Informatica

Informatica® MDM Registry Edition
10.0 HotFix 1

Populations and Controls

Informatica MDM Registry Edition Populations and Controls
10.0 HotFix 1
April 2017

Publication Date: 2018-06-25

Table of Contents

Preface 5

Informatica Resources 5

- Informatica Network 5
- Informatica Knowledge Base 5
- Informatica Documentation 5
- Informatica Product Availability Matrixes 6
- Informatica Velocity 6
- Informatica Marketplace 6
- Informatica Global Customer Support 6

Chapter 1: Updates to Population Files 7

Chapter 2: SSA-NAME3 Controls 8

KEY-LOGIC Controls 8

SEARCH-LOGIC Controls 10

SCORE-LOGIC Controls 11

- PURPOSE Control 12
- MATCH_LEVEL Control 13
- NAMEFORMAT Control 13
- UNICODE_ENCODING Control 14
- MATCH_OPTIONS Control 14
- SEARCH and FILE Controls 15
- FILTER_SEARCHVALUES Control 18
- COMBINE Control 19

Advanced Controls 20

Chapter 3: Standard Population Choices 25

Population Files Reference 26

A Primer on Keys and Search Strategies 28

Key Fields 29

Key Levels 32

Search Levels 33

Match Purposes 34

Match Levels 51

Generic Field 52

Managing Population Rule Sets 53

Index 54

Preface

Welcome to the Informatica MDM Registry Edition Populations and Controls Guide.

This guide describes SSA-Name3 populations and the controls they support. The latter are added to the Controls statement used within an IDX-Definition or Search-Definition section of the SDF.

Informatica Resources

Informatica Network

Informatica Network hosts Informatica Global Customer Support, the Informatica Knowledge Base, and other product resources. To access Informatica Network, visit https://network.informatica.com.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- Review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit https://kb.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com.

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions.

If you are an Informatica Network member, you can access Informatica Velocity resources at http://velocity.informatica.com.

If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that augment, extend, or enhance your Informatica implementations. By leveraging any of the hundreds of solutions from Informatica developers and partners, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through Online Support on Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
http://www.informatica.com/us/services-and-training/support-services/global-support-centers.

If you are an Informatica Network member, you can use Online Support at http://network.informatica.com.

CHAPTER 1

Updates to Population Files

MDM Registry Edition 9.5.2 affects the following changes in population files:

- You need to rebuild all keys as many fields are enhanced.
  Affected population: Chinese
- The Organisation_Name field has an updated edit list.
  Affected population: Korean
- The address edit list changes RUE from Street Skip to Street Delete.
  Affected population: France
- The Person_Name field removes the ST prefix rule and changes the SANTOS to SAINT replacement rule to a SAINT to SANTOS secondary name rule.
  Affected populations: Australia, Canada, Gaelic, India, Indonesia, International, Ireland, New Zealand, South Africa, Singapore, UK, USA, and SFMS.
- The Address_part1 Conservative match field rule corrects a problem when cloning purposes using the Population Override Manager.
  Affected populations: Arabic, Chinese_i, OFAC, SFMS, and Taiwan_r.
- The Person_Name field adds the missing variation YEKATERINA To KATHERINE.
  Affected populations: Arabic_m, Australia, Canada, Chinese, Gaelic, India, Indonesia, International, Ireland, New Zealand, South Africa, Singapore, UK, USA, and SFMS.
- The Organisation_Name field removes the Person name rules.
  Affected populations: Australia, Canada, Gaelic, Indonesia, International, Irish, New Zealand, Philippines, South Africa, UK, USA, and SFMS.
- The Organsation_Name field does not treat words with both digits and characters as codes.
  Affected populations: Australia, Belgium, Brasil, Bulgaria_C, Bulgaria_R, Canada, Chile, Colombia, Croatia, Czech, Denmark, Estonia, Finland, France, Gaelic, Germany, Hk_R, Hungary, India, Indo_Chin_R, Indonesia, International, Ireland, Italy, Luxembourg, Malaysia , Malta, Mexico, Netherlands, New Zealand, Norway, Peru, Philippines, Poland, Portugal, Puerto Rico, Romania, Russia, Singapore, Slovakia, South Africa, Spain, Sweden, Switzerland, Taiwan, Thailand, Turkey, UK, USA, and Vietnam_R.

CHAPTER 2

SSA-NAME3 Controls

In MDM Registry Edition, when the Key-Logic, Search-Logic, and Score-Logic definitions are set to SSA, the setting starts the SSA-NAME3 key-building, search, and match services at runtime.

When you start the SSA-NAME3 services, specify the controls that you want to use. The format and the content of the controls depend on the service that you start.

Conventions

Use the following conventions when you specify a control:

- The syntax of a control is `keyword=value`.
- Separate multiple controls with spaces. The keyword or the value does not contain any spaces.
- Enclose a control within square brackets ([ ]) if the control is optional.
- The controls are not case sensitive.
- Control values such as field names, search names, and purpose names depend upon the Population Rules that you use. For a definitive list of the control values for your population, use the SSA-NAME3 Workbench.

KEY-LOGIC Controls

The `KEY-LOGIC` controls define the field and options to build the SSA-NAME3 key. The `KEY-LOGIC` controls uses the following syntax:

```
FIELD=field name
[KEY_LEVEL=key level]
[NAMEFORMAT=name format]
[UNICODE_ENCODING=Unicode type]
[COMBINE=field name[:combine option]]
[GEOCODE_FORMAT=0|1]
```

Configure the following definitions:

FIELD (Required)

Specifies the field for which you want to build keys. With most standard populations, you can use one of the following predefined fields:

- Person_Name
- Organization_Name
- Address_Part1
- Geocode

You can use only one key field in a function call. If you want to build a key on another field that contains a different type of data, use two separate functions so that you can store the keys in two separate indexes.

KEY_LEVEL (Optional)

Specifies the type of key level that you want to build. With most standard populations, you can use one of the following predefined values:

- Standard
- Extended
- Limited

The default value is Standard.

NAMEFORMAT L/R (Optional)

Defines whether the major word in a name or address is on the left end or the right end. For example, in Western person names, the family name is on the right end of the names.

Use the `NAMEFORMAT` control to override the default name format. For more information about the default name format for a given standard population field and its effects, from the **Help** menu of SSA-NAME3 Workbench, click **Population Documentation**.

UNICODE_ENCODING (Optional)

Instructs MDM Registry Edition to accept Unicode data input, and specifies the Unicode format of the data that you want to pass.

COMBINE (Optional)

The COMBINE option concatenates multiple data fields. Use the `COMBINE` option to generate keys for the multiple data fields that you want to concatenate before the key building process. The value of the `COMBINE` field must be the same as the FIELD parameter.

Use the following optional COMBINE option to associate with the field name:

`DELIM-NONE`: If you set this option when you concatenate multiple data fields, the data fields do not have any space delimiter between the fields.

`DELIM-SPACE`: If you set this option when you concatenate multiple data fields, the data fields have a space delimited between the fields. The default value is `DELIM-SPACE`.

For example, if the person names are John and Smith, for `COMBINE=Person_Name:DELIM-SPACE` the result is John SMITH, and for `COMBINE=Person_Name:DELIM-NONE`, the result is JOHNSMITH.

GEOCODE_FORMAT (Optional)

Indicates the order of the latitude and longitude coordinates that you specify to generate keys.

If `GEOCODE_FORMAT=1`, SSA-NAME3 considers the geographic coordinates in the following order:

```
*Geocode*<Longitude> <Latitude>***
```

If `GEOCODE_FORMAT=0`, SSA-NAME3 considers the geographic coordinates in the following order:

```
*Geocode*<Latitude> <Longitude>***
```

The default value is 0.

SEARCH-LOGIC Controls

The SEARCH-LOGIC controls define the field and options to build the SSA-NAME3 key range. The SEARCH-LOGIC controls use the following syntax:

```
FIELD=field name
[SEARCH_LEVEL=search level]
[NAMEFORMAT=name format]
[UNICODE_ENCODING=Unicode type]
[SEARCH_LIMIT=n]
[COMBINE=field name[:combine option]]
[GEOCODE_FORMAT=0|1]
[GEOCODE_RADIUS=n]
```

Configure the following definitions:

FIELD (Required)

Indicates the field on which you want to build search ranges. With most standard populations, you can use one of the following predefined fields:

- Person_Name
- Organization_Name
- Address_Part1
- Geocode

You can use only one key field in a function call.

SEARCH_LEVEL (Optional)

Specifies the level of search to perform. With most standard populations, you can use one of the following predefined values:

- Typical
- Exhaustive
- Extreme
- Narrow

The default value is Typical.

Note: The SEARCH_LEVEL control is not applicable to the Geocode field. If you specify the SEARCH_LEVEL control with a Geocode field, SSA-NAME3 ignores the SEARCH_LEVEL control.

NAMEFORMAT L/R (Optional)

Defines whether the major word in a name or address is on the left end or the right end. For example, in Western person names, the family name is on the right end of the names.

Use the NAMEFORMAT control to override the default name format. For more information about the default name format for a given standard population field and its effects, from the **Help** menu of SSA-NAME3 Workbench, click **Population Documentation**.

UNICODE_ENCODING (Optional)

Instructs MDM Registry Edition to accept Unicode data input, and specifies the Unicode format of the data that you want to pass.

SEARCH_LIMIT (Optional)

Specifies the maximum percentage of the file that is allowed to be returned in any one search. This can be helpful in preventing costly searches. The percentage value is a number up to 2 decimal places. When the function is called, the Search Strategy is evaluated before any database I/O is done. If it is calculated

that the search would return a greater percentage of the file than allowed, an error message will be returned to the calling program.

Note: For the `Search_Limit` to be useful, the Population must include a Scalar Frequency Table. See the Population Override Manager for more details.

COMBINE (Optional)

This Control option concatenates multiple data fields. If `COMBINE` option is specified, any multiple data fields for which ranges need to be generated, are concatenated together before the range building process. The `COMBINE` fieldname must be the same as the FIELD parameter.

An optional combine option may be associated with the fieldname and can have the following possible values:

`DELIM-NONE`: If this option is set, when concatenating multiple data fields there is no space embedded.

`DELIM-SPACE`: If this option is set, when concatenating multiple data fields there is a space embedded. This is the default and need not be explicitly specified.

GEOCODE_FORMAT (Optional)

Indicates the order of the latitude and longitude coordinates that you specify to build key ranges.

If `GEOCODE_FORMAT=1`, SSA-NAME3 considers the geographic coordinates in the following order: `*Geocode*<Longitude> <Latitude>***`

If `GEOCODE_FORMAT=0`, SSA-NAME3 considers the geographic coordinates in the following order: `*Geocode*<Latitude> <Longitude>***`

The default value is 0.

GEOCODE_RADIUS (Optional)

Indicates the search radius to build key ranges according to the latitude and longitude coordinates that you specify. The default value is 1000 m, and the default unit is meter.

For example, `FIELD=GEOCODE GEOCODE_FORMAT=1 GEOCODE_RADIUS=1000`

Use the following guidelines when you specify the geocode radius:

- SSA-NAME3 considers the radius to be in meters even if you specify any other unit.
- SSA-NAME3 ignores the decimal values and uses the whole number as the radius. For example, if you specify the radius as 850.8, SSA-NAME3 considers the radius as 850.

SCORE-LOGIC Controls

The `SCORE-LOGIC` controls define the match purpose and options for a SSA-NAME3 matching.

You can use the following `SCORE-LOGIC` controls:

- PURPOSE
- MATCH_LEVEL
- NAMEFORMAT
- UNICODE_ENCODING
- MATCH_OPTIONS

- SEARCH and FILE
- FILTER_SEARCHVALUES
- COMBINE

The SCORE-LOGIC controls use the following syntax:

```
PURPOSE=<expression>
[MATCH_LEVEL=matchlevel][+/-nn][+/-nn]
[NAMEFORMAT=name format]
[UNICODE_ENCODING=Unicode type]
[MATCH_OPTIONS=(fieldname:option,...)
[SEARCH=field1[(matching type expression1)],offset1,length1,... ,
fieldn[(matching type expressionn)],offsetn,lengthn]
[FILE=field1[(matching type expression1)],offset1,length1,... ,
fieldn[(matching type expressionn)],offsetn,lengthn]
[FILTER_SEARCHVALUES=<Filter Purpose Number>,<List of Flags>,(<List of Values>)|{<Value>,
(<List of Values>)}]
[COMBINE=fieldname1[:combine option1],… ,
fieldnamen[:combine optionn]]
```

PURPOSE Control

Use the PURPOSE control to specify the name of the matching purpose to use in a match call.

The PURPOSE control uses the following syntax:

PURPOSE=<expression>.

Use one of the following formats for <expression>:

```
<expression> := <Purpose_Name>
<expression> := <Purpose_Name>(<Match_level>)
<expression> := (not<expression>)
<expression> := (<expression> or <expression>)
<expression> := (<expression> and <expression>)
<expression> := (<expression>)
```

In the ssan3_match section of SSA-NAME3 Workbench, under **Mandatory Controls**, you can find a list of the supported purpose names. With most standard populations, you can use one of the following predefined purposes:

- Address
- Contact
- Division
- Fields
- Household
- Individual
- Organization
- Person_Name
- Resident
- Wide_Contact
- Geocode

For more information about each purpose and its required and optional fields, from the **Help** menu of SSA-NAME3 Workbench, click **Population Documentation**.

The simple form of <expression> is PURPOSE=<Purpose Name>. For example, PURPOSE=Address performs matching on the specified address fields.

MATCH_LEVEL Control

Use the MATCH_LEVEL control to specify the level of matching to perform.

With most standard populations, you can use one of the following predefined match levels:

- Typical
- Conservative
- Loose

The default match level is Typical. You can use the `MATCH_LEVEL` control to adjust the predefined accept or reject score limits that affect the match decisions.

The `MATCH_LEVEL` control uses the following guidelines:

- If one of the adjusted limits is greater than 100, the limit is set to 100.
- If one of the adjusted limits is less than 0, the limit is set to 0.
- If the adjusted accept limit is less than the reject limit, the accept limit is set to the reject limit.
- If the accept match adjustment is 101, the accept limit is set to 101. As a result, SSA-NAME3 does not accept any record, and SSA-NAME3 either rejects a record or marks it as undecided.

  For example, if `MATCH_LEVEL=Loose+101-10`, the accept limit is set to 101.

  If a match call returns a score of 100, SSA-NAME3 marks the record as undecided.

  If a match call returns a score that is less than 100 and less than the reject limit, SSA-NAME3 rejects the record. If the score is less than 100 but greater than or equal to the reject limit, SSA-NAME3 marks the record as undecided.

Note: The `MATCH_LEVEL` control is not applicable to the `Geocode` purpose. If you specify the `MATCH_LEVEL` control with a `Geocode` purpose, SSA-NAME3 ignores the `MATCH_LEVEL` control.

You can use one of the following match level expressions:

matchlevel+/-nn

Adjusts the accept and reject limits by using the same specified number. For example:

```
MATCH_LEVEL=Typical+20
```

If the predefined accept limit is 79 and reject limit is 50, the adjusted accept limit is 99 and reject limit is 70.

matchlevel+/-nn+/-nn

Adjusts the accept and reject limits by using the specified numbers. For example:

```
MATCH_LEVEL=Typical-40+40
```

If the predefined accept limit is 79 and reject limit is 50, the calculated accept limit is 39, which is less than the reject limit of 90. The adjusted accept limit is set to 90, and the reject limit remains at 90.

NAMEFORMAT Control

Use the NAMEFORMAT control to define whether the major word in the name or address is on the left end or the right end. For example, in Western person names, the family name is on the right end of the names, and you can specify `NAMEFORMAT=R`.

Use the `NAMEFORMAT` control to override the default name format. For more information about the default name format for a given standard population field and its effects, from the **Help** menu of SSA-NAME3 Workbench, click **Population Documentation**.

You can provide extra weight to match the major words in the Person_Name field in some purposes such as Person_Name, Household, Family, and Wide_Household.

Note: The NAMEFORMAT control in a match call overrides the name format for all name and address fields in the match purpose, so use the NAMEFORMAT control with caution.

UNICODE_ENCODING Control

Use the UNICODE_ENCODING control to instruct MDM Registry Edition to accept Unicode data input, and specify the Unicode format of the data that you want to pass.

MATCH_OPTIONS Control

Use the MATCH_OPTOINS control to define the options for a specific field.

The MATCH_OPTIONS control uses the following syntax:

```
MATCH_OPTIONS=(fieldname:option,...)
```

Use the MATCH_OPTIONS control for the following fields:

Date

The Date field accepts the Date_Format option. The MATCH_OPTIONS control uses the following format for the Date_Format option:

```
MATCH_OPTIONS=DATE:Date_Format=<DDMMYY>|<MMDDYY>|<YYMMDD>
```

The values are not case sensitive.

Geocode

The Geocode field accepts the FORMAT and RADIUS options. The MATCH_OPTIONS control uses the following format for the FORMAT and RADIUS options:

```
MATCH_OPTIONS=GEOCODE:FORMAT=1,GEOCODE:RADIUS=10000
```

The FORMAT option indicates the order of the latitude and longitude coordinates that you specify.

If GEOCODE:FORMAT=1, SSA-NAME3 considers the geographic coordinates in the following order:

```
*Geocode*<Longitude> <Latitude>***
```

If GEOCODE:FORMAT=0, SSA-NAME3 considers the geographic coordinates in the following order:

```
*Geocode*<Latitude> <Longitude>***
```

The default value is 0.

The RADIUS option indicates the search radius to generate keys according to the latitude and longitude coordinates that you specify. The default value is 1000 m, and the default unit is meter.

Use the following guidelines when you specify the geocode radius:

- SSA-NAME3 considers the radius to be in meters even if you specify any other unit.
- SSA-NAME3 ignores the decimal values and uses the whole number as the radius. For example, if you specify the radius as 850.8, SSA-NAME3 considers the radius as 850.

Note: SSA-NAME3 ignores any incorrect or unrecognised options in the MATCH_OPTIONS control and proceeds with matching, which might result in unexpected scores.

SEARCH and FILE Controls

Use the SEARCH and FILE controls to specify the field names and their offset and length pairs in the scatter or gather format for the data in the `Search Data` and `File Data` fields. If you do not specify the offset and length pairs for the field names, SSA-NAME3 considers the data in the tagged data format.

You can extend any of the key fields and set the weight for the extended fields. The extended fields use the algorithm of the key fields. Use the extended fields to override the weight of the key fields in the run time. You can also specify the type of matching to perform between the data in the `Search Data` and `File Data` fields.

You can use the following matching types:

- Exact matching. Returns 100% score only if the search data values match with the file data values.
- Inexact matching. Returns 100% score if the search data values do not match with the file data values.
- Range matching. Returns 100% score if the search data and file data values are within the specified range.

Note: If you specify different matching types in the SEARCH and FILE parameters, the matching type that you specify in the SEARCH parameter takes precedence.

Exact Matching

The exact matching returns 100% score if the search data values match with the file data values. You can perform exact matching on any data type.

Use the following format to perform exact matching:

```
PURPOSE=<Purpose Name> SEARCH=<Field Name>[(<Field ID>;<Weight>;Exact[:B|Z|
ZB])],<Offset>,<Length> FILE=[(<Field ID>;<Weight>;Exact)],<Offset>,<Length>
```

If you want to extend a key field, use any number as the `Field ID` value. If you do not want to extend any key field, use 0 as the `Field ID` value. If you extend a key field in the search or file data, you must extend the corresponding key field in the search or file data.

The value `B` indicates a null value, the value `Z` indicates a zero value, and the value `ZB` indicates a null or zero value.

The following sample expressions perform exact matching between the search data and file data values:

- `PURPOSE=Person_Name SEARCH=Person_Name(Exact),0,16 FILE=Person_Name,0,16`. The expression returns 100% score only if the search data value exactly matches with the file data value.
- `PURPOSE=Fields SEARCH=Person_Name(2;5;Exact),0,16 FILE=Person_Name(2;Exact),0,16`. The expression creates an extended field for the `Person_Name` field named `Person_Name(2)` and sets the weight for the extended field to 5. The search data value returns 100% score if the search data value exactly matches with the file data value.
- `PURPOSE=Person_Name SEARCH=Person_Name(0;5;Exact),0,16 FILE=Person_Name,0,16`. The expression sets the weight of the `Person_Name` field to 5 without creating any extended field. The search data value returns 100% score if the search data value exactly matches with the file data value.
- `PURPOSE=Person_Name SEARCH=Person_Name(1;Exact),0,16 FILE=Person_Name(1;Exact),0,16`. The expression creates an extended field for the `Person_Name` field named `Person_Name(1)`. The extended field uses the weight of the `Person_Name` field. The search data value returns 100% score if the search data value exactly matches with the file data value.
- `PURPOSE=Fields SEARCH=Date,0,8 FILE=Date(Exact:Z),0,8`. The expression returns 100% score if the file data value is 0.

Inexact Matching

The inexact matching returns 100% score if the search data values do not match with the file data values. You can perform inexact matching on any data type.

Use the following format to perform inexact matching:

```
PURPOSE=<Purpose Name> SEARCH=<Field Name>[(<Field ID>;<Weight>;!Exact[:B|Z|
ZB])],<Offset>,<Length> FILE=[(<Field ID>;<Weight>;!Exact)],<Offset>,<Length>
```

If you want to extend a key field, use any number as the `Field ID` value. If you do not want to extend any key field, use 0 as the `Field ID` value. If you extend a key field in the search or file data, you must extend the corresponding key field in the search or file data.

The value `B` indicates a null value, the value `Z` indicates a zero value, and the value `ZB` indicates a null or zero value.

The following sample expressions perform inexact matching between the search data and file data values:

- `PURPOSE=Person_Name SEARCH=Person_Name(!Exact),0,16 FILE=Person_Name,0,16`. The expression returns 100% score if the search data value does not match with the file data value.
- `PURPOSE=Fields SEARCH=Person_Name(1;5;!Exact),0,16 FILE=Person_Name(1;!Exact),0,16`. The expression creates an extended field for the `Person_Name` field named `Person_Name(1)` and sets the weight for the extended field to 5. The search data value returns 100% score if the search data value does not match with the file data value.
- `PURPOSE=Person_Name SEARCH=Person_Name(0;5;!Exact),0,16 FILE=Person_Name,0,16`. The expression sets the weight of the `Person_Name` field to 5 without creating any extended field. The search data value returns 100% score if the search data value does not match with the file data value.
- `PURPOSE=Fields SEARCH=Person_Name(3;!Exact),0,16 FILE=Person_Name(3;!Exact),0,16`. The expression creates an extended field for the `Person_Name` field named `Person_Name(3)`, and the extended field uses the weight of the `Person_Name` field. The search data value returns 100% score if the search data value does not match with the file data value.
- `PURPOSE=Fields SEARCH=Date,0,8 FILE=Date(!Exact:ZB),0,8`. The expression returns 100% score if the file data value is not 0 or null.

Range Matching

The range matching returns 100% score if the search data and file data values are within the specified range. You can specify a range for the search data, the file data, or both to perform matching. You can perform range matching on numeric data and dates.

Use one of the following formats to perform range matching on numeric fields:

- `PURPOSE=<Purpose Name> SEARCH=<Field Name>[(<Field ID>;<Weight>;Range:<Negative Limit>;<Positive Limit>)],<Offset>,<Length> FILE=<Field Name>[(<Field ID>;<Weight>;Range:<Negative Limit>;<Positive Limit>)],<Offset>,<Length>`

  If you want to extend a key field, use any number as the `Field ID` value. If you do not want to extend any key field, use 0 as the `Field ID` value. If you extend a key field in the search or file data, you must extend the corresponding key field in the search or file data.

  For example, the `PURPOSE=Person_Name SEARCH=Person_Name,0,16,Attribute1(1;5;Range:20;10),16,2 FILE=Person_Name,0,16,Attribute1(1;Range),16,2` expression creates an extended field for the `Attribute1` field named `Attribute1(1)` and sets the weight for the extended field to 5. If the `Attribute1(1)` value in the search data is 100, the range for the search data is 100-20 to 100+10, which is 80 to 110. The range matching returns 100% score if the `Attribute1(1)` value in the file data is within the search data range.

  If both the positive and negative limits are the same, you can specify the range in the `(Range:<Limit>)` format. For example, `(Range:20)` equals `(Range:20;20)`.

- `PURPOSE=<Purpose Name> SEARCH=<Field Name>[(<Field ID>;<Weight>;Range:%<Negative Limit>;<Positive Limit>)],<Offset>,<Length> SEARCH=<Field Name>[(<Field ID>;<Weight>;Range:%<Negative Limit>;<Positive Limit>)],<Offset>,<Length>`

  If you want to extend a key field, use any number as the `Field ID` value. If you do not want to extend any key field, use 0 as the `Field ID` value. If you extend a key field in the search or file data, you must extend the corresponding key field in the search or file data.

  For example, the `PURPOSE=Person_Name SEARCH=Person_Name,0,16,Attribute1,16,2 FILE=Person_Name,0,16,Attribute1(Range:%10;20),16,2` expression indicates that if the `Attribute1` value in the file data is 100, the range for the file data is 100-10% of the `Attribute1` value to 100+20% of the `Attribute1` value, which is 90 to 120. The range matching returns 100% score if the `Attribute1` value in the search data is within the file data range.

  If both the positive and negative limits are the same, you can specify the range in the `(Range:%<Limit>)` format. For example, `(Range:%20)` equals `(Range:%20;20)`.

- `PURPOSE=<Purpose Name> SEARCH=<Field Name>[(Range:F|S)],<Offset>,<Length> FILE=<Field Name>[(Range:F|S])<Offset>,<Length>`

  The value F indicates to use the file data to create the range, and the value S indicates to use the search data to create the range.

  For example, the `PURPOSE=Fields SEARCH=Date(Range:F),0,4 FILE=Date(Range:F),0,4,Date(Range:F),4,4` expression indicates that if the file data is 50008000, the range for the search data is 5000 to 8000. The range matching returns 100% score if the `Date` value in the search data is within the file data range.

Use one of the following formats to perform range matching on date fields:

- `PURPOSE=<Purpose Name> SEARCH=<Date Field Name>[(Range:F|S)],<Offset>,<Length> FILE=<Date Field Name>[(Range:F|S]),<Offset>,<Length>`

  The value `F` indicates to use the file data to create the range, and the value `S` indicates to use the search data to create the range.

  For example, the `PURPOSE=Fields SEARCH=Date(Range:F),0,8 FILE=Date(Range:F),0,8,Date(Range:F),8,8` expression indicates that if the file data is 1999101019991020, the range for the search data is 19991010 to 19991020. The range matching returns 100% score if the Date value in the search data is within the file data range.

- `PURPOSE=<Purpose Name> SEARCH=<Date Field Name>[(DateRange:<Negative Limit>;<Positive Limit>)],<Offset>,<Length> SEARCH=<Date Field Name>[(DateRange:<Negative Limit>;<Positive Limit>)],<Offset>,<Length>`

  For example, the `PURPOSE=Fields SEARCH=Date(DateRange:F),0,8 FILE=Date(DateRange:5;10),8,8` expression indicates that if the `Date` value in the file data is 20150410, the range for the file data is 20150405 to 20150420. The range matching returns 100% score if the `Date` value in the search data is within the file data range.

  If both the positive and negative limits are the same, you can specify the range in the `(DateRange:<Limit>)` format. For example, `(DateRange:7)` equals `(DateRange:7;7)`.

- `PURPOSE=<Purpose Name> SEARCH=<Date Field Name>[(DateRange:%<Negative Limit>;<Positive Limit>)],<Offset>,<Length> SEARCH=<Date Field Name>[(DateRange:%<Negative Limit>;<Positive Limit>)],<Offset>,<Length>`

  For example, the `PURPOSE=Fields SEARCH=Date,0,8 FILE=Date(DateRange:%10;20),0,8` expression indicates that if the `Date` value in the file data is 19991010, the range for the file data is 10-10% of the `Date` value to 10+20% of the `Date` value, which is 19991009 to 19991012. The range matching returns 100% score if the `Date` value in the search data is within the file data range.

  If both the positive and negative limits are the same, you can specify the range in the `(DateRange:%<Limit>)` format. For example, `(DateRange:%20)` equals `(DateRange:%20;20)`.

FILTER_SEARCHVALUES Control

Use the FILTER_SEARCHVALUES control to specify a list of values to match with the data in the Search Data field, File Data field, or both the fields. Use the Filter purpose to specify the data in the Search Data and File Data fields.

The FILTER_SEARCHVALUES control uses the following format to specify a list of search values:

```
PURPOSE=(Filter<Filter Purpose Number>,...) FILTER_SEARCHVALUES=<Filter Purpose Number>,<List of Flags>,(<List of Values>)|{<Value>,(<List of Values>)}
```

For example:

```
PURPOSE=(Filter2 AND Filter3) FILTER_SEARCHVALUES=2,SP,(WILLIAM,BILL),3,FP,{ROBERT,(ROB,ROBBIE)}
```

The FILTER_SEARCHVALUES control uses the following parameters:

Filter Purpose Number

Indicates the Filter purpose number for which you define a list of values.

For example, if you use the Filter5 purpose to specify the search data, use 5 as the Filter purpose number.

(List of Values)|{Value,(List of Values)}

Indicates the list of values to match with the search data, file data, or both. You can specify a list of values, or you can pair a value with a list of values.

For example:

```
FILTER_SEARCHVALUES=2,SP,(WILLIAM,BILL),3,FP,{ROBERT,(ROB,ROBBIE)}
```

Use appropriate flags to indicate whether to match the specified values with the search data, file data, or both.

List of Flags

Indicates whether to match the specified values with the search data, file data, or both. You can use multiple flags. For example, FP.

If you specify a list of values, you can use the following flags:

- S. Indicates to match the values with the search data.
- F. Indicates to match the values with the file data.
- B. Indicates to match the values with both the search and file data.
- P. Indicates to match the values in the list with the search data, file data, or both based on the length of the values in the list. This flag functions in conjunction with other flags.

  For example, specify `FILTER_SEARCHVALUES=2,SP,Rob` as one of the SCORE-LOGIC controls and `*Filter2*Rose***` in the search data. The length of `Rob` is three characters, so the matching considers only the first three characters of the search data. The matching does not return 100% score because `Rob` in the list does not exactly match with the first three characters of the search data, which are `Ros`.

If you pair a value with a list of values, you can use the following flags:

- S. Indicates to match the value with the search data and the list of values with the file data.
- F. Indicates to match the value with the file data and the list of values with the search data.
- P. Indicates to match the values in the list with the search data, the file data, or both based on the length of the values in the list. This flag functions in conjunction with other flags.
  For example, the `FILTER_SEARCHVALUES=3,S,{ROBERT,(ROB,ROBBIE)}` control indicates to match `Robert` with the search data and the list of values, `Rob and Robbie`, with the file data.

You can also perform inexact matching to get 100% score if the list values do not match with the search or file data.

To perform inexact matching, use the following format when you define the Filter purpose:

```
PURPOSE=(NOT Filter<Number>, NOT Filter<Number>,...)
```

The usage of NOT indicates that you want to perform inexact matching. For example, the `PURPOSE=(NOT Filter2) FILTER_SEARCHVALUES=2,SP,(WILLIAM,BILL)` expression returns 100% score if the list values do not match with the search data.

COMBINE Control

If you specify the `COMBINE` field, it must be present in the list of Matching-Fields. Multiple fields may be specified for the `COMBINE` option. When `COMBINE` is defined for a field, multiple data fields in `Matching-Fields` of that `Field` type are concatenated together before the matching can occur.

An optional combine option may be associated with the fieldname which can have the following possible values:

DELIM-NONE: If this option is set, when concatenating multiple data fields there is no space embedded.

DELIM-SPACE: If this option is set, when concatenating multiple data fields there is a space embedded. This is the default and need not be explicitly specified.

For example, with the following controls specified,

```
Controls ("PURPOSE=Generic COMBINE=Person_Name,Address_Part1:DELIM-
NONE,Telephone_Number")
```

All fields of type Person_Name will be concatenated into a single field with a space between them. Similarly, all fields of type Telephone_Number will be concatenated into a single field with a space between them. All field of type Address_Part1 will be concatenated with no space embedded between them.

Advanced Controls

The following section explains about the more advanced SSA-NAME3 Controls.

UNICODE_ENCODING

The ssan3_get_keys_encoded, ssan3_get_ranges_encoded and ssan3_match_encoded calls have a Field Data Type parameter that can be used to specify the encoding type. You can use this control with the ssan3_get_keys, ssan3_get_ranges and ssan3_match function calls. The section on UTF-8 considerations is relevant to all API function calls.

This Control is used in the ssan3_get_keys, ssan3_get_ranges and ssan3_match function calls, and instructs MDM Registry Edition to accept Unicode data input. It can also be used to specify non-Unicode encodings as described in the following table.

Value	Description	Short form
TEXT	Data is not in Unicode encoding	
UTF-8 or UTF8	Unicode UTF-8 format	Y or 8
UTF-16 or UTF16	Unicode UTF-16 format	6
UTF-16LE or UTF16LE	Unicode UTF-16 format Little Endian	L
UTF-16BE or UTF16BE	Unicode UTF-16 format Big Endian	B
UTF-32 or UTF32	Unicode UTF-32 format	4
UCS-2 or UCS2	Same as Unicode UTF-16 format	
UCS-4 or UCS4	Unicode UTF-32 format	4
CP932	Japanese CP932 code page (shift-JIS)	J
CP936	Chinese CP936 code page (GBK or Simplified Chinese)	S
CP949	Korean CP949 code page	K
CP950	Chinese CP950 code page (Big5 or Traditional Chinese)	T

CP300	DBCSHost Japanese	D
DBCSHOST	DBCSHost Japanese	
EUC	Extended UNIX Code	E

Note: If the Field Data Type is W, the UTF-16 Unicode encoding is automatically set.

All call parameters use single-byte except for `Key Field Data` in the `ssan3_get_keys` and `ssan3_get_ranges` calls and `Search Data` and `File Data` in the `ssan3_match` call.

When passing Unicode data, all length and offset values must be number of bytes, not number of characters. UTF-8 is a variable length encoding so the number of characters represented will have varying lengths based on the content of the data.

Scatter/Gather Data Format must be used except for UTF-8 encoding in which case the following notes must be considered.

UTF8 Considerations

If the UNICODE UTF-8 encoding is being used and care is taken, all parameters can be treated as Unicode.

All return parameters are single character UTF-8 except for data from the MDM Registry Edition Info calls which might contain national characters.

For the Tagged Data Format, the UTF-8 encoding can be used if the delimiter character consists of the single-byte UTF-8 characters. All the standard (unaccented) characters (A-Z and a-z) and digits (0-9) and many punctuation characters are represented in UTF-8 as a single 8-bit character.

In UTF-8, if the character code is less than 128 (00 to 7F in hexadecimal), it is a single-byte character. Otherwise, it could be a 2-byte to 6-byte character.

Everything between the delimiter characters will be passed unchanged.

It is also possible to use a multibyte UTF-8 character or characters for the `DELIMITER`.

PURPOSE <expression>

The `PURPOSE` Control specifies the name of the Matching Purpose to use in the Match call.

It takes the following form:

```
PURPOSE=(<expression>)
```

Where `<expression>` is one of the following formats:

```
<expression> := <Purpose_Name>
<expression> := <Purpose_Name>(<Match_level>)
<expression> := not <expression>
<expression> := <expression> or <expression>
<expression> := <expression> and <expression>
<expression> := (<expression>)
```

Note: The `<expression>` requires parentheses () whenever there are embedded spaces, especially when you use `not`, `and`, and `or`.

The simple form of the Purpose `<expression>`:

```
PURPOSE=<Purpose_Name>
```

is also the most commonly used format. For example: `PURPOSE=Address`

will cause a match to be done on the supplied Address fields to determine the match purpose "same address."

The form of the <expression>:

```
PURPOSE=<Purpose_Name>(<Match_level>)
```

For example: `PURPOSE=Address(Conservative)`

is the same as specifying: `PURPOSE=Address MATCH_LEVEL=Conservative`

However, if both ways of specifying a Match Level are used, the value specified locally associated

with each of the purposes will take precedence over the match level specified by the `MATCH_LEVEL` keyword. Purposes that do not have an individual match level specified will adopt the match level specified by the `MATCH_LEVEL` keyword.

Combining multiple expressions, gives the application designer more flexibility in choosing the method in which match decisions and scores are computed.

Multi-Purpose Matching

Another use for combining multiple expressions will be to achieve Multi-Purpose Matching. When multiple Purposes are used, it is important to note the following:

- The Multi-Purpose expression is evaluated in a strict left to right order.
- Early exit from the match process is possible after evaluation of the first Purpose.

All Purposes in the expression share the same data as passed in the `Search Data` and `File Data` fields.

If two Purposes share the same field and both Purposes must be evaluated, the field is evaluated twice. It is because the field might have been defined with different match options based on which Purpose it is in.

One example of Multi-Purpose matching is to create an early exit condition that is likely to increase performance. This would be either an early "accept" where Purposes are joined by an `OR`, or an early "reject" in the case of an `AND`. Again, note that early exit from the match process is possible after evaluation of the first Purpose.

For example, if in a typical Resident purpose, it is logically possible to reject on the basis of "Address" not passing a Conservative match. The following expression might be used:

```
PURPOSE=(Address(Conservative) AND Resident(Typical))
```

If the Address purpose receives a "Reject" decision from the Conservative match, the Resident Purpose is not evaluated as the AND has failed. However, if the Address purpose does not result in a Reject condition, the Resident purpose is thus evaluated. Note that the address data will be rematched as part of the Resident purpose, in addition to the `Person_Name` field.

In this example, overall performance of an online system or the run-time of a batch job improves, the more often early exit occurs. Conversely, performance can decrease the more often both Purposes need to be evaluated (because the address field must be evaluated for each Purpose).

Another example of Multi-Purpose matching is to return a superset of matches, such as:

```
PURPOSE=(Individual OR Resident)
```

This expression accepts the matches where the same Individual (Name + (Date of Birth or ID Number)) or the same Resident (Name + Address) is present. In this example, the `Match_Level` Control will be used to apply to both Purposes.

A third example is for mixing Match Levels. For example:

```
PURPOSE=(Contact(Typical)OR

Wide_Contact(Conservative))
```

This expression accepts matches if either `Person_Name, Organization_Name` and `Address` match at the `Typical` level, or `Person_Name` and `Organization_Name` match at the `Conservative` level.

In the previous examples, again note the performance impact of both Purposes being evaluated. It is based on the fact that certain fields would be evaluated twice).

When combining Purposes with `<and, or, not>`, the Purposes are evaluated in a left-to-right order. If the score/decision from the first Purpose invalidates the expression, no further processing is done. For more information about the score/decision processing, see the *Design Guide*.

Lightweight Matching

Lightweight matching uses a fast score estimate to reject the obvious mismatches. SSA-NAME3 performs full scoring for the remaining records, which results in improved performance.

Use the following controls to configure lightweight matching:

LWM=Y/N/ONLY

Enables or disables lightweight matching. Use the value `Y` to enable lightweight matching. Lightweight matching uses a fast score estimate to reject the obvious mismatches. The records that lightweight matching passes go to the full scoring for robust scoring and ranking. SSA-NAME3 returns the full score and the decision to the caller.

Note: If you create system definition files by using the SDF Wizard, the lightweight matching is enabled by default.

Use the value `N` to disable lightweight matching. SSA-NAME3 matching performs full scoring on all the matching records.

Use the value `ONLY` to enable lightweight matching and disable full scoring. Lightweight matching returns the estimate as the final score to the caller.

LWM_FIELDS

Specifies the fields to which you want to apply lightweight matching and their weights. These values override the values that you have defined in the match purpose during the run time. Based on the lightweight matching scores, SSA-NAME3 rejects the obvious mismatches. If you do not set any value, SSA-NAME3 retrieves the fields from the match purpose and assigns equal weight to them.

The syntax of the LWM_FIELDS control is as follows:

```
LWM_FIELDS=<field1>,<weight1>[,...,<fieldn>,<weightn>]
```

where `field` is a valid field name that you have defined in the Purpose control, and `weight` is the relative significance of the specified field (0-100) when compared to the other fields.

For example, `LWM_FIELDS=Person_Name,5,Address_Part1,1`

Lightweight matching is useful when you apply it to the fields that have low variations such as addresses. Lightweight matching is not efficient for the fields with high variations, where SSA-NAME3 handles the variations through Edit-list, and lightweight matching might incorrectly reject the records.

LWM_LIMIT

Specifies the accept and reject limits for the lightweight matching score. Based on the limits, SSA-NAME3 accepts or rejects the search results.

The syntax of the LWM_LIMIT control is as follows:

```
LWM_LIMIT=<Reject>[,<Accept>]
```

where `Reject` and `Accept` are the integer values ranging from 0 through 100.

For example, `LWM_LIMIT=50,90`

If `LWM=N`, the `LWM_LIMIT` control has no effect.

If `LWM=Y`, SSA-NAME3 rejects the lightweight matching scores that are less than the reject limit. The accept limit has no effect, and you can omit it.

If `LWM=ONLY`, SSA-NAME3 rejects the lightweight matching scores that are less than the reject limit. It accepts the scores that are greater than the accept limit. It marks the scores of the records that are greater than or equal to the reject limit and less than the accept limit as undecided.

The default reject limit is 65, and the default accept limit is 90. If you have not set the accept limit and the reject limit is greater than 90, the accept limit is equal to the reject limit.

CHAPTER 3

Standard Population Choices

This section is designed to help the analyst, designer or developer make the right choices when choosing the Standard Population and the search and match controls, levels and data to use in the search application.

Standard Populations

MDM - Registry Edition is delivered with over 60 Standard Populations covering different countries, languages and regions.

As new Standard Populations are added regularly, the most current list is that which is shown by the Informatica IR Product Installer.

Before installing MDM - Registry Edition, an analysis should be done of the data that is to be searched and matched. The following questions need to be addressed:

- Which country(ies) is it from?
- What codepage is it in?
- Does it contain mixed scripts?

When installing MDM - Registry Edition, choose the Standard Population(s) that suit the data you will be searching and matching.

An Informatica Corporation consultant can be contacted for assistance with the decision. In many cases the decision will be simple (example, a USA company doing business in the USA alone would choose the USA Standard Population).

Note: All standard populations currently supported by Informatica Corporation are delivered with the MDM - Registry Edition install. However, some require a separate license to use.

If you have selected a Population during the install process that requires a separate license, a license warning screen will be shown prompting verification that the license is held.

Currently, the Standard Populations requiring a separate license are:

- The Chinese, Japanese and Korean double-byte populations;
- The Arabic Mixed population (supporting bi-directional Arabic / Latin searching and matching)

Population Files Reference

The current population information files.

Country/Language/ Purpose	Population File Name	Notes	Unicode Support	Cross Script
AML	aml.ysp	Special Purpose Anti-Money-Laundering Population	Yes	N/A
Arabic	arabic.ysp	Arabic Script, CP708	No	No
	arabic_m.ysp	Arabic & Romanized Script, CP1256	Yes	Yes
	arabic_r.ysp	Romanized Arabic	Yes	No
Argentina	argentina.ysp		Yes	N/A
Australia	australia.ysp		Yes	N/A
Belgium	belgium.ysp		Yes	N/A
Brasil	brasil.ysp		Yes	N/A
Bulgaria	bulgaria_c.ysp	Bulgarian Cyrillic Script	Yes	No
	bulgaria_r.ysp	Bulgarian Romanized Script, CP1251	Yes	N/A
Canada	canada.ysp		Yes	N/A
Chile	chile.ysp		Yes	N/A
Chinese	chinese.ysp	New Chinese Multi-Script (introduced in version 9.0.1)	Yes	Yes
	chinese_r.ysp	Chinese Romanized Script	Yes	N/A
	chinese_s.ysp	Chinese in Romanized and Simplified Script, CP936	Yes	Yes
	chinese_t.ysp	Chinese in Romanized and Traditional Script, CP950	Yes	Yes
Colombia	colombia.ysp		Yes	N/A
Croatia	croatia.ysp	CP1250	Yes	N/A
Czech	czech.ysp	CP1250	Yes	N/A
Denmark	denmark.ysp		Yes	N/A
Estonia	estonia.ysp		Yes	N/A
Finland	finland.ysp		Yes	N/A
France	france.ysp		Yes	N/A

Gaelic	gaelic.ysp		Yes	N/A
Germany	germany.ysp		Yes	N/A
Greek	greek.ysp	CP928	Yes	No
	greek_l.ysp	CP928L	Yes	No
Hebrew	hebrew.ysp	CP1255	Yes	No
Hong Kong	hk_r.ysp	Chinese in Romanized Script Localized for Hong Kong	Yes	N/A
Hungary	hungary.ysp	CP1250	Yes	N/A
India	india.ysp		Yes	N/A
INDO_CHIN_R	indo_chin_r.ysp		Yes	N/A
Indonesia	indonesia.ysp		Yes	N/A
International	international.ysp		Yes	N/A
Ireland	ireland.ysp		Yes	N/A
Italy	italy.ysp		Yes	N/A
Japan	japan.ysp	Japanese Mixed Script (Kanji, Kana, Romanized) , CP932, Shift-JIS	Yes	Yes
	japan_r.ysp	Japanese in Romanized Script	Yes	N/A
Kazakhst	kazakhstan.ysp		Yes	Yes
Korean	korean.ysp	Korean Mixed Script (Korean, Romanized), CP949	Yes	Yes
	korean_r.ysp	Korean in Romanized Script	Yes	N/A
Luxembourg	luxembourg.ysp		Yes	N/A
Malaysia	malaysia.ysp		Yes	N/A
Malta	malta.ysp		Yes	N/A
Mexico	mexico.ysp		Yes	N/A
Netherlands	netherlands.ysp		Yes	N/A
New Zealand	new_zealand.ysp		Yes	N/A
Norway	norway.ysp		Yes	N/A
OFAC	ofac.ysp	Special Purpose tuned for data in US OFAC list	Yes	N/A

Peru	peru.ysp		Yes	N/A
Philippines	philippines.ysp		Yes	N/A
Poland	poland.ysp	CP1250	Yes	N/A
Portugal	portugal.ysp		Yes	N/A
Puerto Rico	puerto_rico.ysp		Yes	N/A
Romania	romania.ysp	CP1250	Yes	N/A
Russia	Russia.ysp	Russian Cyrillic (Renamed from Cyrillic.ysp in version 9.2), CP1251	Yes	Yes
Singapore	singapore.ysp		Yes	N/A
Slovakia	slovakia.ysp	CP1250	Yes	N/A
South Africa	south_africa.ysp		Yes	N/A
Spain	spain.ysp		Yes	N/A
Sweden	sweden.ysp		Yes	N/A
Switzerland	switzerland.ysp		Yes	N/A
Taiwan	taiwan_r.ysp	Chinese in Romanized Script Localized for Taiwan	Yes	N/A
Thai	thai.ysp	Thai Native Script, CP874	Yes	No
	thai_r.ysp	Thai in Romanized script	Yes	N/A
Turkey	turkey.ysp	CP1254	Yes	N/A
UK	uk.ysp		Yes	N/A
USA	usa.ysp		Yes	N/A
Vietnam	vietnam_r.ysp	Vietnamese in Romanized Script	Yes	N/A

A Primer on Keys and Search Strategies

The safest way of finding a name match in a database is to first perform a search on an index built from name alone, thus building a candidate list of possible matches, and then to refine, rank or select the matches in that candidate list based on other identification data.

Name only keys are built from one or more parts of the name field (words & words, words & initials). Of course the method used for constructing the database keys must match the method used for constructing the search keys.

The more name parts used in the key, and the greater the number of keys built per name, the greater the variety of search strategies which can be supported.

A name key for "ANN JACKSON-SMITH" built from family name plus first initial, "SMITH A", can support search strategies using the family name word and initial and also using only the single family word. A name key built from family name and first name, "SMITH ANN" can support search strategies using two words from the name (at the "two word level" or wider). The fewer words used in the key the larger or wider the set of responses will be.

An extra name key, say "JACKSON ANN", supports a search where the search name is missing a certain part or the parts are in a certain different order.

The choice of keys and search strategies together defines the width or depth of the search (by the number of name parts used in the search keys) and the degree of sequence variations and missing parts overcome (by the number of different keys).

The greater the number of name parts used in a search key, the fewer candidates on average will be returned, and the quicker the search. A search strategy which uses the full name makes sense when the name is expected to be generally reliable, when the match is expected to be in the database, or when the search will be stopped, or at least interrupted, at the first match. This type of search strategy is thought of as a Typical search and is used to find data that is expected to be on file.

As confidence in the quality of the search or database names declines, or as the risk of missing a match increases, so will the need for a different search strategy arise. A high-risk search, or a search using poor quality data, should use a wider search strategy to compensate for severe spelling errors and more sequence variations, missing and extra words in the names. This type of search strategy is thought of as an Exhaustive search and is frequently used to prove that data is not on file.

In large scale systems the choice and sophistication of the search strategy is consequential to both performance demands, risk of missing critical data, need to avoid duplication of data and the volume of data under indexing.

The choice of search strategy should match the business needs of the search. The search strategy used for one set of data or one system may be very different from that used in another.

A search strategy is affected by decisions on the following Standard Population components:

- Key Field - the field to use for indexing and search
- Key Level - the type of keys built
- Search Level - the breadth of search performed

Matching, filtering and ranking of the candidates returned from a search is affected by decisions on the following Standard Population components:

- Match Purpose - the fields used in Matching and the business purpose of the Match
- Match Level - the degree of Match chosen

Key Fields

By using standard populations, you can set up an application to index and search the following field types:

- Person Names
- Organization Names
- Addresses

- Code Fields
- Geocodes

Person Names

Pass `FIELD=Person_Name` to SSA-NAME3 in the Controls parameter of the get_keys or get_ranges calls. The field runs an algorithm that builds the keys and the search ranges for person names.

Using MDM Registry Edition, it is passed in the `KEY-LOGIC` and `SEARCH-LOGIC` Controls parameters.

The `Person_Name` algorithm fixes the errors and variations that are found in a person's full name. They include salutations and honorifics, special characters, embedded spaces, nicknames, different word orders, use of initials, spelling errors, concatenated words, localized words, or foreign words.

An application passes the full person name to the SSA-NAME3 functions. The word order represents the position of the first name, middle names, and family names. It must be the normal word order used in your data population. For example, in English speaking countries, the normal word order is as follows:

```
First Name + Middle Name(s) + Family Name(s)
```

Based on your table design, your application might have to concatenate these separate fields into one field before calling SSA-NAME3. Using MDM Registry Edition this can be done with `TRANSFORM` statements.

SSA-NAME3 includes Search Strategies that overcome word order variations. However, the word order does have some significance in the quality of Narrow and Typical searches and when matching using the Purposes "same Household," "same Family," or "same `Wide_Household`."

The application (or MDM Registry Edition) might pass multiple names (such as a married name and a former name) in the one call to SSA-NAME3.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Person_Name` field.

Organization Names

Pass `FIELD=Organization_Name` or `FIELD=Organisation_Name` to SSA-NAME3 in the Controls parameter of the get_keys or get_ranges calls. It invokes the algorithm that builds the keys and search ranges for the organization names.

Using MDM Registry Edition, it is passed in the `KEY-LOGIC` and `SEARCH-LOGIC` Controls parameters.

The `Organization_Name` algorithm overcomes the error and variation that would be found in a business, company, institution, or other organization name. This algorithm also caters for multiple names in the one field, and a mixture of Organization and Person names in the data population. The error and variation might include different legal endings, abbreviations, salutations and honorifics, special characters, embedded spaces, nicknames, different word orders, missing and extra words, spelling errors, concatenated words, use of initials, mixed use of numbers and words, foreign words, or localization.

This field supports matching on a single name, or a compound name (such as a legal name and its trading style).

MDM Registry Edition might also pass multiple names (such as a current name and a former name) in the one call to SSA-NAME3.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Organization_Name` field.

Addresses

Pass `FIELD=Address_Part1` to SSA-NAME3 in the Controls parameter of the get_keys or get_ranges calls. It invokes the algorithm that builds the keys and search ranges addresses. MDM Registry Edition passes the value to the `KEY-LOGIC` and `SEARCH-LOGIC` Controls parameters.

The `Address_Part1` algorithm overcomes the error and variation that would be found in addresses. The error and variation can include the presence of care of information, abbreviations, special characters, embedded spaces, different word orders, spelling errors, concatenated words and numbers, use of initials, mixed use of numbers and words, foreign words, missing words, and extra words and sequence variations.

An application should pass the part of address up to, but not including, the locality "last line". The word order, i.e. the position of the address components, should be the normal word order used in your data population. These should be passed in one field. Depending on your table design, your application may need to concatenate these attributes into one field before calling SSA-NAME3.
Using MDM Registry Edition this can be done with TRANSFORM statements. For example, in the U.S., a typical string to pass would comprise of:

```
Care-of + Building Name + Street Number + Street Name + Street Type + Apartment Details
```

But not including `City`, `State`, `Zip`, `Country`.

The application (or MDM Registry Edition) might pass multiple addresses (such as a residential address and a postal address) in the one call to SSA-NAME3. For more information, see the *API Reference Guide*.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Address_Part1` field.

Code Fields

The code fields are a group of fields that support numeric or alphanumeric values. The code fields include the following fields:

- Code
- Telephone_Number
- Date
- CreditCard
- VIN
- ISBN10 or ISBN13

The code fields support the following Match Purposes:

- CC_Owner
- CC_Issuer
- VIN_Owner
- VIN_Manufacturer
- AuthorISBN
- PublisherISBN

Note: The code fields are available for all the standard populations except the OFAC population.

Pass `FIELD=<Code Fields>` to SSA-NAME3 in the Controls parameter of the get_keys or get_ranges calls, where `Code Fields` can be one or more of the supported code fields. It invokes the algorithm that builds the keys and search ranges for the code fields. MDM Registry Edition passes the value to the `KEY-LOGIC` and `SEARCH-LOGIC` Controls parameters.

The algorithms of the code fields overcome any errors or variations that are common across the code entities, such as missing digits, different digit orders, and special characters.

MDM Registry Edition can pass multiple code fields as one code field in the form of a match purpose to SSA-NAME3.

For example, if you search for a credit card to find the customer name and the credit card issuer, you can use the `CC_Issuer` purpose.

The `Code` field matches any numeric or alphanumeric data. The algorithms of the `Telephone_Number`, `Date`, `VIN`, `ISBN10` or `ISBN13`, and `CreditCard` fields are specific to their processing domain. They carry less overhead than the generic `Code` field. However, the `Code` field algorithm reduces the overhead associated with the creation and maintenance of separate indexes. The application simplifies the search experience by not categorizing the search data.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Code` field.

Geocode

Pass `Field=Geocode` to SSA-NAME3 in the Controls parameter of the get_keys or get_ranges call to build keys and search ranges based on the latitude and longitude geographic coordinates. You can pass the Geocode field in the `KEY-LOGIC` and `SEARCH-LOGIC` Controls parameter.

The Geocode field matches the location or creates a search range based on the geographic coordinates that you specify.

You can specify the latitude and longitude geographic coordinates with an optional elevation. Specify the elevation value with its unit after the geographic coordinates. The supported units are feet (ft), kilometer (km), and meter (m). Negative elevation represents depths below the sea level.

For example, -23.399437,-52.090904, 203m

Use one of the following formats to specify the geographic coordinates:

- -23.399437,-52.090904, 203m
- 40:26:46.302N 079:56:55.903W 192ft
- 40°26'47"N 079°58'36"W 0.203km
- 40d 26' 47" N 079d 58' 36" W 203
- 40.446195N 79.948862W
- 40.446195 -79.948862
- 40° 26.7717, -79° 56.93172
- N40:26:46.302 W079:56:55.903
- N40°26'47 W079°58'36"
- N40d 26' 47" W079d 58' 36"
- N40.446195 W79.948862

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Geocode` field. Define a geocode override rule as a character rule, which replaces or removes strings or numbers that include special characters or delimiters.

Key Levels

Using Standard Populations, a user's database may be indexed on Person Names, Organization Names and Addresses using one of three Key Levels:

- Standard
- Extended

- Limited

The choice of Key Level is passed to the SSA-NAME3 "get keys" function directly by the user's application, or via the `KEY-LOGIC` Controls parameter in MDM Registry Edition, using the `KEY_LEVEL= Control`.

Standard

Standard is the recommended Key Level for typical applications. Its use overcomes most variations in word order, missing words and extra words.

It also maximizes the likelihood of finding candidates in cases of severe spelling error in multi-word names.

Standard is the default if no Key Level is specified.

Standard Keys or Extended Keys should be implemented if the Edit Rule Wizard is being used.

Extended

For high-risk or critical search applications, SSA-NAME3 can generate "Extended" Keys. Extended Keys extend Standard Keys by adding more keys based on token concatenation. The designer/developer should be aware that the use of Extended Keys will increase disk space requirements and result in larger candidate sets at search time. However, the intended use of Extended Keys is to improve reliability by finding matches regardless of word order variation and concatenation.

Standard Keys or Extended Keys should be implemented if the Edit Rule Wizard is being used.

Limited

If disk space is limited, SSA-NAME3 can generate "Limited" Keys. Limited Keys are a subset of Standard Keys. The designer/developer should be aware that the use of Limited Keys, while saving on disk space, may also reduce search reliability.

Search Levels

Using Standard Populations, an application may be set up to search on Person Names, Organization Names and Addresses using four different Search Levels:

- Typical
- Exhaustive
- Narrow
- Extreme

The choice of Search Level is passed to the SSA-NAME3 "get ranges" function directly by the user's application, or through the `SEARCH-LOGIC` Controls parameter in MDM Registry Edition, using the `SEARCH_LEVEL=` Control.

It is good practice to test using different Search Levels on real production data and volumes to measure both the response time and the reliability differences.

Typical

A Typical search level for most applications will provide a practical balance between quality and response time. It should be used in typical online or batch transaction searches. It is the default if no Search Level is specified.

For `Person_Name` searches, it is designed to find common, but not extreme, error and variation including cases where initials are present instead of full given names and where the initial of a name has changed due to the internal rules applied.

For `Organization_Name` searches, it is designed to find common, but not extreme, error and variation including instances of word concatenation.

For `Address_Part1` searches, it is designed to find common, but not extreme, error and variation.

Exhaustive

An Exhaustive search level is provided for applications that have an increased risk associated with missing a match, where data quality is a concern or where data volumes are low enough to make it the default search. It increases the number of candidates returned and consequently response times may be extended. An Exhaustive search will occasionally find matches that a Typical search misses, however, these will generally be where there is more extreme error and variation.

For `Person_Name` searches, it is designed to find more error and variation than a Typical search, especially where there is extreme spelling error in the family or middle names.

For `Organization_Name` searches, it is designed to find more error and variation than a Typical search, especially where there is extreme spelling error in the major word or trailing words.

For `Address_Part1` searches, it is designed to find more error and variation than a Typical search, especially where there are more cases of missing words, extra words or sequence differences.

Narrow

A Narrow search level compromises on completeness of search in favor of faster and more direct answers. It may be an option in search applications that do not have a high risk associated with missing a match, require very tight levels of matching, or where data volumes are extreme and response time is a critical factor.

For `Person_Name` searches, it is designed to find the very common error and variation including cases where initials are present instead of full given names.

For `Organization_Name` searches, it is designed to find the very common error and variation and primarily where the words are in a stable order.

For `Address_Part1` searches, it is designed to find the very common error and variation and primarily where the tokens are in a stable order.

Extreme

An Extreme search level uses every possibility to discover a candidate match; consequently response times may be extended. It is provided for applications that have a critical need to find a match if one is present in the database, despite the error and variation.

An Extreme search may only occasionally find matches that an Exhaustive search misses, however, because the risk is very high, every possible match is deemed important.

The types of candidates returned for all Field types is the same when using an Extreme search. Extreme spelling error is picked up in names or addresses with two or more words or tokens.

Match Purposes

MDM Registry Edition uses the SSA-NAME3's matching services to filter, rank, and match the candidate records returned from a search.

The identity data from the search is compared to the identity data from the candidate record, and a score or a ruling is returned. Pre-built matching algorithms are provided to address today's common business purposes. These are called "Match Purposes."

In combination with the Match Purpose, a selectable Match Level determines the tightness or looseness of the match. The application might also override the Score threshold, which determines the match ruling returned.

SSA-NAME3 Matching compensates for the error and variation in identity data. The matching logic consists of heuristic algorithms that are optimized for each class of data such as name, organization, address, dates, and codes. The algorithms include numerous rules and switches to handle initials, aliases, common variations, prefixes, suffixes, transpositions, and word order.

Additionally, all Match Purposes use string cleaning routines, Edit-Lists, different matching Methods for different data types, optimized Matching options, field and token level weighting, and phonetic or orthographic stabilization.

Each Match Purpose supports a combination of required and optional fields, and each field is weighted according to its influence in the match decision. Some fields in some Purposes might be "grouped." Two types of grouping exist:

- A "Required" group requires at least one of the field members to have a value;
- A "Best of" group contributes the best score from the fields in the group to the overall match score.

For example, in the "Individual" Match Purpose:

- `Person_Name` is a required field.
- One of either ID number or date of birth is required.
- Other attributes are optional.

The overall score returned by each Purpose is calculated by adding the participating field scores multiplied by their respective weight and divided by the total of all field weights. If a field is optional and is not provided, it is not included in the weight calculation.

The weights and matching options used in the Standard Populations are internally set by Informatica's Population experts based on years of tuning experience. They are not available to be overridden by the application. However, if a user has a different need not supported by the Standard Population, Informatica Corporation might offer to build a Custom Population for that client.

Field Types

The description of the fields supported by the various Match Purposes are listed as follows:

Address_Part1

Includes the part of address up to, but not including, the locality "last line." The word order, which represents the position of the address components, must be the normal word order used in your data population.

These must be passed in one field. Based on table design, your application might need to concatenate these attributes into one field before calling SSA-NAME3. For example, in the U.S., a typical string to pass consists of the following attributes:

```
Care-of + Building Name + Street Number + Street Name + Street Type + Apartment Details
```

The default key length of the `Address_Part1` field is 8 bytes. You can change the key length to 5 bytes by using the `KEY_SIZE` control. Matching on `Address_Part1` uses methods and options designed specifically for addresses. The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Address_Part1` field.

It is also possible to supply the entire address in the `Address_Part1` field for matching. The application might pass multiple addresses (such as a residential address and a postal address) in the one call to SSA-NAME3.

Address_Part2

Includes the "locality" line in an address. For example, in the U.S., a typical string to pass consists of the following attributes:

```
City + State + Zip (+ Country)
```

Matching on `Address_Part2` uses methods and options designed specifically for addresses. It uses the same Edit-List as `Address_Part1`. The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Address_Part2` field.

The default key length of the `Address_Part2` field is 8 bytes. You can change the key length to 5 bytes by using the `KEY_SIZE` control.

Attribute1, Attribute2

Use the `Attribute 1` and `Attribute 2` fields for general purpose. The fields use a general purpose string matching algorithm that compensates for transpositions and missing characters or digits.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Attribute 1` and `Attribute 2` fields. Define any rules for the `Attribute 1` and `Attribute 2` fields as character rules, which replace or remove strings or numbers that include special characters or delimiters.

The default key length of the `Attribute 1` and `Attribute 2` fields is 8 bytes. You can change the key length to 5 bytes by using the `KEY_SIZE` control.

Code

The `Code` field indexes and searches for any value that is in the alphanumeric or numeric format, such as credit card number, telephone number, Vehicle Identification Number, date, and International Standard Book Number. The `Code` field uses an algorithm that overcomes any errors or variations that are common across the code entities, such as missing digits, different digit orders, and special characters.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Code` field. Define any rules for the `Code` field as character rules, which replace or remove strings or numbers that include special characters or delimiters.

The key length of the `Code` field is 12 bytes.

Company_Name

Matches the names of organizations, which can be business names, institution names, department names, agency names, or trading names. The `Company_Name` field matches a single name or a compound name such as a legal name and its trading style. The `Company_Name` does not include the person name rules.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Company_Name` field.

The default key length of the `Company_Name` field is 8 bytes. You can change the key length to 5 bytes by using the `KEY_SIZE` control.

CreditCard

The `CreditCard` field indexes and searches the credit card numbers, which are generally a 16-digit number. A credit card is electronically associated with a bank, a customer, and the customers' accounts. The `CreditCard` field uses an algorithm that overcomes any errors or variations, such as different industry identifiers, issuer identifiers, account numbers, special characters, embedded spaces, different digit orders, and missing or extra digits.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `CreditCard` field. Define any rules for the `CreditCard` field as character rules, which replace or remove strings or numbers that include special characters or delimiters.

The key length of the `CreditCard` field is 12 bytes.

Date

The `Date` field indexes and searches for any types of date, such as expiry date, date of contract, date of change, and creation date. Use the `Day+Month+Year` format for a date. The algorithm uses the methods and options designed specifically for dates to overcome any errors or variations that are common in the date format.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Date` field. Define any rules for the `Date` field as character rules, which replace or remove strings or numbers that include special characters or delimiters.

The default key length of the `Date` field is 8 bytes. You can change the key length to 5 bytes by using the `KEY_SIZE` control.

Geocode

Use the Geocode field to index and search the records based on latitude and longitude geographic coordinates with an optional elevation. Specify the elevation value with its unit after the geographic coordinates. The supported units for elevation are ft (feet), km (kilometer), and m (meter). Negative elevation represents depths below the sea level.

The default key length of the `Geocode` field is 8 bytes. You can change the key length to 5 bytes by using the `KEY_SIZE` control.

ID

Use the `ID` field to match any type of ID number such as account number, customer number, credit card number, drivers license number, passport, policy number, SSN or other identity code, or VIN. The `ID` field uses a string-matching algorithm that compensates for transpositions and missing characters or digits.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `ID` field. Define any rules for the `ID` field as character rules, which replace or remove strings or numbers that include special characters or delimiters.

The key length of the `ID` field is 12 bytes.

ISBN10 and ISBN13

The `ISBN10` and `ISBN13` fields index and search the International Standard Book Number (ISBN). The International Standard Book Number is a unique code that identifies the commercial books. Use the `ISBN10` field to search for a 10-digit number, and use the `ISBN13` field to search for a 13-digit number.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `ISBN10` and `ISBN13` fields. Define any rules for the `ISBN10` and `ISBN13` fields as character rules, which replace or remove strings or numbers that include special characters or delimiters.

The key length of the `ISBN10` and `ISBN13` fields is 12 bytes.

Organization_Name

Matches the names of organizations, which can be business names, institution names, department names, agency names, or trading names.

The `Organization_Name` field matches a single name or a compound name such as a legal name and its trading style. The `Organization_Name` field includes the person name rules. The Population Override

Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Organization_Name` field.

The default key length of the `Organization_Name` field is 8 bytes. You can change the key length to 5 bytes by using the `KEY_SIZE` control.

Person_Name

Used to match the names of people. An application must pass the full person name. The word order represents the position of the first name, middle names, and family names. It must be the normal word order used in your data population. For example, in English speaking countries, the normal word order is as follows:

```
First Name + Middle Name(s) + Family Name(s)
```

Based on table design, your application might have to concatenate these separate fields into one field before calling SSA-NAME3.

The default key length of the `Person_Name` field is 8 bytes. You can change the key length to 5 bytes by using the `KEY_SIZE` control.

This field supports matching on a single name, or an account name such as `JOHN and MARY SMITH`. The application might pass multiple names such as a married name and a former name in the one call to SSA-NAME3.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Person_Name` field.

Postal_Area

Use the `Postal_Area` field to place more emphasis on the postal code than if it were included in the `Address_Part2` field. Use this field for all types of postal codes, including ZIP codes. This field uses a string matching algorithm that compensates for transpositions and missing characters or digits.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Postal_Area` field. Define any rules for the `Postal_Area` field as character rules, which replace or remove strings or numbers that include special characters or delimiters.

The default key length of the `Postal_Area` field is 8 bytes. You can change the key length to 5 bytes by using the `KEY_SIZE` control.

Telephone_Number

The `Telephone_Number` field indexes and searches different variations of the telephone numbers. The `Telephone_Number` field uses a string-matching algorithm that overcomes any errors or variations, such as incorrect area code or country code, special characters, embedded spaces, different digit orders, use of symbols such as plus (+), transpose errors, concatenated digits, deletion or omission of digits, insertion errors, substitution errors, and transpositions.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `Telephone_Number` field. Define any rules for the `Telephone_Number` field as character rules, which replace or remove strings or numbers that include special characters or delimiters.

The key length of the `Telephone_Number` field is 12 bytes.

VIN

The `VIN` field indexes and searches the Vehicle Identification Number (VIN). It is a unique code that identifies individual motor vehicles, towed vehicles, motorcycles, scooters, or mopeds. The `VIN` field uses an algorithm that overcomes any errors or variations, such as vehicle manufacturer codes, special

characters, embedded spaces, different letter or digit orders, transpose errors, concatenated digits, transpositions, and sequence variations.

The Population Override Manager or the Edit Rule Wizard can override the Edit-List rules of the algorithm that uses the `VIN` field. Define any rules for the `VIN` field as character rules, which replace or remove strings or numbers that include special characters or delimiters.

The key length of the `VIN` field is 12 bytes.

Purpose Types

The descriptions of the Purposes supported by the Standard Populations are listed as follows:

Address

This Purpose identifies an address match. The address might be postal, residential, delivery, descriptive, formal, or informal.

This Match purpose is used after a search by `Address_Part1`.

Field	Required
Address_Part1	Yes
Address_Part2	No
Postal_Area	No
Telephone_Number	No
ID	No
Date	No
Attribute1	No
Attribute2	No
Geocode	No
Code	No

If you use the name of a city, state, or both as `Address_Part2`, the field helps differentiate between a common street address [100 Main Street] in different locations.

To achieve a best score between `Address_Part2` and `Postal_Area`, pass `Postal_Area` as a repeat value in the `Address_Part2` field.

For example:

```
*Address_Part2*100 Main St*Address_Part2*06870***
```

In this example, the `Address_Part2` score uses the highest of the two scored fields.

AuthorISBN

The `AuthorISBN` purpose identifies and matches an author for the specified book with an International Standard Book Number. You can use this purpose after a search by book number, author, or both.

Field	Required
Person_Name	Yes
ISBN10/ISBN13	No

You can use the ISBN 10-digit number or 13-digit number based on your requirement to achieve a best score.

CC_Issuer

The `CC_Issuer` purpose identifies and matches the organization that issues credit card to the customers. You can use this purpose when you search for a credit card or the credit card-issuing organization.

Field	Required
CreditCard	Yes
Organization_Name	Yes, unless you specify Company_Name
Company_Name	Yes, unless you specify Organization_Name

CC_Owner

The `CC_Owner` purpose identifies and matches the credit card owner and the address. You can use this purpose after a search by person name, credit card number, or telephone number. You can provide the locality, state, or post or ZIP code in the `Address_Part2` field to achieve a best score. You can increase the weight of the post or ZIP code by using the `Postal_Area` field.

Field	Required
Person_Name	Yes
CreditCard	Yes
Telephone_Number	Yes
Address_Part1	Yes
Postal_Area	Yes
Address_Part2	No
Date	No
Geocode	No

To achieve a best score between the Address_Part1 and Postal_Area fields, specify Postal_Area as a repeat value in the Address_Part2 field.

Contact

This purpose identifies a contact within an organization at a specific location. This Match purpose is used after a search by Person_Name. However, either Organization_Name or Address_Part1 could be used as the search criteria.

For ultimate quality, a tiered search using two or all three of these fields could be used in the search. An example of a tiered search is a Person_Name search followed by a Address_Part1 search).

Field	Required
Person_Name	Yes
Organization_Name	Yes, unless you specify Company_Name
Company_Name	Yes, unless you specify Organization_Name
Address_Part1	Yes
Address_Part2	No
Postal_Area	No
Telephone_Number	No
ID	No
Date	No
Attribute1	No
Attribute2	No
Geocode	No
Code	No

To achieve a best score between Address_Part2 and Postal_Area, pass Postal_Area as a repeat value in the Address_Part2 field.

For example:

```
*Address_Part2*100 Main St*Address_Part2*06870***
```

In this example, the Address_Part2 score uses the highest of the two scored fields.

Corp_Entity

The Corp_Entity purpose identifies an organization based on its legal corporate name, including the legal endings such as INC and LTD. Use the Corp_Entity purpose in applications that honor differences between names such as ABC TRADING INC and ABC TRADING LTD.

Use this purpose after you perform a search based on `Organization_Name`.

Field	Required
Organization_Name	Yes, unless you specify Company_Name
Company_Name	Yes, unless you specify Organization_Name
Address_Part1	No
Address_Part2	No
Postal_Area	No
Telephone_Number	No
ID	No
Date	No
Attribute1	No
Attribute2	No
Geocode	No
Code	No

It is in essence the same purpose as Organization, except that tighter matching is performed and legal endings are not treated as noise.

To achieve a best score between `Address_Part2` and `Postal_Area`, pass `Postal_Area` as a repeat value in the `Address_Part2` field.

For example:

```
*Address_Part2*100 Main St*Address_Part2*06870***
```

In this example, the `Address_Part2` score uses the highest of the two scored fields.

Division

The `Division Purpose` identifies an Organization at an Address. It is used after a search by `Organization_Name` or by `Address_Part1`, or both.

Field	Required
Organization_Name	Yes, unless you specify Company_Name
Company_Name	Yes, unless you specify Organization_Name
Address_Part1	Yes
Address_Part2	No
Postal_Area	No

Field	Required
Telephone_Number	No
ID	No
Attribute1	No
Attribute2	No
Geocode	No
Code	No

It is in essence the same purpose as Organization, except that `Address_Part1` is a required field. Thus, this purpose matches organization X at address Y or Z if multiple addresses are supplied.

To achieve a best score between `Address_Part2` and `Postal_Area`, pass `Postal_Area` as a repeat value in the `Address_Part2` field.

For example:

```
*Address_Part2*100 Main St*Address_Part2*06870***
```

In this example, the `Address_Part2` score uses the highest of the two scored fields.

Family

The `Family` purpose identifies matches where individuals with the same or similar family names share the same address `or` the same telephone number.

This purpose is used after a tiered search (multisearch) by `Address_Part1` and `Telephone_Number`.

Note: It is not practical to search by `Person_Name`. It is because ultimately one word from the `Person_Name` needs to match, and a one-word search does not perform well in most situations.

Field	Required
Person_Name	Yes
Address_Part1	Yes
Telephone_Number	Yes
Address_Part2	No
Postal_Area	No
Attribute1	No
Attribute2	No
Geocode	No
Code	No

Note: Score is based on the best of the above group.

Emphasis is placed on the Last Name, or "Major Word" of the `Person_Name` field. This is one of the few cases where word order is important in the way the records are passed to SSA-NAME3 for matching.

However, a reasonable score is generated provided that a match occurs between the major word in one name and any other word in the other name.

To achieve a best score between `Address_Part2` and Postal_Area, pass `Postal_Area` as a repeat value in the `Address_Part2` field.

For example:

```
*Address_Part2*100 Main St*Address_Part2*06870***
```

In this example, the `Address_Part2` score uses the highest of the two scored fields.

Fields

The Purpose is provided for general nonspecific use. It is designed in such a way that there are no required fields. All field types are available as optional input fields.

Field	Required
Person_Name	No
Organization_Name	No
Company_Name	No
Address_Part1	No
Address_Part2	No
Postal_Area	No
Telephone_Number	No
ID	No
Date	No
Attribute1	No
Attribute2	No
Geocode	No
Code	No
Generic_Field	No

One way this Purpose could be used is as a non-exact match filter before applying some other Match Purpose. For exact match filters, use the `Filter` Purpose. For example, before passing a record to the `Division` Purpose, use the `Fields` Purpose to eliminate any organization with `ID` numbers which do not score above 80%. To do this, the application first passes the `ID` numbers to SSA-NAME3 for matching using `PURPOSE=FIELDS`. It then decides based on the score returned whether to pass the full records for matching by the `Division` Purpose.

Filter1-9

The `Filter` Purpose is provided so that the application can perform exact match filtering based on the setting of one or more flags in the records. One call to `ssan3_match` can use up to nine Filters `(Filter1-9)`.

Field	Required
Filter1-9	Yes

For example, say an index supported searching and matching across two types of names: Organization names (identified by a Name-Type-Flag of "C"), and Person names (identified by a Name-Type-Flag of "P"). A search application might need to support searches across both name types and within each name type. To support the "within each name type" search, the application can use the Filter Purpose to filter out exact matches based on the name type flag.

The fields `Filter1-9` can be any code or flag.

For non-exact filtering, use the `Fields` Purpose.

Household

The Household purpose identifies matches where individuals with the same or similar family names share the same address.

This purpose is used after a search by `Address_Part1`. It is not practical to search by `Person_Name`. It is because ultimately one word from the `Person_Name` needs to match, and a one-word search does not perform well in most situations.

Field	Required
Person_Name	Yes
Address_Part1	Yes
Address_Part2	No
Postal_Area	No
Telephone_Number	No
Attribute1	No
Attribute2	No
Geocode	No
Code	No

Emphasis is placed on the Last Name, or "Major Word" of the `Person_Name` field. This is one of the few cases where word order is important in the way the records are passed to SSA-NAME3 for matching.

However, a reasonable score is generated provided that a match occurs between the major word in one name and any other word in the other name.

To achieve a best score between `Address_Part2` and `Postal_Area`, pass `Postal_Area` as a repeat value in the `Address_Part2` field.

For example:

```
*Address_Part2*100 Main St*Address_Part2*06870***
```

In this example, the `Address_Part2` score uses the highest of the two scored fields.

Individual

This Purpose identifies a specific individual by name and with either the same ID number or Date of Birth attributes.

It is typically used after a search by `Person_Name`.

Field	Required
Person_Name	Yes
ID	At least one
Date	of these two
Attribute1	No
Attribute2	No
Code	No

Organization

The Organization Purpose matches organizations primarily by name. It is targeted at online searches when a name-only lookup is required and a human is available to make the choice. Matching in batch requires other attributes in addition to name to make match decisions.

Field	Required
Organization_Name	Yes, unless you specify Company_Name
Company_Name	Yes, unless you specify Organization_Name
Address_Part1	No
Address_Part2	No
Postal_Area	No
Telephone_Number	No
ID	No
Date	No
Attribute1	No
Attribute2	No

Field	Required
Geocode	No
Code	No

To achieve a best score between `Address_Part2` and `Postal_Area`, pass `Postal_Area` as a repeat value in the `Address_Part2` field.

For example:

```
*Address_Part2*100 Main St*Address_Part2*06870***
```

In this example, the `Address_Part2` score uses the highest of the two scored fields.

Person_Name

This Purpose identifies a person by name. It is targeted at online searches when a name-only lookup is required and a human is available to make the choice. Matching in batch requires other attributes in addition to name to make match decisions.

Field	Required
Person_Name	Yes
Address_Part1	No
Address_Part2	No
Postal_Area	No
Telephone_Number	No
ID	No
Date	No
Attribute1	No
Attribute2	No
Geocode	No
Code	No

To achieve a best score between `Address_Part2` and `Postal_Area`, pass `Postal_Area` as a repeat value in the `Address_Part2` field.

For example:

```
*Address_Part2*100 Main St*Address_Part2*06870***
```

In this example, the `Address_Part2` score uses the highest of the two scored fields.

PublisherISBN

The `PublisherISBN` purpose identifies and matches the publishing organization for a specified book with an International Standard Book Number (ISBN). The ISBN can be a 10-digit number or a 13-digit number. You can use this purpose after a search by book number, publisher, or both.

Field	Required
Organization_Name	Yes, unless you specify Company_Name
Company_Name	Yes, unless you specify Organization_Name
ISBN10/ISBN13	No
Address_Part1	No
Address_Part2	No
Postal_Area	No
Geocode	No

You can provide the locality, state, or post or ZIP code in the `Address_Part2` field to achieve a best score. You can increase the weight of the post or ZIP code by using the `Postal_Area` field.

Resident

The Resident Purpose identifies a person at an address.

This purpose is used after a search by either `Person_Name` or `Address_Part1`, or both in a multi-search.

Field	Required
Person_Name	Yes
Address_Part1	Yes
Address_Part2	No
Postal_Area	No
Telephone_Number	No
ID	No
Date	No
Attribute1	No
Attribute2	No
Geocode	No
Code	No

To achieve a best score between `Address_Part2` and `Postal_Area`, pass `Postal_Area` as a repeat value in the `Address_Part2` field.

For example:

```
*Address_Part2*100 Main St*Address_Part2*06870***
```

In this example, the `Address_Part2` score uses the highest of the two scored fields.

VIN_Manufacturer

The `VIN_Manufacturer` purpose identifies and matches the manufacturer of a vehicle for a specified vehicle number. You can use this purpose after a search by vehicle number, vehicle manufacturer, or both. You can provide the locality, state, or post or ZIP code in the `Address_Part2` field to achieve a best score. You can increase the weight of the post or ZIP code by using the `Postal_Area` field.

Field	Required
VIN	Yes
Organization_Name	Yes, unless you specify Company_Name
Company_Name	Yes, unless you specify Organization_Name
Address_Part1	No
Address_Part2	No
Postal_Area	No
Geocode	No

You can provide the locality, state, or post or ZIP code in the `Address_Part2` field to achieve a best score. You can increase the weight of the post or ZIP code by using the `Postal_Area` field.

VIN_Owner

The `VIN_Owner` purpose identifies and matches the owner of a vehicle. You can use this purpose after a search by the person name who is the owner of the vehicle, vehicle number, or both. You can provide the locality, state, or post or ZIP code in the `Address_Part2` field to achieve a best score. You can increase the weight of the post or ZIP code by using the `Postal_Area` field.

Field	Required
VIN	Yes
Organization_Name	Yes, unless you specify Company_Name
Company_Name	Yes, unless you specify Organization_Name
Person_Name	Yes
Address_Part1	No
Address_Part2	No
Postal_Area	No
Attribute1	No

Attribute2	No
Date	No
Geocode	No
Code	No

You can provide the color and the body type of the vehicle as the `Attribute1` and `Attribute2` fields for a best score. You can increase the weight of the post or ZIP code by using the `Postal_Area` field.

Wide_Contact

This Purpose loosely identifies a contact within an organization, without regard to the location.

It is used after a search by `Person_Name`. However, a second search by `Organization_Name` can be used to get better quality.

Field	Required
Person_Name	Yes
Organization_Name	Yes, unless you specify Company_Name
Company_Name	Yes, unless you specify Organization_Name
ID	No
Attribute1	No
Attribute2	No
Code	No

Wide_Household

The `Wide_Household` purpose identifies matches where the same address is shared by individuals with the same family name or with the same telephone number.

This purpose is used after a search by `Address_Part1`.

Note: It is not practical to search by `Person_Name`. It is because ultimately one word from the `Person_Name` needs to match, and a one-word search will not perform well in most situations.

Field	Required
Address_Part1	Yes
Telephone_Number	Yes
Address_Part2	No
Postal_Area	No

Field	Required
Attribute1	No
Attribute2	No
Geocode	No
Code	No

Note: Score will be based on best of this group.

Emphasis is placed on the Last Name, or "Major Word" of the `Person_Name` field. This is one of the few cases where word order is important in the way the records are passed to SSA-NAME3 for matching.

However, a reasonable score is generated provided that a match occurs between the major word in one name and any other word in the other name.

To achieve a best score between `Address_Part2` and `Postal_Area`, pass `Postal_Area` as a repeat value in the `Address_Part2` field.

For example:

```
*Address_Part2*100 Main St*Address_Part2*06870***
```

In this example, the `Address_Part2` score uses the highest of the two scored fields.

Match Levels

Using Standard Populations, an application may be set up to match on any of the defined Match Purposes using one of three different Match Levels:

- Typical
- Conservative
- Loose

The choice of Match Level is passed to the SSA-NAME3 "match" function directly by the user's application, or using the `MATCH_LEVEL=` Control in MDM - Registry Edition.

It is good practice to test using different Match Levels on real production data and volumes to measure the reliability differences.

Typical

A Typical match level for most applications delivers "reasonable" matches. It should be used in typical online or batch transaction searches. It is the default if no Match Level is specified.

Conservative

A Conservative match level for most applications delivers "close" matches. It is generally used in batch systems where accuracy of match is paramount.

Loose

A Loose match level for most applications delivers matches with a higher degree of variation than Typical. It is generally used in systems where the risk of missing a match is high and manual review is available.

Generic Field

Some standard populations contain a Key Field called the `Generic_Field`, and a Match Purpose called `Generic`. Currently, this is available in the following standard populations:

- USA
- UK
- AUSTRALIA

The algorithm that builds keys and search ranges for Generic Data is invoked by calling SSA-NAME3 and by passing `FIELD=Generic_Field` in the Controls parameter of the `get_keys` or `get_ranges` calls. Generic Data may contain either Person Names, Organization Names or Addresses. It may also contain a combination of these.

Using MDM - Registry Edition, it is passed in the KEY-LOGIC and SEARCH-LOGIC Controls parameters.

The `Generic_Field` algorithm is designed to overcome some of the errors and variation that are common across entities such as Names of people, Names of organizations and Addresses. For Person Name, this may include salutations and honorifics, special characters, embedded spaces, nicknames, different word orders, use of initials, spelling errors, concatenated words, localized words, and foreign words. For Organizations, this may include different legal endings, abbreviations, salutations and honorifics, special characters, embedded spaces, nicknames, different word orders, missing and extra words, spelling errors, concatenated words, use of initials, mixed use of numbers and words, foreign words, and localization. For Addresses, this may include presence of care of information, abbreviations, special characters, embedded spaces, different word orders, spelling errors, concatenated words and numbers, use of initials, mixed use of numbers and words, foreign words, missing words, extra words and sequence variations.

MDM - Registry Edition may pass multiple Generic Fields (such as names and addresses) in the one call to SSA-NAME3.

`Generic_Field` is normally used to homogenize indexes of various types of data such as Person Names, Addresses, and Organization names. This allows them to be stored in a single index table by the application (or MDM - Registry Edition). Using such an index, application may implement a generic search where the type of data need not be available.

For example, if search for 'GreenHorn' is expected to find the person 'Peter GreenHorn', as well as the Organization 'GreenHorn Industries', then the `Generic_Field` may be used to build keys and ranges.

The Generic Match Purpose is designed to match general non-specific data. The only required field for this purpose is the `Generic_Field`.

Field	Required?
`Generic_Field`	Yes

Algorithms designed for specific type of data such as `Person_Names`, `Address_Part1` and `Organization_Name` carry significantly less overhead than the `Generic_Field` due to their targeted processing domain. On the other hand, the `Generic_Field` algorithm reduces the overhead associated with creation and maintenance of separate indexes, and allows applications such as MDM-Registry Edition to simplify the search experience by not having to categorize search data.

The `Generic_Field` algorithm has an Edit-List whose rules may be overridden by the Population Override Manager or Edit Rule Wizard.

Managing Population Rule Sets

A Population rule-set is a file used by the SSA-NAME3 callable routine to modify its behavior for different countries, languages or data populations.

Population rule-sets may be one of three types:

- Standard Populations are provided with the product.
- A Custom Population may be built by an Informatica Corporation consultant for a customer with unusual or special needs.
- A Local Population is the result of local rules modifications done via the Population Override Manager or Edit RuleWizard.

It is possible for a system to have all three types of Population rule-sets. If so, there is an order of precedence in loading by SSA-NAME3. If a Local Population (file extension of `.YLP`) is present in the folder identified by the "System" Control, it is loaded; else if a Custom Population is present (file extension of `.YCP`), it is loaded; else the Standard Population is loaded (file extension of `.YSP`).

INDEX

C

Code 29
Corp_Entity 34
Custom Population 53

E

Edit Rule Wizard 32
Extended Keys 32

G

Geocode 29

K

Key Field 28
Key Levels 32
Key-Logic 8

M

Match Level 28
Match Levels
  Conservative 51
  Loose 51
  Typical 51
Match Purpose 28, 34
Multi-Purpose Matching 20

O

Organization Names 29

P

Person Names 29
Population rule-set 53

S

Score-Logic 8
Search Levels
  Exhaustive 33
  Extreme 33
  Narrow 33
  Typical 33
Search-Logic 8
SSA-NAME3 Controls 20
Standard Keys 32
Standard Population 25
Standard Populations 53

T

Tagged Data Format 8

U

Unicode 20
UTF8 20