Refugee Outcomes Transition Fund

Data Specification

Version 7539216

Generated on 16 September 2021

# About this document

This document outlines the Refugee Outcomes Transition Fund (RTOF) Data Specification.

The document outlines the data fields, types, and descriptions for all required data collection during the lifetime of the Refugee Transition Outcome Fund. The description for each field outlines the context, collection frequency and milestone at which it should be collected.

The validation rules that apply for each field are noted in the grey tables for each record. These will be applied on submission of data, if any validation rules are not met, a report will be produced with details of the fields and specified errors for the given fields. The images below outline the key milestones and data collection points during the program.





# Changelog

Summary of git tags?

# Table of Contents

[About this document 2](#_Toc78278753)

[Changelog 3](#_Toc78278754)

[Table of Contents 4](#_Toc78278755)

[Records 5](#_Toc78278756)

[Validation Rules 9](#_Toc78278760)

# Records

Records are the top-level entities of the RTOF data specification and correspond to tables in relational databases.

Records capture information about a specific step in the programme flow and contains essential information for the outcomes validation and payment metrics.



## PERSON Record

This form outlines the required information for all service users, the data is to be collected once at enrolment.

Workflow: Enrolment

### Fields

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Type | Decision status |
| unique\_id [PK] | Unique person identifier | string | Decided |
| year\_of\_birth | Year of birth | year | Decided |
| gender | Gender | categorical | Decided |
| dispersal\_area | Dispersal area | categorical | Decided |
| date\_started\_service | Date started with service | monthyear | Decided |

#### unique\_id: Unique person identifier

Person identifier provided in the person form. This can be a NINO or a unique provider ID. This must be unique for each individual supported on RTOF.

|  |  |
| --- | --- |
| Type | string |
| Primary Key | True |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'} |

**Status:** *Decided*

#### year\_of\_birth: Year of birth

Year of birth of participant. To be collected once at enrolment

|  |  |
| --- | --- |
| Type | year |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'} |

**Status:** *Decided*

#### gender: Gender

Gender of participant. Must be one of the five categories provided. To be collected once at enrolment.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | Man, Woman, Non-binary, Other, Prefer not to say |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'gender', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(gender)'} |

**Status:** *Decided*

#### dispersal\_area: Dispersal area

Local authority where participant was located once asylum status was granted. A list of all possible dispersal areas are listed within the dimensions. To be collected once at enrolment into the program.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | Bolton Council, Bury Council, Manchester City Council, Oldham Council, Rochdale Borough Council, Salford City Council, Stockport Metropolitan Borough Council, Tameside Metropolitan Borough Council, Trafford Council, Wigan Council, Wolverhampton Council, Coventry City Council, Sandwell Council, Birmingham City Council, Plymouth City Council, Darlington Borough Council, Gateshead Council, Hartlepool Borough Council, Middlesbrough Council, Newcastle City Council, North Tyneside Council, Northumberland County Council, Redcar and Cleveland Council, South Tyneside Council, Stockton Council, Sunderland City Council |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'dispersal\_area', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(dispersal\_area)'} |

**Status:** *Decided*

#### date\_started\_service: Date started with service

Month and year the participant enrolled onto the program. To be collected once at enrolment.

|  |  |
| --- | --- |
| Type | monthyear |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'date\_after', 'args': 'date\_asylum\_status\_granted', 'rule': ValidationRule(id='date\_after', description='Only used for fields of type date, this validator ensures that the provided value is after the date indicated. When\nmultiple\n', args=[{'name': 'field\_id', 'type': 'array', 'items': {'type': 'string'}, 'help': 'The ID of the field that this date has to be after'}]), 'description': 'date\_after(date\_asylum\_status\_granted)'} |

**Status:** *Decided*

## HOUSING\_ENTRY Record

This form outlines the required data for participants who achieve the housing entry outcome. Data should be collected when achieving the housing entry outcome.

Workflow: Housing

### Fields

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Type | Decision status |
| unique\_id [PK] | Unique person identifier | string | Decided |
| housing\_entry\_date | Date of housing entry | date | Decided |
| housing\_entry\_accomodation | Accomodation type entry | categorical | Decided |

#### unique\_id: Unique person identifier

Person identifier provided in the person form. This can be a NINO or a unique provider ID. This must be unique for each individual supported on RTOF.

|  |  |
| --- | --- |
| Type | string |
| Primary Key | True |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'unique', 'args': True, 'rule': ValidationRule(id='unique', description="A field with 'unique: true' should be unique within the dataset for this provider.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'unique(True)'} |

**Status:** *Decided*

#### housing\_entry\_date: Date of housing entry

Date of housing entry outcome achieved. To be collected once at entry to housing outcome submission.

|  |  |
| --- | --- |
| Type | date |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'date\_after', 'args': 'date\_started\_service', 'rule': ValidationRule(id='date\_after', description='Only used for fields of type date, this validator ensures that the provided value is after the date indicated. When\nmultiple\n', args=[{'name': 'field\_id', 'type': 'array', 'items': {'type': 'string'}, 'help': 'The ID of the field that this date has to be after'}]), 'description': 'date\_after(date\_started\_service)'} |

**Status:** *Decided*

#### housing\_entry\_accomodation: Accomodation type entry

Safe and secure accommodation - Safe accommodation is defined by compliance with the Decent Homes Standard including the absence of category 1 and 2 hazards under the Housing Health and Safety Rating System, and with all relevant gas, electricity and fire safety checks in place. Within this safe accommodation, the individual refugee/family must have appropriate individual space. This is defined as one bedroom for - every adult couple; any other adult aged 16 or over; any two children of the same sex; any two children regardless of sex under the age of 10; any other child; and access to self-contained facilities, including bathroom and kitchen, if in a shared facility. Secure accommodation is defined by security of tenure for at least six months.  
Under RTOF, accommodation may include - Private rental sector; Social housing; Supported housing N.B. Accommodation may be let under a Periodic Licence agreement where appropriate, but use of an excluded licence is not permissible Note that planned and anticipated moves between properties during the 6 months will be allowed, provided the individual is still within secure, stable accommodation at the end of the 6-month sustainment period.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | RPC, RLA, other |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'housing\_accomodation', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(housing\_accomodation)'} |

**Status:** *Decided*

## HOUSING\_SUSTAIN Record

This form outlines the required data for participants who achieve the housing sustainment outcome. Data should be collected when achieving the housing sustainment outcome.

Workflow: Housing

### Fields

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Type | Decision status |
| unique\_id [PK] | Unique person identifier | string | Decided |
| housing\_sustainment\_date | Date housing sustainment achieved | date | Decided |
| housing\_sustain\_accomodation | Accomodation type sustainment | categorical | Decided |

#### unique\_id: Unique person identifier

Person identifier provided in the person form. This can be a NINO or a unique provider ID. This must be unique for each individual supported on RTOF.

|  |  |
| --- | --- |
| Type | string |
| Primary Key | True |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'unique', 'args': True, 'rule': ValidationRule(id='unique', description="A field with 'unique: true' should be unique within the dataset for this provider.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'unique(True)'} |

**Status:** *Decided*

#### housing\_sustainment\_date: Date housing sustainment achieved

Date of housing sustainment outcome achieved. To be collected once at housing sustainment outcome submission.

|  |  |
| --- | --- |
| Type | date |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'date\_after', 'args': 'housing\_entry\_date', 'rule': ValidationRule(id='date\_after', description='Only used for fields of type date, this validator ensures that the provided value is after the date indicated. When\nmultiple\n', args=[{'name': 'field\_id', 'type': 'array', 'items': {'type': 'string'}, 'help': 'The ID of the field that this date has to be after'}]), 'description': 'date\_after(housing\_entry\_date)'} |

**Status:** *Decided*

#### housing\_sustain\_accomodation: Accomodation type sustainment

Safe and secure accommodation - Safe accommodation is defined by compliance with the Decent Homes Standard including the absence of category 1 and 2 hazards under the Housing Health and Safety Rating System, and with all relevant gas, electricity and fire safety checks in place. Within this safe accommodation, the individual refugee/family must have appropriate individual space. This is defined as one bedroom for - every adult couple; any other adult aged 16 or over; any two children of the same sex; any two children regardless of sex under the age of 10; any other child; and access to self-contained facilities, including bathroom and kitchen, if in a shared facility. Secure accommodation is defined by security of tenure for at least six months.  
Under RTOF, accommodation may include - Private rental sector; Social housing; Supported housing N.B. Accommodation may be let under a Periodic Licence agreement where appropriate, but use of an excluded licence is not permissible Note that planned and anticipated moves between properties during the 6 months will be allowed, provided the individual is still within secure, stable accommodation at the end of the 6-month sustainment period.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | RPC, RLA, other |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'housing\_accomodation', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(housing\_accomodation)'} |

**Status:** *Decided*

## BASELINE Record

This form outlines the required baseline data for all service users, the data is to be collected across interactions with participants and submitted within 3 months of enrolment.

Workflow: Integration

### Fields

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Type | Decision status |
| unique\_id [PK] | Unique person identifier | string | Decided |
| ni\_number | NI Number | string | Decided |
| nationality | Nationality | categorical | Decided |
| referral\_source | Referral source | categorical | Decided |
| current\_dependents\_uk | Current dependents in UK | categorical | Decided |
| number\_u18\_HH | Number of children under 18 living in HH | integer | Decided |
| number\_over\_18\_HH | Number of adults over 18 living in HH | integer | Decided |
| age\_profile\_of\_dependents\_uk | Age profile of dependents in UK | list | Decided |
| date\_arrived\_in\_uk | Date arrived in UK | monthyear | Decided |
| date\_asylum\_status\_granted | Date asylum status granted | monthyear | Decided |
| language\_level\_on\_entry | Language level on entry | categorical | Decided |
| highest\_qualification\_achieved | Highest qualification level achieved | categorical | Decided |
| age\_finished\_study | Age when finished study | integer | Decided |
| employed\_in\_home\_country | Employed in home country | categorical | Decided |
| occupation\_type | Occupation type | string | Decided |
| occupation\_sector | Sector | string | Decided |
| occupation\_goal | Occupation goal at baseline | string | Decided |
| prior\_employment | Prior employment | categorical | Decided |
| employment\_status | Main economic status at baseline | categorical | Decided |
| hours\_of\_employment | Hours of employment | categorical | Decided |
| wage\_baseline | Wage baseline | categorical | Pending consideration |
| eduction\_training\_status | Eduction / training status | categorical | Decided |
| housing\_baseline\_accommodation | Accommodation type baseline | categorical | Decided |
| homelessness\_baseline | Homelessness status baseline | categorical | Decided |

#### unique\_id: Unique person identifier

Person identifier provided in the person form. This can be a NINO or a unique provider ID. This must be unique for each individual supported on RTOF.

|  |  |
| --- | --- |
| Type | string |
| Primary Key | True |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'} |

**Status:** *Decided*

#### ni\_number: NI Number

UK National Insurance number. Must be unique. To be collected at once the NINO for a participant is available. All upper case and no spaces [regular expressions for NiNos].

|  |  |
| --- | --- |
| Type | string |
| Required |  |
| Validators | {'id': 'unique', 'args': True, 'rule': ValidationRule(id='unique', description="A field with 'unique: true' should be unique within the dataset for this provider.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'unique(True)'}, {'id': 'national\_insurance\_number', 'args': True, 'rule': ValidationRule(id='national\_insurance\_number', description='UK National Insurance Number - uppercase with all whitespace removed. Validated according to format given in\nhttps://github.com/dwp/nino-format-validation\n', args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is a national insurance number'}]), 'description': 'national\_insurance\_number(True)'} |

**Status:** *Decided*

Comments

*This data field is not validated as 'required', as we are aware some NINOs may not be available when submitting the baseline record. However, this field is ultimately required to be submitted and missing NINOs will be followed up.*

#### nationality: Nationality

Nationality of participant. A list of nationalities are provided. Must select one of the given categories, please see nationality descriptions to link with the relevant code. To be collected once and submitted within 3 months of enrolment.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | AD, AE, AF, AG, AI, AL, AM, AO, AQ, AR, AS, AT, AU, AW, AX, AZ, BA, BB, BD, BE, BF, BG, BH, BI, BJ, BL, BM, BN, BO, BQ, BR, BS, BT, BV, BW, BY, BZ, CA, CC, CD, CF, CG, CH, CI, CK, CL, CM, CN, CO, CR, CU, CV, CW, CX, CY, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, EH, ER, ES, ET, FI, FJ, FK, FM, FO, FR, GA, GB, GD, GE, GF, GG, GH, GI, GL, GM, GN, GP, GQ, GR, GS, GT, GU, GW, GY, HK, HM, HN, HR, HT, HU, ID, IE, IL, IM, IN, IO, IQ, IR, IS, IT, JE, JM, JO, JP, KE, KG, KH, KI, KM, KN, KP, KR, KW, KY, KZ, LA, LB, LC, LI, LK, LR, LS, LT, LU, LV, LY, MA, MC, MD, ME, MF, MG, MH, MK, ML, MM, MN, MO, MP, MQ, MR, MS, MT, MU, MV, MW, MX, MY, MZ, NA, NC, NE, NF, NG, NI, NL, NO, NP, NR, NU, NZ, OM, PA, PE, PF, PG, PH, PK, PL, PM, PN, PR, PS, PT, PW, PY, QA, RE, RO, RS, RU, RW, SA, SB, SC, SD, SE, SG, SH, SI, SJ, SK, SL, SM, SN, SO, SR, SS, ST, SV, SX, SY, SZ, TC, TD, TF, TG, TH, TJ, TK, TL, TM, TN, TO, TR, TT, TV, TW, TZ, UA, UG, UM, US, UY, UZ, VA, VC, VE, VG, VI, VN, VU, WF, WS, YE, YT, ZA, ZM, ZW |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'nationality', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(nationality)'} |

**Status:** *Decided*

#### referral\_source: Referral source

The referral source that led the participant to RTOF.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | LA, Home Office, Local VCS, Primary Care, Other Health, HO providers, Migrant Help, Self-referrals, DWP, Peer, Other |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'referral\_source', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(referral\_source)'} |

**Status:** *Decided*

#### current\_dependents\_uk: Current dependents in UK

Do you currently have any dependents in the UK? Yes / No to understand whether the participant has any depedents curently in the UK. To be collected once and submitted within 3 months of enrolment.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | Yes, No |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'current\_dependents\_uk', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(current\_dependents\_uk)'} |

**Status:** *Decided*

#### number\_u18\_HH: Number of children under 18 living in HH

How many children under the age of 18 are currently living in the participants household? To be collected once and submitted within 3 months of enrolment.

|  |  |
| --- | --- |
| Type | integer |
| Required |  |
| Validators |  |

**Status:** *Decided*

#### number\_over\_18\_HH: Number of adults over 18 living in HH

How many adults over the age of 18 are currently living in the participants household? To be collected once and submitted within 3 months of enrolment.

|  |  |
| --- | --- |
| Type | integer |
| Required |  |
| Validators |  |

**Status:** *Decided*

#### age\_profile\_of\_dependents\_uk: Age profile of dependents in UK

If answered Yes to 'current dependents UK', select every option that applies to the participants dependents (this does not record how many dependents are in each age bracket.) To be collected once and submitted within 3 months of enrolment.

|  |  |
| --- | --- |
| Type | list |
| Required |  |
| One of | 0-4, 5-11, 11-17, Caring responsibilities for an adult |
| Validators | {'id': 'dimension', 'args': 'age\_profile\_of\_dependents\_uk', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(age\_profile\_of\_dependents\_uk)'}, {'id': 'conditional', 'args': [{'expression': "one\_of(current\_dependents\_uk, ['Yes'])", 'validation': {'required': True}}], 'rule': ValidationRule(id='conditional', description='Allows for conditional application of validators', args=[{'name': 'expression', 'type': 'string', 'help': 'The evaluation expression'}, {'name': 'validation', 'type': 'validation', 'help': 'The validation that should be applied'}]), 'description': 'conditional rules apply'} |

**Status:** *Decided*

#### date\_arrived\_in\_uk: Date arrived in UK

Month and year the participant entered the UK, relating to their latest and current arrival. To be collected once and submitted within 3 months of enrolment.

|  |  |
| --- | --- |
| Type | monthyear |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'date\_after', 'args': 'date\_of\_birth', 'rule': ValidationRule(id='date\_after', description='Only used for fields of type date, this validator ensures that the provided value is after the date indicated. When\nmultiple\n', args=[{'name': 'field\_id', 'type': 'array', 'items': {'type': 'string'}, 'help': 'The ID of the field that this date has to be after'}]), 'description': 'date\_after(date\_of\_birth)'} |

**Status:** *Decided*

#### date\_asylum\_status\_granted: Date asylum status granted

Month and year the participant was granted asylum status, relating to their latest and current asylum status. To be collected once and submitted within 3 months of enrolment.

|  |  |
| --- | --- |
| Type | monthyear |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'date\_after', 'args': 'date\_arrived\_in\_uk', 'rule': ValidationRule(id='date\_after', description='Only used for fields of type date, this validator ensures that the provided value is after the date indicated. When\nmultiple\n', args=[{'name': 'field\_id', 'type': 'array', 'items': {'type': 'string'}, 'help': 'The ID of the field that this date has to be after'}]), 'description': 'date\_after(date\_arrived\_in\_uk)'} |

**Status:** *Decided*

#### language\_level\_on\_entry: Language level on entry

The language level of participant when entering RTOF. Selection of one of the specified categories. To be collected once and submitted within 3 months of enrolment.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | Below formal levels, Pre-entry, Entry level 1, Entry level 2, Entry level 3, Level 1, Level 2, Beyond Level 2 |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'language\_level\_on\_entry', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(language\_level\_on\_entry)'} |

**Status:** *Decided*

Comments

*Information regardimg these levels can be found at the following link - https://esol.britishcouncil.org/sites/default/files/attachments/informational-page/ESOL%20level%20descriptors.pdf*

#### highest\_qualification\_achieved: Highest qualification level achieved

The highest level of qualification of participant when entering RTOF. Selection of one of the specified categories. To be collected once and submitted within 3 months of enrolment.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | Below formal levels, Entry level, Level 1-2, Level 3-4, Level 5+, Unknown |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'highest\_qualification\_achieved', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(highest\_qualification\_achieved)'} |

**Status:** *Decided*

Comments

*Information regardimg these levels can be found at the following link - https://www.gov.uk/what-different-qualification-levels-mean/list-of-qualification-levels#:~:text=Level%205%20qualifications%20are%3A,higher%20national%20diploma%20(%20HND%20)*

#### age\_finished\_study: Age when finished study

This should only be submitted if the response to the field "highest qualification level achieved" was "unknown"

|  |  |
| --- | --- |
| Type | integer |
| Required |  |
| Validators | {'id': 'conditional', 'args': [{'expression': "one\_of(highest\_qualification\_achieved, ['Unknown'])", 'validation': {'required': True}}], 'rule': ValidationRule(id='conditional', description='Allows for conditional application of validators', args=[{'name': 'expression', 'type': 'string', 'help': 'The evaluation expression'}, {'name': 'validation', 'type': 'validation', 'help': 'The validation that should be applied'}]), 'description': 'conditional rules apply'} |

**Status:** *Decided*

#### employed\_in\_home\_country: Employed in home country

When participant left home country were they in any form of employment - yes / no response. To be collected once and submitted within 3 months of enrolment.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | Yes, No |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'employed\_in\_home\_country', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(employed\_in\_home\_country)'} |

**Status:** *Decided*

#### occupation\_type: Occupation type

If answered 'Yes' to employed in home country field, state job role of their last employment. To be collected once and submitted within 3 months of enrolment. For some providers this may be a categorical data field - please confirm with your nominated data lead.

|  |  |
| --- | --- |
| Type | string |
| Required |  |
| Validators | {'id': 'character\_limit', 'args': 255, 'rule': ValidationRule(id='character\_limit', description='Maximum number of unicode characters in string.\n', args=[{'name': 'characters', 'type': 'integer', 'help': 'The number of characters'}]), 'description': 'character\_limit(255)'}, {'id': 'conditional', 'args': [{'expression': "one\_of(employed\_in\_home\_country, ['Yes'])", 'validation': {'required': True}}], 'rule': ValidationRule(id='conditional', description='Allows for conditional application of validators', args=[{'name': 'expression', 'type': 'string', 'help': 'The evaluation expression'}, {'name': 'validation', 'type': 'validation', 'help': 'The validation that should be applied'}]), 'description': 'conditional rules apply'} |

**Status:** *Decided*

#### occupation\_sector: Sector

If answered 'Yes' to employed in home country field, state the sector of their last employment. To be collected once and submitted within 3 months of enrolment. For some providers this may be a categorical data field - please confirm with your nominated data lead.

|  |  |
| --- | --- |
| Type | string |
| Required |  |
| Validators | {'id': 'character\_limit', 'args': 255, 'rule': ValidationRule(id='character\_limit', description='Maximum number of unicode characters in string.\n', args=[{'name': 'characters', 'type': 'integer', 'help': 'The number of characters'}]), 'description': 'character\_limit(255)'}, {'id': 'conditional', 'args': [{'expression': "one\_of(employed\_in\_home\_country, ['Yes'])", 'validation': {'required': True}}], 'rule': ValidationRule(id='conditional', description='Allows for conditional application of validators', args=[{'name': 'expression', 'type': 'string', 'help': 'The evaluation expression'}, {'name': 'validation', 'type': 'validation', 'help': 'The validation that should be applied'}]), 'description': 'conditional rules apply'} |

**Status:** *Decided*

#### occupation\_goal: Occupation goal at baseline

All participants to be asked what their employment goals. This can be sector, job role or employment-type. To be collected once and submitted within 3 months of enrolment. For some providers this may be a categorical data field - please confirm with your nominated data lead.

|  |  |
| --- | --- |
| Type | string |
| Required |  |
| Validators | {'id': 'character\_limit', 'args': 255, 'rule': ValidationRule(id='character\_limit', description='Maximum number of unicode characters in string.\n', args=[{'name': 'characters', 'type': 'integer', 'help': 'The number of characters'}]), 'description': 'character\_limit(255)'}, {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'} |

**Status:** *Decided*

#### prior\_employment: Prior employment

Since arriving in the UK, have you ever undertaken any paid employment? To be collected once and submitted within 3 months of enrolment.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | Yes, No, Don't know |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'prior\_employment', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(prior\_employment)'} |

**Status:** *Decided*

#### employment\_status: Main economic status at baseline

What is your current employment status? Selection of a single category describing the employment status of the participant prior to entering RTOF.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | PC, TC, ZHC, CIH, SE, ULA, UNLA, Other |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'employment\_status\_baseline', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(employment\_status\_baseline)'} |

**Status:** *Decided*

#### hours\_of\_employment: Hours of employment

In your main job, how many hours a week do you usually work? (Include paid and unpaid overtime). For employed and self-employed.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | 0 to 7 hours, 8 to 15 hours, 16 to 30 hours, 31 to 48 hours, 49 or more hours, Don't know |
| Validators | {'id': 'dimension', 'args': 'hours\_of\_employment', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(hours\_of\_employment)'} |

**Status:** *Decided*

#### wage\_baseline: Wage baseline

tbc

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | tbc |
| Validators | {'id': 'dimension', 'args': 'wage', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(wage)'} |

**Status:** *Pending consideration*

Comments

*Categories are currently pending. Ecorys to confirm.*

#### eduction\_training\_status: Eduction / training status

What is your current education status?

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | formal education, Not in formal education |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'education\_training\_status', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(education\_training\_status)'} |

**Status:** *Decided*

#### housing\_baseline\_accommodation: Accommodation type baseline

Rough sleepers - Rough sleepers are defined for the purposes of rough sleeping counts and estimates as - 1) People sleeping, about to bed down (sitting on/in or standing next to their bedding) or actually bedded down in the open air (such as on the strets, in tents, doorways, parks, bus shelters or encampments). 2) People in buildings or other places not designed for habitation (such as stairwells, barns, sheds, car parks, cars, derelict boats, stations, or bushes). The definition does not include people in hostels or shelters, people in campsites or other sites used for recreational purposes or organised protest, squatters or travellers. Bedded down is taken to mean either lying down or sleeping. About to bed down includes those who are sitting in/on or near a sleeping bag or other bedding.  
Safe and secure accommodation - Safe accommodation is defined by compliance with the Decent Homes Standard including the absence of category 1 and 2 hazards under the Housing Health and Safety Rating System, and with all relevant gas, electricity and fire safety checks in place. Within this safe accommodation, the individual refugee/family must have appropriate individual space. This is defined as one bedroom for - every adult couple; any other adult aged 16 or over; any two children of the same sex; any two children regardless of sex under the age of 10; any other child; and access to self-contained facilities, including bathroom and kitchen, if in a shared facility. Secure accommodation is defined by security of tenure for at least six months.  
Under RTOF, accommodation may include - Private rental sector; Social housing; Supported housing N.B. Accommodation may be let under a Periodic Licence agreement where appropriate, but use of an excluded licence is not permissible Note that planned and anticipated moves between properties during the 6 months will be allowed, provided the individual is still within secure, stable accommodation at the end of the 6-month sustainment period.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | Homeless and rough sleeping, Homeless but not rough sleeping (e.g. living with friends or family), Asylum accommodation, Self-funded temporary accommodation (e.g. hotel), Publicly funded temporary accommodation, Renting - private sector, Renting - Local authority or housing association, Other safe and secure accommodation |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'housing\_baseline\_accommodation', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(housing\_baseline\_accommodation)'} |

**Status:** *Decided*

#### homelessness\_baseline: Homelessness status baseline

This data field is only applicable if either 'Homeless and rough sleeping' or 'Homeless but not rough sleeping (e.g. living with friends or family)' was selected in the housing\_baseline\_accommodation data field'.  
Statutory homelessness requires a household to be unintentionally homeless AND demonstrate a priority category need, such as dependent children. Non-statutory homelessness may include single homeless people without specific vulnerabilities, or those people who do not approach their local authority for help.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | Statutory homelessness, Non-statutory homelessness |
| Validators | {'id': 'dimension', 'args': 'homelessness\_baseline', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(homelessness\_baseline)'}, {'id': 'conditional', 'args': [{'expression': "one\_of(housing\_baseline\_accommodation, ['Homeless and rough sleeping', 'Homeless but not rough sleeping (e.g. living with friends or family)'])", 'validation': {'required': True}}], 'rule': ValidationRule(id='conditional', description='Allows for conditional application of validators', args=[{'name': 'expression', 'type': 'string', 'help': 'The evaluation expression'}, {'name': 'validation', 'type': 'validation', 'help': 'The validation that should be applied'}]), 'description': 'conditional rules apply'} |

**Status:** *Decided*

Comments

*More information - https://www.gov.uk/guidance/homelessness-data-notes-and-definitions and https://www.legislation.gov.uk/ukpga/1996/52/section/175?view=plain*

## INTEGRATION\_PLAN Record

This form outlines the required data for participants who achieve each integration plan outcome - one submission for each creation, 6-month progress and 12-month progress. Data should be collected when achieving each integration plan outcome.

Workflow: Integration

### Fields

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Type | Decision status |
| unique\_id [PK] | Unique person identifier | string | Decided |
| integration\_outcome\_type [PK] | Integration outcome type | categorical | Decided |
| integration\_outcome\_achieved\_date | Date integration plan outcome acheived | date | Decided |
| integration\_social | Social bonds / bridges / links | categorical | Blocked |
| integration\_comms\_language | Language and communication | categorical | Blocked |
| integration\_digital | Digital skills | categorical | Blocked |

#### unique\_id: Unique person identifier

Person identifier provided in the person form. This can be a NINO or a unique provider ID. This must be unique for each individual supported on RTOF.

|  |  |
| --- | --- |
| Type | string |
| Primary Key | True |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'unique', 'args': True, 'rule': ValidationRule(id='unique', description="A field with 'unique: true' should be unique within the dataset for this provider.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'unique(True)'} |

**Status:** *Decided*

#### integration\_outcome\_type: Integration outcome type

Selection of which integration outcome has been achieved. To be collected at each outcome submission.

|  |  |
| --- | --- |
| Type | categorical |
| Primary Key | True |
| Required |  |
| One of | Creation, 6 month, 12 month |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'integration\_outcome\_type', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(integration\_outcome\_type)'} |

**Status:** *Decided*

#### integration\_outcome\_achieved\_date: Date integration plan outcome acheived

Date of integration plan outcome achievement. To be collected at each outcome submission.

|  |  |
| --- | --- |
| Type | date |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'date\_after', 'args': 'date\_started\_service', 'rule': ValidationRule(id='date\_after', description='Only used for fields of type date, this validator ensures that the provided value is after the date indicated. When\nmultiple\n', args=[{'name': 'field\_id', 'type': 'array', 'items': {'type': 'string'}, 'help': 'The ID of the field that this date has to be after'}]), 'description': 'date\_after(date\_started\_service)'} |

**Status:** *Decided*

#### integration\_social: Social bonds / bridges / links

tbc. To be collected at each outcome submission.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | tbc |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'integration\_social', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(integration\_social)'} |

**Status:** *Blocked*

Comments

*tbc - Ecorys to confirm categories after provider engagement*

#### integration\_comms\_language: Language and communication

tbc. To be collected at each outcome submission.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | tbc |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'integration\_comms\_language', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(integration\_comms\_language)'} |

**Status:** *Blocked*

Comments

*tbc - Ecorys to confirm categories after provider engagement*

#### integration\_digital: Digital skills

tbc. To be collected at each outcome submission.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | tbc |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'integration\_digital', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(integration\_digital)'} |

**Status:** *Blocked*

Comments

*tbc - Ecorys to confirm categories after provider engagement*

## EMPLOYMENT\_INTERMEDIATE Record

This form outlines the required data for participants who achieve the intermediate employment outcome. Data should be collected when achieving the intermediate outcome.

Workflow: Employment

### Fields

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Type | Decision status |
| unique\_id [PK] | Unique person identifier | string | Decided |
| date\_intermediate\_employment\_outcome | Date achieved intermediate outcome | date | Decided |
| intermediate\_employment\_outcome\_type | Type of intermediate outcome | list | Decided |

#### unique\_id: Unique person identifier

Person identifier provided in the person form. This can be a NINO or a unique provider ID. This must be unique for each individual supported on RTOF.

|  |  |
| --- | --- |
| Type | string |
| Primary Key | True |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'unique', 'args': True, 'rule': ValidationRule(id='unique', description="A field with 'unique: true' should be unique within the dataset for this provider.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'unique(True)'} |

**Status:** *Decided*

#### date\_intermediate\_employment\_outcome: Date achieved intermediate outcome

Date of intermediate employment outcome achievement. To be collected once at intermediate employment outcome submission.

|  |  |
| --- | --- |
| Type | date |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'date\_after', 'args': 'date\_started\_service', 'rule': ValidationRule(id='date\_after', description='Only used for fields of type date, this validator ensures that the provided value is after the date indicated. When\nmultiple\n', args=[{'name': 'field\_id', 'type': 'array', 'items': {'type': 'string'}, 'help': 'The ID of the field that this date has to be after'}]), 'description': 'date\_after(date\_started\_service)'} |

**Status:** *Decided*

#### intermediate\_employment\_outcome\_type: Type of intermediate outcome

Selection of at least 3 intermediate outcomes achieved from the provided list [link to list]. To be collected once at intermediate employment outcome submission.

|  |  |
| --- | --- |
| Type | list |
| Required |  |
| One of | qualifications or skills, training course, mentoring and/or coaching, employability, Volunteering, work experience, Internship, sector-specific language training, business plan for self-employment, Registered business for self-employment, another specified activity |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'intermediate\_employment\_outcome\_type', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(intermediate\_employment\_outcome\_type)'}, {'id': 'count\_min', 'args': 3, 'rule': ValidationRule(id='count\_min', description='Minimum numbers of entries in a list\n', args=[{'name': 'number', 'type': 'integer', 'help': 'The number of entries'}]), 'description': 'count\_min(3)'} |

**Status:** *Decided*

## EMPLOYMENT\_ENTRY Record

This form outlines the required data for participants who achieve the employment OR self-employment entry outcome. A participant can only achieve either the employment or self-employment outcome. Data should be collected when achieving the employment entry outcome.

Workflow: Employment

### Fields

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Type | Decision status |
| unique\_id [PK] | Unique person identifier | string | Decided |
| employment\_entry\_outcome\_type | Type of employment entry outcome | categorical | Decided |
| date\_employment\_entry | Date entered employment | date | Decided |
| primary\_employment\_entry\_status | Primary employment entry status | categorical | Decided |
| secondary\_employment\_entry\_status | Seconday employment entry status | categorical | Decided |
| hours\_of\_employment\_entry | Hours of employment | categorical | Decided |
| wage\_employment\_entry | Wage employment entry | categorical | Pending consideration |
| eduction\_training\_status\_entry | Eduction / training status | categorical | Decided |
| employment\_entry\_occupation | Occupation type | string | Decided |
| employment\_entry\_sector | Sector | string | Decided |

#### unique\_id: Unique person identifier

Person identifier provided in the person form. This can be a NINO or a unique provider ID. This must be unique for each individual supported on RTOF.

|  |  |
| --- | --- |
| Type | string |
| Primary Key | True |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'} |

**Status:** *Decided*

#### employment\_entry\_outcome\_type: Type of employment entry outcome

Employment entry can either be an employment outcome or self-employment outcome. To be collected once at entry to employment outcome submission.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | Employment, Self-employment |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'employment\_entry\_outcome\_type', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(employment\_entry\_outcome\_type)'} |

**Status:** *Decided*

#### date\_employment\_entry: Date entered employment

Date of employment entry submission. To be collected once at entry to employment outcome submission.

|  |  |
| --- | --- |
| Type | date |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'date\_after', 'args': 'date\_started\_service', 'rule': ValidationRule(id='date\_after', description='Only used for fields of type date, this validator ensures that the provided value is after the date indicated. When\nmultiple\n', args=[{'name': 'field\_id', 'type': 'array', 'items': {'type': 'string'}, 'help': 'The ID of the field that this date has to be after'}]), 'description': 'date\_after(date\_started\_service)'} |

**Status:** *Decided*

#### primary\_employment\_entry\_status: Primary employment entry status

What is your primary current employment status?

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | PC, TC, ZHC, CIH, SE, Other |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'employment\_status', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(employment\_status)'} |

**Status:** *Decided*

#### secondary\_employment\_entry\_status: Seconday employment entry status

If you have more than one economic status, what is your current secondary employment status?

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | PC, TC, ZHC, CIH, SE, Other |
| Validators | {'id': 'dimension', 'args': 'employment\_status', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(employment\_status)'} |

**Status:** *Decided*

#### hours\_of\_employment\_entry: Hours of employment

In your main job, how many hours a week do you usually work? (Include paid and unpaid overtime). For employed and self-employed.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | 0 to 7 hours, 8 to 15 hours, 16 to 30 hours, 31 to 48 hours, 49 or more hours, Don't know |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'hours\_of\_employment', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(hours\_of\_employment)'} |

**Status:** *Decided*

Comments

*Ecorys addition (based on Census wording) to ensure emploment questions are MECE / required for CBA*

#### wage\_employment\_entry: Wage employment entry

Estimated hourly wage

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | tbc |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'wage', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(wage)'} |

**Status:** *Pending consideration*

#### eduction\_training\_status\_entry: Eduction / training status

What is your education status?

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | formal education, Not in formal education |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'education\_training\_status', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(education\_training\_status)'} |

**Status:** *Decided*

Comments

*Ecorys addition (based on Census wording) to ensure emploment questions are MECE*

#### employment\_entry\_occupation: Occupation type

State participants employment type for the achievement of entry to employment outcome. To be collected once at entry to employment outcome submission. For some providers this may be a categorical data field - please confirm with your nominated data lead.

|  |  |
| --- | --- |
| Type | string |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'character\_limit', 'args': 255, 'rule': ValidationRule(id='character\_limit', description='Maximum number of unicode characters in string.\n', args=[{'name': 'characters', 'type': 'integer', 'help': 'The number of characters'}]), 'description': 'character\_limit(255)'} |

**Status:** *Decided*

#### employment\_entry\_sector: Sector

State the participants employment sector for the achievement of entry to employment outcome. To be collected once at entry to employment outcome submission. For some providers this may be a categorical data field - please confirm with your nominated data lead.

|  |  |
| --- | --- |
| Type | string |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'character\_limit', 'args': 255, 'rule': ValidationRule(id='character\_limit', description='Maximum number of unicode characters in string.\n', args=[{'name': 'characters', 'type': 'integer', 'help': 'The number of characters'}]), 'description': 'character\_limit(255)'} |

**Status:** *Decided*

## EMPLOYMENT\_SUSTAIN Record

This form outlines the required data for participants who achieve the employment sustainment outcome. Data should be collected when achieving the employment sustainment outcome.

Workflow: Employment

### Fields

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Type | Decision status |
| unique\_id [PK] | Unique person identifier | string | Decided |
| employment\_sustainment\_date | Date employment sustainment outcome achieved | date | Decided |
| employment\_sustain\_status | Employment status at sustainment outcome | categorical | Decided |
| hours\_of\_employment\_sustain | Hours of employment at sustain | categorical | Decided |
| wage\_employment\_sustain | Wage employment at sustain | categorical | Pending consideration |
| eduction\_training\_status\_sustain | Eduction / training status at sustain | categorical | Decided |
| employment\_susatin\_occupation | Occupation type | string | Decided |
| employment\_sustain\_sector | Sector | string | Decided |

#### unique\_id: Unique person identifier

Person identifier provided in the person form. This can be a NINO or a unique provider ID. This must be unique for each individual supported on RTOF.

|  |  |
| --- | --- |
| Type | string |
| Primary Key | True |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'unique', 'args': True, 'rule': ValidationRule(id='unique', description="A field with 'unique: true' should be unique within the dataset for this provider.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'unique(True)'} |

**Status:** *Decided*

#### employment\_sustainment\_date: Date employment sustainment outcome achieved

Date of employment sustainment outcome achieved. This can be either be for the self-employment or employment route. To be collected once at employment sustained outcome submission.

|  |  |
| --- | --- |
| Type | date |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'date\_after', 'args': 'date\_employment\_entry', 'rule': ValidationRule(id='date\_after', description='Only used for fields of type date, this validator ensures that the provided value is after the date indicated. When\nmultiple\n', args=[{'name': 'field\_id', 'type': 'array', 'items': {'type': 'string'}, 'help': 'The ID of the field that this date has to be after'}]), 'description': 'date\_after(date\_employment\_entry)'} |

**Status:** *Decided*

#### employment\_sustain\_status: Employment status at sustainment outcome

What is your current employment status?

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | PC, TC, ZHC, CIH, SE, Other |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'employment\_status', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(employment\_status)'} |

**Status:** *Decided*

#### hours\_of\_employment\_sustain: Hours of employment at sustain

In your main job, how many hours a week do you usually work? (Include paid and unpaid overtime). For employed and self-employed.

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | 0 to 7 hours, 8 to 15 hours, 16 to 30 hours, 31 to 48 hours, 49 or more hours, Don't know |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'hours\_of\_employment', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(hours\_of\_employment)'} |

**Status:** *Decided*

Comments

*Ecorys addition (based on Census wording) to ensure emploment questions are MECE / required for CBA*

#### wage\_employment\_sustain: Wage employment at sustain

Estimated hourly wage

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | tbc |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'wage', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(wage)'} |

**Status:** *Pending consideration*

#### eduction\_training\_status\_sustain: Eduction / training status at sustain

What is your education status?

|  |  |
| --- | --- |
| Type | categorical |
| Required |  |
| One of | formal education, Not in formal education |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'dimension', 'args': 'education\_training\_status', 'rule': ValidationRule(id='dimension', description='Only used for fields of type categorical, this validator ensures that the provided value is part of category\nlist identified.\n', args=[{'name': 'category\_id', 'type': 'string', 'help': 'The ID of the category that this list has to be a member of.'}]), 'description': 'dimension(education\_training\_status)'} |

**Status:** *Decided*

Comments

*Ecorys addition (based on Census wording) to ensure emploment questions are MECE*

#### employment\_susatin\_occupation: Occupation type

State participants employment type for the achievement of entry to employment outcome. To be collected once at entry to employment outcome submission. For some providers this may be a categorical data field - please confirm with your nominated data lead.

|  |  |
| --- | --- |
| Type | string |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'character\_limit', 'args': 255, 'rule': ValidationRule(id='character\_limit', description='Maximum number of unicode characters in string.\n', args=[{'name': 'characters', 'type': 'integer', 'help': 'The number of characters'}]), 'description': 'character\_limit(255)'} |

**Status:** *Decided*

#### employment\_sustain\_sector: Sector

State the participants employment sector for the achievement of entry to employment outcome. To be collected once at entry to employment outcome submission. For some providers this may be a categorical data field - please confirm with your nominated data lead.

|  |  |
| --- | --- |
| Type | string |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'character\_limit', 'args': 255, 'rule': ValidationRule(id='character\_limit', description='Maximum number of unicode characters in string.\n', args=[{'name': 'characters', 'type': 'integer', 'help': 'The number of characters'}]), 'description': 'character\_limit(255)'} |

**Status:** *Decided*

## DATE\_LAST\_SEEN Record

This form outlines the required data for participants who are no longer engaged with the service. Data should be collected during the annual caseload review.

Workflow: Annual review

### Fields

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Type | Decision status |
| unique\_id [PK] | Unique person identifier | string | Decided |
| date\_last\_seen | Date last seen | date | Decided |
| reason\_for\_disengagement | Reason for disengagement | string | Decided |

#### unique\_id: Unique person identifier

Person identifier provided in the person form. This can be a NINO or a unique provider ID. This must be unique for each individual supported on RTOF.

|  |  |
| --- | --- |
| Type | string |
| Primary Key | True |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'unique', 'args': True, 'rule': ValidationRule(id='unique', description="A field with 'unique: true' should be unique within the dataset for this provider.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'unique(True)'} |

**Status:** *Decided*

#### date\_last\_seen: Date last seen

Date of last interaction with participant. Month and Year required. To be collected annually during annual caseload review.

|  |  |
| --- | --- |
| Type | date |
| Required |  |
| Validators | {'id': 'required', 'args': True, 'rule': ValidationRule(id='required', description="A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand\nfor 'notnull: true' and 'notblank: true'.\n", args=[{'name': 'enabled', 'type': 'boolean', 'help': 'true/false value indicating if the current field is required'}]), 'description': 'required(True)'}, {'id': 'date\_after', 'args': 'date\_started\_service', 'rule': ValidationRule(id='date\_after', description='Only used for fields of type date, this validator ensures that the provided value is after the date indicated. When\nmultiple\n', args=[{'name': 'field\_id', 'type': 'array', 'items': {'type': 'string'}, 'help': 'The ID of the field that this date has to be after'}]), 'description': 'date\_after(date\_started\_service)'} |

**Status:** *Decided*

#### reason\_for\_disengagement: Reason for disengagement

Optional field. If the reason for a participant disengagement from RTOF is known, please provide details. ie. 'Moved out of area'

|  |  |
| --- | --- |
| Type | string |
| Required |  |
| Validators | {'id': 'character\_limit', 'args': 255, 'rule': ValidationRule(id='character\_limit', description='Maximum number of unicode characters in string.\n', args=[{'name': 'characters', 'type': 'integer', 'help': 'The number of characters'}]), 'description': 'character\_limit(255)'} |

**Status:** *Decided*

# Dimensions

The following categorical lists are referenced throughout the specification.

* hours\_of\_employment
* gender
* intermediate\_employment\_outcome\_type
* referral\_source
* homelessness\_baseline
* integration\_social
* employment\_status\_baseline
* nationality
* current\_dependents\_uk
* housing\_baseline\_accommodation
* employed\_in\_home\_country
* housing\_accomodation
* employment\_entry\_outcome\_type
* employment\_status
* integration\_outcome\_type
* highest\_qualification\_achieved
* education\_training\_status
* integration\_digital
* wage
* age\_profile\_of\_dependents\_uk
* prior\_employment
* integration\_comms\_language
* dispersal\_area
* language\_level\_on\_entry

#### hours\_of\_employment

|  |  |
| --- | --- |
| Value | Description |
| 0 to 7 hours | None |
| 8 to 15 hours | None |
| 16 to 30 hours | None |
| 31 to 48 hours | None |
| 49 or more hours | None |
| Don't know | None |

#### gender

|  |  |
| --- | --- |
| Value | Description |
| Man | None |
| Woman | None |
| Non-binary | None |
| Other | None |
| Prefer not to say | None |

#### intermediate\_employment\_outcome\_type

|  |  |
| --- | --- |
| Value | Description |
| qualifications or skills | Successful formal recognition and/or comparison of qualifications or skills |
| training course | Completion of relevant training course |
| mentoring and/or coaching | Completion of 10 hours+ mentoring and/or coaching |
| employability | Completion of 10 hours+ employability training |
| Volunteering | Volunteering - at least 10 days |
| work experience | Work experience - at least 5 days |
| Internship | Internship- maximum 10 days, unless paid for at the National Minimum Wage (or above) for people aged 18-22 or the National Living Wage (or above) for people aged 23+\* |
| sector-specific language training | Completion of sector-specific language training which is additional to the mainstream. |
| business plan for self-employment | Developed business plan for self-employment |
| Registered business for self-employment | Registered business for self-employment |
| another specified activity | another specified activity directly relevant to achieving the individual’s employment goals |

#### referral\_source

|  |  |
| --- | --- |
| Value | Description |
| LA | Local Authority |
| Home Office | None |
| Local VCS | Local voluntary and community sector |
| Primary Care | None |
| Other Health | None |
| HO providers | Home Office Asylum Housing Providers (e.g. Serco) |
| Migrant Help | None |
| Self-referrals | None |
| DWP | Governmentment Departmnet for Work and Pensions |
| Peer | Peer to peer referrals |
| Other | Any other referral source |

#### homelessness\_baseline

|  |  |
| --- | --- |
| Value | Description |
| Statutory homelessness | i.e. main duty/priority group |
| Non-statutory homelessness | i.e. NOT main duty/priority group |

#### integration\_social

|  |  |
| --- | --- |
| Value | Description |
| tbc | None |

#### employment\_status\_baseline

|  |  |
| --- | --- |
| Value | Description |
| PC | Working as an employee on a permanent contract (or on leave from employment, for example maternity or paternity leave) |
| TC | Working as an employee on a temporary contract |
| ZHC | Working as an employee on a zero hours contract |
| CIH | Working as an employee cash in hand |
| SE | Self employed or freelance (Freelance means that you are self-employed and work for different companies or people on particular pieces of work) |
| ULA | Unemployed and looking/available for work immediately (in the next two weeks) |
| UNLA | Unemployed and NOT looking/available for work immediately (for example, retired, looking after family or long term sick or disabled) |
| Other | None |

#### nationality

|  |  |
| --- | --- |
| Value | Description |
| AD | Andorra |
| AE | United Arab Emirates |
| AF | Afghanistan |
| AG | Antigua and Barbuda |
| AI | Anguilla |
| AL | Albania |
| AM | Armenia |
| AO | Angola |
| AQ | Antarctica |
| AR | Argentina |
| AS | American Samoa |
| AT | Austria |
| AU | Australia |
| AW | Aruba |
| AX | Åland Islands |
| AZ | Azerbaijan |
| BA | Bosnia and Herzegovina |
| BB | Barbados |
| BD | Bangladesh |
| BE | Belgium |
| BF | Burkina Faso |
| BG | Bulgaria |
| BH | Bahrain |
| BI | Burundi |
| BJ | Benin |
| BL | Saint Barthélemy |
| BM | Bermuda |
| BN | Brunei |
| BO | Bolivia |
| BQ | Caribbean Netherlands |
| BR | Brazil |
| BS | Bahamas |
| BT | Bhutan |
| BV | Bouvet Island |
| BW | Botswana |
| BY | Belarus |
| BZ | Belize |
| CA | Canada |
| CC | Cocos (Keeling) Islands |
| CD | Democratic Republic of the Congo |
| CF | Central African Republic |
| CG | Congo (Republic of) |
| CH | Switzerland |
| CI | Côte d'Ivoire (Ivory Coast) |
| CK | Cook Islands |
| CL | Chile |
| CM | Cameroon |
| CN | China |
| CO | Colombia |
| CR | Costa Rica |
| CU | Cuba |
| CV | Cape Verde |
| CW | Curaçao |
| CX | Christmas Island |
| CY | Cyprus |
| CZ | Czech Republic |
| DE | Germany |
| DJ | Djibouti |
| DK | Denmark |
| DM | Dominica |
| DO | Dominican Republic |
| DZ | Algeria |
| EC | Ecuador |
| EE | Estonia |
| EG | Egypt |
| EH | Western Saharan |
| ER | Eritrea |
| ES | Spain |
| ET | Ethiopia |
| FI | Finland |
| FJ | Fiji |
| FK | Falkland Islands |
| FM | Micronesia |
| FO | Faroe Islands |
| FR | France |
| GA | Gabon |
| GB | United Kingdom |
| GD | Grenada |
| GE | Georgia |
| GF | French Guiana |
| GG | Guernsey |
| GH | Ghana |
| GI | Gibraltar |
| GL | Greenland |
| GM | Gambia |
| GN | Guinea |
| GP | Guadeloupe |
| GQ | Equatorial Guinea |
| GR | Greece |
| GS | South Georgia and the South Sandwich Islands |
| GT | Guatemala |
| GU | Guam |
| GW | Guinea-Bissau |
| GY | Guyana |
| HK | Hong Kong |
| HM | Heard and McDonald Islands |
| HN | Honduras |
| HR | Croatia |
| HT | Haiti |
| HU | Hungary |
| ID | Indonesia |
| IE | Ireland |
| IL | Israel |
| IM | Isle of Man |
| IN | India |
| IO | British Indian Ocean Territory |
| IQ | Iraq |
| IR | Iran |
| IS | Iceland |
| IT | Italy |
| JE | Jersey |
| JM | Jamaica |
| JO | Jordan |
| JP | Japan |
| KE | Kenya |
| KG | Kyrgyzstan |
| KH | Cambodia |
| KI | Kiribati |
| KM | Comoros |
| KN | Saint Kitts and Nevis |
| KP | North Korea |
| KR | South Korea |
| KW | Kuwait |
| KY | Cayman Islands |
| KZ | Kazakhstan |
| LA | Laos |
| LB | Lebanon |
| LC | Saint Lucia |
| LI | Liechtenstein |
| LK | Sri Lanka |
| LR | Liberia |
| LS | Lesotho |
| LT | Lithuania |
| LU | Luxembourg |
| LV | Latvia |
| LY | Libya |
| MA | Morocco |
| MC | Monaco |
| MD | Moldova |
| ME | Montenegro |
| MF | Saint Martin (France) |
| MG | Madagascar |
| MH | Marshall Islands |
| MK | Macedonia |
| ML | Mali |
| MM | Burma (Republic of the Union of Myanmar) |
| MN | Mongolia |
| MO | Macau |
| MP | Northern Mariana Islands |
| MQ | Martinique |
| MR | Mauritania |
| MS | Montserrat |
| MT | Malta |
| MU | Mauritius |
| MV | Maldives |
| MW | Malawi |
| MX | Mexico |
| MY | Malaysia |
| MZ | Mozambique |
| NA | Namibia |
| NC | New Caledonia |
| NE | Niger |
| NF | Norfolk Island |
| NG | Nigeria |
| NI | Nicaragua |
| NL | Netherlands |
| NO | Norway |
| NP | Nepal |
| NR | Nauru |
| NU | Niue |
| NZ | New Zealand |
| OM | Oman |
| PA | Panama |
| PE | Peru |
| PF | French Polynesia |
| PG | Papua New Guinea |
| PH | Philippines |
| PK | Pakistan |
| PL | Poland |
| PM | St. Pierre and Miquelon |
| PN | Pitcairn |
| PR | Puerto Rico |
| PS | Palestine |
| PT | Portugal |
| PW | Palau |
| PY | Paraguay |
| QA | Qatar |
| RE | Réunion |
| RO | Romania |
| RS | Serbia |
| RU | Russian Federation |
| RW | Rwanda |
| SA | Saudi Arabia |
| SB | Solomon Islands |
| SC | Seychelles |
| SD | Sudan |
| SE | Sweden |
| SG | Singapore |
| SH | Saint Helena |
| SI | Slovenia |
| SJ | Svalbard and Jan Mayen Islands |
| SK | Slovakia |
| SL | Sierra Leone |
| SM | San Marino |
| SN | Senegal |
| SO | Somalia |
| SR | Suriname |
| SS | South Sudan |
| ST | São Tome and Príncipe |
| SV | El Salvador |
| SX | Saint Martin (Netherlands) |
| SY | Syria |
| SZ | Swaziland |
| TC | Turks and Caicos Islands |
| TD | Chad |
| TF | French Southern Territories |
| TG | Togo |
| TH | Thailand |
| TJ | Tajikistan |
| TK | Tokelau |
| TL | Timor-Leste |
| TM | Turkmenistan |
| TN | Tunisia |
| TO | Tonga |
| TR | Turkey |
| TT | Trinidad and Tobago |
| TV | Tuvalu |
| TW | Taiwan |
| TZ | Tanzania |
| UA | Ukraine |
| UG | Uganda |
| UM | United States Minor Outlying Islands |
| US | United States of America |
| UY | Uruguay |
| UZ | Uzbekistan |
| VA | Vatican |
| VC | Saint Vincent and Grenadines |
| VE | Venezuela |
| VG | British Virgin Islands |
| VI | United States Virgin Islands |
| VN | Vietnam |
| VU | Vanuatu |
| WF | Wallis and Futuna Islands |
| WS | Samoa |
| YE | Yemen |
| YT | Mayotte |
| ZA | South Africa |
| ZM | Zambia |
| ZW | Zimbabwe |

#### current\_dependents\_uk

|  |  |
| --- | --- |
| Value | Description |
| Yes | None |
| No | None |

#### housing\_baseline\_accommodation

|  |  |
| --- | --- |
| Value | Description |
| Homeless and rough sleeping | None |
| Homeless but not rough sleeping (e.g. living with friends or family) | None |
| Asylum accommodation | None |
| Self-funded temporary accommodation (e.g. hotel) | None |
| Publicly funded temporary accommodation | None |
| Renting - private sector | None |
| Renting - Local authority or housing association | None |
| Other safe and secure accommodation | None |

#### employed\_in\_home\_country

|  |  |
| --- | --- |
| Value | Description |
| Yes | None |
| No | None |

#### housing\_accomodation

|  |  |
| --- | --- |
| Value | Description |
| RPC | Renting - private sector |
| RLA | Renting - Local authority or housing association |
| other | Other safe and secure accommodation |

#### employment\_entry\_outcome\_type

|  |  |
| --- | --- |
| Value | Description |
| Employment | None |
| Self-employment | None |

#### employment\_status

|  |  |
| --- | --- |
| Value | Description |
| PC | Working as an employee on a permanent contract (or on leave from employment, for example maternity or paternity leave) |
| TC | Working as an employee on a temporary contract |
| ZHC | Working as an employee on a zero hours contract |
| CIH | Working as an employee cash in hand |
| SE | Self employed or freelance (Freelance means that you are self-employed and work for different companies or people on particular pieces of work) |
| Other | None |

#### integration\_outcome\_type

|  |  |
| --- | --- |
| Value | Description |
| Creation | None |
| 6 month | None |
| 12 month | None |

#### highest\_qualification\_achieved

|  |  |
| --- | --- |
| Value | Description |
| Below formal levels | None |
| Entry level | None |
| Level 1-2 | None |
| Level 3-4 | None |
| Level 5+ | None |
| Unknown | None |

#### education\_training\_status

|  |  |
| --- | --- |
| Value | Description |
| formal education | In formal education or training (for example, registered in college or university, an apprenticeship, or a government employment and training programme. This may be combined with paid employment) |
| Not in formal education | Not in formal education or training (i.e. not formally registered) |

#### integration\_digital

|  |  |
| --- | --- |
| Value | Description |
| tbc | None |

#### wage

|  |  |
| --- | --- |
| Value | Description |
| tbc | None |

#### age\_profile\_of\_dependents\_uk

|  |  |
| --- | --- |
| Value | Description |
| 0-4 | None |
| 5-11 | None |
| 11-17 | None |
| Caring responsibilities for an adult | None |

#### prior\_employment

|  |  |
| --- | --- |
| Value | Description |
| Yes | None |
| No | None |
| Don't know | None |

#### integration\_comms\_language

|  |  |
| --- | --- |
| Value | Description |
| tbc | None |

#### dispersal\_area

|  |  |
| --- | --- |
| Value | Description |
| Bolton Council | None |
| Bury Council | None |
| Manchester City Council | None |
| Oldham Council | None |
| Rochdale Borough Council | None |
| Salford City Council | None |
| Stockport Metropolitan Borough Council | None |
| Tameside Metropolitan Borough Council | None |
| Trafford Council | None |
| Wigan Council | None |
| Wolverhampton Council | None |
| Coventry City Council | None |
| Sandwell Council | None |
| Birmingham City Council | None |
| Plymouth City Council | None |
| Darlington Borough Council | None |
| Gateshead Council | None |
| Hartlepool Borough Council | None |
| Middlesbrough Council | None |
| Newcastle City Council | None |
| North Tyneside Council | None |
| Northumberland County Council | None |
| Redcar and Cleveland Council | None |
| South Tyneside Council | None |
| Stockton Council | None |
| Sunderland City Council | None |

#### language\_level\_on\_entry

|  |  |
| --- | --- |
| Value | Description |
| Below formal levels | None |
| Pre-entry | None |
| Entry level 1 | None |
| Entry level 2 | None |
| Entry level 3 | None |
| Level 1 | None |
| Level 2 | None |
| Beyond Level 2 | None |

# Validation Rules

To ensure high-quality and accurate data, the specification includes a number of validation rules. These rules will be checked upon processing of received data, and an error-report may be generated including references to the record and field that contained the error and which rule was found in violation. These rules will help trace issues.

## required

A field with 'required: true' must be present in the data record, and must have a non-blank value. It is short-hand  
for 'notnull: true' and 'notblank: true'.

## unique

A field with 'unique: true' should be unique within the dataset for this provider.

## notnull

A field with 'notnull: true' must be present in the data record.

## notblank

A field with 'notblank: true' must have a value that's not blank (i.e. an empty string or a string  
consisting entirely of whitespace).

## dimension

Only used for fields of type categorical, this validator ensures that the provided value is part of category  
list identified.

## date\_after

Only used for fields of type date, this validator ensures that the provided value is after the date indicated. When  
multiple

## national\_insurance\_number

UK National Insurance Number - uppercase with all whitespace removed. Validated according to format given in  
https://github.com/dwp/nino-format-validation

## character\_limit

Maximum number of unicode characters in string.

## count\_min

Minimum numbers of entries in a list

## conditional

Allows for conditional application of validators