



## NATIONAL DATA ARCHIVE ON NDACAN CHILD ABUSE AND NEGLECT

### **Joining two data files from the RPG-2 dataset, using SPSS.**

This document is intended to assist data users with joining/merging two data files from RPG-2, dataset #218. The instructions are generally applicable for use with other data files with similar characteristics.

The two data files used in this example are the `rpg_esl_person_level` and the `rpg_esl_case_level`. These were selected because they feature the joining of two datasets where the unit of observation differs, resulting in different record counts for each file.

The `rpg_esl_person_level` dataset contains information about each person who participated in the RPG-2 data collection, including substance using adults and their children. As shown in the User's Guide Table 1, you will note that the variables that identify a unique record for the `rpg_esl_person_level` data file are: `Grantee_ID`, `Case_ID`, and `Ind_ID`. In this file, because each case has multiple people and the unit of observation is the person, the `Case_ID` repeats across records for each person represented in that case. The variables that identify a unique record in the `rpg_esl_case_level` data file are: `Grantee_ID` and `Case_ID`. Since `Grantee_ID` and `Case_ID` are unique to both data files, we will merge/join the records based on those unique identifiers.

In this example, we are interested in preserving all records from the `rpg_esl_person_level` data file that contains 8,626 records pertaining to each individual who participated in the RPG-2 study. Due to the `rpg_esl_case_level` data file containing information about the "case," there are only 2,887 records. To summarize, each **case** represents multiple **individuals** (substance using adults and their children). When we merge/join these two datasets together we want the information about the case properly associated with records for the individuals. This is known as a many-to-one relationship. The Star Join procedure in SPSS performs a left outer join of the two files where the resulting file will consist of all records from the `rpg_esl_person_level` and the matching records from the `rpg_esl_case_level` file.

The instructions are broken out into sections and should be followed in the order in which they appear. Data users should copy each section of syntax into the Syntax Editor window of SPSS, omitting section headings and other explanatory text. You will need to update all file paths to point to the location of the data file on your own computer and provide locations to where the modified data files should be saved.

The instructions were written based on Version 24 of the IBM SPSS software package.

## **Section One: Assign a file nickname to the rpg\_esl\_person\_level data file**

This section will assist data users with opening and assigning file nickname that will be used during the Star Join procedure. The Star Join procedure does not require sorting the data file by the key IDs that will be used during the join/merge.

\*Assigning the nickname t0 to the rpg\_esl\_person\_level data file.

```
GET FILE = 'c:\temp\change-the-file-path-location\rpg_esl_person_level_data.sav'.
```

```
EXECUTE.
```

```
DATASET NAME t0.
```

```
EXECUTE.
```

```
DATASET ACTIVATE t0.
```

```
EXECUTE.
```

Please leave the rpg\_esl\_person\_level data file open and do not close it.

## **Section Two: Assign a nickname to the rpg\_esl\_case\_level data file**

In this section, you will assign the nickname “t1” to the rpg\_esl\_case\_level data file. This nickname will be leveraged during the Star Join procedure in Section Three.

\*assign the nickname t1 to the rpg\_esl\_case\_level data file.

```
GET FILE = 'c:\temp\change-the-file-path-location\rpg_esl_case_level.sav'.
```

```
DATASET NAME t1.
```

```
EXECUTE.
```

```
DATASET ACTIVATE t1.
```

```
EXECUTE.
```

Please leave the rpg\_esl\_case\_level file open and do not close it.

## **Section Three: Merge/join the rpg\_esl\_person\_level data file with the rpg\_esl\_case\_level data file.**

In this section, you will perform the data join/merge of the rpg\_esl\_person\_level data with the rpg\_esl\_case\_level data file. The data file nickname of “t0” will point to the rpg\_esl\_person\_level data file and the data file nickname of “t1” will point to the rpg\_esl\_case\_level data file. After pasting the Star Join syntax, you will need to insert hard returns so that each line of syntax contains fewer than 251 bytes per line. There are two syntax examples of the Star Join procedure presented below. The first set of syntax includes all

variables from both data files in the resulting joined dataset. The second set of syntax lists out each variable from the two data files that you want to include in the resulting joined dataset. This way is handy if you are interested in including only a subset of variables from one or both files.

\*Section Three, first example—includes all variables from both files in the resulting joined data file.

```
DATASET ACTIVATE t0.
```

```
EXECUTE.
```

```
STAR JOIN
```

```
  /SELECT *
```

```
    /FROM * AS t0
```

```
    /JOIN 't1' AS t1
```

```
      ON t0.Grantee_ID=t1.Grantee_ID AND t0.CASE_ID=t1.CASE_ID
```

```
  /OUTFILE FILE= "c:\temp\change-the-file-path-location-and-name-the-newly-merged-file.sav".
```

```
EXECUTE.
```

```
GET FILE = "c:\temp\change-the-file-path-location-and-name-the -newly-merged-file.sav".
```

```
EXECUTE.
```

\*Section Three, second example—lists out each variable to include in resulting joined data file.

```
DATASET ACTIVATE t0.
```

```
EXECUTE.
```

```
STAR JOIN
```

```
  /SELECT
```

```
t0.IND_ID, t0.RPG_ESL_P_income_wage, t0.RPG_ESL_P_income_pa,  
t0.RPG_ESL_P_income_ret, t0.RPG_ESL_P_income_dis, t0.RPG_ESL_P_income_oth,  
t0.RPG_ESL_P_income_none, t0.RPG_ESL_P_residence, t0.RPG_ESL_P_language,  
t0.RPG_ESL_P_relationship_to_fc, t0.RPG_ESL_P_fc, t0.RPG_ESL_P_rda,  
t0.RPG_ESL_P_ffa, t0.RPG_ESL_P_adult, t0.RPG_ESL_P_child, t0.RPG_ESL_P_age,  
t0.RPG_ESL_P_race, t0.RPG_ESL_P_ethnicity, t0.RPG_ESL_P_report_bio_parent,  
t0.RPG_ESL_P_education, t0.RPG_ESL_P_income, t0.RPG_ESL_P_adult_relationship,  
t0.RPG_ESL_P_gender, t1.RPG_ESL_C_in_SUT, t1.RPG_ESL_C_in_CCT,  
t1.RPG_ESL_C_in_CS, t1.RPG_ESL_C_in_FS, t1.RPG_ESL_C_in_FTDC,  
t1.RPG_ESL_C_in_RTT, t1.RPG_ESL_C_EBP_enr_lngth,  
t1.RPG_ESL_C_ebp_enrollments, t1.RPG_ESL_C_num_ebps,
```

```
t1.RPG_ESL_C_RPG_enr_lngth, t1.RPG_ESL_C_case_closed,  
t1.RPG_ESL_C_close_prg_comp, t1.RPG_ESL_C_close_fam_move,  
t1.RPG_ESL_C_close_unable_loc, t1.RPG_ESL_C_close_unresp,  
t1.RPG_ESL_C_close_fam_dec, t1.RPG_ESL_C_close_transfer,  
t1.RPG_ESL_C_close_other, t1.RPG_ESL_C_close_death,  
t1.RPG_ESL_C_treatment_fac_sut, t1.RPG_ESL_C_num_children,  
t1.RPG_ESL_C_num_adults, t1.RPG_ESL_C_num_ind_case,  
t1.RPG_ESL_C_only_bio_fam, t1.RPG_ESL_C_in_ebp, t1.RPG_ESL_C_in_CF,  
t1.RPG_ESL_C_in_NPP, t1.RPG_ESL_C_in_SF, t1.RPG_ESL_C_in_LIB,  
t1.RPG_ESL_C_in_MM, t1.RPG_ESL_C_in_SS, t1.RPG_ESL_C_in_CPP,  
t1.RPG_ESL_C_in_CBT, t1.RPG_ESL_C_in_TFCBT, t1.RPG_ESL_C_in_PCIT,  
t1.RPG_ESL_C_time_to_EBP, t1.RPG_ESL_C_proportion_EBP,  
t1.RPG_ESL_C_state_sut, t1.RPG_ESL_C_grantee_sut, t1.RPG_ESL_C_adult_sut,  
t1.RPG_ESL_C_rpg_enr_date, t1.RPG_ESL_C_rpg_close_date
```

```
/FROM * AS t0
```

```
/JOIN 't1' AS t1
```

```
ON t0.Grantee_ID=t1.Grantee_ID AND t0.CASE_ID=t1.CASE_ID
```

```
/OUTFILE FILE= "c:\temp\change-the-file-path-location-and-name-the-newly-merged-file.sav".
```

```
EXECUTE.
```

```
GET FILE = "c:\temp\update-the-file-path-location-to-point-to-your-newly-merged-file.sav".
```

```
EXECUTE.
```

In this example, you should have a combined data file that contains 8,626 records and 67 variables. You should refer to each respective data file's codebook for information pertaining to the variables in the newly linked/merged dataset.

This document was prepared by NDACAN Analyst, Holly M. Larrabee. Email questions to [NDACANsupport@cornell.edu](mailto:NDACANsupport@cornell.edu).