
LONGITUDINAL EMPLOYER - HOUSEHOLD DYNAMICS

INTERNAL DOCUMENT No. IP-LEHD-BRB-1.1.10

LEHD Business Register Bridge Technical documentation

Code version : 1.1.10
Author : Hyowook Chiang, Kristin Sandusky, and Lars Vilhuber
Date : June 9, 2005

Contents

- 1 LEHD Business Register Bridge Technical documentation 2**
- 1.1 Overview 3
 - 1.1.1 Definition of BRB 3
 - 1.1.2 Update frequency 3
 - 1.1.3 Acquisition process 3
 - 1.1.4 Processing description 4
 - 1.1.5 Naming conventions 4
- 1.2 Characterizing the Bridge Between LEHD Data and Census Business Data 4
 - 1.2.1 General Description 4
 - 1.2.2 File Structure and Contents 4
 - 1.2.3 How to Use The BRB 6
 - 1.2.4 Some warnings and caveats 9
- 1.3 Data set description 9
 - 1.3.1 Crosswalk 9
 - 1.3.2 BR list 10
 - 1.3.3 ECF list 10
- 1.4 Data dictionary 11
 - 1.4.1 Crosswalk 11
 - 1.4.2 BR list 36
 - 1.4.3 ECF list 45
- 1.5 Details 54
 - 1.5.1 SAS programs used 55
 - 1.5.2 SAS macros used 73
 - 1.5.3 SAS layouts used 84
 - 1.5.4 SAS formats used 85
 - 1.5.5 Scripts used 86
 - 1.5.6 Total code lines for this process 87

Chapter 1

LEHD Business Register Bridge Technical documentation

Contents

- 1.1 Overview 3**
 - 1.1.1 Definition of BRB 3
 - 1.1.2 Update frequency 3
 - 1.1.3 Acquisition process 3
 - 1.1.4 Processing description 4
 - 1.1.5 Naming conventions 4
- 1.2 Characterizing the Bridge Between LEHD Data and Census Business Data 4**
 - 1.2.1 General Description 4
 - 1.2.2 File Structure and Contents 4
 - 1.2.2.1 Identifiers 4
 - 1.2.2.2 Unit of observation 5
 - 1.2.2.3 Structure of output files 5
 - 1.2.3 How to Use The BRB 6
 - 1.2.3.1 Some considerations 6
 - 1.2.3.2 Choosing the Link Variables 6
 - 1.2.3.3 Choosing the Base File and Unit of Observation 7
 - 1.2.3.4 Example 7
 - 1.2.3.5 Summary of How to Use This File: 7
 - 1.2.4 Some warnings and caveats 9
 - 1.2.4.1 Active establishments 9
 - 1.2.4.2 Coverage 9
 - 1.2.4.3 Discrepancies in geo and industry codes 9
- 1.3 Data set description 9**
 - 1.3.1 Crosswalk 9
 - 1.3.2 BR list 10
 - 1.3.3 ECF list 10
- 1.4 Data dictionary 11**
 - 1.4.1 Crosswalk 11
 - 1.4.2 BR list 36
 - 1.4.3 ECF list 45
- 1.5 Details 54**
 - 1.5.1 SAS programs used 55
 - 1.5.1.1 library/sasprogs/01_brb.sas 55
 - 1.5.1.2 library/sasprogs/02_brb.sas 56
 - 1.5.1.3 library/sasprogs/03_brb.sas 56
 - 1.5.1.4 library/sasprogs/04_brb.sas 57

1.5.1.5	library/sasprogs/05_brb.sas	63
1.5.1.6	library/sasprogs/06_brb.sas	65
1.5.1.7	library/sasprogs/07_brb.sas	66
1.5.1.8	library/sasprogs/08_brb.sas	67
1.5.1.9	library/sasprogs/01_brbqa.sas	69
1.5.1.10	library/sasprogs/02_brbqa.sas	70
1.5.1.11	library/sasprogs/03_brbqa.sas	70
1.5.2	SAS macros used	73
1.5.2.1	library/macros/brb_formats.sas	73
1.5.2.2	library/macros/brbqa_geo.sas	74
1.5.2.3	library/macros/capture_index.sas	75
1.5.2.4	library/macros/capture_sort.sas	75
1.5.2.5	library/macros/ces_transfer_appendixa.sas	76
1.5.2.6	library/macros/ces_transfer_appendixb.sas	77
1.5.2.7	library/macros/ces_transfer_cover.sas	78
1.5.2.8	library/macros/ces_transfer_script.sas	79
1.5.2.9	library/macros/ces_transfer_settitle.sas	79
1.5.2.10	library/macros/read_ecf.sas	79
1.5.2.11	library/macros/read_ssel.sas	80
1.5.2.12	library/macros/recreate_indexes.sas	81
1.5.3	SAS layouts used	84
1.5.4	SAS formats used	85
1.5.5	Scripts used	86
1.5.6	Total code lines for this process	87

1.1 Overview

1.1.1 Definition of BRB

The LEHD employer-level data are at an establishment-level and can be linked to Census Bureau establishment and firm level micro data (e.g. Economic Censuses and Surveys) via this bridge . The bridge provides a crosswalk at various levels of business-unit aggregation. The most detailed exact crosswalk is at an EIN, State, 4-digit industry, county level. The bridge includes the full list of establishments in the LEHD data and in the SSEL that are associated with the business units (e.g., EIN/4-digit SIC/State/County) in the crosswalk and measures of activity (e.g., employment, sales) that can be used to aggregate the establishment level data.

The BRB is a research file. As such, certain caveats apply. Please be certain to read Section 1.2.3 on how to use the BRB, and in particular, Section 1.2.4 for some important cautions. Also, check back regularly for updates to this document. If you as an active user of this file notice discrepancies or problems, please notify the authors of this document and/or your data providers.

1.1.2 Update frequency

Updates occur annually.

1.1.3 Acquisition process

The BRB requires presence of BR (Business Register, also known as SSEL, see the BR codebook, yearly acquisition) and the ECF (quarterly updates).

1.1.4 Processing description

Extracts of the Business Register (formerly known as the SSEL) and the ECF are built, properly aggregated, and output into a single file. A more detailed description is available in Section 1.2.

1.1.5 Naming conventions

Three data files are produced and transferred. The data files from this process conform to LEHD naming conventions and are called

- `brb_us_xwalk`: the actual bridge, by year, at different levels of aggregation.
- `brb_us_ecflist`: list of SEINUNITs on the ECF, by quarter
- `brb_us_brlist`: list of CFNs on the BR, by year

Once on the CES system, these files are identified as “LEHD-BRB” and renamed to

- `lehd_us_brb`: the actual BR Bridge, by year, at different levels of aggregation.
- `lehd_us_brb_ecflist`: list of SEINUNITs on the ECF, by quarter
- `lehd_us_brb_brlist`: list of CFNs on the BR, by year

1.2 Characterizing the Bridge Between LEHD Data and Census Business Data

1.2.1 General Description

One of the many advantages of the LEHD database is the ability it offers to link information from Census Bureau business-level surveys (such as the Annual Survey of Manufacturers, ASM, or Business Expenditure Survey, BES) to the set of workers employed by these units. The *LEHD Business Register Bridge* (BRB) provides researchers with the tool to make this linkage.

The LEHD data available through the RDCs is described in detail in other documents. In this document, we provide a description of the BRB itself as well as an overview of the business data that is available through this link.

The primary bridge between the LEHD data and the business data is the U.S. Census Bureau’s annual Business Register (BR), a list of establishments the Bureau uses to develop the initial mailing list for the economic censuses and surveys. The BR contains data from several different sources. Primarily, however, the BR contains very reliable information on business identifiers, business organizational structure, and business location. All of this detail helps us to form links to the LEHD data.

Unfortunately, we are not able to directly form establishment-to-establishment linkages between LEHD data and the Business Register. The establishment identification system for the Business Register is the same as for all other Census Bureau business data products but different from the LEHD establishment identifier. Unfortunately, there is no one best way to form linkages between these data sources. There are many alternatives, and the optimal linking strategy depends on the research objective. To provide researchers with as much flexibility as possible, the BRB has been constructed as a crosswalk that allows for a number of different ways to integrate these data sources

1.2.2 File Structure and Contents

1.2.2.1 Identifiers

We use three types of identifying variables in the construction of the crosswalk file. These are: business identifier, geographic information, and industry code. Employer Identifier Number (EIN) is a nine-digit

taxpayer identification number assigned by the Internal Revenue Service. It is a unique identifier for single units but not for multi-units. Geographic information such as state or county is available. 1987 Standard Industrial Classification codes are available in 4-digit level

1.2.2.2 Unit of observation

The unit of observation on the BRB file is a unique EIN-State-SIC4-County record. All unique combinations of these identifiers that are found on either the LEHD data, the Business Register, or both will appear on the crosswalk.

We will use the EIN with 15 different combinations of geographic and industry information as follows:

- EIN
- EIN/SIC1
- EIN/SIC2
- EIN/SIC3
- EIN/SIC4
- EIN/STATE
- EIN/STATE/SIC1
- EIN/STATE/SIC2
- EIN/STATE/SIC3
- EIN/STATE/SIC4
- EIN/STATE/COUNTY
- EIN/STATE/COUNTY/SIC1
- EIN/STATE/COUNTY/SIC2
- EIN/STATE/COUNTY/SIC3
- EIN/STATE/COUNTY/SIC4

1.2.2.3 Structure of output files

For complete technical description, see Section 1.4.

Crosswalk file The crosswalk file contains the following variables

Alpha: Ten-digit enterprise identifier

County: Three-digit county identifier

EIN: Nine-digit employer identifier

Flag_XXX: 15 one-digit match type flags. These flags represent the status of the match corresponding to each level of aggregation, one flag per unique id variable combination. Each flag takes on the value “M”, “L”, or “B”. For example, `flag_e_c_2` is the flag variable when we use EIN/County/SIC2 as linking unit and `flag_e_s_4` is the flag when we use EIN/State/SIC4 as unit of matching.

M means that the business unit at this level of aggregation is matched.

L means that this business unit is observed only in the LEHD database.

B means that this business unit is observed only in the Business Register.

EIN/County/SIC4 is the most disaggregate level of aggregation we can use to match. So, it is obvious that if the flag value for the EIN State SIC4 County match is “M” then all 15 flags are “M”.

SIC1: one-digit 1987 SIC

SIC2: two-digit 1987 SIC

SIC3: three-digit 1987 SIC

SIC4: four-digit 1987 SIC

State: Two-digit state abbreviation (e.g. md)

Stgeo: Two-digit State FIPS code (e.g. 24)

Year: Calendar year

List files List files link different business identifiers that are only observed in one database to one of the fifteen level of aggregation used in matching

Business Register Identifiers: Census File Number (CFN), Permanent Plant Number (PPN), Alpha, etc. are captured on `brb_us_brlist.sas7bdat`

LEHD Identifiers: State Employer Identification Number (SEIN), SEINUNIT, etc. are captured on `brb_us_ecflist.sas7bdat`. *IMPORTANT:* The ECF list file is a quarterly file, not a yearly file!

1.2.3 How to Use The BRB

1.2.3.1 Some considerations

Because there are many ways to use the BRB, finding the optimal way to use it for any given research project requires some researcher planning. The steps outlined here summarize the experience of the LEHD researcher staff in using this crosswalk for a variety of different research projects.

Before making use of the BRB, there are a number of questions a researcher must address, and the answer to these questions will determine how the BRB will be used. To illustrate this point, we take a research topic and cover, step by step, the decisions a researcher would face when identifying the best way to make use of the BRB to build an analytic dataset for the project.

Because there is not a common establishment ID variable on the Census and LEHD business files, the link between the files must be formed at a level of aggregation that is, for many multi-unit businesses, higher than an establishment. Both the set of link variables as well as the unit of observation for the analytic dataset must be selected by the researcher. Note that these need not be the same.

1.2.3.2 Choosing the Link Variables

As noted, the BRB identifies all variables – state, county, and one-, two-, three-, and four-digit industry code – on which an EIN found among LEHD workforce traits and an EIN found on Census business data agree. In addition to EIN, any (or none) of these link variables may be chosen to merge together workforce traits and other business traits for analysis. When making this decision, observation counts in the resulting matched dataset as well as other factors should be considered. For example, if the analysis will focus on particular industries or regions, researchers may wish to require that linked business data agree on SIC or on state and county. In this case, any EIN the BRB indicates to be found on both files but not agreeing on state or industry would not be included in analysis. As we will discuss in the next section, it should be noted that the link variables do not necessarily determine to unit of observation for the analysis

1.2.3.3 Choosing the Base File and Unit of Observation

There are three types of ways the LEHD data may be combined with Census business data to create research-ready datasets. These types are:

1. LEHD workforce traits (such as worker churning rates) at the establishment level (or higher) may be linked to more highly aggregated Census business traits (such as labor productivity). In this situation, the LEHD establishments form the “base file.”
2. Establishment-level Census business traits (such as technology spending relative to a scaling measure) may be linked to more highly aggregated LEHD workforce traits. Here, Census data (such as the Annual Survey of Manufacturers) form the base file.
3. Traits aggregated on both sides may be combined using the crosswalk.

In short, researchers linking LEHD workforce traits to other Census business files may choose to aggregate LEHD traits, other business traits, or both. Note that by “aggregate,” we mean that establishment-level data should be aggregated to the level of the link (EIN-county, 4-digit SIC, for example), or higher. The researcher must decide on which side (workforce traits or other business characteristics) it is important to preserve more detail.

1.2.3.4 Example

Suppose we are interested in exploring how worker turnover impacts labor productivity for restaurants. There are two key decisions regarding identifier variables that must be made. Link traits as well as the unit of observation must be chosen. We hope to make it very clear that the two decisions need not be the same, and both will vary with the nature of the research question

A. Choosing the Link Level Because we are interested in restaurants only and because this is a two-digit industry trait, we will most likely want to use records from the BRB where `flag.e.2` (at a minimum)='M'. Noting that county-level match rates are high and that a franchised restaurant in downtown Chicago may behave very differently from the same franchise in rural Illinois, we may also require that observations entering our sample all agree on state and county as well and thus choose only to keep those BRB records where `flag.e.c.2`='M'

B. Choosing the Base File, Supplemental Files, and Unit of Observation From which data source – Census business data or LEHD data – would we like to preserve the most detailed data? Some researchers may in fact choose to sacrifice detail on both sides in favor of uniformity of variable construction. Regardless, because all variables that will characterize the unit of observation are available from other files, the BRB is not needed to obtain them. However, the unit of observation and the linkage unit are closely tied.

For this example, our goal is to characterize variation in labor productivity across restaurants and to determine if those with more worker turnover are less productive. Thus, we will most likely select Census business data as our “base” data source and will link establishment detail on productivity and other business traits to more aggregated workforce traits (in this example, the LEHD workforce aggregated would be the “supplemental” file). Note that this linking assumes that worker churning is similar for all restaurants in the same EIN state and county

1.2.3.5 Summary of How to Use This File:

Step 1 Make an extract from the BRB sub-setting on all records for each state and year that have a value of “M” for the match flag corresponding to these match variables. The variables to keep include all ID variables needed to make the match. For example, if we sub-set on `flag.e.c.2`='M', the match variables we keep from the BRB will be:

- EIN
- State
- SIC2
- County
- Year

CAUTION: One should make sure to sort the crosswalk to get unique observation per linking unit when using linking units less detailed than EIN/County/SIC4.

Step 2 Some Base and supplemental files have all the necessary variables such as EIN, State, SIC2, County, and Year. For example, the Annual Survey of Manufactures (ASM) and Census of Manufactures (CM) have these link variables as well as establishment identifiers such as CFN and PPN. In this case we can match the extract from the BRB with the base file or supplement file (after some aggregation). However, some files do have their own identifiers but not all link variables. In this case, we have to get additional variables from the list files we provide before matching the extract from the BRB with the files of interest. For example, some files have only CFN and year but not EIN. . Thus, we must first obtain the CFN from Business Register list file before matching.

Step 3 Select a base file (Census business data or LEHD business data), a supplemental file, and a unit of observation for the final analytic dataset. The base file should be the file from which we could like to preserve the most detail (if aggregation level differences within the same observation are desired).

Step 4 From the base file, pull off all observations (matching by the link variables selecting in Step 2) that match to the BRB extract. If allowing for more detailed data from the base file, the dataset resulting from this step will have more observations than the BRB extract.

Step 5 If the LEHD data has been selected as the base file in Step 3, some collection of Census business data files will be used to construct the supplemental file. The most dis-aggregated level of detail that may be maintained on the supplemental file is defined by the link variables (EIN, state, one-digit SIC, county and year in this example). If the business data variables of interest are reported on the files at this level of aggregation or smaller (typically this will be at an establishment level), then the business data must be aggregated to the level defined by the link variables. Note that some business files contain records reported at higher levels of aggregation. In these cases, no additional aggregation is needed in the construction of the supplemental file. **CAUTION:** When adding variables at a higher level of aggregation, it is often preferred to use ratios of variables from the same dataset rather than levels of one variable only. For example, rather than using aggregated sales, it is often preferable to weight sales at each establishment by some fraction (where the fractions for each establishment sum to one) before aggregating across establishments.

Step 6 Link the supplemental file to the joined base file and BRB extract. Use the same variables used to define the level of aggregation of the supplemental file.

Step 7 The construction of the analytic dataset is now complete. Note, however, that not all records in the dataset need be constructed at the same level of aggregation.

1.2.4 Some warnings and caveats

1.2.4.1 Active establishments

In the construction of the crosswalk, duplicates and non-active establishments are deleted (not included) based on Census activity flags from the BR. Invalid, missing, illegible, or out-of-scope industry code, invalid geography, or even zero payroll are *not* grounds for exclusion from the BRB. In particular, invalid codes can and do appear on the Business Register, and are carried through unchanged.

1.2.4.2 Coverage

The states covered in the BRB will vary across years depending on ECF availability in each year.

1.2.4.3 Discrepancies in geo and industry codes

There may be discrepancies between the BR and ES-202 based data for a variety of reasons. Processing issues imply that geo and industry codes may be missing or out of bounds on BR, especially for entrants. It is an open question whether the ES-202 might get information on industry and geo faster than the BR. Note that there is some evidence that suggests that this is the case: Census currently receives a list of industry codes from BLS (from its ES202 processing) for new EINs. Census has typically found these industry codes to be more reliable than either the PBA (principal business activity from income tax forms for businesses) codes they receive from IRS or the industry codes from SSA extracted from the SS-4 form (the form used to apply for an EIN).

In terms of preference, the preference ordering that Census typically uses for industry codes is:

1. Direct Census collection in economic censuses or annual surveys (for most businesses this means only once every five years).
2. BLS codes
3. SSA codes
4. IRS codes

1.3 Data set description

1.3.1 Crosswalk

Field name	Data dictionary reference name	Starting position	Field size	Data type
Cleaned EIN	EIN	00026	9	A/N
Cleaned ES202 FIPS County CCC	COUNTY	00035	3	A/N
Cleaned SIC Code IIII	SIC4	00020	4	A/N
ES202 FIPS State SS	STGEO	00024	2	A/N
Enterprise identifier	ALPHA	00008	10	A/N
Match flag EIN level	FLAG_E	00058	1	A/N
Match flag EIN/SIC1 level	FLAG_E_1	00057	1	A/N
Match flag EIN/SIC2 level	FLAG_E_2	00056	1	A/N
Match flag EIN/SIC3 level	FLAG_E_3	00055	1	A/N
Match flag EIN/SIC4 level	FLAG_E_4	00054	1	A/N
Match flag EIN/STATE level	FLAG_E_S	00053	1	A/N
Match flag EIN/STATE/COUNTY level	FLAG_E_S_C	00048	1	A/N
Match flag EIN/STATE/COUNTY/SIC1	FLAG_E_S_C_1	00047	1	A/N
Match flag EIN/STATE/COUNTY/SIC2	FLAG_E_S_C_2	00046	1	A/N

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match flag EIN/STATE/COUNTY/SIC3	FLAG_E_S_C_3	00045	1	A/N
Match flag EIN/STATE/COUNTY/SIC4	FLAG_E_S_C_4	00044	1	A/N
Match flag EIN/STATE/SIC1 level	FLAG_E_S_1	00052	1	A/N
Match flag EIN/STATE/SIC2 level	FLAG_E_S_2	00051	1	A/N
Match flag EIN/STATE/SIC3 level	FLAG_E_S_3	00050	1	A/N
Match flag EIN/STATE/SIC4 level	FLAG_E_S_4	00049	1	A/N
One-digit SIC code	SIC1	00043	1	A/N
State FIPS code	STATE	00018	2	A/N
Three-digit SIC Code	SIC3	00038	3	A/N
Two-digit SIC code	SIC2	00041	2	A/N
Year YYYY	YEAR	00000	8	N

1.3.2 BR list

Field name	Data dictionary reference name	Starting position	Field size	Data type
Census File Number	CFN	00008	10	A/N
Employer Identification Number	EIN	00036	9	A/N
Enterprise identifier	ALPHA	00018	10	A/N
FIPS State SS	STATE	00028	2	A/N
FIPS State xx	STGEO	00034	2	A/N
FIPS county xxx	COUNTY	00055	3	A/N
Four-digit 1987 SIC	SIC4	00030	4	A/N
Permanent Plant Number	PPN	00045	10	A/N
Year YYYY	YEAR	00000	8	N

1.3.3 ECF list

Field name	Data dictionary reference name	Starting position	Field size	Data type
Cleaned EIN	EIN	00026	9	A/N
Cleaned ES202 FIPS County CCC	COUNTY	00023	3	A/N
Cleaned SIC Code IIII	SIC4	00019	4	A/N
ES202 FIPS State SS	STGEO	00017	2	A/N
FIPS State SS	STATE	00035	2	A/N
Quarter QQ	QUARTER	00040	3	N
State Employer ID Number	SEIN	00000	12	A/N
State UI Reporting Unit Number	SEINUNIT	00012	5	A/N
Year YYYY	YEAR	00037	3	N

1.4 Data dictionary

1.4.1 Crosswalk

Field name	Data dictionary reference name	Starting position	Field size	Data type
Cleaned EIN	EIN	00026	9	A/N

CHARACTERISTICS

UNITS	Identifier
ALGORITHM	read-in
SOURCEFILE	BR and ECF, if available
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

Field name	Data dictionary reference name	Starting position	Field size	Data type
Cleaned ES202 FIPS County CCC	COUNTY	00035	3	A/N

CHARACTERISTICS

UNITS	Geography
ALGORITHM	read-in
SOURCEFILE	ECF/BR
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

Field name	Data dictionary reference name	Starting position	Field size	Data type
Cleaned SIC Code IIII	SIC4	00020	4	A/N

CHARACTERISTICS

UNITS	Industry
ALGORITHM	read-in
SOURCEFILE	ECF/BR
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

Field name	Data dictionary reference name	Starting position	Field size	Data type
ES202 FIPS State SS	STGEO	00024	2	A/N

CHARACTERISTICS

UNITS	Geography
ALGORITHM	read-in
SOURCEFILE	BR
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

Field name	Data dictionary reference name	Starting position	Field size	Data type
Enterprise identifier	ALPHA	00008	10	A/N

CHARACTERISTICS

	UNITS	Identifier
	ALGORITHM	read-in
	SOURCEFILE	BR
	ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match flag EIN level	FLAG_E	00058	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match flag EIN/SIC1 level	FLAG_E_1	00057	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/SIC1 level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match flag EIN/SIC2 level	FLAG_E_2	00056	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/SIC2 level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match flag EIN/SIC3 level	FLAG_E_3	00055	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/SIC3 level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match flag EIN/SIC4 level	FLAG_E_4	00054	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/SIC4 level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match flag EIN/STATE level	FLAG_E_S	00053	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/STATE level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match flag EIN/STATE/COUNTY level	FLAG_E_S_C	00048	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/STATE/COUNTY level

Field name		Data dictionary reference name	Starting position	Field size	Data type
Match EIN/STATE/COUNTY/SIC1	flag	FLAG_E_S_C_1	00047	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/STATE/COUNTY/SIC1 level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match EIN/STATE/COUNTY/SIC2	flag FLAG_E_S_C_2	00046	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/STATE/COUNTY/SIC2 level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match EIN/STATE/COUNTY/SIC3	flag FLAG_E_S_C_3	00045	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/STATE/COUNTY/SIC3 level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match EIN/STATE/COUNTY/SIC4	flag FLAG_E_S_C_4	00044	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/STATE/COUNTY/SIC4 level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match flag EIN/STATE/SIC1 level	FLAG_E_S_1	00052	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/STATE/SIC1 level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match flag EIN/STATE/SIC2 level	FLAG_E_S_2	00051	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/STATE/SIC2 level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match flag EIN/STATE/SIC3 level	FLAG_E_S_3	00050	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/STATE/SIC3 level

Field name	Data dictionary reference name	Starting position	Field size	Data type
Match flag EIN/STATE/SIC4 level	FLAG_E_S_4	00049	1	A/N

CHARACTERISTICS

UNITS	Flag
ALGORITHM	computed
SOURCEFILE	see 04_brb.sas
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

B: only in BR
L: only in LEHD
M: matched in EIN/STATE/SIC4 level

Field name	Data dictionary reference name	Starting position	Field size	Data type
One-digit SIC code	SIC1	00043	1	A/N

CHARACTERISTICS

UNITS	Industry
ALGORITHM	derived
SOURCEFILE	sic1=substr(sic4,1,1)
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

Field name	Data dictionary reference name	Starting position	Field size	Data type
State FIPS code	STATE	00018	2	A/N

CHARACTERISTICS

UNITS	Geography
ALGORITHM	read-in
SOURCEFILE	ECF
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

Field name	Data dictionary reference name	Starting position	Field size	Data type
Three-digit SIC Code	SIC3	00038	3	A/N

CHARACTERISTICS

UNITS	Industry
ALGORITHM	derived
SOURCEFILE	sic3=substr(sic4,1,3)
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

Field name	Data dictionary reference name	Starting position	Field size	Data type
Two-digit SIC code	SIC2	00041	2	A/N

CHARACTERISTICS

UNITS	Industry
ALGORITHM	derived
SOURCEFILE	sic2=substr(sic4,1,2)
ALTERNATE DOCUMENTATION	n.a.

VALUE TABLE

Field name	Data dictionary reference name	Starting position	Field size	Data type
Year YYYY	YEAR	00000	8	N

CHARACTERISTICS

UNITS	Calendar
ALGORITHM	read-in
SOURCEFILE	ECF and BR
ALTERNATE DOCUMENTATION	n.a.

1.4.2 BR list

Field name	Data dictionary reference name	Starting position	Field size	Data type
Census File Number	CFN	00008	10	A/N

CHARACTERISTICS

UNITS	Identifier
ALGORITHM	read-in
SOURCEFILE	BR
ALTERNATE DOCUMENTATION	n.a.

Field name	Data dictionary reference name	Starting position	Field size	Data type
Employer Identification Number	EIN	00036	9	A/N

CHARACTERISTICS

UNITS	Identifier
ALGORITHM	read-in
SOURCEFILE	BR
ALTERNATE DOCUMENTATION	n.a.

Field name	Data dictionary reference name	Starting position	Field size	Data type
Enterprise identifier	ALPHA	00018	10	A/N

CHARACTERISTICS

UNITS	Identifier
ALGORITHM	read-in
SOURCEFILE	BR
ALTERNATE DOCUMENTATION	n.a.

Field name	Data dictionary reference name	Starting position	Field size	Data type
FIPS State SS	STATE	00028	2	A/N

CHARACTERISTICS

UNITS	Geography
ALGORITHM	state=lowercase(fipstate(stgeo))
SOURCEFILE	derived
ALTERNATE DOCUMENTATION	n.a.

Field name	Data dictionary reference name	Starting position	Field size	Data type
FIPS State xx	STGEO	00034	2	A/N

CHARACTERISTICS

UNITS	Geography
ALGORITHM	read-in
SOURCEFILE	BR
ALTERNATE DOCUMENTATION	n.a.

Field name	Data dictionary reference name	Starting position	Field size	Data type
FIPS county xxx	COUNTY	00055	3	A/N

CHARACTERISTICS

UNITS	Geography
ALGORITHM	read-in
SOURCEFILE	BR
ALTERNATE DOCUMENTATION	n.a.

Field name	Data dictionary reference name	Starting position	Field size	Data type
Four-digit 1987 SIC	SIC4	00030	4	A/N

CHARACTERISTICS

UNITS	Industry
ALGORITHM	read-in
SOURCEFILE	BR
ALTERNATE DOCUMENTATION	n.a.

Field name	Data dictionary reference name	Starting position	Field size	Data type
Permanent Plant Number	PPN	00045	10	A/N

CHARACTERISTICS

UNITS	Identifier
ALGORITHM	read-in
SOURCEFILE	BR
ALTERNATE DOCUMENTATION	n.a.

Field name	Data dictionary reference name	Starting position	Field size	Data type
Year YYYY	YEAR	00000	8	N

CHARACTERISTICS

UNITS	Calendar
ALGORITHM	read-in
SOURCEFILE	BR
ALTERNATE DOCUMENTATION	n.a.

1.4.3 ECF list

Field name	Data dictionary reference name	Starting position	Field size	Data type
Cleaned EIN	EIN	00026	9	A/N

CHARACTERISTICS

	UNITS	Identifier
	ALGORITHM	read-in
	SOURCEFILE	ECF (availability differs across states)
	ALTERNATE DOCUMENTATION	ECF documentation

Field name	Data dictionary reference name	Starting position	Field size	Data type
Cleaned ES202 FIPS County CCC	COUNTY	00023	3	A/N

CHARACTERISTICS

UNITS	Geography
ALGORITHM	read-in
SOURCEFILE	ECF
ALTERNATE DOCUMENTATION	ECF documentation

Field name	Data dictionary reference name	Starting position	Field size	Data type
Cleaned SIC Code IIII	SIC4	00019	4	A/N

CHARACTERISTICS

UNITS	Industry
ALGORITHM	read-in
SOURCEFILE	ECF
ALTERNATE DOCUMENTATION	ECF documentation

Field name	Data dictionary reference name	Starting position	Field size	Data type
ES202 FIPS State SS	STGEO	00017	2	A/N

CHARACTERISTICS

UNITS	Geography
ALGORITHM	rename es.state=stgeo
SOURCEFILE	ECF
ALTERNATE DOCUMENTATION	ECF documentation

Field name	Data dictionary reference name	Starting position	Field size	Data type
FIPS State SS	STATE	00035	2	A/N

CHARACTERISTICS

UNITS	Geography
ALGORITHM	state=lowercase(fipstate(stgeo))
SOURCEFILE	derived
ALTERNATE DOCUMENTATION	n.a.

Field name	Data dictionary reference name	Starting position	Field size	Data type
Quarter QQ	QUARTER	00040	3	N

CHARACTERISTICS

UNITS	Calendar
ALGORITHM	read-in
SOURCEFILE	ECF
ALTERNATE DOCUMENTATION	n.a.

Field name	Data dictionary reference name	Starting position	Field size	Data type
State Employer ID Number	SEIN	00000	12	A/N

CHARACTERISTICS

UNITS	(State-specific) Identifier
ALGORITHM	read-in
SOURCEFILE	ECF
ALTERNATE DOCUMENTATION	ECF documentation

Field name	Data dictionary reference name	Starting position	Field size	Data type
State UI Reporting Unit Number	SEINUNIT	00012	5	A/N

CHARACTERISTICS

UNITS	(State-specific) Identifier
ALGORITHM	read-in
SOURCEFILE	ECF
ALTERNATE DOCUMENTATION	ECF documentation

Field name	Data dictionary reference name	Starting position	Field size	Data type
Year YYYY	YEAR	00037	3	N

CHARACTERISTICS

UNITS	Calendar
ALGORITHM	read-in
SOURCEFILE	ECF
ALTERNATE DOCUMENTATION	ECF documentation

1.5 Details

1.5.1 SAS programs used

Modules can be either SAS programs that are '%include' in the main control program, or SAS macros that are made available by a '%include' or by a system-wide macro library, and executed by the control program. The preferred method are macros, but legacy programs use SAS programs as well.

1.5.1.1 library/sasprogs/01_brb.sas

```
/* +--< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >--+ */
/* +--< Location: /programs/production/dev7/current/brb >--+ */
/* +--< File: library/sasprogs/01_brb.sas >--+ */
/* Time-stamp: <05/01/14 13:21:40 vilhuber> */
```

First module: concatenate relevant SSEL files.

1. CFN, EIN, County, SIC4 list from SSEL.

```
%read_ssel(level=su);
%read_ssel(level=mu);

proc sort data=ssel_su noequals;
by cfn year;
run;

proc sort data=ssel_mu noequals;
by cfn year;
run;

data ssel (keep=cfn ppn alpha ein year stgeo state county sic4);
merge ssel_su(in=a rename=(stgeo=stgeo_su ctygeo=county_su ppn=ppn_su
ein=ein_su sic4=sic4_su))
ssel_mu(in=b rename=(stgeo=stgeo_mu ctygeo=county_mu ppn=ppn_mu
ein=ein_mu sic4=sic4_mu));
by cfn year;
length alpha $10 state $2 sic4 $4;
if a and not b then do;
state=lowcase(fipstate(stgeo_su));
stgeo=stgeo_su;
ein =ein_su;
ppn =ppn_su;
county=county_su;
sic4=substr(sic4_su,1,4);
alpha=cfn;
end;
else do;
state=lowcase(fipstate(stgeo_mu));
stgeo=stgeo_mu;
ein =ein_mu;
ppn =ppn_mu;
county=county_mu;
sic4=substr(sic4_mu,1,4);
alpha=substr(cfn,1,6)||'0000';
end;
label alpha='Enterprise identifier';
run;
```

2. Unique EIN/State/County/SIC4/year list from SSEL.


```

/* proc sort data=ssel nodupkey out=INTERWRK.ssellist; */
proc sort data=ssel nodupkey out=OUTPUTS.brb_&state._brlist ;
    by year ein state county sic4 cfn;
run;

```

1.5.1.2 library/sasprogs/02_brb.sas

```

/* ---< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >---+ */
/* ---< Location: /programs/production/dev7/current/brb >---+ */
/* ---< File: library/sasprogs/02_brb.sas >---+ */
/* Time-stamp: <05/01/14 13:22:08 vilhuber> */

```

Second module: Unique EIN/State/County/SIC4/year list from SSEL.

```

data INTERWRK.ssel_01(drop=cfn ppn);
/*    set INTERWRK.ssellist; */
    set OUTPUTS.brb_&state._brlist;
    by year ein state county sic4;
    length sic3 $3 sic2 $2 sic1 $1;
    sic3=substr(sic4,1,3);
    sic2=substr(sic4,1,2);
    sic1=substr(sic4,1,1);
    if last.sic4 then output;
run;

```

1.5.1.3 library/sasprogs/03_brb.sas

```

/* ---< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >---+ */
/* ---< Location: /programs/production/dev7/current/brb >---+ */
/* ---< File: library/sasprogs/03_brb.sas >---+ */
/* Time-stamp: <04/10/26 10:35:35 vilhuber> */

```

Second module: Reading from ECF .

```

** 3. Reading from ECF **;

```

Reading ECF and including only those born in 1991 or after.

```

%read_ecf;

/* proc sort data=ecf nodupkey out=INTERWRK.ecflist; */
proc sort data=ecf nodupkey out=OUTPUTS.brb_&state._ecflist;
    by year ein state county sic4 sein seinunit quarter;
run;

data INTERWRK.ecf_02(drop=sein seinunit quarter);
/* set INTERWRK.ecflist; */
    set OUTPUTS.brb_&state._ecflist;
    by year ein state county sic4;
    length sic3 $3 sic2 $2 sic1 $1;
    sic3=substr(sic4,1,3);
    sic2=substr(sic4,1,2);
    sic1=substr(sic4,1,1);
    if last.sic4 then output;
run;

```

1.5.1.4 library/sasprogs/04_brb.sas

```

/* ---< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >---+ */
/* ---< Location: /programs/production/dev7/current/brb >---+ */
/* ---< File: library/sasprogs/04_brb.sas >---+ */
/* Time-stamp: <05/01/14 13:15:24 vilhuber> */

```

Merging files at different levels.

```
** 4. Merging;
```

4.1 By County level

4.1.1 By SIC4

```

data INTERWRK.xwalk_county_sic4;
    merge INTERWRK.ssel_01(in=aa) INTERWRK.ecf_02(in=bb);
    by year ein state county sic4;
    if aa and bb then flag_e_s_c_4="M";
    else if aa and not bb then flag_e_s_c_4="B";
    else if not aa and bb then flag_e_s_c_4="L";
    label flag_e_s_c_4="M if matched in EIN/STATE/COUNTY/SIC4 level, =B if
only in BR, =L if only in LEHD";
    label state="State FIPS code";
    label sic3="Three-digit SIC Code";
    label sic2="Two-digit SIC code";
    label sic1="One-digit SIC code";
run;

```

4.1.2 By SIC3

```

proc sort data=INTERWRK.ssel_01 nodupkey out=sssel_county_sic3
  (keep=year ein state county sic3);
  by year ein state county sic3;
run;
proc sort data=INTERWRK.ecf_02 nodupkey out=ecf_county_sic3
  (keep=year ein state county sic3);
  by year ein state county sic3;
run;

data INTERWRK.xwalk_county_sic3;
  merge ssel_county_sic3(in=aa) ecf_county_sic3(in=bb);
  by year ein state county sic3;
  if aa and bb then flag_e_s_c_3="M";
  else if aa and not bb then flag_e_s_c_3="B";
  else if not aa and bb then flag_e_s_c_3="L";
  label flag_e_s_c_3="M if matched in EIN/STATE/COUNTY/SIC3 level, =B if
only in BR, =L if only in LEHD";
run;

```

4.1.3 By SIC2

```

proc sort data=INTERWRK.ssel_01 nodupkey out=sssel_county_sic2
  (keep=year ein state county sic2);
  by year ein state county sic2;
run;
proc sort data=INTERWRK.ecf_02 nodupkey out=ecf_county_sic2
  (keep=year ein state county sic2);
  by year ein state county sic2;
run;

data INTERWRK.xwalk_county_sic2;
  merge ssel_county_sic2(in=aa) ecf_county_sic2(in=bb);
  by year ein state county sic2;
  if aa and bb then flag_e_s_c_2="M";
  else if aa and not bb then flag_e_s_c_2="B";
  else if not aa and bb then flag_e_s_c_2="L";
  label flag_e_s_c_2="M if matched in EIN/STATE/COUNTY/SIC2 level, =B if
only in BR, =L if only in LEHD";
run;

```

4.1.4 By SIC1

```

proc sort data=INTERWRK.ssel_01 nodupkey out=sssel_county_sic1
  (keep=year ein state county sic1);
  by year ein state county sic1;
run;
proc sort data=INTERWRK.ecf_02 nodupkey out=ecf_county_sic1
  (keep=year ein state county sic1);
  by year ein state county sic1;
run;

data INTERWRK.xwalk_county_sic1;
  merge ssel_county_sic1(in=aa) ecf_county_sic1(in=bb);
  by year ein state county sic1;
  if aa and bb then flag_e_s_c_1="M";
  else if aa and not bb then flag_e_s_c_1="B";
  else if not aa and bb then flag_e_s_c_1="L";
  label flag_e_s_c_1="M if matched in EIN/STATE/COUNTY/SIC1 level, =B if
only in BR, =L if only in LEHD";
run;

```

4.1.5 No industry code

```

proc sort data=INTERWRK.ssel_01 nodupkey out=ssel_county
  (keep=year ein state county);
  by year ein state county;
run;
proc sort data=INTERWRK.ecf_02 nodupkey out=ecf_county
  (keep=year ein state county);
  by year ein state county;
run;

data INTERWRK.xwalk_county;
  merge ssel_county(in=aa) ecf_county(in=bb);
  by year ein state county;
  if aa and bb then flag_e_s_c="M";
  else if aa and not bb then flag_e_s_c="B";
  else if not aa and bb then flag_e_s_c="L";
  label flag_e_s_c="M if matched in EIN/STATE/COUNTY level, =B if only in
BR, =L if only in LEHD";
run;

```

4.2 By state level

4.2.1 By SIC4

```

proc sort data=INTERWRK.ssel_01 nodupkey out=ssel_state_sic4
  (keep=year ein state sic4 sic3 sic2 sic1);
  by year ein state sic4;
run;
proc sort data=INTERWRK.ecf_02 nodupkey out=ecf_state_sic4
  (keep=year ein state sic4 sic3 sic2 sic1);
  by year ein state sic4;
run;

data INTERWRK.xwalk_state_sic4;
  merge ssel_state_sic4(in=aa) ecf_state_sic4(in=bb);
  by year ein state sic4;
  if aa and bb then flag_e_s_4="M";
  else if aa and not bb then flag_e_s_4="B";
  else if not aa and bb then flag_e_s_4="L";
  label flag_e_s_4="M if matched in EIN/STATE/SIC4 level, =B if only in BR,
=L if only in LEHD";
  * drop sic3 sic2 sic1;
run;

```

4.2.2 By SIC3

```

proc sort data=ssel_state_sic4 nodupkey out=ssel_state_sic3
  (keep=year ein state sic3 sic2 sic1);
  by year ein state sic3;
run;
proc sort data=ecf_state_sic4 nodupkey out=ecf_state_sic3
  (keep=year ein state sic3 sic2 sic1);
  by year ein state sic3;
run;

data INTERWRK.xwalk_state_sic3;
  merge ssel_state_sic3(in=aa) ecf_state_sic3(in=bb);
  by year ein state sic3;
  if aa and bb then flag_e_s_3="M";
  else if aa and not bb then flag_e_s_3="B";
  else if not aa and bb then flag_e_s_3="L";
  label flag_e_s_3="M if matched in EIN/STATE/SIC3 level, =B if only in BR,
=L if only in LEHD";
  * drop sic2 sic1;
run;

```

4.2.3 By SIC2

```

proc sort data=ssel_state_sic3 nodupkey out=ssel_state_sic2
  (keep=year ein state sic2 sic1);
  by year ein state sic2;
run;
proc sort data=ecf_state_sic3 nodupkey out=ecf_state_sic2
  (keep=year ein state sic2 sic1);
  by year ein state sic2;
run;

data INTERWRK.xwalk_state_sic2;
  merge ssel_state_sic2(in=aa) ecf_state_sic2(in=bb);
  by year ein state sic2;
  if aa and bb then flag_e_s_2="M";
  else if aa and not bb then flag_e_s_2="B";
  else if not aa and bb then flag_e_s_2="L";
  label flag_e_s_2="M if matched in EIN/STATE/SIC2 level, =B if only in BR,
=L if only in LEHD";
  * drop sic1;
run;

```

4.2.4 By SIC1

```

proc sort data=ssel_state_sic2 nodupkey out=ssel_state_sic1
  (keep=year ein state sic1);
  by year ein state sic1;
run;
proc sort data=ecf_state_sic2 nodupkey out=ecf_state_sic1
  (keep=year ein state sic1);
  by year ein state sic1;
run;

data INTERWRK.xwalk_state_sic1;
  merge ssel_state_sic1(in=aa) ecf_state_sic1(in=bb);
  by year ein state sic1;
  if aa and bb then flag_e_s_1="M";
  else if aa and not bb then flag_e_s_1="B";
  else if not aa and bb then flag_e_s_1="L";
  label flag_e_s_1="M if matched in EIN/STATE/SIC1 level, =B if only in BR,
=L if only in LEHD";
run;

```

4.2.5 No industry code

```

proc sort data=ssel_state_sic1 nodupkey out=ssel_state
  (keep=year ein state);
  by year ein state;
run;
proc sort data=ecf_state_sic1 nodupkey out=ecf_state
  (keep=year ein state);
  by year ein state;
run;

data INTERWRK.xwalk_state;
  merge ssel_state(in=aa) ecf_state(in=bb);
  by year ein state ;
  if aa and bb then flag_e_s="M";
  else if aa and not bb then flag_e_s="B";
  else if not aa and bb then flag_e_s="L";
  label flag_e_s="M if matched in EIN/STATE level, =B if only in BR, =L if
only in LEHD";
run;

```

4.3 By national level

4.3.1 By SIC4

```

proc sort data=INTERWRK.ssel_01 nodupkey out=ssel_sic4
  (keep=year ein sic4 sic3 sic2 sic1);
  by year ein sic4;
run;
proc sort data=INTERWRK.ecf_02 nodupkey out=ecf_sic4
  (keep=year ein sic4 sic3 sic2 sic1);
  by year ein sic4;
run;

data INTERWRK.xwalk_sic4;
  merge ssel_sic4(in=aa) ecf_sic4(in=bb);
  by year ein sic4;
  if aa and bb then flag_e_4="M";
  else if aa and not bb then flag_e_4="B";
  else if not aa and bb then flag_e_4="L";
  label flag_e_4="M if matched in EIN/SIC4 level, =B if only in BR, =L if
only in LEHD";
  * drop sic3 sic2 sic1;
run;

```

4.3.2 By SIC3

```

proc sort data=ssel_sic4 nodupkey out=ssel_sic3
  (keep=year ein sic3 sic2 sic1);
  by year ein sic3;
run;
proc sort data=ecf_sic4 nodupkey out=ecf_sic3
  (keep=year ein sic3 sic2 sic1);
  by year ein sic3;
run;

data INTERWRK.xwalk_sic3;
  merge ssel_sic3(in=aa) ecf_sic3(in=bb);
  by year ein sic3;
  if aa and bb then flag_e_3="M";
  else if aa and not bb then flag_e_3="B";
  else if not aa and bb then flag_e_3="L";
  label flag_e_3="M if matched in EIN/SIC3 level, =B if only in BR, =L if
only in LEHD";
  * drop sic2 sic1;
run;

```

4.3.3 By SIC2

```

proc sort data=ssel_sic3 nodupkey out=ssel_sic2
  (keep=year ein sic2 sic1);
  by year ein sic2;
run;
proc sort data=ecf_sic3 nodupkey out=ecf_sic2
  (keep=year ein sic2 sic1);
  by year ein sic2;
run;

data INTERWRK.xwalk_sic2;
  merge ssel_sic2(in=aa) ecf_sic2(in=bb);
  by year ein sic2;
  if aa and bb then flag_e_2="M";
  else if aa and not bb then flag_e_2="B";
  else if not aa and bb then flag_e_2="L";
  label flag_e_2="M if matched in EIN/SIC2 level, =B if only in BR, =L if
only in LEHD";
  * drop sic1;
run;

```

4.3.4 By SIC1

```

proc sort data=ssel_sic2 nodupkey out=ssel_sic1
  (keep=year ein sic1);
  by year ein sic1;
run;
proc sort data=ecf_sic2 nodupkey out=ecf_sic1
  (keep=year ein sic1);
  by year ein sic1;
run;

data INTERWRK.xwalk_sic1;
  merge ssel_sic1(in=aa) ecf_sic1(in=bb);
  by year ein sic1;
  if aa and bb then flag_e_1="M";
  else if aa and not bb then flag_e_1="B";
  else if not aa and bb then flag_e_1="L";
  label flag_e_1="M if matched in EIN/SIC1 level, =B if only in BR, =L if
only in LEHD";
run;

```

4.3.5 No industry code

```

proc sort data=ssel_sic1 nodupkey out=ssel_
  (keep=year ein );
  by year ein ;
run;
proc sort data=ecf_sic1 nodupkey out=ecf_
  (keep=year ein );
  by year ein ;
run;

data INTERWRK.xwalk;
  merge ssel_(in=aa) ecf_(in=bb);
  by year ein ;
  if aa and bb then flag_e="M";
  else if aa and not bb then flag_e="B";
  else if not aa and bb then flag_e="L";
  label flag_e="M if matched in EIN level, =B if only in BR, =L if only in
LEHD";
run;

```

1.5.1.5 library/sasprogs/05_brb.sas

```

/* +--< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >--+ */
/* +--< Location: /programs/production/dev7/current/brb >--+ */
/* +--< File: library/sasprogs/05_brb.sas >--+ */
/* Time-stamp: <05/01/15 16:03:28 vilhuber> */

```

5. Combining all. This used to be one giant SQL step. Kind of slow. This was changed in January 2005. Note that this could be accelerated by parallelizing each level, and using SAS 9.1 to use TCP libraries.

5.1. County level

```

data county_4_3;
  merge INTERWRK.xwalk_county_sic4 INTERWRK.xwalk_county_sic3;
  by year ein state county sic3;
run;

data county_4_3_2;
  merge county_4_3 INTERWRK.xwalk_county_sic2;
  by year ein state county sic2;
run;

data county_4_3_2_1;
  merge county_4_3_2 INTERWRK.xwalk_county_sic1;
  by year ein state county sic1;
run;

data INTERWRK.county_all;
  merge county_4_3_2_1 INTERWRK.xwalk_county;
  by year ein state county;
run;

```

5.2. State level


```

data state_4_3;
  merge INTERWRK.xwalk_state_sic4 INTERWRK.xwalk_state_sic3;
  by year ein state sic3;
run;

data state_4_3_2;
  merge state_4_3 INTERWRK.xwalk_state_sic2;
  by year ein state sic2;
run;

data state_4_3_2_1;
  merge state_4_3_2 INTERWRK.xwalk_state_sic1;
  by year ein state sic1;
run;

data INTERWRK.state_all;
  merge state_4_3_2_1 INTERWRK.xwalk_state;
  by year ein state ;
run;

```

5.3. National level

```

data national_4_3;
  merge INTERWRK.xwalk_sic4 INTERWRK.xwalk_sic3;
  by year ein sic3;
run;

data national_4_3_2;
  merge national_4_3 INTERWRK.xwalk_sic2;
  by year ein sic2;
run;

data national_4_3_2_1;
  merge national_4_3_2 INTERWRK.xwalk_sic1;
  by year ein sic1;
run;

data INTERWRK.national_all;
  merge national_4_3_2_1 INTERWRK.xwalk;
  by year ein ;
run;

```

5.4. Combining County, State, and National files

```

/* block 1 */
proc sort data=INTERWRK.county_all;
  by year ein state sic4;
run;

proc sort data=INTERWRK.state_all;
  by year ein state sic4;
run;

data county_state;
  merge INTERWRK.county_all INTERWRK.state_all;
  by year ein state sic4;
run;

/* block 2 */
proc sort data=county_state;
  by year ein sic4;
run;

proc sort data=INTERWRK.national_all;
  by year ein sic4;
run;

data INTERWRK.xwalk_&state.;
  merge county_state INTERWRK.national_all;
  by year ein sic4;
run;

```

1.5.1.6 library/sasprogs/06_brb.sas

```

/* ---< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >---+ */
/* ---< Location: /programs/production/dev7/current/brb >---+ */
/* ---< File: library/sasprogs/06_brb.sas >---+ */
/* Time-stamp: <05/01/15 16:03:12 vilhuber> */

```

6. Final sort.

```

proc sort data=INTERWRK.xwalk_&state. out=OUTPUTS.brb_&state._xwalk;
  by year ein state county sic4;
run;

```

Final labelling.

```

proc datasets library=OUTPUTS;
  modify brb_&state._xwalk;
  label
    flag_e      = "Match flag  EIN level"
    flag_e_1    = "Match flag  EIN/SIC1 level"
    flag_e_2    = "Match flag  EIN/SIC2 level"
    flag_e_3    = "Match flag  EIN/SIC3 level"
    flag_e_4    = "Match flag  EIN/SIC4 level"
    flag_e_s    = "Match flag  EIN/STATE level"
    flag_e_s_1  = "Match flag  EIN/STATE/SIC1 level"
    flag_e_s_2  = "Match flag  EIN/STATE/SIC2 level"
    flag_e_s_3  = "Match flag  EIN/STATE/SIC3 level"
    flag_e_s_4  = "Match flag  EIN/STATE/SIC4 level"
    flag_e_s_c  = "Match flag  EIN/STATE/COUNTY level"
    flag_e_s_c_1 = "Match flag  EIN/STATE/COUNTY/SIC1"
    flag_e_s_c_2 = "Match flag  EIN/STATE/COUNTY/SIC2"
    flag_e_s_c_3 = "Match flag  EIN/STATE/COUNTY/SIC3"
    flag_e_s_c_4 = "Match flag  EIN/STATE/COUNTY/SIC4"
  ;
quit;

```

1.5.1.7 library/sasprogs/07_brb.sas

```

/* +---< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >---+ */
/* +---< Location: /programs/production/dev7/current/brb >---+ */
/* +---< File: library/sasprogs/07_brb.sas >---+ */
/* Time-stamp: <05/03/23 13:34:33 vilhuber> */

```

Capture the sort information.

```

%capture_sort(OUTPUTS.brb_&state._xwalk);
data brb_sort;
  set _sort;
run;

%capture_sort(OUTPUTS.brb_&state._brlist);
proc append base=brb_sort data=_sort;
run;

%capture_sort(OUTPUTS.brb_&state._ecflist);
proc append base=brb_sort data=_sort;
run;

```

Create a dataset with the names on the CES system.

```

data newlocalnames;
  set brb_sort(keep=memname)
;
  length localname newname $ 50;
  label localname="Name in LEHD export process"
  newname="Name on RDC system"
  memname="Original name"
;
  if memname="brb_&state._xwalk" then newname="lehd_&state._brb";
  if memname="brb_&state._ecflist" then newname="lehd_&state._brb_ecflist";
  if memname="brb_&state._brlist" then newname="lehd_&state._brb_brlist";
run;

proc sort data=newlocalnames nodupkey;
by memname;
run;

```

Output file to reconstruct indexes and sort order at remote end. First, sort the list of indexes by dataset and index name. Then output a SAS program file to set sortedby options.

```

proc sort data=brb_sort;
  by memname;
run;

data INTERWRK.brb_sorts;
  merge brb_sort(in=a)
  newlocalnames;
  by memname;
  if a;
run;

%recreate_indexes(basename=brb,index=no);

```

1.5.1.8 library/sasprogs/08_brb.sas

```

/* ---< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >--- */
/* ---< Location: /programs/production/dev7/current/brb >--- */
/* ---< File: library/sasprogs/08_brb.sas >--- */
/* Time-stamp: <05/03/23 16:00:56 vilhuber> */

```

This program preps the file for transfer to the RDC network

```

%let outputs=%trim(%sysfunc(pathname(OUTPUTS)));

```

Date. We will need to construct the memo no.

```

data _null_;
  tday=put(today(),yymmdd10.);
  call symput("today",tday);
run;

%put today=&today.;

```

The memorandum number is composed of date + type + state.

Create data for appendix B

```
data logical_brb_xwalk;
  length variable content $ 100;
  variable="Unit of analysis"; content="Multiple (EIN, SEIN, SIC, STATE,
County)"; output;
  variable="Geographic coverage"; content="&ecfstates."; output;
  variable="Periodicity of the data"; content="Yearly"; output;
  variable="Period covered";
content="&ecf_start_year.:&ecf_start_quarter.-&ecf_end_year.:&ecf_end_quarter.";
  output;
  variable="Documentation source(s)"; content="See data description"; output;
  variable="Variable list, sort order, indexes"; content="See PROC CONTENTS";
output;
run;

data logical_brb_ecflist;
  length variable content $ 100;
  variable="Unit of analysis"; content="Establishment (SEINUNIT,
State-specific identifier)"; output;
  variable="Geographic coverage"; content="&ecfstates."; output;
  variable="Periodicity of the data"; content="Quarterly"; output;
  variable="Period covered";
content="&ecf_start_year.:&ecf_start_quarter.-&ecf_end_year.:&ecf_end_quarter.";
  output;
  variable="Documentation source(s)"; content="See data description"; output;
  variable="Variable list, sort order, indexes"; content="See PROC CONTENTS";
output;
run;

data logical_brb_brlist;
  length variable content $ 100;
  variable="Unit of analysis"; content="Company (EIN)"; output;
  variable="Geographic coverage"; content="National"; output;
  variable="Periodicity of the data"; content="Yearly"; output;
  variable="Period covered";
content="&br_start_year.:&br_start_quarter.-&br_end_year.:&br_end_quarter.";
  output;
  variable="Documentation source(s)"; content="See data description"; output;
  variable="Variable list, sort order, indexes"; content="See PROC CONTENTS";
output;
run;
```

Now start building the memo.

```
%let type=LEHD-BRB;
%let memono=&today.-&type.-&state.;
%let memoname=ces_transfer.&memono..txt;
%let reroute=&outputs/&memoname.;

filename reroutep "&reroute." new;
proc printto print=reroutep new;
run;
```

Set the titles.

```
%ces_transfer_settitle(memono=&memono.);
```

8.1 The coversheet.

```
%ces_transfer_cover(type=&type.);
```

8.2 Create appendix A: ls -l

```
%ces_transfer_appendixa(files=&process_id._&state.*);
```

8.2 Create appendix B: proc contents

```
%ces_transfer_appendixb(file=OUTPUTS.brb_&state._xwalk,extrainfo=logical_brb_xwalk);  
  
%ces_transfer_appendixb(file=OUTPUTS.brb_&state._ecflist,extrainfo=logical_brb_ecflist);  
  
%ces_transfer_appendixb(file=OUTPUTS.brb_&state._brlist,extrainfo=logical_brb_brlist);  
  
proc printto;  
run;
```

Now we need to write out the file transfer script. Add on the memo - it will be transferred together with the data files. Note that the macro assumes that all files are in OUTPUTS.

```
data tmp;  
length permissions $ 10 inode $ 8 owner $ 16 group $ 16 bytes $ 30  
month $ 10 day $ 2 time $ 5 name $ 80 ;  
name="&memoname";  
output;  
run;  
  
proc append base=appendixA data=tmp;  
run;  
  
%ces_transfer_script(appendixA,suffix=_brb);
```

This finishes LEHD preparation for transfer.

1.5.1.9 library/sasprogs/01_brbqa.sas

```
/* +--< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >--+ */  
/* +--< Location: /programs/production/dev7/current/brb >--+ */  
/* +--< File: library/sasprogs/01_brbqa.sas >--+ */  
/* Time-stamp: <05/03/04 23:16:33 vilhuber> */
```

7. Final analytics.

```
proc contents data=OUTPUTS.brb_&state._brlist;  
proc contents data=OUTPUTS.brb_&state._ecflist;  
proc contents data=OUTPUTS.brb_&state._xwalk;  
  
proc print data=OUTPUTS.brb_&state._xwalk(obs=50);  
run;
```

Diagnostics on some of the input files.

```

%brbqa_geo();
%brbqa_geo(_state);
%brbqa_geo(_county);

```

1.5.1.10 library/sasprogs/02_brbqa.sas

```

/* +--< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >--+ */
/* +--< Location: /programs/production/dev7/current/brb >--+ */
/* +--< File: library/sasprogs/02_brbqa.sas >--+ */
/* Time-stamp: <05/01/15 12:54:39 vilhuber> */

```

Diagnostics on some of the output files.

```

%brb_formats;

proc freq data=OUTPUTS.brb_&state._xwalk;
title2 "OUTPUTS.brb_&state._xwalk";
format
  flag_e_s_c_4 $escfour.
  flag_e_s_c_3 $escthree.
  flag_e_s_c_2 $esctwo.
  flag_e_s_c_1 $escone.
  flag_e_s_c $esc.
  flag_e_s_4 $esfour.
  flag_e_s_3 $esthree.
  flag_e_s_2 $estwo.
  flag_e_s_1 $esone.
  flag_e_s $es.
  flag_e_4 $efour.
  flag_e_3 $ethree.
  flag_e_2 $etwo.
  flag_e_1 $eone.
  flag_e $e.
;
table flag_e / out=INTERWRK.brbqa_flag_e;
table flag_e_1 / out=INTERWRK.brbqa_flag_e_1;
table flag_e_2 / out=INTERWRK.brbqa_flag_e_2;
table flag_e_3 / out=INTERWRK.brbqa_flag_e_3;
table flag_e_4 / out=INTERWRK.brbqa_flag_e_4;
table flag_e_s / out=INTERWRK.brbqa_flag_e_s;
table flag_e_s_1 / out=INTERWRK.brbqa_flag_e_s_1;
table flag_e_s_2 / out=INTERWRK.brbqa_flag_e_s_2;
table flag_e_s_3 / out=INTERWRK.brbqa_flag_e_s_3;
table flag_e_s_4 / out=INTERWRK.brbqa_flag_e_s_4;
table flag_e_s_c / out=INTERWRK.brbqa_flag_e_s_c;
table flag_e_s_c_1 / out=INTERWRK.brbqa_flag_e_s_c_1;
table flag_e_s_c_2 / out=INTERWRK.brbqa_flag_e_s_c_2;
table flag_e_s_c_3 / out=INTERWRK.brbqa_flag_e_s_c_3;
table flag_e_s_c_4 / out=INTERWRK.brbqa_flag_e_s_c_4;
run;

```

1.5.1.11 library/sasprogs/03_brbqa.sas

```

/* +--< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >--+ */
/* +--< Location: /programs/production/dev7/current/brb >--+ */
/* +--< File: library/sasprogs/03_brbqa.sas >--+ */
/* Time-stamp: <05/02/24 11:19:03 vilhuber> */

```

The BRB can cover multiple ECF files. It is defined on the largest possible ECF range, so that some states may not contribute to certain years of data if the ECF range for that particular state is shorter.

```

/* brbqa_input_tables: has state and year range for ECF and SSEL */

%let module_number=03;

data qa_one;
  set INTERWRK.brbqa_input_tables
  (keep=state start_year start_quarter end_year end_quarter process_id
  where=(process_id='ecf')
  );
  keep yyq state_flag;
  do year=start_year to end_year;
if year=start_year then a=start_quarter;
else a=1;
if year=end_year then b=end_quarter;
else b=4;
do quarter=a to b;
  yyq=put(yyq(year,quarter),yyq6.);
  state_flag=1;
  output;
end; /* end quarter loop */
end; /* end year loop */
run;

proc summary data=qa_one nway;
  class yyq;
  var state_flag;
  output out=qa_two n=n;
run;

proc print data=qa_two;
run;

```

We will consider that if more than 50% of all periods cover less than 75% of states, that a warning should be output. Is this reasonable?


```

    data _null_;
set qa_two end=eof;
retain maxn 0;
maxn=max(maxn,n);
if eof then call symput('maxn',trim(left(maxn)));
    run;

%let warnlevel=75;
%let warnperiods=50;

    data INTERWRK.brbqa_coverage_02;
set qa_two end=eof;
keep yyq coverage note status;
length note $ 52 status $ 7;
retain flag 0;
note="";
status="OK";
coverage=n/&maxn.;
label coverage = "% states covered";
if coverage < &warnlevel./100 then do;
    note="Less than &warnlevel.% of states covered";
    status="Warning";
    flag+1;
end;
    if eof then do;
        if flag/_n_ > &warnperiods./100 then do;
call symput('note',"More than &warnperiods.% of states have
warnings");
call symput('status',"%upcase(warn)");
end;
    else do;
call symput('note',"");
call symput('status',"%upcase(ok)");
end;
    end; /* end eof condition */
run;

```

Register the test.

```

%register_qa_test(
    qa_test_name=Test &module_number.-1: Verify state coverage across time,
    qa_module_number=&module_number.,
    status=&status.,
    result_files=INTERWRK.brbqa_coverage_02,
    notes=&note );

```

We also provide the raw date ranges, for verification.

```

data INTERWRK.brbqa_ecf_dateranges_02
    (keep=state start_yyq end_yyq);
set INTERWRK.brbqa_input_tables(where=(process_id='ecf'));
start_yyq=put(yyq(start_year,start_quarter),yyq6.);
end_yyq=put(yyq(end_year,end_quarter),yyq6.);
run;

%register_qa_test(
    qa_test_name=Test &module_number.-2: Date ranges of multiple ECFs,
    qa_module_number=&module_number.,
    status=Verify,
    result_files=INTERWRK.brbqa_ecf_dateranges_02,
    notes=List of dateranges for ECF by state );

```

1.5.2 SAS macros used

1.5.2.1 library/macros/brb_formats.sas

```
/* +--< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >--+ */
/* +--< Location: /programs/production/dev7/current/brb >--+ */
/* +--< File: library/macros/brb_formats.sas >--+ */
/* Time-stamp: <05/01/15 16:19:21 vilhuber> */
```

Based on formats provided by Hyowook Chiang and Kristin Sandusky. January 14, 2004.

```
%macro brb_formats();
```

```
proc format;
```

Fourth-level interactions

```
/*----- flag_e_s_c_4 -----*/
value $escfour
'M'='M: matched in EIN/STATE/COUNTY/SIC4 level'
'B'='B: only in BR'
'L'='L: only in LEHD';
/*----- flag_e_s_c_3 -----*/
value $escthree
'M'='M: matched in EIN/STATE/COUNTY/SIC3 level'
'B'='B: only in BR'
'L'='L: only in LEHD';
/*----- flag_e_s_c_2 -----*/
value $esctwo
'M'='M: matched in EIN/STATE/COUNTY/SIC2 level'
'B'='B: only in BR'
'L'='L: only in LEHD';
/*----- flag_e_s_c_1 -----*/
value $escone
'M'='M: matched in EIN/STATE/COUNTY/SIC1 level'
'B'='B: only in BR'
'L'='L: only in LEHD';
```

Third-level interactions.

```
/*----- flag_e_s_c -----*/
value $esc
'M'='M: matched in EIN/STATE/COUNTY level'
'B'='B: only in BR'
'L'='L: only in LEHD';
/*----- flag_e_s_4 -----*/
value $esfour
'M'='M: matched in EIN/STATE/SIC4 level'
'B'='B: only in BR'
'L'='L: only in LEHD';
/*----- flag_e_s_3 -----*/
value $esthree
'M'='M: matched in EIN/STATE/SIC3 level'
'B'='B: only in BR'
'L'='L: only in LEHD';
/*----- flag_e_s_2 -----*/
value $estwo
'M'='M: matched in EIN/STATE/SIC2 level'
'B'='B: only in BR'
'L'='L: only in LEHD';
/*----- flag_e_s_1 -----*/
value $esone
'M'='M: matched in EIN/STATE/SIC1 level'
'B'='B: only in BR'
'L'='L: only in LEHD';
```

Second-level interactions

```

/*----- flag_e_s -----*/
  value $es
'M'='M: matched in EIN/STATE level'
'B'='B: only in BR'
'L'='L: only in LEHD';
/*----- flag_e_4 -----*/
  value $efour
'M'='M: matched in EIN/SIC4 level'
'B'='B: only in BR'
'L'='L: only in LEHD';
/*----- flag_e_3 -----*/
  value $ethree
'M'='M: matched in EIN/SIC3 level'
'B'='B: only in BR'
'L'='L: only in LEHD';
/*----- flag_e_2 -----*/
  value $etwo
'M'='M: matched in EIN/SIC2 level'
'B'='B: only in BR'
'L'='L: only in LEHD';
/*----- flag_e_1 -----*/
  value $eone
'M'='M: matched in EIN/SIC1 level'
'B'='B: only in BR'
'L'='L: only in LEHD';

```

Top level

```

/*----- flag_e -----*/
  value $e
'M'='M: matched in EIN level'
'B'='B: only in BR'
'L'='L: only in LEHD';
run;
%mend;

```

1.5.2.2 library/macros/brbqa_geo.sas

```

/* ---< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >--- */
/* ---< Location: /programs/production/dev7/current/brb >--- */
/* ---< File: library/macros/brbqa_geo.sas >--- */
/* Time-stamp: <04/10/15 08:00:23 vilhuber> */

```

Diagnostics on some of the input files.

```

%macro brbqa_geo(geo);
  %macro ind(ind);
proc freq data=INTERWRK.xwalk&geo.&ind;
  tables flag; ;
  title "By Year, EIN, &geo, &ind";
run;
%mend;
  %ind();
  %ind(_sic1);
  %ind(_sic2);
  %ind(_sic3);
  %ind(_sic4);
%mend brbqa_geo;

```

1.5.2.3 library/macros/capture_index.sas

```
/* +--< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >---+ */
/* +--< Location: /programs/production/dev7/current/brb >---+ */
/* +--< File: library/macros/capture_index.sas >---+ */
/* Time-stamp: <04/10/12 14:17:48 vilhuber> */
```

Captures index information to a file.

```
%macro capture_index(dataset);

%let memname=%scan(&dataset.,2,.);

proc datasets library=work;
delete tmp;
run;

proc contents data=&dataset. noprint out2=tmp;
run;

%if %sysfunc(exist(tmp)) %then %do;

data _null_;
nobs=0;
ds=open("tmp");
nobs=attrn(ds,'nobs');
call symput('checkobs',trim(left(nobs)));
run;
%put TMP has &checkobs obs;

%if (&checkobs > 0 ) %then %do;
data _index(keep=memname recreate name type);
length memname $ 32;
set tmp;
if type="Index";
memname="&memname";
run;
%end;
%else %do;
data _index;
length memname name $ 32 type $ 12 recreate $ 154;
memname="&memname";
type="None";
recreate="";
name="";
run;
%end;
%end; /* end exist _index condition */
%mend;
```

1.5.2.4 library/macros/capture_sort.sas

```
/* +--< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >---+ */
/* +--< Location: /programs/production/dev7/current/brb >---+ */
/* +--< File: library/macros/capture_sort.sas >---+ */
/* Time-stamp: <04/10/15 07:08:24 vilhuber> */
```

Captures sort information to a file.

```
%macro capture_sort(dataset);

%let memname=%scan(&dataset.,2,.);

proc datasets library=work;
delete tmp;
run;

proc contents data=&dataset. noprint out=tmp;
run;

    %if %sysfunc(exist(tmp)) %then %do;

data _null_;
nobs=0;
ds=open("tmp");
nobs=attrn(ds,'nobs');
call symput('checkobs',trim(left(nobs)));
run;
%put TMP has &checkobs obs;

%if (&checkobs > 0 ) %then %do;
    data _sort(keep=memname sortedby name );
        length memname $ 32;
        set tmp;
        if sortedby ne .;
        memname="&memname";
    run;
%end;
%else %do;
    data _sort;
length memname name $ 32 sortedby 3;
memname="&memname";
sortedby=.;
name="";
run;
%end;
%end; /* end exist _sort condition */
%mend;
```

1.5.2.5 library/macros/ces_transfer_appendixa.sas

```
/* +--< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >--+ */
/* +--< Location: /programs/production/dev7/current/brb >--+ */
/* +--< File: library/macros/ces_transfer_appendixa.sas >--+ */
/* Time-stamp: <05/02/24 22:28:11 vilhuber> */
```

This creates Appendix A for a file transfer memo for CES. Essentially, Appendix A is a directory listing. There probably is a better way to do this. This is OS sensitive.

```

%macro ces_transfer_appendixa(files=);

options ls=78;
title4 "Appendix A: File location and description ";
title5 "Directory: &outputs.";

filename manifest pipe "(cd &outputs.; \ls -l &files. )";

data appendixA;
    length permissions $ 10 inode $ 8 owner $ 16 group $ 16 bytes $ 30 month $
10 day $ 2 time $ 5 name $ 80 ;
    infile manifest;
%if ( "&syssscpl" = "SunOS" ) %then %do;
    %put Using SOLARIS syntax for ls readin;
    input permissions inode owner group bytes month day time name;
%end;
%else %if ( "&sysscp" = "LINUX" ) %then %do;
    %put Using LINUX syntax for ls readin;
    input permissions inode owner group bytes month time name;
    day="";
%end;
%else %do;
    %put THIS PROGRAM DOES NOT KNOW WHAT TO DO;
%end;
run;

proc print data=appendixa;
var name month day bytes;
run;

%mend;

```

1.5.2.6 library/macros/ces_transfer_appendixb.sas

```

/* +--< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >--+ */
/* +--< Location: /programs/production/dev7/current/brb >--+ */
/* +--< File: library/macros/ces_transfer_appendixb.sas >--+ */
/* Time-stamp: <05/02/24 22:08:26 vilhuber> */

```

This creates Appendix B for a file transfer memo for CES. Essentially, Appendix B is a proc contents

```

%macro ces_transfer_appendixb(file=,extrainfo=);

options ls=78;

title4 "Appendix B: Logical attributes";

%if ( "&extrainfo" ne "" ) %then %do;

    proc print data=&extrainfo.;
run;

%end;

proc contents data=&file. varnum;
run;

%mend;

```

1.5.2.7 library/macros/ces_transfer_cover.sas

```

/* +---< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >---+ */
/* +---< Location: /programs/production/dev7/current/brb >---+ */
/* +---< File: library/macros/ces_transfer_cover.sas >---+ */
/* Time-stamp: <05/02/24 21:35:33 vilhuber> */

```

This creates a cover sheet for a file transfer to CES. It needs to be augmented with at least two, if not three appendixes.

```

%macro ces_transfer_cover(type=);

options ls=78;

data coversheet;
    set INPUTS.parms_us_ces_coversheet;
    keep variable content;
    /* FILE TYPE */
    if field_no = 1 then content="&type.";
    /* Authorizing transfer */
    if field_no = 11 then content="Louis Schwarz";
    /* Date authorized - assumes QA the same day */
    if field_no = 13 then content="&today";
run;

proc print data=coversheet;
id variable;
var content;
run;

%mend;

```

1.5.2.8 library/macros/ces_transfer_script.sas

```
/* ---< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >---+ */
/* ---< Location: /programs/production/dev7/current/brb >---+ */
/* ---< File: library/macros/ces_transfer_script.sas >---+ */
/* Time-stamp: <05/03/04 16:56:38 vilhuber> */
```

This creates Appendix C for a file transfer memo for CES. Essentially, Appendix C is a script for SFTP (OpenSSH version)

```
%macro ces_transfer_script(indata,suffix=);

options ls=78;

/* specify SFTP options here */
/* -P as per Kristin McCue, 2005-02 */

%let sftp_opts=-P;

filename script "&outputs/ces_transfer.&state.&suffix..sftp" new;

data _null_;
  set &indata.;
  length fullpath $ 255;
  file script;
  fullpath=trim(left("&outputs./"))|| trim(left(name));
  put "get &sftp_opts. " fullpath;
run;

%mend;
```

1.5.2.9 library/macros/ces_transfer_settitle.sas

```
/* ---< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >---+ */
/* ---< Location: /programs/production/dev7/current/brb >---+ */
/* ---< File: library/macros/ces_transfer_settitle.sas >---+ */
/* Time-stamp: <05/02/24 21:34:28 vilhuber> */
```

```
%macro ces_transfer_settitle(memono=);

title1 "TITLES 13 AND/OR 26 U.S.C.: DISCLOSURE PROHIBITED";
title2 "STAFF RESTRICTED: DO NOT CIRCULATE";
title3 "CES/LEHD Data Transfer Memorandum No. &memono";
%mend;
```

1.5.2.10 library/macros/read_ecf.sas

```
/* ---< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >---+ */
/* ---< Location: /programs/production/dev7/current/brb >---+ */
/* ---< File: library/macros/read_ecf.sas >---+ */
/* Time-stamp: <04/10/26 15:09:00 vilhuber> */
```


Read in all the ECFs. Should work both for a single ECF or all ECFs.

```
%macro read_ecf();  
  
%let i=1;  
  
    data ecf;  
    set
```

Loop through all datasets.

```
%do %until ("%scan(&ecfstates,&i)=""");  
    %let st=%scan(&ecfstates,&i);  
    INPUTS.ecf_&st._seinunit  
    (keep=year quarter es_state es_county es_ein  
        es_sic sein seinunit es_state  
        rename=(es_ein=ein es_state=stgeo  
            es_county=county es_sic=sic4)  
        where =(&start_year. <= year  
            and year <= &end_year.  
            and ein~='')  
        )  
    )  
    %let i=%eval(&i+1);  
%end; /* end of scan loop */  
;  
length state $2;  
state=lowcase(fipstate(stgeo));  
run;  
%mend;
```

1.5.2.11 library/macros/read_ssel.sas

```
/* +--< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >--- */  
/* +--< Location: /programs/production/dev7/current/brb >--- */  
/* +--< File: library/macros/read_ssel.sas >--- */  
/* Time-stamp: <04/10/26 15:18:02 vilhuber> */
```

Read in all the BR/SSELS, with some modifications.

```
%macro read_ssel(level=);  
  
%if ( &level. = mu ) %then %do;  
    %let xkeep=;  
    %let xcond=substr(cfn,7,3) ne '999' and act ne 'G' ;  
%end;  
%if ( &level. = su ) %then %do;  
    %let xkeep=pdiv;  
    %let xcond=pdiv ne 'M';  
%end;  
  
    data ssel_&level.(keep= year cfn ppn ein stgeo ctygeo sic4);  
    set
```

Loop through all datasets.

```

%do year=&start_year. %to &end_year.;
  INPUTS.br_us_&level._&year.
  (keep= cfn ein ppn stgeo &xkeep. act ctygeo sic
  where=&xcond.
/*          and stgeo in ( &stgeos ) */
          and act ne 'D'
          and ein not in ('','0','000000000')
          and cfn not in ('0000000000', ''))
  )
  in=ssel&year.
  )
%end;
;

```

Set the year variable.

```

%do year=&start_year. %to &end_year.;
  if ssel&year. then year=&year.;
%end;
sic4=substr(sic,1,4);
run;

proc sort data=ssel_&level. nodupkey;
  by year cfn;
  run;

%mend;

```

1.5.2.12 library/macros/recreate_indexes.sas

```

/* +--< LEHD-QWI brb 1.1.07 2005-05-19 schwa305 >--+ */
/* +--< Location: /programs/production/dev7/current/brb >--+ */
/* +--< File: library/macros/recreate_indexes.sas >--+ */
/* Time-stamp: <05/03/23 13:32:03 vilhuber> */

```

Builds a SAS program that recreates indexes and sorts. Requires type of file (ecf, uff)

```

%macro recreate_indexes(basename=rdcru,suffix=,newname=,sort=yes,index=yes);
%let outputs=%trim(%sysfunc(pathname(OUTPUTS)));
%let filename=&outputs/&process_id._&state.&suffix._recreate_indexes.sas;

%if ( "&sort" = "yes" ) %then %do;
proc sort data=INTERWRK.&basename._sorts&suffix. nodupkey;
by memname sortedby;
run;

data _null_;
file "&filename.";
set INTERWRK.&basename._sorts&suffix.(where=(sortedby ne .)) end=eof;
by memname sortedby;
memname=trim(left(memname));
localname=trim(left(localname));

if _n_ = 1 then do;
put "/*-----*/";
put "/* Automatically created          */";
put "/* Will recreate indexes on remote system as they  */";
put "/* existed on the source system.          */";
put "/*          */";
put "/* If placed in the same directory as the data     */";
put "/* files, does not need editing. Otherwise, adjust */";
put "/* libname statement to point to data files.     */";
put "/*-----*/";

put "libname here '.'";
end;

```

NEWNAME may contain the name on the destination system, if known. If empty, we simply reuse the name on LEHD's system.

```

if newname="" then newname=trim(left(memname));
else newname=trim(left(newname));

if first.memname then do;

```

LOCALNAME is the name in this process. MEMNAME is the original name.

```

if localname="" then
put "proc sort data=here." newname ";";
else
put "proc sort data=here." localname " out=here." newname ";";
put "by";
end;
put " " name;
if last.memname then do;
put ";";
put "run;";
end;
put " ";
run;
%end;

```

Now append to a SAS program file to reconstruct indexes.

```

%if ( "&index." = "yes" ) %then %do;
proc sort data=INTERWRK.&basename._indexes&suffix. nodupkey;
by memname name;
run;

data _null_;
file "&filename." mod;
set INTERWRK.&basename._indexes&suffix.(where=(type="Index")) end=eof;
by memname name;
if _n_ = 1 then do;
put "/*-----*/";
put "/* This recreates the indexes */";
put "/*-----*/";
put "proc datasets library=here;";
end;

```

NEWNAME may contain the name on the destination system, if known. If empty, we simply reuse the name on LEHD's system.

```

if newname="" then newname=trim(left(memname));
else newname=trim(left(newname));
if first.memname then do;
put "modify " newname ";";
end;
if type="Index" then do;
put recreate;
end;
if last.memname then do;
put "contents data=" newname ";";
end;
if eof then do;
put "quit;";
end;
run;
%end;

%mend;

```

1.5.3 SAS layouts used

1.5.4 SAS formats used

1.5.5 Scripts used

1.5.6 Total code lines for this process

SAS	:	1283
KSH	:	0