

Using the Data Quality Accelerator for Crisis Response

Abstract

The Data Quality Accelerator for Crisis Response is a set of preconfigured rules that accelerate your implementation of a data quality process in healthcare and crisis management scenarios. Import the rules into your Informatica environment to analyze, validate, and improve the accuracy of your data, enabling more reliable decision making and better tracking of patients and healthcare issues.

Supported Versions

- Informatica Data Quality 10.4.0 and later

Table of Contents

Introduction.	3
Deploying the Crisis Response Accelerator.	3
Accelerator Components.	4
Before You Import the Solution Files.	4
Importing the Accelerator Files.	4
After You Import the Files.	5
Verifying the Project in the Model Repository.	5
Installing Reference Data for Address Verification and Identity Matching.	6
Updating Reference Tables with CPT and ICD-10 Data.	6
Strategies for Using the Crisis Response Accelerator.	7
Creating a Strategy: Defining a Sequence of Rules.	7
Applying Rules to Your Data.	8
Crisis Response Accelerator Rules.	8
Address Data Cleansing Rules.	8
Contact Data Cleansing Rules.	9
General Data Cleansing Rules.	12
Healthcare Rules.	16
Matching and Deduplication Rules.	17
Appendix 1: Installing Reference Data for Address Verification and Identity Matching.	18
Downloading the Reference Data.	18
Installing the Reference Data.	18
Configuring the File Properties on the Content Management Service.	19
Appendix 2: Updating CPT and ICD-10 Reference Tables.	19
Downloading the ICD-10 Data.	20
Downloading the CPT data.	20
Updating the Reference Tables.	20

Introduction

The Data Quality Accelerator for Crisis Response can help you meet the challenges that medical crises and emergency management bring to data handling and decision making. The accelerator adds business rules to your Informatica environment that speed up data analysis, improve data quality, and enable you to track patients, diagnoses, outcomes, and healthcare issues.

The rules address the most common data quality issues, with a focus on issues that can arise in healthcare and patient management. Use the rules to quickly implement a data quality strategy that can make your data more reliable and useful. The rules are ready to use out of the box. You can customize the rules to suit your data requirements.

The Crisis Response Accelerator includes rules that perform the following tasks:

- **Discover**
Discover facts about your data - for example, determine the levels of completeness in your columns, and establish the conformity of the column data to the structures and types of data that you expect.
- **Standardize and cleanse**
Standardize the form and structure of common data values, such as dates, telephone numbers, Social Security numbers, and country identifiers. Additionally, standardize the use of character case and diacritic characters in the data. You can also remove extraneous symbols, characters, and character spaces from the data.
- **Calculate and identify**
Calculate and derive a range of facts from your source data, including patient age, gender, time elapsed since diagnosis or other milestones, physical distance from a given location, and presence within a given target area.
- **Parse**
Parse important data values from fields that contain strings or multiple values. For example, parse telephone numbers, CPT codes, comorbidity factors, Social Security numbers, and healthcare facility types from source data fields, and write each type of value to a discrete new field.
- **Match**
Identify records that contain significant duplicate information, so that you can fix or remove the duplicate records. Match rules analyze the information that the records represent and therefore can find duplicates when records are non-identical.
- **Validate**
Verify that your data is accurate or present in the expected form. You can validate medical data, such as principal diagnoses, CPT codes, and ICD-10 data. You can also validate common business and personal data, such as patient ages, state names, and ZIP codes.

Deploying the Crisis Response Accelerator

The Crisis Response Accelerator contains a set of rules and dictionaries that you download from the Informatica marketplace in a compressed file.

A **rule** is equivalent to a mapplet in the Informatica environment. Each rule defines a series of data analysis or data transformation operations that you can run on your data. You add a rule to a mapping, and you connect the mapping to a data source.

A **dictionary** is a text file that contains the correct or expected versions of a set of data values. Typically, a dictionary also contains one or more incorrect or alternative versions of each data value. A rule can use the dictionary values as reference data in operations that parse, standardize, or validate your data.

You import the rules and the dictionary data to the Model repository database in an Informatica on-premises environment. Use Informatica Developer to import the rules and the dictionary data. You also use the Developer tool to add a rule to a mapping and to run a mapping.

Accelerator Components

The accelerator rules are packaged in a single XML file named

DataQuality_Accelerator_For_CrisisResponse_1040.xml.

The accelerator also includes multiple dictionary files in a compressed file named

DataQuality_Accelerator_For_CrisisResponse_1040.zip.

You import the rule XML file and the compressed dictionary file contents in a single operation.

The import operation copies the rules to a series of folders in the Model repository. The import operation also copies the dictionary file data to a series of tables in a database that the Informatica domain specifies. Each table has a corresponding object in the Model repository folder structure.

Rules can also read reference data objects called content sets. The content set data is included in the rule XML file that you import. Find the content sets in the Model repository after you import the rule XML file.

Before You Import the Solution Files

The repository objects and database tables that the Crisis Response Accelerator creates operate in the same way as other objects and tables in an Informatica system.

Consider the following rules and guidelines before you import the accelerator:

- Extract the accelerator file that you download to a local directory. You'll select files from the extracted file set during the import operation.
- You'll import the rules and dictionary file data to a single project in the Model repository. The import operation can create the project for you. Or, you can create the project before you import the files.
- Verify that you have all privileges on the following Informatica application services: Analyst Service, Content Management Service, and Data Integration Service.

Note: The Content Management Service specifies the default database to which the import operation will write the dictionary data. The repository tables that contain the dictionary data are known as *reference tables*.

Importing the Accelerator Files

Use the Developer tool to import the rules and reference data for the Crisis Response Accelerator. The import operation prompts you to select the rule XML file and the compressed dictionary file that contains the reference data.

To import the rules and reference data, complete the following steps:

1. In the Developer tool, connect to the Model repository to which you'll import the accelerator.
2. Select the Model repository, and select **File > Import**.
3. In the **Import** pane, select **Informatica > Import Object Metadata File (Basic)**.
4. Click **Next**.
5. Browse to the rule XML file in the accelerator directories that you extracted, and select the XML file.
6. Click **Open**, and click **Next**.
7. In the **Source** pane, select a project node, or select *<project name> Project Content*.

- To create a project for the accelerator, select the project node in the **Source** pane.
 - To import the accelerator to a current project, select *<project name> Project Content* in the **Source** pane and select the destination project in the **Target** pane.
8. Click **Add to Target**.
 9. Expand the **Resolution** option, and select *Replace object in target*.
The **Resolution** options specify how the import operation handles any duplicate object that it encounters.
 10. Click **Next**.
 11. In the **Content Import Settings** pane, verify that the *Import Content* option is checked.
 12. Browse to the compressed dictionary file in the accelerator directories that you extracted, and select the file.
The import operation copies the dictionary file data to database tables and adds a reference table object for each table to the Model repository.
 13. Click **Open**.
 14. Verify that the code page is UTF-8, and click **Next**.
 15. Accept the default options on the **Import Summary** page.
 16. Click **Finish**.

After You Import the Files

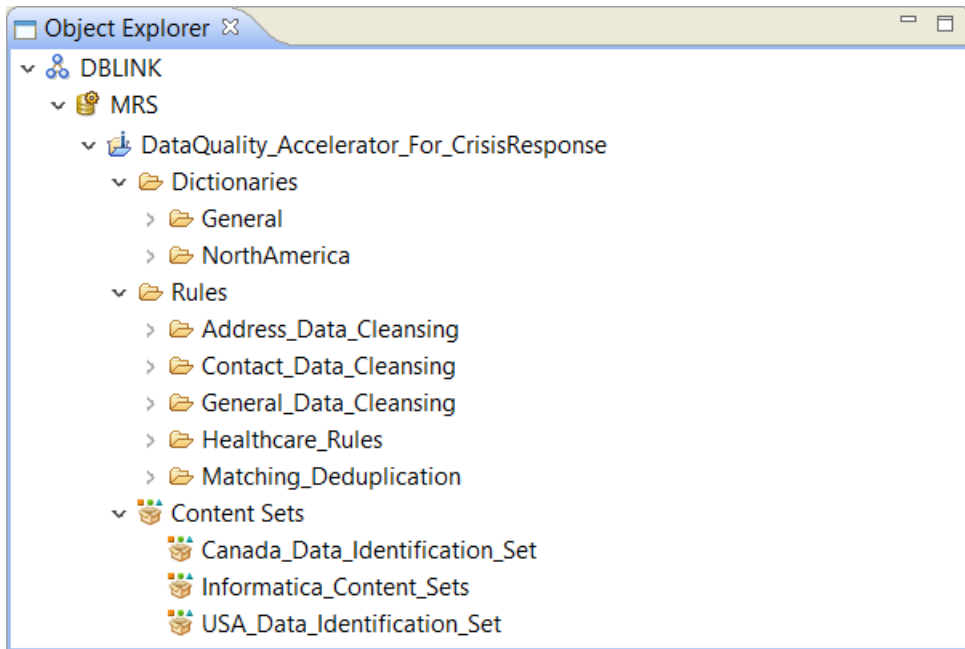
After you import the Crisis Response Accelerator, complete the following tasks:

- Verify the project contents in the Model repository.
- Install the Informatica reference data for address verification and identity match rules.
- Update the contents of the reference tables that contain CPT data and ICD-10 data.

Verifying the Project in the Model Repository

Browse the Model repository and verify that the project contains folders for dictionaries, rules, and content sets.

The following image shows an example of the accelerator project in the Model repository after you import the files:



Installing Reference Data for Address Verification and Identity Matching

Some of the rules in the Crisis Solution Accelerator read reference data files that you must download independently from Informatica. The reference data files contain data that allows you to verify postal addresses and to find duplicate identities in your source data.

Note: The identity reference data files are known as population files.

Download the address reference data files and the identity population files to the machine that hosts the Informatica application services.

For full instructions on how to download the reference data files, see [“Appendix 1: Installing Reference Data for Address Verification and Identity Matching” on page 18](#).

Updating Reference Tables with CPT and ICD-10 Data

A number of rules in the Crisis Solution Accelerator read reference tables that contain CPT or ICD-10 data. The accelerator installs reference tables that include a sample of the data. Before you run a mapping with a rule that uses the data, you must obtain the complete data from an appropriate source and add the data to reference tables in the Model repository.

Note: To use the complete sets of CPT and ICD-10 data, you may need to pay a royalty to the organization that maintains the data.

For full instructions on how to obtain the data and update the reference tables, see [“Appendix 2: Updating CPT and ICD-10 Reference Tables” on page 19](#).

Strategies for Using the Crisis Response Accelerator

To get the most from the Crisis Response Accelerator, define a data quality strategy that applies the rules to your data.

The strategy might include the following steps:

- Identify the data challenges that you face and the data objectives that you want to achieve.
For example, you might be interested in the distribution of people who test positive for COVID-19 within a particular area.
- Identify the rules in the accelerator that can address the challenges and meet the objectives.
To check if a set of patients live within an area that you define, use the following rule:
rule_Geocoordinate_In_Polygon
- Define the sequence in which to run the rules.
The sequence in which you run the rules that you select is a key element in the strategy, and one that is easy to overlook.

Creating a Strategy: Defining a Sequence of Rules

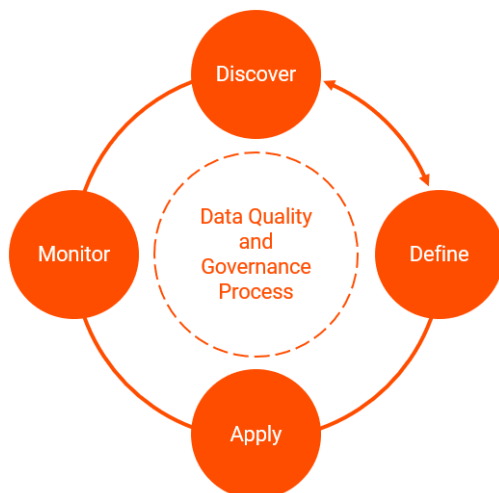
The following guidelines may help you to decide on the sequence in which to run data quality rules:

- In a comprehensive data quality project, you are likely to run rules to analyze and confirm your understanding of your data structure before you perform any tasks on the data.
- You are likely to cleanse and standardize your data, so that it is uniform and well-formatted.
- Once you have established a data quality baseline and defined your data quality goals, you are likely to run rules to validate the data to ensure that it is accurate and free of duplicates. You might parse and derive new information from your data. You may take other actions to address data quality, including manual intervention in the data.
At this stage, your data may be in good shape to enable effective and informed decision making.
- You may re-run some rules on a schedule or in an iterative manner to track the progress of patients or other metrics through the system.

This sequence of rules - from analysis to action - can form the basis of a data quality life cycle that runs continually on your data.

Sample Data Quality Life Cycle

The following diagram shows a data quality life cycle that you might adopt:



You might organize the life cycle stages in the following way:

- In the **Discover** phase, you run rules to analyze your data and establish the baseline data quality.
- In the **Define** phase, you identify a set of data quality and data governance goals for the data, and you select the rules that will verify or update your data to meet the goals.
- In the **Apply** phase, you configure mappings to run the rules on your data in the sequence that you decided.
- In the **Monitor** phase, you evaluate the results of the mappings. You can use the mapping results to aid in business decision making, and you can use the results to plan the steps that you'll take to maintain and further enhance your data quality.

Running data quality rules in an iterative matter creates a data quality life cycle, wherein you continually improve the quality of your data and you build a set of statistics that can demonstrate your levels of success over time.

Note: When you define a data quality strategy, consider also the users who may use the data after you. For example, a billing department or an insurance company may need access to some of the data. Consider the end-to-end users of the data - who may include your patients - and select the rules that can make your data most fit for purpose.

Applying Rules to Your Data

To apply a rule to your data, add it to an open mapping in the Developer tool. In addition to selecting the rule to use, you connect the mapping to a data source, and you select a data target to receive the mapping results.

You can add multiple rules in a series in a single mapping, and you can use a rule in multiple mappings. Design the mappings in the way that is most suitable to your project.

Note: You do not need to add a reference table to an out-of-the-box rule that you use in a mapping. The out-of-the-box rules are preconfigured to find the reference tables that they need in the Model repository.

Crisis Response Accelerator Rules

The following sections describe the rules in the Informatica Crisis Response Accelerator.

Address Data Cleansing Rules

Use the address data cleansing rules to parse, standardize, and validate address data. Find the address data cleansing rules in the following repository location:

[Project_Name]\Rules\Address_Data_Cleansing

The following table describes the address data cleansing rules:

Name	Description
rule_Country_Identification	Identifies a country identifier in the column(s) that the rule specifies.
rule_Country_Name_Standardization	Standardizes country names. The rule returns a country name, a two-character ISO country code, and a three-character ISO country code.
rule_Geocoordinate_In_Polygon	Verifies the presence of geocoordinate points within an area that three or more geocoordinate points define.

Name	Description
rule_Global_Address_Validation_Hybrid	Validates the deliverability of address records from multiple countries. The rule corrects errors in the input addresses where possible. Use the rule when you can connect the input address fields to the Hybrid input fields on the Address Validator transformation.
rule_USA_County_Validation	Validates United States county names. The rule compares input data against county names in all states. The rule returns "Valid" or "Invalid."
rule_USA_State_Validation	Validates United States state names. The rule returns "Valid" or "Invalid."
rule_USA_ZIPCode_Validation	Validates five-digit United States ZIP Codes. The rule returns "Valid" or "Invalid."

Contact Data Cleansing Rules

Use the contact data cleansing rules to parse, standardize, and validate data about business contacts and individuals. Find the contact address data cleansing rules in the following repository location:

[Project_Name]\Rules>Contact_Data_Cleansing

The following table describes the contact data cleansing rules:

Name	Description
rule_Email_Parse	Parses email addresses from data fields.
rule_Email_Parse_and_Validate	Parses email addresses from data fields and validates the format of each email address.
rule_Email_Parse_Into_Mailbox_Domain	Parses email addresses into mailbox, domain, and subdomain fields. For example, the rule parses info@informatica.com in the following manner: - Mailbox: info - Subdomain: informatica - Domain: com
rule_Email_Validation	Validates the format of email addresses. The rule does not verify that the email addresses are accurate or active. The rule returns "Valid" or "Invalid."
rule_Identify_Suspect_Names	Identifies names that might not be genuine person names. The rule compares the input values to a reference table of names that are unlikely to be genuine. For example, the reference table includes the names of fictional characters.
rule_Prenome_Assignment	Generates an honorific according to the gender. You can change the female_prenome expression variable from Ms. to Mrs.
rule_Salutation_Assignment	Generates formal and casual greetings from prenames and name tokens. For example, when input data contains "Mr. John Smith," the rule generates the formal greeting "Dear Mr. Smith," and the casual greeting "Dear John,". You can change the prefix and punctuation by editing the variables in the dq_Generate_Salutation Expression transformation.

Name	Description
rule_USA_Gender_Assignment	Assigns gender according to first name. The rule returns "M" for male names, "F" for female names, and "U" if the gender is unknown. For example, the rule assigns the name "John Smith" a gender of "M" for male.
rule_USA_Given_Name_Standard	Generates given names from U.S. nicknames. For example, the rule standardizes the nickname "Bob" to the given name "Robert."
rule_USA_Multi_Person_Name_Parse	Parses person name values into separate fields. The rule creates fields for values such as title, first name, middle name, and surname. The rule output includes a field that contains the full name of the person in the record. You can use the full name field as an input to a Match transformation in an identity match analysis mapping. When the name data identifies more than one person, the rule creates an output field for each full name. For example, the rule can read the name "John and Jane Smith" and create output fields for "John Smith" and "Jane Smith."
rule_USA_Personal_Name_Parse_and_Standardize_FML	Parses the values in a person name into separate fields. The rule also standardizes the name values. The rule creates the fields in the following sequence: First name, middle name, last name The rule output also includes a field that contains the full name of the person in the record. You can use the full name field as an input to a Match transformation in an identity match analysis mapping.
rule_USA_Personal_Name_Parse_and_Standardize_LFM	Parses the values in a person name into separate fields. The rule also standardizes the name values. The rule creates the fields in the following sequence: Last name, first name, middle name The rule output also includes a field that contains the full name of the person in the record. You can use the full name field as an input to a Match transformation in an identity match analysis mapping.
rule_USA_Personal_Name_Parse_Validation	Validates the gender assignment for a name. The rule calculates the probabilities that a data value is a male name or a female name. If the gender is unknown, the rule uses the probability calculations to assign a gender to the name.
rule_USA_Personal_Name_Parsing_FML	Parses the values in a person name into separate fields. The rule creates the fields in the following sequence: - First name, middle name, last name The rule output also includes a field that contains the full name of the person in the record. You can use the full name field as an input to a Match transformation in an identity match analysis mapping. Note: The rule does not standardize the name values. To standardize and parse United States name values in the sequence that the rule defines, select rule_USA_Personal_Name_Parse_and_Standardize_FML.
rule_USA_Personal_Name_Parsing_LFM	Parses the values in a person name into separate fields. The rule creates the fields in the following sequence: - Last name, first name, middle name The rule output also includes a field that contains the full name of the person in the record. You can use the full name field as an input to a Match transformation in an identity match analysis mapping. Note: The rule does not standardize the name values. To standardize and parse United States name values in the sequence that the rule defines, select rule_USA_Personal_Name_Parse_and_Standardize_LFM.

Name	Description
rule_USA_Phone_Number_Parse	Parses a United States telephone number from a string. The rule parses the first telephone number in the data, reading from right to left. The rule returns a telephone number and also returns a string that contains the input text with the telephone number removed.
rule_USA_Phone_Number_Standardization	Standardizes United States telephone numbers. The rule returns the telephone number in the following formats: - Standard: (nnn) nnn-nnnn - Dashes: nnn- nnn -nnnn - No Spaces: nnnnnnnnn
rule_USA_Phone_Number_Validation	Validates the area code and length of United States telephone numbers. The rule returns values that indicate if the area code and length of a telephone number are valid.
rule_USA_Phone_Parse_Standardize_Validate	Parses a telephone number from a string of text and verifies that the area code is valid for the United States. If the area code is valid, the rule returns the telephone number in three formats. The rule also returns a status value to indicate whether the data conforms to the standard format for a United States telephone number.
rule_USA_Phone_w_Extension_Parse	Parses a number from a string of text if the number conforms to the standard format for a United States telephone number. The rule includes any telephone extension data when it parses the telephone number.
rule_USA_SSN_Parse	Parses United States Social Security numbers (SSN) from data.
rule_USA_SSN_Parse_Standardize_and_Validate	Parses, standardizes, and validates United States Social Security numbers from a larger string of text. The rule can parse numbers that include or omit dashes. By default, the rule writes Social Security numbers without any punctuation. To change the standardization format, open the dq_SSN_Format transformation in the rule and update the expression on the SSN_Format field.
rule_USA_SSN_Standardization	Standardizes United States Social Security numbers. The rule can output the following formats: - No Punctuation - nnnnnnnnn - Space - nnn nnn nnn - Dash - nnn- nnn -nnn To change the format, edit the SSN_Format expression variable in the dq_SSN_Format Expression transformation. Default is "No_Punctuation."

Name	Description
rule_USA_SSN_Validation	Validates United States Social Security numbers. The rule validates each Social Security number for length, numeric values, and known minimum and maximum values in the Area, Group, and Serial Number sections. The Area section comprises the first three digits of the number, and the Group section comprises the fourth and fifth digits. The Serial Number section comprises the final four digits. If the number was issued prior to June 2011, the rule also verifies that the Area value and Group value are a valid combination. The rule does not verify that the number is an issued number. The rule returns "Valid" or "Invalid."
rule_USA_SSN_Validation_post_June2011	Validates United States Social Security numbers. The rule validates each Social Security number for length, numeric values, and known minimum and maximum values in the Area, Group, and Serial Number sections. The Area section comprises the first three digits of the number, and the Group section comprises the fourth and fifth digits. The Serial Number section comprises the final four digits. The rule does not verify that the Area value and Group value are a valid combination. The rule does not verify that the number is an issued number. The rule returns "Valid" or "Invalid."

General Data Cleansing Rules

Use the general data cleansing rules to parse, standardize, and validate data. Find the general data cleansing rules in the following repository location:

[Project_Name]\Rules\General_Data_Cleansing

The following table describes the general data cleansing rules:

Name	Description
rule_Assign_DQ_Mailability_Score_Description	Assigns a description to the Mailability Score output from the Address Validator transformation. The description corresponds to the output from Data Quality transformations in releases prior to Data Quality 9.0.
rule_Assign_DQ_Match_Code_Description	Assigns a description to the Match Code output from the Address Validator transformation. The description corresponds to the output from Data Quality transformations in releases prior to Data Quality 9.0.
rule_CAN_NER_Field_Identification	Identifies the type of information that an input field contains. The rule can identify names, personal IDs, company names, dates, and Canadian address data. The rule returns a label that describes the type of input data. The rule uses reference data to identify the types of information. The rule uses probabilistic matching techniques to identify the types of information.
rule_Compare_Dates	Calculates the difference between two dates. The rule uses the following units of measure: - Hours - Days - Months - Years Each output value is exclusive from the other values. The outputs cannot be added to represent the difference between the data values.

Name	Description
rule_Completeness	Checks a single field for NULL values. Returns "Complete" if the field contains data. Returns "Incomplete" if the field is empty or contains a NULL value.
rule_Completeness_Multi_Port	Checks multiple fields for NULL values. Returns "Complete" if all fields contain data. Returns "Incomplete" if any field is empty or contains a NULL value.
rule_Date_Complete	Verifies that the input string conforms to a date format that the rule recognizes. The rule reads the following reference table: user_defined_dates_infa
rule_Date_of_Birth_Validation	Checks the number of years between a date of birth and the current date. Returns "Adult" or "Minor" in addition to "Valid" if the number of years is 120 or lower. Returns "Invalid" if the number of years is greater than 120.
rule_Date_Parse	Parses date data from a string to a field that the rule specifies. The rule recognizes dates in the following formats: - dd/mm/yyyy - mm/dd/yyyy - yyyy/dd/mm The rule returns a date and also returns a string that contains the input text without the date.
rule_Date_Standardization	Standardizes date strings to an output format that you specify. To set the output format, open the dq_FormatDate Expression transformation in the rule and update the Output_Date_Format expression variable and the Delimiter expression variable. If the input data does not describe a valid date, the rule returns the digit 0 for each input character.
rule_Date_Validation	Validates date strings that appear in a single format in a data column. To configure the date format that the rule uses for validation, open the dq_ValidateDate Expression transformation in the rule and update the In_Date_Format expression variable. The default format is "MM/DD/YYYY." The rule returns "Valid" or "Invalid."
rule_Date_Validation_Variable_Format	Validates date strings that appear in multiple formats in a data column. Use the rule when a data source includes the following columns: - A column that contains date values in multiple formats. - A column that identifies the format of the date value in each row. If the column does not identify a date format for a row, the rule applies the format "MM/DD/YYYY" to the date value. The rule reads all data values that the is_date() function recognizes. The rule returns "Valid" or "Invalid."
rule_Days_Between_Dates	Calculates the number of days between two dates.
rule_Days_From_Current_Date	Calculates the number of days between a specified date and the current date.
rule_Field_North_American_Data	Identifies the following types of fields: name, occupation title, company, address, city, state or province, postcode, country, personal ID, email, telephone, credit card, and date. The rule generates a score that indicates the degree of confidence in the field identification. Higher scores indicate greater levels of confidence. If the rule cannot assign a field type, the rule writes the data on the Out_Undetermined field.
rule_IsNumeric	Verifies that the input data is numeric. The rule returns "True" or "False."

Name	Description
rule_LowerCase	Returns all alphabetic characters in lower case.
rule_Negative_Number_Validation	Validates that the input data is a negative number.
rule_Numeric_Completeness	Checks for NULL values in numeric inputs.
rule_Parse_Alpha_Chars_from_Non_Alpha_Chars	Identifies the alphabetic characters and the non-alphabetic characters in an input string and writes each set of characters to different output fields. For example, the rule parses the following values from the input string teststring_123: teststring_123
rule_Parse_First_Word	Parses the first word in an input string to a field that the rule specifies.
rule_Parse_Number_At_End_Of_Line	Parses any number that occurs at the end of an input string to a field that the rule specifies. The rule reads strings from left to right.
rule_Parse_Number_At_Start_Of_Line	Parses any number that occurs at the start of an input string to a field that the rule specifies. The rule reads strings from left to right.
rule_Parse_Text_Between_Parentheses	Parses strings that are enclosed in parentheses to a field that the rule specifies. The rule contains an output field for the parsed strings and an output field for the input text without the parsed strings.
rule_Parse_Text_in_Single_Quotes	Parses strings that are enclosed in quotation marks to a field that the rule specifies. When the input data contains multiple quoted elements, the rule parses the final element. The rule reads the input strings from left to right. The rule contains an output field for the parsed strings and an output field for the input text without the parsed strings.
rule_Past_Date_Label	Determines whether an input date is earlier than the system date or later than the system date.
rule_Personal_Company_Identification	Parses person names and company names to different fields that the rule specifies. The rule has the following outputs: - Person name - Company name - Data category, such as person name or company name- - Data that the rule cannot parse
rule_Positive_Number_Validation	Verifies that the input data is a positive number.
rule_Prepend_Zero_to_Single_Digit	Prepends the numeral "0" to single numeric characters.
rule_Remove_All_Leading_Zeros	Removes all instances of the numeric character "0" from the beginning of a string.
rule_Remove_Apostrophe	Removes apostrophes. The rule merges the text strings on either side of the apostrophe.
rule_Remove_Control_Characters	Removes control characters from text strings. The rule returns a string that contains the control characters and a string that contains the input text without the control characters.
rule_Remove_Extra_Spaces	Replaces all consecutive spaces with a single space and trims leading and trailing spaces.
rule_Remove_Hyphen	Removes hyphens from anywhere in the input string.

Name	Description
rule_Remove_Leading_Zero	Removes a single instance of the numeric character "0" from the beginning of a string.
rule_Remove_Limited_Punctuation	Removes extraneous characters. Extraneous characters include slashes, back slashes, periods, exclamation marks, and underscores. The rule also replaces multiple consecutive spaces with a single space.
rule_Remove_Non_Numbers	Removes all characters that are not numeric.
rule_Remove_Parentheses	Removes right and left parenthesis symbols.
rule_Remove_Period	Remove periods.
rule_Remove_Period_Parentheses	Removes the following characters: - Left and right parentheses - Periods
rule_Remove_Punctuation	Removes punctuation symbols.
rule_Remove_Punctuation_and_Space	Removes all punctuation and all space characters.
rule_Remove_Quotation	Removes quotation marks.
rule_Remove_Slashes	Removes forward slashes and back slashes.
rule_Remove_Space	Removes all character spaces.
rule_Replace_Hyphen_with_Space	Replaces hyphens with spaces.
rule_Replace_Limited_Punct_with_Space	Replaces the following punctuation characters with a single space: dash, back slash, period, exclamation mark, and underscore The rule also replaces two, three, and four consecutive spaces with a single space.
rule_Replace_Non_Alphabetic_with_Space	Replaces numerals and punctuation characters with a single space.
rule_String_Completeness	Checks a string for completeness. The rule also searches the input strings for values in the reference table <i>string_default_values_infa</i> . The reference table contains values such as NA, DEFAULT, and XX. If an input string contains a value in the reference table, the rule identifies the string as incomplete.
rule_TitleCase	Converts strings to title case. In title case strings, the first letter of each word is capitalized.
rule_Translate_Diacritic_Characters	Replaces diacritic characters with ASCII equivalents. For example, the rule converts "ã" to "a".
rule_UpperCase	Returns all alphabetic characters in upper case. The input and output fields in the rule use a precision of 200.
rule_UpperCase1000	Returns alphabetic characters in upper case. The input and output fields in the rule use a precision of 1,000.

Name	Description
rule_USA_NER_Field_Identification	Identifies the type of information that an input field contains. The rule can identify names, personal IDs, company names, dates, and United States address data. The rule returns a label that describes the type of input data. The rule uses reference data and probabilistic matching techniques to identify the types of information.
rule_Years_Since_Date_of_Birth	Calculates the number of years since the input date.

Healthcare Rules

Use the healthcare rules to parse, standardize, and validate data in patient records.

Find the patient rules in the following repository location:

[Project_Name]\Rules\Healthcare

The following table describes the patient rules:

Name	Description
mplt_Association_btw_Infection_and_Blood_Type	Standardizes blood type and diagnosis data, and returns the corresponding ICD-10 codes and description for each diagnosis.
mplt_Create_Institution_MatchFld	Standardizes Institution names in preparation for use in a rule that performs match analysis.
mplt_Parse_ICD_Codes_frm_Strings	Parses ICD codes from strings
rule_Blood_Type_Standardization	Standardizes the blood type identifiers in the input column(s) that the rule specifies.
rule_CPT_Code_Parse	Parses Current Procedural Terminology (CPT) codes for all categories.
rule_CPT_Code_Validate	Validates CPT codes and returns the respective section code in each case.
rule_ICD10_Adult_Code_Check	Checks if an ICD-10 code is for an adult patient, that is, for a patient in the age range of 15 through 124 inclusive. Returns 'True' if it is an adult code and 'False' if not.
rule_ICD10_Code_Definition	Returns the Title and Block definitions for an ICD-10 code.
rule_ICD10_Code_Validation	Validates the ICD-10 code by means of the list built by the user.
rule_ICD10_Questionable_Principal_Diagnosis_Code_Check	Checks if an ICD-10 code is determined to be insufficient for admission to an acute care hospital when used as a principal diagnosis. Returns 'True' if it is a Questionable Principal Diagnosis Code, 'False' if not.

Name	Description
rule_Match_Diag_Single	Compares morbidity values in two input strings, and returns a match score that represents the degree of similarity between them.
rule_Parse_Comorbidities_frm_Diagnosis_Strings	Parses comorbidities from diagnosis strings.
rule_Parse_FacilityType_from_FacilityName	Parses the healthcare facility type from the field that that contains the facility name.
rule_USA_Patient_Gender_Assignment	Assigns gender to a record according to the first-name information that the record contains. The rule returns "M" for male names, "F" for female names, and "U" if the gender is unknown. For example, the rule assigns the name "John Smith" a gender of "M" for male.
rule_USA_Provider_Gender_Assignment	Assigns gender to a record according to the first-name information that the record contains. The rule returns "M" for male names, "F" for female names, and "U" if the gender is unknown. For example, the rule assigns the name "John Smith" a gender of "M" for male.

Matching and Deduplication Rules

Use the matching and deduplication rules to measure the levels of similarity between the records in data sets.

Find the matching and deduplication rules in the following repository location:

[Project_Name]\Rules\Matching_Deduplication

The following table describes the matching and deduplication rules:

Name	Description
mplt_Individual_Name_and_Address_Match	Uses field match strategies to identify duplicate rows based on person names and United States address data. The maplet uses a combination of characters from the surname values and the postal code values to generate group keys.
mplt_Individual_Name_and_Email_Match	Uses field match strategies to identify duplicate rows based on person names and email addresses. The maplet generates group keys from the email address data.
mplt_Individual_Name_and_SSN_Match	Uses field match strategies to identify duplicate rows based on person names and United States Social Security numbers. The maplet generates keys generated from the Social Security number data.
mplt_USA_IMO_Individual_Name_and_Address_Match	Uses identity match strategies to identify duplicate rows in United States data based on person names and addresses. The maplet generates group keys from the postal code data.

Name	Description
mplt_USA_IMO_Personal_Name_and_Data_Match	Uses identity match strategies to identify duplicate rows in United States data based on person names and personal data. The fields in the personal data column must contain a single type of data, such as telephone number, email, or Social Security number. The maplet generates group keys from the personal data.
rule_Individual_Name_and_Address_MatchScore	Generates a match score based on person names and United States address data.
rule_Individual_Name_and_Email_MatchScore	Generates a match score based on person names and email addresses.
rule_Individual_Name_and_SSN_MatchScore	Generates a match score based on person names, Social Security numbers, and identification data.

Appendix 1: Installing Reference Data for Address Verification and Identity Matching

Address reference data files contain comprehensive data about the postal addresses in a given country. Identity population files contain metadata about personal, household, and corporate identities in a given region.

To run rules that perform address verification or identity matching, perform the following tasks:

- Download the reference data and population files from Informatica, and extract the files.
- Install the files to separate directories that the Data Integration Service can read.
- Verify the reference data properties on the Content Management Service.

Note: The Data Integration Service and the Content Management Service are application services in the Informatica domain.

Downloading the Reference Data

The address reference data files that you download for the Crisis Response Accelerator are called *Batch/Interactive* files. The file names include "BI" to indicate Batch/Interactive processing.

The address reference data file names end in **.mb**.

The identity population file names end in **.yxp**.

Work with your Informatica representative to identify the files that you need and to confirm the download site from which you can access your files.

Extract the files after you download them.

Installing the Reference Data

The installation process consists of verifying the file locations or moving the files to usable locations. There are no executable installation steps.

Verify or move the address reference data files to a directory that the Data Integration Service can read.

Verify or move the population files to the following directory on the Data Integration Service machine:

`[Informatica_installation_directory]/services/DQContent/INFA_Content/identity/default`

Note: Before you install any address reference data file or identity population file, stop the Data Integration Service and the Content Management Service on the Informatica domain. Restart the services after you install the files and configure the file properties.

Configuring the File Properties on the Content Management Service

Before you run any rule that performs address verification or identity match analysis, verify the reference data properties on the Content Management Service.

To verify the properties, complete the following steps:

1. Log in to Informatica Administrator, and select the Content Management Service in the Informatica domain.
2. Open the **Process Properties** on the Content Management Service.
3. Select **Address Validation Properties**.
 - On the *License* property, verify or update the license key for the address reference data files.
 - On the *Reference Data Location* property, verify or update the path to the address reference data files.
4. Select **Identity Properties**.
5. On the *Reference Data Location* property, verify or update the path to the parent directory for the /default/ directory that contains the population files.

Rules and Guidelines for Address Reference Data

Consider the following rules and guidelines for address reference data:

- Informatica releases updates to the address reference data files at regular intervals. You can overwrite the reference data files in your installation with any new version of the file that you download from Informatica.
- You can update the address reference data files without entering a new license key.

Note: The Content Management Service contains additional properties that you can optionally set for the address reference data files and identity population files. For more information about the files, follow the link to the Informatica Content Guide in the ["Further Reading" on page 22](#) section in this document.

Appendix 2: Updating CPT and ICD-10 Reference Tables

After you import the Crisis Solution Accelerator, you can update the reference tables that contain CPT and ICD-10 data.

Update the following reference tables:

- *icd10cm_codes_2020_descr_infa*
Contains the ICD-10 disease names and corresponding disease codes.
- *icd10cm_codes_2020_infa*
Contains the ICD-10 disease codes in two identical columns and also contains the corresponding disease names.
- *icd10dm_codes_2020_classification*
Contains the chapter titles and block codes from the international version of the ICD-10 classification.
- *cpt_section_by_code*
Contains the CPT codes and their corresponding descriptions.
- *cpt_codes_infa*
Contains the CPT codes in two identical columns.

To update the reference tables, you download the complete sets of CPT data and ICD-10 data that the tables contain.

Downloading the ICD-10 Data

Find the data for the *icd10cm_codes_2020_infa* and *icd10cm_codes_2020_descr_infa* tables at the following address:

<https://www.cms.gov/Medicare/Coding/ICD10/2020-ICD-10-CM>

Download the following file:

- 2020 Code Descriptions in Tabular Order (ZIP)

Extract the file, and find the following file in the extracted contents:

- *icd10cm_codes_2020.txt*

Find the data for the *icd10dm_codes_2020_classification* table on the following Wikipedia page:

<https://en.wikipedia.org/wiki/ICD-10>

Download the data from section 1. *Chapters* on the Wikipedia page.

Downloading the CPT data

Find the CPT data at the following address:

<https://coder.aapc.com/cpt-codes>

Copy the CPT data for the following subject areas:

- 00100-01999 Anesthesia
- 10004-69990 Surgery
- 70010-79999 Radiology Procedures
- 80047-89398 Pathology and Laboratory Procedures
- 90281-99756 Medicine Services and Procedures
- 99201-99499 Evaluation and Management Services
- 0001F-9007F Category II Codes
- 0001U-0202U Laboratory Analyses
- 0002M-0014M Multianalyte Assay
- 0042T-0593T Category III Code
- *cpt-modifiers* Modifiers

Updating the Reference Tables

You can copy data values directly into a reference table that you open in the Developer tool.

However, because the CPT and ICD-10 reference tables contain large quantities of sensitive data, Informatica recommends the following best practice:

1. Export the reference table from the Model repository to your local system.
2. Update the reference data in Microsoft Excel.
3. Import the reference table that you updated to the Crisis Solution Accelerator project.

The export operation writes an XML file and a compressed dictionary file to a location that you specify. The XML file represents the reference table repository object. The dictionary file is a text file that contains the reference data values.

You'll update the values in the dictionary file, and you'll import the XML file back into the repository. The XML import process identifies the dictionary file and writes the dictionary file contents to the corresponding reference database table.

Exporting a Reference Table from the Model Repository

1. In the Developer tool, connect to the Model repository that contains the accelerator.
2. Select **File > Export**.
3. Select **Informatica > Export Object Metadata File**.
4. Click **Next**.
5. Select the Crisis Solution Accelerator project, and click **Next**.
6. Select the reference table object to export.
You can find the ICD-10 and CPT reference tables at the following location:
Dictionaries/General.
7. Browse to the directory to which you'll export the reference table. Click **Save**, and click **Next**.
The wizard displays any dependent object for the metadata object.
8. Click **Next** to accept the dependent objects.
9. Verify that the **Export content** option is checked, and verify the export settings:
 - Verify a name and location for the file that will contain the reference data values. The Developer tool Service exports the reference data file to a zip file.
By default, the wizard exports the ZIP file and the XML metadata file to the same directory.
 - Verify that the code page is UTF-8.
 - Do not update the probabilistic model settings.
10. Click **Finish**.

Updating a Reference Data in Microsoft Excel

To update the dictionary data associated with a reference table, complete the following steps:

1. Extract the compressed dictionary file, and navigate to the dictionary file. The file has the following extension:
.dic
Note: Do not move the dictionary file to another location.
2. Open the dictionary file in Microsoft Excel.
3. Open the ICD-10 file that you downloaded in another instance of Microsoft Excel.
Or, copy the data from the appropriate web page into another instance of Microsoft Excel.
4. Copy the data that you require from the CPT or ICD-10 worksheet to the dictionary file.
Note: Take care to preserve the column structure of the dictionary file. The rules that read the reference data are configured to treat one of the columns as the valid or preferred version of the data.
5. Save the dictionary file. Do not change the file name or file type.
6. After you save the file, create a zip file from the directories that you extracted in step 1.
Take care to preserve the directory structure of the file that you exported.

Importing a Reference Table to the Model Repository

The steps to import a reference table are similar to the steps to import a repository project that the document described earlier.

To import a single reference table, complete the following steps:

1. In the Developer tool, select the Crisis Solution Accelerator repository project and select **File > Import**.

2. In the **Import** pane, select **Informatica > Import Object Metadata File (Advanced)**.
3. Click **Next**.
4. Browse to the reference table XML file.
5. Click **Open**, and click **Next**.
6. In the **Source** pane, expand the directory structure and select the reference table object.
7. Click **Automatch** to Target to associate the updated reference table object with the current object in the repository.
8. Expand the **Resolution** option, and select *Replace object in target*.
The **Resolution** options specify how the import operation handles any duplicate object that it encounters.
9. Click **Next**. The Developer tool lists any dependent objects for the import file.
10. Add any dependent objects to the target project.
11. Click **Next**.
12. In the **Content Import Settings** pane, verify that the *Import Content* option is checked.
13. Browse to the compressed file that contains the dictionary file that you updated, and select the file.
The import operation updates the dictionary file data in the database tables and refreshes the reference table object in the Model repository.
14. Click **Open**.
15. Verify that the code page is UTF-8, and click **Next**.
16. Accept the default options on the **Import Summary** page.
17. Click **Finish**.

When the import is complete, refresh the Model repository connection.

Further Reading

Informatica provides a full suite of user documentation on its documentation portal:

<https://docs.informatica.com> .

The following links highlight some of the useful information that you can find on the portal:

- To learn about Informatica mappings in the on-premises environment, consult the Developer Mapping Guide:
<https://docs.informatica.com/data-engineering/shared-content-for-data-engineering/10-4-0/developer-mapping-guide/preface.html>
- To learn about the Developer tool, consult the Developer Tool Guide:
<https://docs.informatica.com/data-engineering/shared-content-for-data-engineering/10-4-0/developer-tool-guide/preface.html>
- To learn about the data analysis and data transformation objects that constitute rules and mappings, consult the Developer Transformation Guide:
<https://docs.informatica.com/data-quality-and-governance/data-quality/10-4-0/developer-transformation-guide/preface.html>
- To learn about the configuration of Informatica application services, consult the Application Service Guide:
<https://docs.informatica.com/data-quality-and-governance/data-quality/10-4-0/application-service-guide/preface.html>
- To learn about the installation and configuration of address reference data and identity population data, consult the Content Guide:
<https://docs.informatica.com/data-quality-and-governance/data-quality/10-4-0/content-guide/preface.html>

- To learn about basic on-premises operations in Informatica Data Quality, consult the Data Quality Getting Started Guide:
<https://docs.informatica.com/data-quality-and-governance/data-quality/10-4-0/getting-started-guide/preface.html>

Author

David Handy