity may only belong to a predetermined set of values. During data migration from stage to the MDM store, records that fail primarily due to these integrity rules are rejected and sent to the rejection table. Test data was created with records lacking such integrity and were expected to land up in the rejection table at this stage. So, here a snapshot of the rejection table is queried by SQL or exported into a spreadsheet to check all such test records created did really arrive. This will ensure that the MDM hub is properly configured to maintain data integrity.

De-Duplication & Enrichment

- If Match & Merge is required to be run, a request is placed with the data team to schedule the same.
- Execution then resumes after the data team confirms the execution of Match & Merge.

Test Point 4

At this point observations are recorded on the following checks:

MDM is properly configured to identify actual matching records.

- Records transmitted from the CRM system or other sources may bring in multiple instances of the same entity. The instances may have some variations in their attribute values. These are to be merged by MDM into a single instance. As an example: "John May | 34 years | Male" and "J. May | 34 years | Male" may refer to the same person. The MDM hub should be well configured to decide if that really is the case -- it can refer other attributes also. In case this is true, MDM identifies this duplicity termed as matching and assigns a match token number internally to each such record. Similar test data scenarios are created to test the match functionality at this point. After the match process tester queries the MDM system table in conjunction with data tables at the hub store to check if all the duplicate test records were identified by MDM and assigned a match token. If configuration passes this check it implies duplicate records which originated from the CRM data store; other internal systems and external data sources can then proceed to be merged for de-duplication and enrichment.

MDM is properly configured to merge records and trust data from the most trusted source for an entity or attribute during this merger.

- The objective is to check records previously consolidated into individual match sets have been merged into a golden copy. Also, the value of every individual attributes therein should have been picked from the most trusted source. A copy of the matched record sets was saved in spreadsheet marking the most trusted value for each column as per requirement.

This is compared with the merged records seen by querying MDM tables. Here, the tester will check if every matched record set has been merged into golden copy per the design and also each golden record was created with the trusted values marked previously in each match set.

CRM Imports Data Published by MDM

- In certain cases records are edited at the MDM user interface to check whether it is finally reflected in CRM when the enriched data is transferred back.
- A request is placed to the data team to publish the data. The execution then resumes only after data team confirms a publication.
- Published records transmit to CRM end again as data files.
- On request the administrator at CRM executes relevant CRM process to consume the enriched records into CRM database.
- The transformed test data is scrutinized from screens of the CRM application.

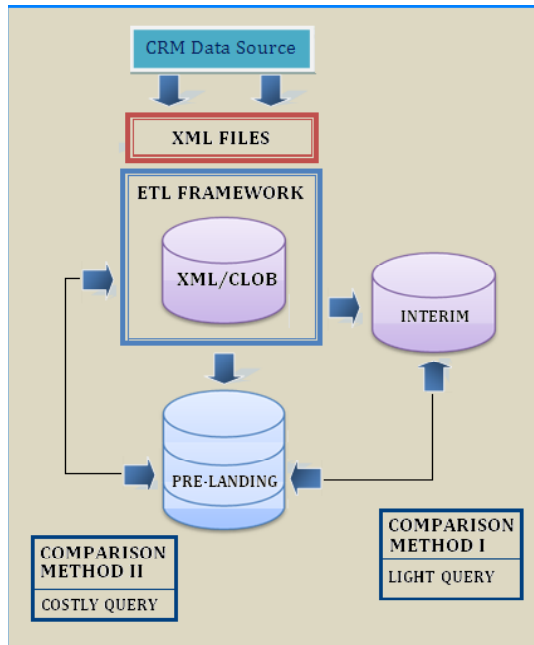
Test Point 5

At this point observations are recorded on the following checks:

CRM shows all the data published by MDM after the purification process.

- The objective here is to check if the golden copy of data generated at the MDM hub is properly transported by the ETL presentation layer to the CRM data store without any change. The set of transformed test records is taken from the published file and compared with records queried at the CRM screens. Say, details of a test customer are fetched on the CRM screen and then his address is compared with the standardized address in the spreadsheets.

XML Data Sources



A source may publish data in XML format to be consumed at the MDM gate. As a general method of the pre-MDM phase, such XML is first fed into an ETL framework, where relevant information is extracted and loaded onto pre-landing tables before they get transported to the first set of LANDING tables in the MDM tool. Here, the goal is to check if the XML data is properly transformed to the pre-landing format.

- Extract the XML data into ETL tables.
- Transform the same to the pre-landing format using transformation routines created by any method but the ETL in actual design. Usually, procedures and queries created with implicit XML functions from the underlying database are used for this purpose.
- First, all XML files are stored in a database table, as XML tables.
- Next, a couple of approaches can be used to compare pre-landing data with XML data, the method can be chosen depending on the bandwidth available and necessary.
- Comparison by Method I has a high design effort but low execution effort. This can be achieved by extracting data from XML tables to an intermediate table containing standard data types using a stored procedure for transformation and load. Then, compare the intermediate data with pre-landing.

- The alternative has low design effort but high execution effort. Here, the data in XML tables is directly compared with pre-landing data, using costly SQL queries.

Challenges

Challenges in this type of testing come in varied flavors. Some of the major challenges hover around the MDM hub and its operations. There are many small but potential hurdles that also need to be accessed in order to align expectations with reality.

Major Challenges

- Preparation of test data.
- Access to relevant database objects.
- Understand third-party cleansers and plug-ins.
- Concurrency issues at Load & Match-Merge may necessitate a tester to await his/her turn to trigger such processing at the MDM hub.
- Limited access to MDM processes or screens.
- Awaiting initiation and completion of critical processes like match-merge and publishing by the MDM administrator.
- Performance of the MDM tool itself especially in loading and merger of complex entities like addresses or extended product information.

Generic Challenges

- Data availability from an external source vendor.
- Understanding and accommodating frequent changes in the requirement.
- Dependency on enterprise messaging systems for data transport.
- Inter dependencies among test cases due to sequence of CRM/MDM processes.
- Unavailability of data file spec slows down data preparation.
- System unavailability due to shared activities contributes to almost 20% delay time.
- System instability due to parallel implementation.

- Lack of real-time support during afterhours narrows down the execution window.
- Low throughput on remote connectivity.
- Delayed defect closure cycles causes almost 40% delay time.
- MDM job runs may not be allowed at times in the case of parallel implementation.
- Rolling resource forecast.
- Requirements flowing in at execution phase.
- Absence of project wind up guidelines.

Final Notes

Testing CRM/MDM integration requires careful planning and preparation of test data. It is more convenient to use sample data from existing systems after modifications, in line with the requirement. While planning test case executions, consider conflict management with MDM batch jobs that use common MDM resources.

About Cognizant

Cognizant (NASDAQ: CTSH) is a leading provider of information technology, consulting, and business process outsourcing services. With over 50 global delivery centers and approximately 88,700 employees, we dedicate our global technology and innovation know-how, our industry expertise and worldwide resources to working together with clients to make their businesses stronger. A member of the NASDAQ-100 Index and S&P 500 Index, Cognizant is a Forbes Global 2000 company and a member of the Fortune 1000 and is ranked among the top information technology companies in BusinessWeek's Hot Growth and Top 50 Performers listings.

Start Today

For more information on how to drive your business results with Cognizant, contact us at inquiry@cognizant.com or visit our website at: www.cognizant.com



Cognizant | **CSP Testing**
Passion for building stronger businesses

World Headquarters

500 Frank W. Burr Blvd.
Teaneck, NJ 07666 USA
Phone: +1 201 801 0233
Fax: +1 201 801 0243
Toll Free: +1 888 937 3277
Email: inquiry@cognizant.com

European Headquarters

Haymarket House
28-29 Haymarket
London SW1Y 4SP UK
Phone: +44 (0) 20 7321 4888
Fax: +44 (0) 20 7321 4890
Email: infouk@cognizant.com

India Operations Headquarters

#5/535, Old Mahabalipuram Road
Okkiyam Pettai, Thoraipakkam
Chennai, 600 096 India
Phone: +91 (0) 44 4209 6000
Fax: +91 (0) 44 4209 6060
Email: inquiryindia@cognizant.com