PanelWhiz: A Flexible Modularized Stata Interface for Accessing Large Scale Panel Data Sets

John P. Haisken-DeNew[1] (RWI Essen) and
Markus Hahn (RWI Essen, Uni Bochum)

This Documentation Version: Sep 2006 (version 1)

Abstract: This paper outlines a panel data retrieval program written for Stata/SE or better, which allows easier accessing of the household panel data sets. Using a drop-down menu system, the researcher selects variables from any and all available years of the panel. The data is automatically retrieved and merged to form a "long file", which can be directly used by the Stata panel estimators. The system implements modular data cleaning programs called "plugins". Yearly updates to the data retrievals can be made automatically. Projects can be stored in libraries allowing modular administration and appending.

Keywords: Panel data storage and retrieval, SOEP, HILDA

JEL: C81, C87, C23

[1]The author wishes to thank the following persons: Michael R. Veall, Klaus F. Zimmermann, Gert G. Wagner, Christoph M. Schmidt, Peter Szmyt, Kristina Liljefors, Joachim R. Frick, Markus Pannenberg, Ingo Sieber, Ingo Geishecker, Mathias Sinning, Matthias Vorell, Patrick Dross, Christopher Baum, Stephen Jenkins, Goetz Rohwer, William Gould, Kevin Crow, Ulrich Kohler.

1. Introduction

Applied social scientists have forever been faced with different data interfaces for different data sets. In most cases, an interface is not even available, forcing the researcher to address data files by name, and extract the information required by hand. However, the specific structure of a panel data can vary dramatically as described in Haisken-DeNew (2001), such that some data sets provide many files per year, differing by their population, or level of aggregation etc.

PanelWhiz is a collection of Stata add-ons or subroutines that allow researchers to use an intuitive "common" graphical interface for accessing many datasets directly within the statistical package Stata, whereby the researcher does not select individual variables, but rather vectors of variables (items) with one mouse click. This allows for an efficient method of selecting information for a data set retrieval, especially if the data set contains many waves (years) of information. With one mouse-click, data can be automatically retrieved, with merging and matching done automatically.

The basic functionality behind PanelWhiz has evolved for some time now. It allows data sets that have not previously been integrated into the PanelWhiz system to be added with minimal additional programming. PanelWhiz establishes a meta-data method of describing panel data sets by injecting item-correspondence information into the data sets themselves, allowing Stata add-ons to read this information and use it in an actual retrieval. This paper outlines the functionality of PanelWhiz and provides an immediate introduction to using the package.

2. Historical Development

The development of PanelWhiz starts[2] with the German Socio-Economic Panel Data (SOEP), which is a collection of data files currently numbering over 250 in the regular distribution from the DIW Berlin. The dataset contains information collected from the longest running European household panel survey, including information at the person, household, and spell level. In addition there have been numerous additional data files provided, which have been generated by the SOEP Group at the DIW Berlin containing derived variables or user-friendly versions of biography data. See Haisken-DeNew and Frick (2005) and Haisken-DeNew (2001) for details.

In order to deal with the complexity of this very large panel data set, an internet based tool called SOEPinfo was established by Dr. John Haisken-DeNew and Ingo Sieber in the SOEP Group, already in 1996 in a rudimentary form[3]. This tool allowed users to "collect" SOEP variables by clicking on them with a mouse, and once variables were "added to basket", actions could be run on the variables, such as displaying pre-generated frequencies, or jumping to the exact place in the questionnaire where the variables could be found. Alternatively, one could automatically create a command file[4] in Stata, SAS or SPSS to retrieve the variables "collected in the basket".

The basic idea was further developed several years later. "SOEP Menu" was written by Dr. John Haisken-DeNew and came into being as a method allowing users of the German Socio-

[2] Of course the mother of all SOEP retrieval programs is PPD/RZoo/TDA from Götz Rohwer, Uni Bochum !

[3] Since 2003 SOEPinfo has been maintained entirely by Ingo Sieber and Jan Goebel at the DIW (SOEP).

[4]The first tool known to the authors that carried ran the retrieval immediately was written for the NLS. This idea was the basis for SOEPinfo generating executable code. Several years later, the PSID added a similar feature to their Data Center which allowed immediate retrievals of data and downloads to the user's computer. The Canadian Survey of Labour and Income Dynamics (SLID) also provided a rudimentary retrieval tool.

Economic Panel (SOEP) to access the data conveniently and directly in their Stata/SE environments. It allowed users to retrieve data directly from their own locally installed data using an interactive interface, such that data could automatically be transposed from wide to long and be made time consistent in a convenient manner. As the name suggests, "SOEP Menu" was a data set specific solution only relevant for the SOEP. See Haisken-DeNew (2005) for more details.

In spring 2006, a similar package to SOEP Menu was written by Dr. John Haisken-DeNew to incorporate the micro data from the Australian Household Income and Labour Dynamics (HILDA) provided by the Melbourne Institute at Melbourne University, Australia. The scale economies were immediately recognized in summer 2006 and the code Stata code for the SOEP and HILDA was merged in a completely rewritten package called "PanelWhiz". In the fall 2006 work on incorporating two other German data sets into the PanelWhiz system was started.

3. Panel Data Structure and "Variable Characteristics"

With the PanelWhiz system, the user can open data files by clicking on a browse page. The idea behind the tool is that because of the intrinsically longitudinal nature of the data, one is typically NOT interested in retrieving a variable in a particular wave, but rather in retrieving the variable for several waves, i.e. an item-correspondence. For all data sets, a variable renaming algorithm (where necessary) is used to ensure time consistent variable names (See Haisken-DeNew (2001) for more information on this). Thus, if one opens a data file and one finds a variable of interest, one clicks on the variable and information for the entire item correspondence is also collected and added to a "PanelWhiz project". Straightforwardly, the object is to collect items and save them into the data project, allowing an automatic data retrieval.

PanelWhiz can function because Stata provides certain key commands or features in its command language and user interface. PanelWhiz requires:

- Stata SE 9.2 or better. The internal limits of Stata Intercooled are insufficient.
- the existence of variable characteristics: Stata command `char`
- the existence of drop-down menus: Stata command `window menu`
- the existence of dialog boxes: Stata command `db file.dlg`
- being able to buffer info in the global-space: Stata command `global`
- being able to deal with long strings in global-space: Stata command `global`
- the existence of the markup language, `SMCL`
- the existence of the Stata `class` environment

The PanelWhiz system requires additional information describing the contents and structure of a dataset. In order to select an item, the dataset must first be prepared with meta data. For instance, in the HILDA, the variable in 2001 refering to "Country of Birth" is "`aancob`". The item name is "`ancob`" (notice the first "`a`" has been removed). For the years 2001-2004, the variable names are as follows: `aancob`, `bancob`, `cancob`, `dancob`. Thus, when reading the meta information from the variable "`aancob`", it is already clear that the variable is available in all waves (2001-2004). Upon selecting the variable for 2001, the entire item is selected for all years. The information stored in the variable corresponds to the Stata commands:

```
char define aancob[numitems]    "1"
char define aancob[itemnameEN1] "ancob"
char define aancob[itemlabelEN1]"Country of birth"
char define aancob[categoryEN1] "AN: Ancestry"
char define aancob[itemvect1]   "aancob | bancob | cancob | dancob |"
char define aancob[iteminfo1]   "A1 | History: | History: | History: |"
```

The reader will notice "EN" as part of the characteristic names. This corresponds to "English", indicating English label names. The number of items per variable is stored (in this case "1") and indicated by the ending "1" in the characteristic names. The language "German" is indicated by "DE".

Essentially, all data sets in the PanelWhiz system are described in this manner and the meta-information is stored in the data files themselves and read whenever it is required. There is also the possibility of storing information for non-time-varying or special items, such as in the SOEP's Biography files, which typically refer to one fixed time point in the past, although not always.

4. Basic Features of PanelWhiz

An integral part of PanelWhiz is the item-correspondence list, i.e. a list of items, such as "Wage" or "Unemployed Status" which contain lists of actual variable names corresponding to the variables in a particular year. The variable naming schemes for panel data sets vary dramatically.

For better or for worse, the naming scheme of variables in the SOEP reflects essentially the order of the particular question a particular questionnaire in a particular year. For instance the actual variable name `AP3301` refers to personal gross wage in 1984. It simply happens to be the first part (`01`) of question `33` in the person (`P`) questionnaire in 1984 (or year `A`). Clearly, questionnaires evolve over time and the particular position of a question does not remain constant. Thus, there is no guarantee whatsoever that the same "wage" question in the following year follows any kind of patterns similar to the previous year. Indeed the variable in 1985 (year `B`) is named `BP4301`. This variable naming scheme has the advantage of immediately identifying to the user where exactly the question can be found in the survey and whether it was a person or a household head answering on behalf of the entire household, who answered the question. On the other hand, when using many waves of information in a panel setting, keeping variables straight becomes very tedious.

The SOEP is of considerable interest to social scientists as it allows researchers to examine the same individuals over as many as 20 years using the most current panel estimation techniques available in statistical packages such as Stata, SAS and SPSS.

Retrieving the data outright contravenes data security laws in Germany and is therefore not viable. However creating the command files to retrieve data locally installed on the user's own machine is indeed legal and efficient. This allowed the user to run a data retrieval on his own local machine and create a rectangularized data file in "wide format", such that there would be a person or household as the unit of analysis and additional years of information would be stored as additional columns in a spreadsheet.

However, to use the advanced techniques, all packages require the data to be stored in a different manner than "wide format" called "long format". This implies transposing the wide data set to a unit of analysis based on person-years or household-years.

There are no longer variable names with year indicators attached to them such as `WAGE2000` or `WAGE2001`, but just simply `WAGE` with an additional indicator `YEAR` indicating the specific year that the information refers to. Thus each row in the data set refers to a person-year or household-year observation, "making the data sets markedly narrower (fewer variables) and at the same time much longer (more observations), illustrated in Table 1.

Table 1: Wide versus Long Format

Wide Format

PERSNR	WAGE2000	WAGE2001	WAGE2002
100	4000	4050	4100
101	-2	-2	400
102	1000	1000	1200

Long Format

PERSNR	YEAR	WAGE
100	2000	4000
100	2001	4050
100	2002	4100
101	2000	-2
101	2001	-2
101	2002	400
102	2000	1000
102	2001	1000
102	2002	1200

However, before one simply changes a data set from wide to long (in Stata, done by the "reshape" command for instance), one must first be assured that the variables have been made time-consistent. For example, in the simple "gender" question, 1=Male and 2=Female. However this coding could theoretically change over time to 0=Male and 1=Female. Thus, all variables corresponding to the particular item "Gender" must first be harmonized before any reshaping is made. In the case of wages, the analogue would be first to deflate values by a price index to create "real wages".

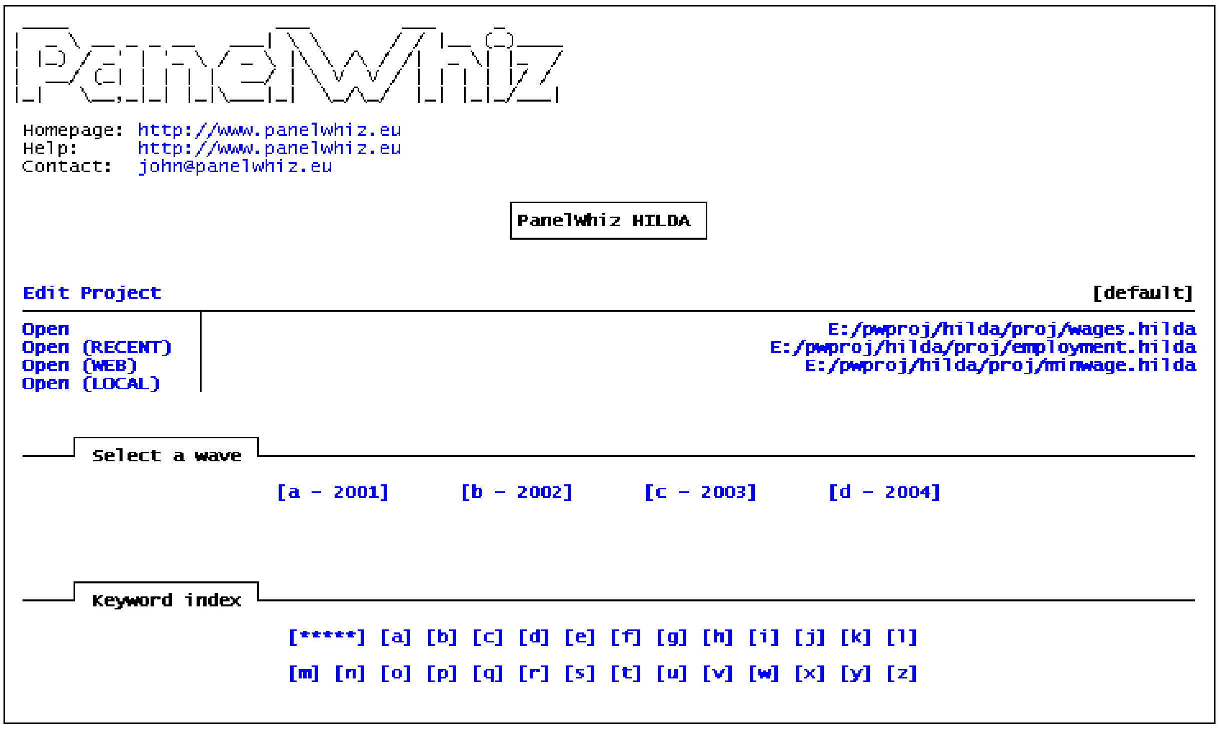
Other data sets have far simpler variable naming schemes and are therefore much simpler to transform from wide to long. The Australian HILDA dataset contains regular speaking variable names that allow easy transformation from wide to long.

4.1 Start Page

As the name suggests, the main functions of the start page are geared toward getting the user started with adding "items" and/or alternatively "specials" to the retrieval. One can open already saved projects that are stored locally, in a local library or from the PanelWhiz homepage library. Furthermore, automatically, the last four saved projects are also immediately clickable (to open and load the contents of the retrieval into the PanelWhiz system).

There are several ways of adding items to the data project. Most intuitively, one opens a data file from a particular year or wave and then selects an item. PanelWhiz, regardless of original file structure, allows the user to open a single browse page for each year. Although data sets like the SOEP have typically up to 10 files per year, only one browse page need be opened. The HILDA uses "combined" files, such that all person and household information is stored in a single year specific file. Once the user selects the year of interest, a browse page is opened sorted by categories. The user then clicks on the appropriate category and then all items pertaining to that category found in that year are listed. The user can obtain information on the variables of interest, such as summary statistics (`summ`), tabulations (`tab`), kernel density (`kdensity`), stored meta information (item correspondence etc). Furthermore, entire items (vectors of variables) can be selected at this stage and added into the data project, or indeed immediately removed if need be. If available, also "plugin" information is displayed and for items with value labels, the listings of all value labels over all variables/years in the item can also be displayed.

Screen Shot 1: Start Page

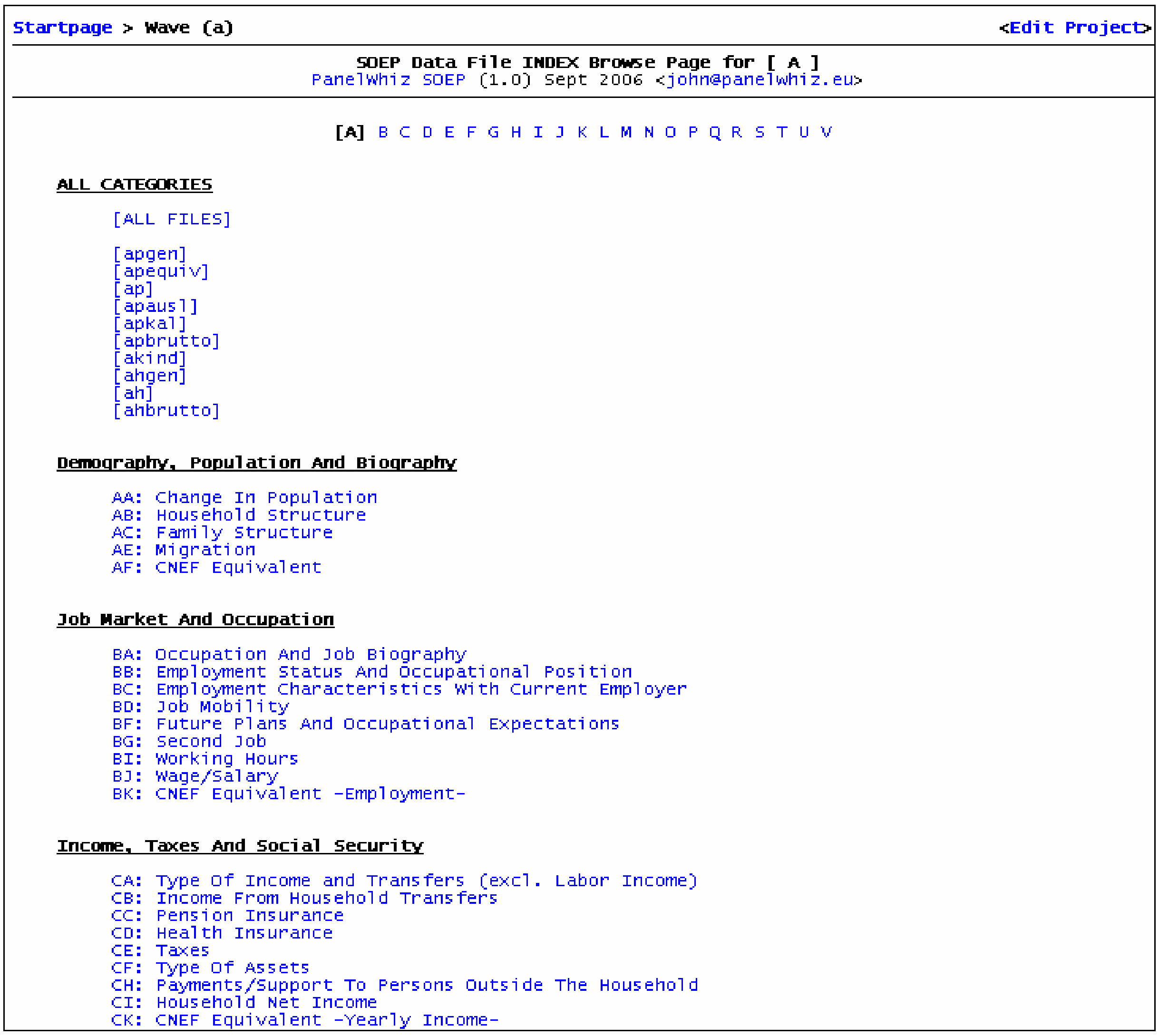


```
Homepage:  http://www.panelwhiz.eu
Help:      http://www.panelwhiz.eu
Contact:   john@panelwhiz.eu

                              PanelWhiz HILDA

Edit Project                                                    [default]

Open            |                      E:/pwproj/hilda/proj/wages.hilda
Open (RECENT)   |                 E:/pwproj/hilda/proj/employment.hilda
Open (WEB)      |                    E:/pwproj/hilda/proj/minwage.hilda
Open (LOCAL)    |

Select a wave

          [a - 2001]    [b - 2002]    [c - 2003]    [d - 2004]

Keyword index

          [*****] [a] [b] [c] [d] [e] [f] [g] [h] [i] [j] [k] [l]
          [m] [n] [o] [p] [q] [r] [s] [t] [u] [v] [w] [x] [y] [z]
```

In the following example, the user has selected the 1984 wave of the SOEP. There is a single entry point for this wave, although it contains information from many different SOEP data files (such as AP, APGEN, etc). The user may chose how he wishes to browse the contents of a particular wave. The user may browse by (a) looking at each original SOEP data file separately, (b) looking at subject categories of variables containing information from all original SOEP data files in that particular wave, or (c) by examining a list of "ALL FILES", i.e. a single list of all variables in all files in that particular wave.

One can see in the heading that wave [A] has been selected. The user may immediately switch to another available wave, such as [V] by simply clicking on the appropriate letter. The categories used here follow the same classification scheme as used by the SOEP's SOEPinfo, allowing the user seamless switching between the two applications.

Screen Shot 2: Example File Index for SOEP Wave A/1984



Startpage > Wave (a) <Edit Project>

SOEP Data File INDEX Browse Page for [A]
PanelWhiz SOEP (1.0) Sept 2006 <john@panelwhiz.eu>

[A] B C D E F G H I J K L M N O P Q R S T U V

ALL CATEGORIES

[ALL FILES]

[apgen]
[apequiv]
[ap]
[apausl]
[apkal]
[apbrutto]
[akind]
[ahgen]
[ah]
[ahbrutto]

Demography, Population And Biography

AA: Change In Population
AB: Household Structure
AC: Family Structure
AE: Migration
AF: CNEF Equivalent

Job Market And Occupation

BA: Occupation And Job Biography
BB: Employment Status And Occupational Position
BC: Employment Characteristics With Current Employer
BD: Job Mobility
BF: Future Plans And Occupational Expectations
BG: Second Job
BI: Working Hours
BJ: Wage/Salary
BK: CNEF Equivalent -Employment-

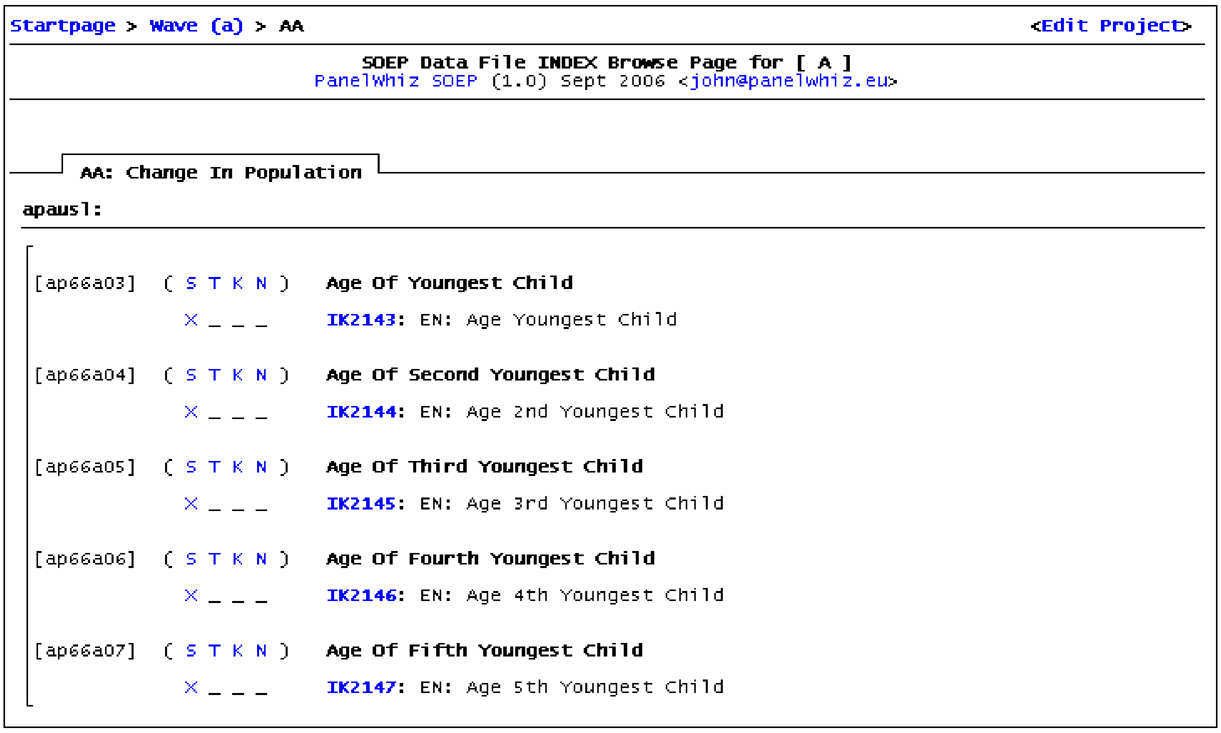
Income, Taxes And Social Security

CA: Type Of Income and Transfers (excl. Labor Income)
CB: Income From Household Transfers
CC: Pension Insurance
CD: Health Insurance
CE: Taxes
CF: Type Of Assets
CH: Payments/Support To Persons Outside The Household
CI: Household Net Income
CK: CNEF Equivalent -Yearly Income-

Once the user has selected a topic category, he is brought to the browse page containing the actual list of variables in that category, as shown in Screen Shot 3. There, the user receives a list of variables such as `[ap66a03]`, then a list of standardized commands (S, T, K, N), in which S provides summary statistics, T provides tabulate output, K produces a kernel density estimate and N displays all meta data information concerning the variable, such as item labels, item-correspondence etc. The variable label is also displayed. In the second line of the variable entry, the user may select the item by clicking on the item name.

The first entry in Screen Shot 3, for instance is `IK2143`. This would select not only the variable `ap66a03` from wave A (1984) but also that information for all other waves. In order to define which waves of data are actually used, the user defines the preferences in the PanelWhiz drop down command “Project | Prefs”.

Screen Shot 3: Example File Index for SOEP Wave A/1984



```
Startpage > Wave (a) > AA                                        <Edit Project>

                 SOEP Data File INDEX Browse Page for [ A ]
              PanelWhiz SOEP (1.0) Sept 2006 <john@panelwhiz.eu>

AA: Change In Population
apausl:

[ap66a03]  ( S T K N )   Age Of Youngest Child
             X _ _ _     IK2143: EN: Age Youngest Child

[ap66a04]  ( S T K N )   Age Of Second Youngest Child
             X _ _ _     IK2144: EN: Age 2nd Youngest Child

[ap66a05]  ( S T K N )   Age Of Third Youngest Child
             X _ _ _     IK2145: EN: Age 3rd Youngest Child

[ap66a06]  ( S T K N )   Age Of Fourth Youngest Child
             X _ _ _     IK2146: EN: Age 4th Youngest Child

[ap66a07]  ( S T K N )   Age Of Fifth Youngest Child
             X _ _ _     IK2147: EN: Age 5th Youngest Child
```

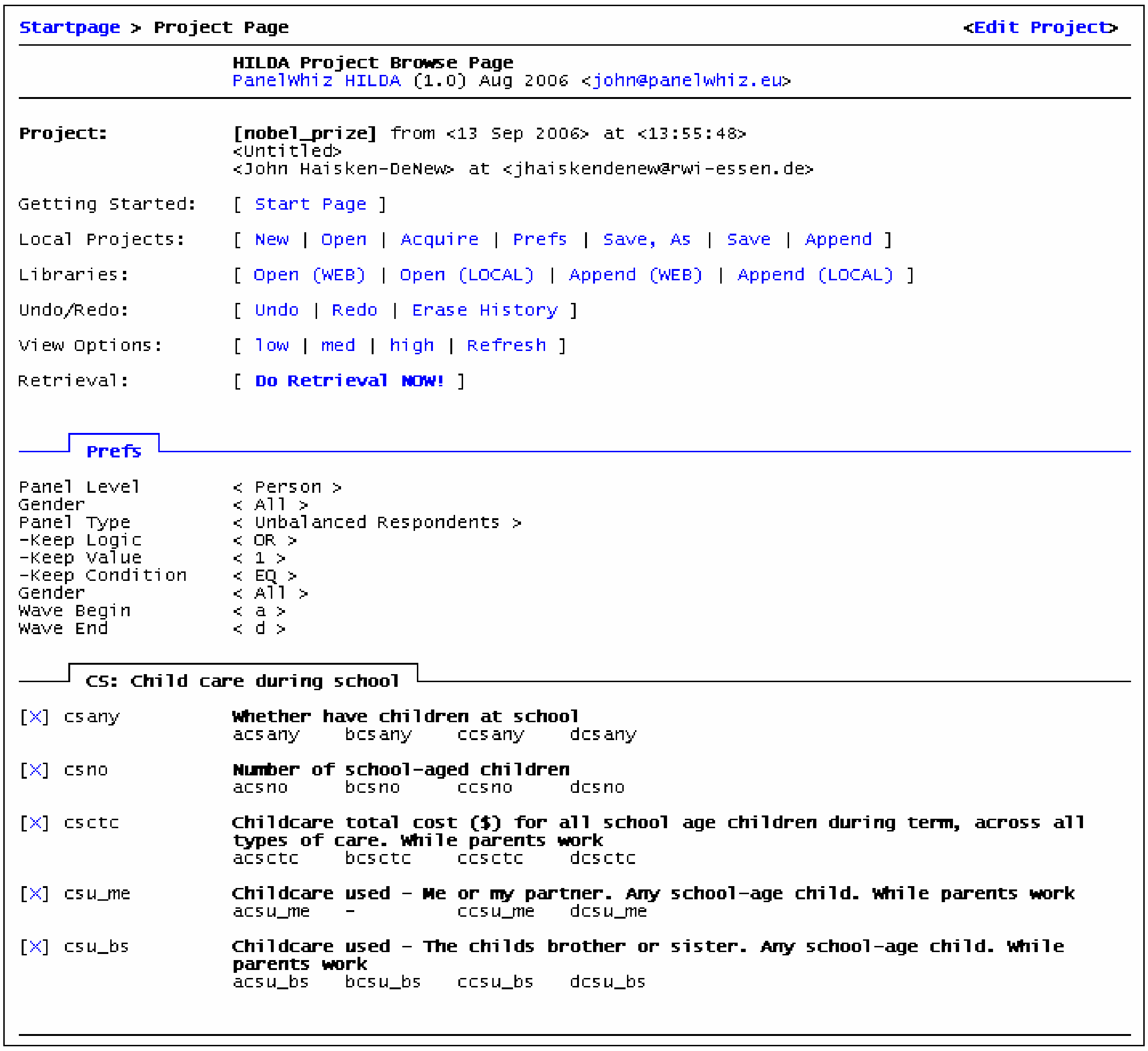
Alternatively, one can add items to the data project by looking up keywords. The keywords searchable in the PanelWhiz system are those words found in the item labels of the variables. The user first clicks on the starting letter of a word, ie. “w” for the keyword “wages”. Then a page containing all keywords starting with the letter “w” is opened. The keyword browse page contains a table of contents of all found keywords at the beginning. The user can click on a keyword, and automatically jump to the position of the keyword in the page or jump back to the beginning. Alternatively, the user can scroll down to a keyword. To add an item to the data project, the user simply clicks on the item name.

4.2 Project Page

In the previous section, the user learned how he/she can directly open a data file and add items. Additionally, the user can simply append other PanelWhiz projects. Data projects can be named, renamed, saved, resaved, deleted etc. The saved PanelWhiz projects are saved as ASCII text files with the filename ending “`.soep`” for SOEP projects, or “`.hilda`” for HILDA projects and so on. If one saves projects in a modular fashion, one can create quickly more complicated projects, e.g. `wages.soep`, `firm.soep`, and `humancapital.soep` get appended together to create `labor.soep`.

Assuming that a project is complete (the user has found all the items of interest), then by clicking on a button, the retrieval can be executed. Taking the HILDA now for an example, the Project Page of an example data project is displayed in Screen Shot 4. The user can see the item-correspondences. When for instance, the item “`csany`” is retrieved, automatically the variables `acsany` though `dcsany` are pulled from the corresponding data sets.

Screen Shot 4: Project Page with View Option "Medium"

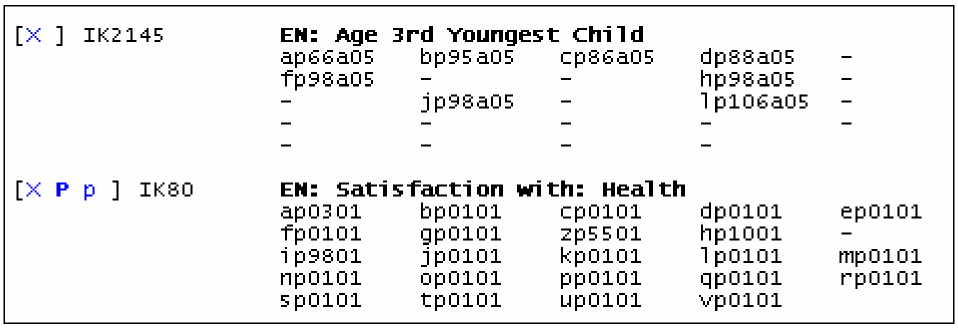


```
Startpage > Project Page                                                              <Edit Project>

                   HILDA Project Browse Page
                   PanelWhiz HILDA (1.0) Aug 2006 <john@panelwhiz.eu>

Project:           [nobel_prize] from <13 Sep 2006> at <13:55:48>
                   <Untitled>
                   <John Haisken-DeNew> at <jhaiskendenew@rwi-essen.de>

Getting Started:   [ Start Page ]

Local Projects:    [ New | Open | Acquire | Prefs | Save, As | Save | Append ]

Libraries:         [ Open (WEB) | Open (LOCAL) | Append (WEB) | Append (LOCAL) ]

Undo/Redo:         [ Undo | Redo | Erase History ]

View Options:      [ low | med | high | Refresh ]

Retrieval:         [ Do Retrieval NOW! ]

    Prefs

Panel Level        < Person >
Gender             < All >
Panel Type         < Unbalanced Respondents >
-Keep Logic        < OR >
-Keep Value        < 1 >
-Keep Condition    < EQ >
Gender             < All >
Wave Begin         < a >
Wave End           < d >

    CS: Child care during school

[X] csany          Whether have children at school
                   acsany     bcsany     ccsany     dcsany

[X] csno           Number of school-aged children
                   acsno      bcsno      ccsno      dcsno

[X] csctc          Childcare total cost ($) for all school age children during term, across all
                   types of care. While parents work
                   acsctc     bcsctc     ccsctc     dcsctc

[X] csu_me         Childcare used - Me or my partner. Any school-age child. While parents work
                   acsu_me    -          ccsu_me    dcsu_me

[X] csu_bs         Childcare used - The childs brother or sister. Any school-age child. While
                   parents work
                   acsu_bs    bcsu_bs    ccsu_bs    dcsu_bs
```

The project page allows the user to edit the data project and refine its contents. For instance, a previously selected item can be removed at this stage simply with a click. The [X] associated with every item in the Project Page allows the user to remove items already in the project.

Furthermore, the user can obtain detailed information as to which variables in which original data files will actually be retrieved. With the "Item View Option" at "low", only selected item names are displayed. With view option "medium", the item names and their item-correspondences are displayed, i.e. the vector of variables associated with the item. With view option "high", one has the output of "medium" plus a listing of each original file and the associated.

Screen Shot 5: Project Page with View Option "Medium"



```
[X ] IK2145         EN: Age 3rd Youngest Child
                    ap66a05    bp95a05    cp86a05    dp88a05    -
                    fp98a05    -          -          hp98a05    -
                    -          jp98a05    -          lp106a05   -
                    -          -          -          -          -
                    -          -          -          -

[X P p ] IK80       EN: Satisfaction with: Health
                    ap0301     bp0101     cp0101     dp0101     ep0101
                    fp0101     gp0101     zp5501     hp1001     -
                    ip9801     jp0101     kp0101     lp0101     mp0101
                    np0101     op0101     pp0101     qp0101     rp0101
                    sp0101     tp0101     up0101     vp0101
```

In this example using the SOEP, there are two items selected, `IK2145` and `IK80`. The hyphens in the variable indicate that for that particular year, there is indeed no variable which corresponds to item. Beside the item names, e.g. `IK80`, the user can see that there are several actions possible: "`[X P p]`". The `X` indicates deleting the item from the data project. The `P` indicates deleting only the associated plugin from the retrieval, but not the item itself. The lowercase `p` indicates being able to view the exact contents of the associated plugin.

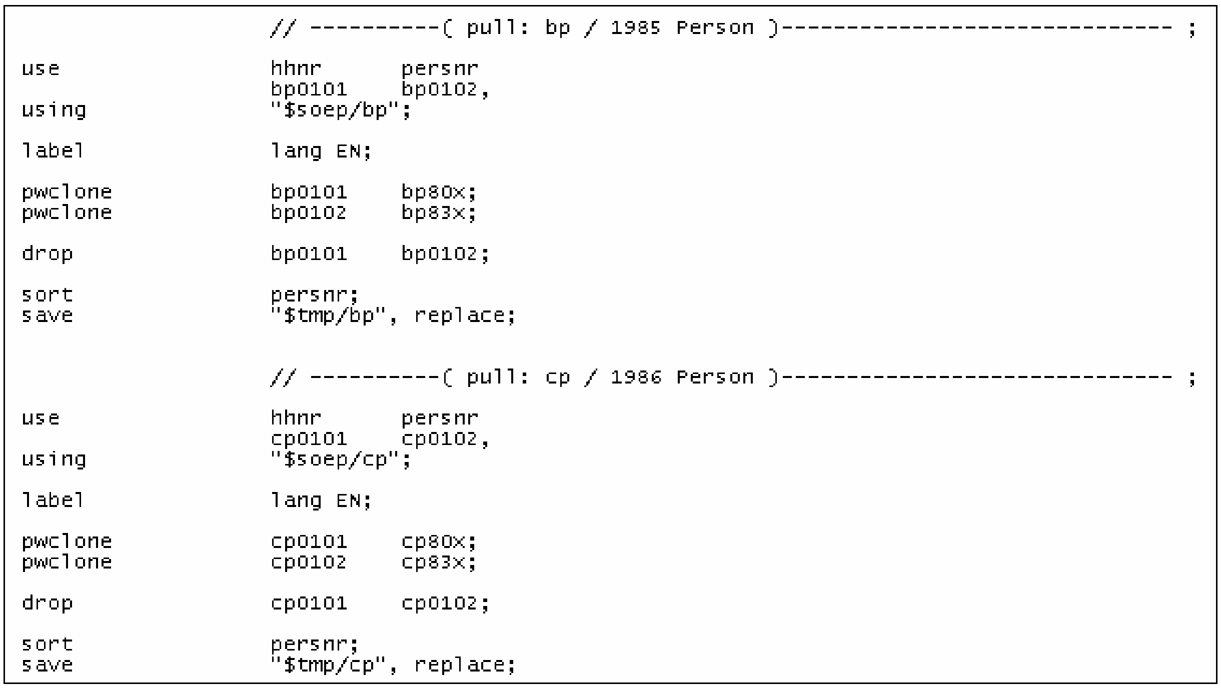
4.2 Retrieval DO File Generator and Custom Variable Names

As a user of panel data, one would eventually like the micro data to be in "long" form. PanelWhiz standardizes variable names over time. For instance in PanelWhiz SOEP, variables have been given a serial number. Thus wide variables from the item number `80` would be `ap80x`, `bp80x`, ..., `vp80x`. The first letter indicates the wave `[a-v]`, the second letter `[p,h]` indicates whether at the person or household level, and "`x`" at the end indicates a PanelWhiz variable. Thus after the reshape command has be executed, the first letter indicating the wave is dropped to create the long variable name for example, `p80x`. The user can access the retrieved data at the "wide", "long" and "custom" level. In the Australian HILDA, all item names are speaking names, already defined by the HILDA group in Melbourne.

As mentioned, PanelWhiz requires a standardized naming scheme to maintain longitudinal consistency (i.e. reshape from wide to long format). In the case of the SOEP, this creates variable names of the following sort: `p80x`. This is perhaps not so intuitive however PanelWhiz allows users to have the option of customizing the system to rename PanelWhiz variables to other arbitrary names. Thus, whenever the PanelWhiz variable `p80x` is retrieved in a "long" file, the system automatically creates an additional "custom" file in which PanelWhiz variables have been renamed to something perhaps more intuitive such as "`healthsat`" or "`sath`". The renaming feature is only activated if the user explicitly requests it. The user can provide custom names for variables extracted. Should a variable be found in the list of variables to be renamed, the variable is renamed according to the instructions of the user. If a PanelWhiz variable is not found in the custom list, then the variable name simply remains as it is. The exact renaming executed in the retrieval is documented in the generated retrieval DO file. Thus, there is unambiguous information as to the origins of each and every variable.

The DO file depends directly on the user's selected data project preferences and is adjusted accordingly, e.g. "household" vs. "person" level retrievals, or "just men" vs "just women" retrievals or "balanced" vs. "unbalanced" retrievals etc. The DO file allows the user to document exactly which data has been extracted from which files. This is important for scientific data documentation, and for prospective journal editors, requiring exact replicable results.

Screen Shot 6: Excerpt from DO File Generator



```
                   // ----------( pull: bp / 1985 Person )------------------------------ ;

use                hhnr      persnr
                   bp0101    bp0102,
using              "$soep/bp";

label              lang EN;

pwclone            bp0101    bp80x;
pwclone            bp0102    bp83x;

drop               bp0101    bp0102;

sort               persnr;
save               "$tmp/bp", replace;


                   // ----------( pull: cp / 1986 Person )------------------------------ ;

use                hhnr      persnr
                   cp0101    cp0102,
using              "$soep/cp";

label              lang EN;

pwclone            cp0101    cp80x;
pwclone            cp0102    cp83x;

drop               cp0101    cp0102;

sort               persnr;
save               "$tmp/cp", replace;
```

4.3 Plugins

Using panel data sets, one immediately is confronted with the fact that values of variables in any particular item may change over time. For instance if 1=Yes and 2=No in 1984, there is no particular reason to assume this will remain for all years following. The following year may have 1=Yes 2=Maybe 3=No, being problematic for long data sets. Hence, using “plugins”, allows the user to "clean up" these value inconsistencies.

A plugin is simply an add-on in its own right with the same name as the long variable name, e.g. in PanelWhiz SOEP `p1234x.ado`, and is executed whenever that item is selected in the retrieval. If desired, these plugins can be added, removed, reloaded, etc. For instance, one can use a plugin to deflate income measures with a price index automatically. Theoretically, there should be a plugin for each item, which would guarantee that all values would be consistent over time. In fact, if there is no plugin available for an item, value labels if present are stripped from the long variable to ensure that the user examines the item consistency over time. PanelWhiz users may write and contribute themselves to the pool of PanelWhiz plugins.

If the user indeed decides to incorporate plugins into the data retrieval, he must make a conscious decision to include the plugins into the data project. Then each plugin author must be cited in any and all documents or presentations derived from the underlying data in which the plugins were used. The user does not have to keep track of the plugin authors himself – this is done by PanelWhiz automatically. The user simply has to copy a generated citation text into his paper text.

The idea behind plugins is that embedded knowledge about the data set can be distributed very efficiently using the PanelWhiz system. If a particular item has been cleaned by one person and this information is embedded in a plugin, it can be distributed freely to all users of the PanelWhiz system, without contravening any data protection laws or regulations. Clearly there is a potential free-rider issue here, as it is effectively costless to use other people’s plugins, but costly to write oneself plugins to share with others. Hopefully, researchers will see the true benefits of sharing their knowledge. The PanelWhiz user contract requires that if a plugin is used in a retrieval and it was not written by the user himself, then the user is required to cite using the intellectual property of the author of the plugin. Not citing the use of other people's ideas is simply unfair and undermines the basis of PanelWhiz.

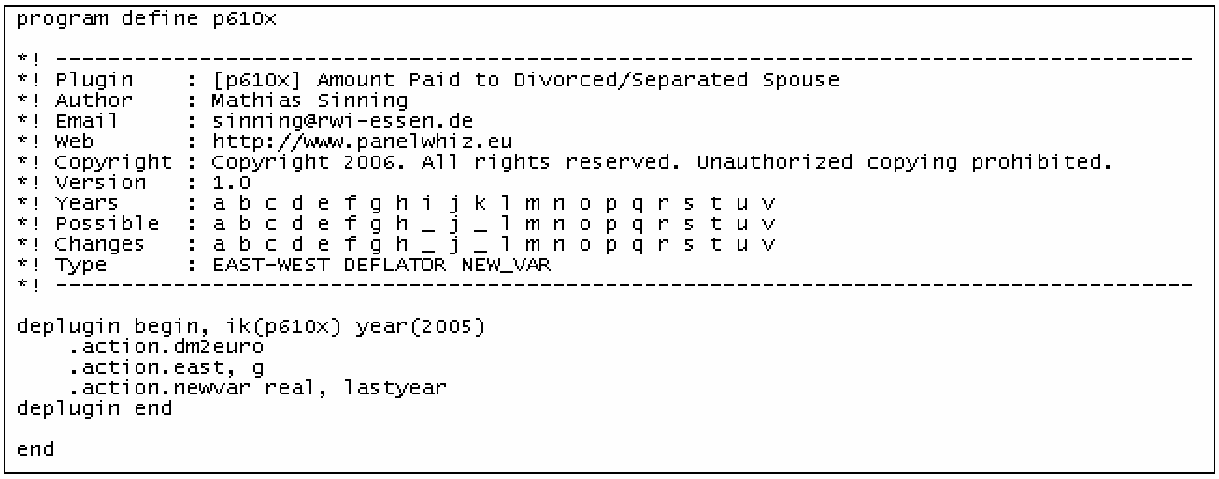
The old SOEP Menu plugin interface was completely rewritten by Markus Hahn in summer 2006 and integrated into the PanelWhiz system. The new method differs significantly from the old in that Stata's newly introduced `class` system is implemented; allowing a much easier and more standardized coding of plugins.

In conjunction with this new plugin interface, a series of generated plugins is provided for PanelWhiz SOEP. Here, a Stata program was written to check the technical time consistency of every item containing variables with value labels in the SOEP item-correspondence list. In so doing, the program compares for each year and outcome value, the contents of the value labels. If for all values and in every variable in a particular item there are no changes, then a "dummy" plugin is created indicating the apparent time consistency. This indicates to the user that the variable can be used immediately. However there are some drawbacks to this methodology. One can only compare exact matches of value labels over time. Small changes or abbreviations of course cannot automatically be recognized. An example plugin is shown in Screen Shot 7.

There are various standard operations within a plugin that are already supported. These are:

(a) recoding values
(b) deflating nominal to real
(c) converting old Deutsche Mark (DM) values to Euro (€)
(d) joining separate East and West German variables in the years 1990/91

Screen Shot 7: Example SOEP Plugin



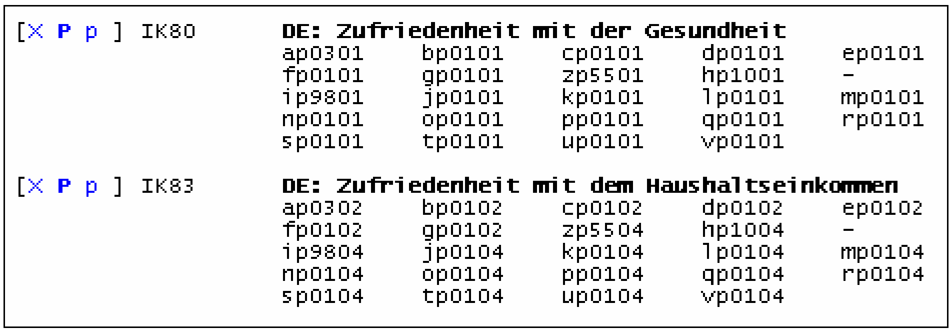
```
program define p610x

*! ------------------------------------------------------------------------------
*! Plugin    : [p610x] Amount Paid to Divorced/Separated Spouse
*! Author    : Mathias Sinning
*! Email     : sinning@rwi-essen.de
*! Web       : http://www.panelwhiz.eu
*! Copyright : Copyright 2006. All rights reserved. Unauthorized copying prohibited.
*! Version   : 1.0
*! Years     : a b c d e f g h i j k l m n o p q r s t u v
*! Possible  : a b c d e f g h _ j _ l m n o p q r s t u v
*! Changes   : a b c d e f g h _ j _ l m n o p q r s t u v
*! Type      : EAST-WEST DEFLATOR NEW_VAR
*! ------------------------------------------------------------------------------

deplugin begin, ik(p610x) year(2005)
    .action.dm2euro
    .action.east, g
    .action.newvar real, lastyear
deplugin end

end
```

However, this system is flexible to allow other additional functions and operations. Currently only PanelWhiz SOEP contains plugins. However, the plugin interface will be expanded to all other data sets within the PanelWhiz system. Plugins in general are only valid for a given distribution year. Variables can change over time and also retroactively be modified by data providers. Therefore in each plugin, an automatic check is made to ensure that the plugins being executed during the data retrieval are all appropriate for that particular data distribution.

4.4 Label Language

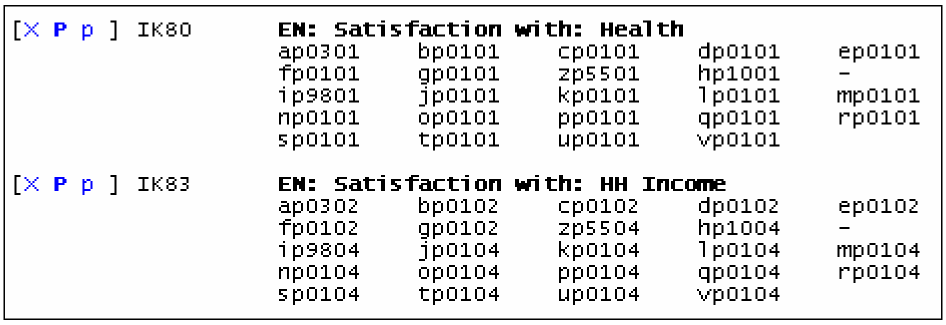
The Stata files used with PanelWhiz are preloaded with all available meta information. In the case of the SOEP, the documentation is available in both German and English. The user can at any time switch between available languages.

Screen Shot 8: Project Page with Label Language in German



```
[X P p ] IK80        DE: Zufriedenheit mit der Gesundheit
                     ap0301    bp0101    cp0101    dp0101    ep0101
                     fp0101    gp0101    zp5501    hp1001    -
                     ip9801    jp0101    kp0101    lp0101    mp0101
                     np0101    op0101    pp0101    qp0101    rp0101
                     sp0101    tp0101    up0101    vp0101

[X P p ] IK83        DE: Zufriedenheit mit dem Haushaltseinkommen
                     ap0302    bp0102    cp0102    dp0102    ep0102
                     fp0102    gp0102    zp5504    hp1004    -
                     ip9804    jp0104    kp0104    lp0104    mp0104
                     np0104    op0104    pp0104    qp0104    rp0104
                     sp0104    tp0104    up0104    vp0104
```

Screen Shot 9: Project Page with Label Language in English



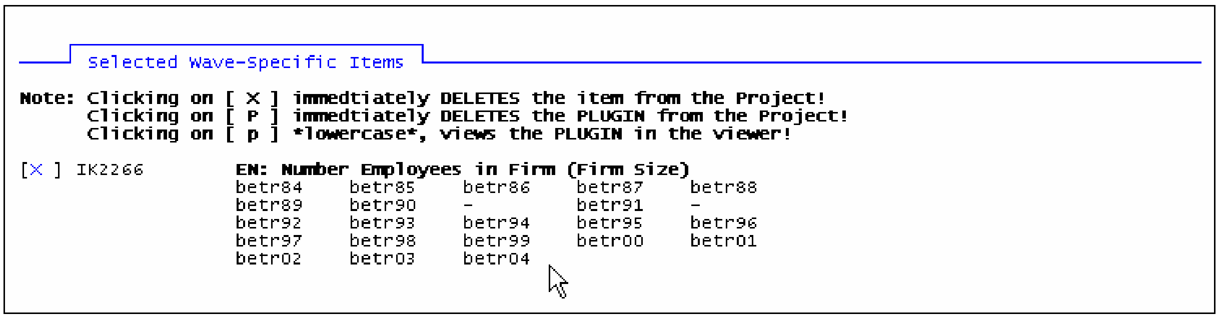
```
[X P p ] IK80        EN: Satisfaction with: Health
                     ap0301    bp0101    cp0101    dp0101    ep0101
                     fp0101    gp0101    zp5501    hp1001    -
                     ip9801    jp0101    kp0101    lp0101    mp0101
                     np0101    op0101    pp0101    qp0101    rp0101
                     sp0101    tp0101    up0101    vp0101

[X P p ] IK83        EN: Satisfaction with: HH Income
                     ap0302    bp0102    cp0102    dp0102    ep0102
                     fp0102    gp0102    zp5504    hp1004    -
                     ip9804    jp0104    kp0104    lp0104    mp0104
                     np0104    op0104    pp0104    qp0104    rp0104
                     sp0104    tp0104    up0104    vp0104
```

In addition to changing the language of the data project, one can also change the language of the data set labels in memory. For instance, in data sets like the SOEP that support at least two languages, one can switch instantaneously from English to German and reverse. Using Stata's `label lang` command, all data labels are switched immediately.

4.5 Automatically Updating Projects / Changing Data Project Languages

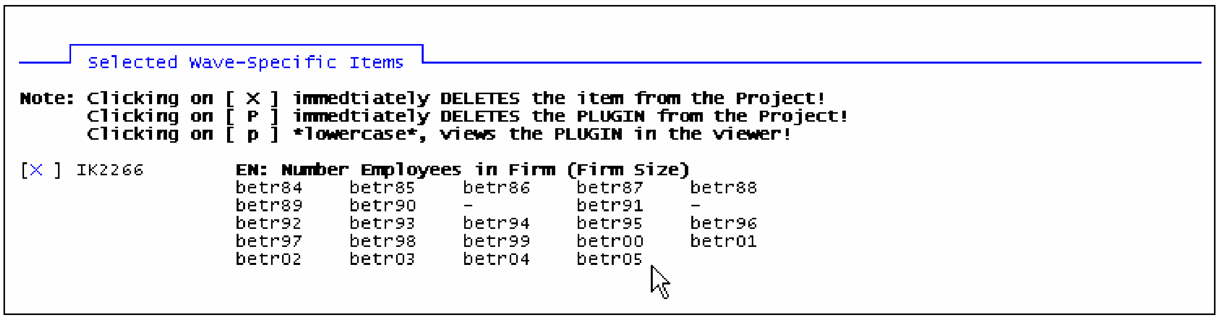
Typically once a year, new waves from the household surveys become available. The user then would have the task of completely renewing any data projects, because they would not longer contain current information, i.e. the information from the most current wave would be missing. PanelWhiz allows the user to update any data projects to the most recent information automatically. This saves the user from having to update the data projects manually or from having to recreate old projects. The work is done once and then automatically updated. The following Screen Shots illustrate this principle. This same procedure can be used to change languages in the SOEP from German to English or the reverse. Additionally it can be used to reconstruct the data project.

Screen Shot 10: Project Page Excerpt – before – Automatic Update



```
Selected Wave-Specific Items

Note: Clicking on [ X ] immedtiately DELETES the item from the Project!
      Clicking on [ P ] immedtiately DELETES the PLUGIN from the Project!
      Clicking on [ p ] *lowercase*, views the PLUGIN in the viewer!

[X ] IK2266        EN: Number Employees in Firm (Firm Size)
                   betr84    betr85    betr86    betr87    betr88
                   betr89    betr90    -         betr91    -
                   betr92    betr93    betr94    betr95    betr96
                   betr97    betr98    betr99    betr00    betr01
                   betr02    betr03    betr04
```

Screen Shot 11: Project Page Excerpt – after – Automatic Update



```
Selected Wave-Specific Items

Note: Clicking on [ X ] immedtiately DELETES the item from the Project!
      Clicking on [ P ] immedtiately DELETES the PLUGIN from the Project!
      Clicking on [ p ] *lowercase*, views the PLUGIN in the viewer!

[X ] IK2266        EN: Number Employees in Firm (Firm Size)
                   betr84    betr85    betr86    betr87    betr88
                   betr89    betr90    -         betr91    -
                   betr92    betr93    betr94    betr95    betr96
                   betr97    betr98    betr99    betr00    betr01
                   betr02    betr03    betr04    betr05
```

4.6 Exporting Data

If for some reason, the data just created should be exported into another format, one can use several commercial tools such as StatTransfer® or DBMSCOPY®, or several tools made available here. PanelWhiz allows exporting data to SPSS®, SAS®, LIMDEP®, GAUSS®, MS Excel®. If the data is in memory, then it can be exported. Where variable labels and value labels have a relevant meaning in the exported format, they are kept. For instance, LIMDEP and GAUSS do not have a concept of value labels and MS Excel does not have a concept of neither value nor variable labels.

4.7 Acquire, Undo and Redo

Data sets created using PanelWhiz have the added feature that the entire data project information has additionally been stored in the finished Stata data set `*.dta` itself. This is achieved using the Stata char command allowing the dataset to be injected with the same information found in the `*.soep` or `*.hilda` data project files. This allows the user to reproduce at all times the data project used to create that particular data set. In addition, the data signature using the new Stata 9.2 command `datasignature` is also injected into the data set, allowing the user complete information as to the content and origin of the data in the Stata data set. This additional security feature allows the user a "last chance", should the data set itself be separated from the data project and generated data documentation. The "Project Acquire" command can be activated in the Project drop down menu, and a Stata data file that was created by PanelWhiz is thereby read in, immediately importing the underlying data project directly into PanelWhiz.

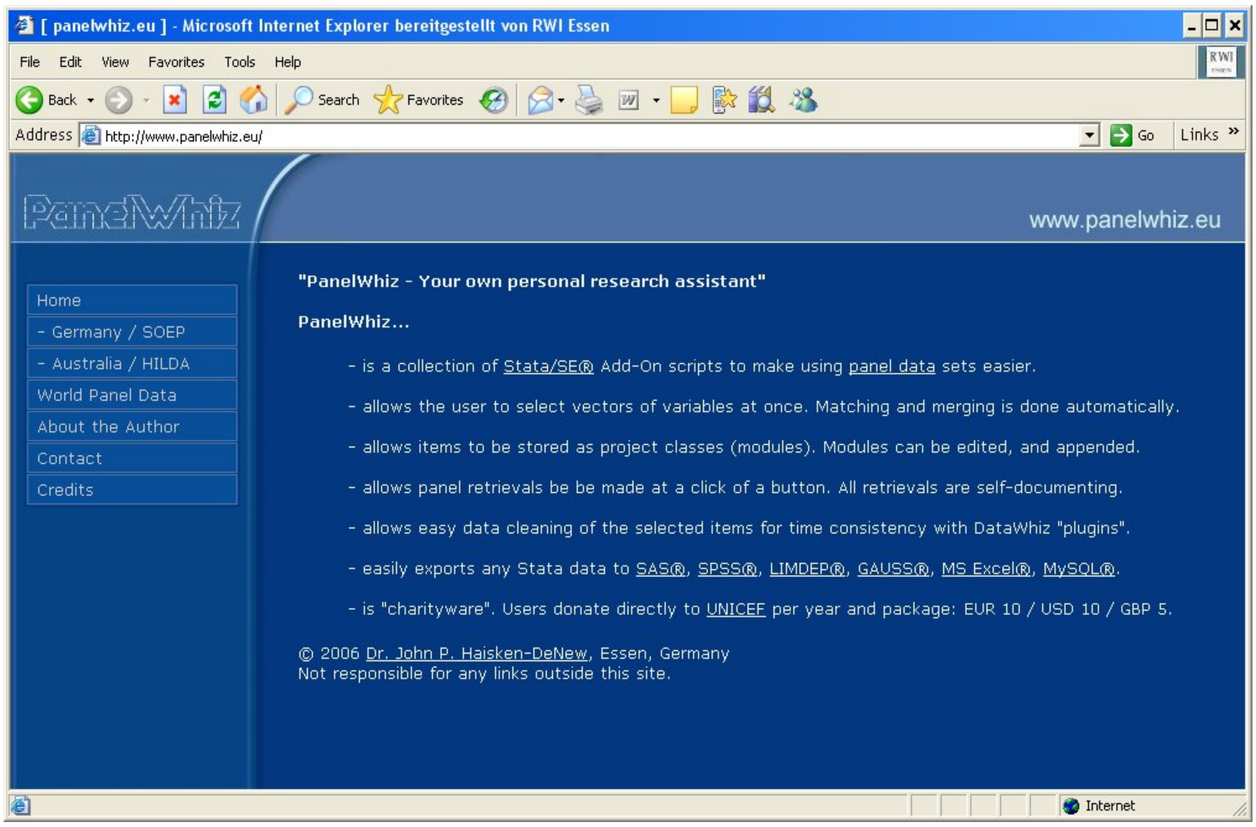
All steps or actions that change the contents of the data project (including preferences) are documented step by step, and as such, may be undone or redone. The user himself decides the number of times that can be undone or redone. Essentially, a global variable is created for each step and stored in global space, containing the entire contents of the data project. This is completely independent of any data sets in memory at the time.

Upon saving a data project, the most current version is saved along with a backup of the previous version. This is done automatically. This allows the user a final chance to recover any work potentially lost due to unwanted changes or data errors on his hard disk.

4.8 Internet Support

The PanelWhiz system is supported by the homepage at `http://www.PanelWhiz.eu`. Here detailed documentation is available. See Screen Shot 12. All PanelWhiz supported data sets are listed and described here.

Screen Shot 12: PanelWhiz Homepage



"PanelWhiz - Your own personal research assistant"

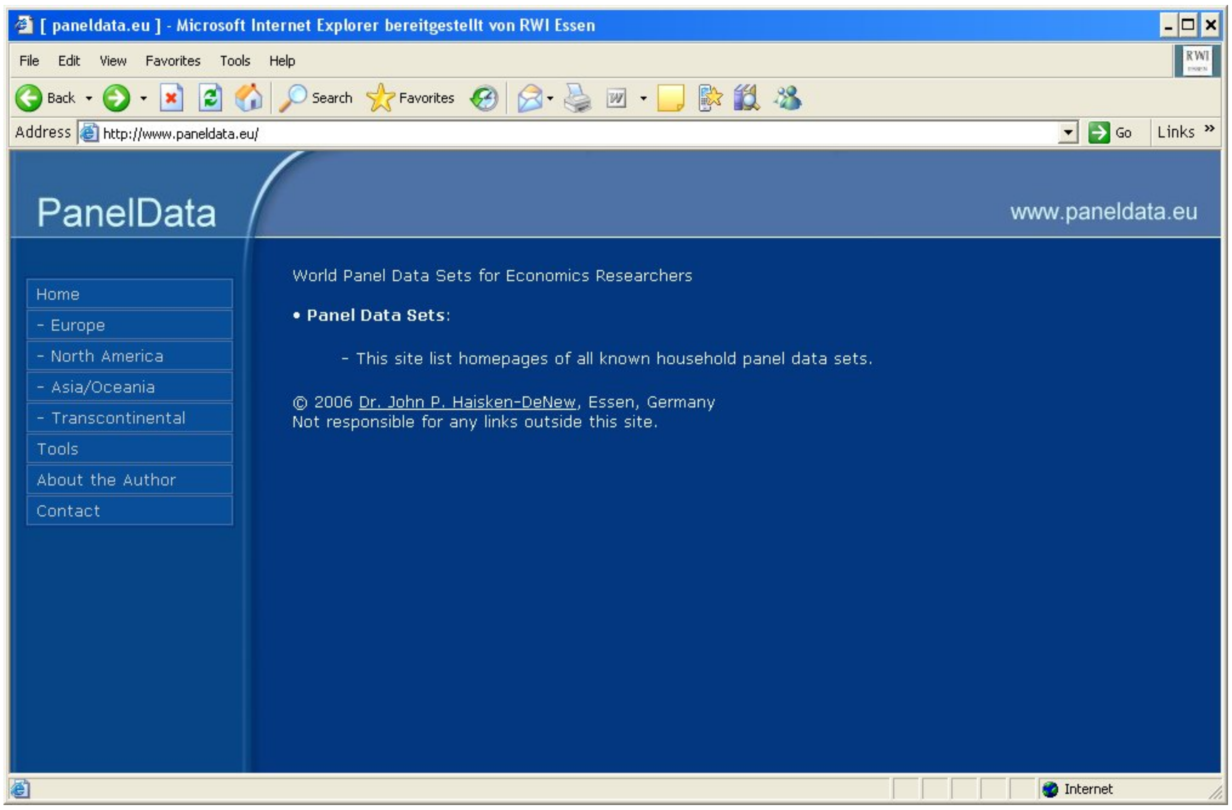
PanelWhiz...

- is a collection of Stata/SE® Add-On scripts to make using panel data sets easier.
- allows the user to select vectors of variables at once. Matching and merging is done automatically.
- allows items to be stored as project classes (modules). Modules can be edited, and appended.
- allows panel retrievals be be made at a click of a button. All retrievals are self-documenting.
- allows easy data cleaning of the selected items for time consistency with DataWhiz "plugins".
- easily exports any Stata data to SAS®, SPSS®, LIMDEP®, GAUSS®, MS Excel®, MySQL®.
- is "charityware". Users donate directly to UNICEF per year and package: EUR 10 / USD 10 / GBP 5.

© 2006 Dr. John P. Haisken-DeNew, Essen, Germany
Not responsible for any links outside this site.

Also available on the homepage `http://www.PanelData.eu` is a list of all known relevant panel datasets in the world. See Screen Shot 13. This site allows the user to peruse a listing of panel data sets, sorted by geographical location.

Screen Shot 13: World Panel Data Homepage


[paneldata.eu] - Microsoft Internet Explorer bereitgestellt von RWI Essen
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.paneldata.eu/ Go Links
PanelData
www.paneldata.eu
Home
- Europe
- North America
- Asia/Oceania
- Transcontinental
Tools
About the Author
Contact
Internet

World Panel Data Sets for Economics Researchers

- **Panel Data Sets:**

  - This site list homepages of all known household panel data sets.

© 2006 Dr. John P. Haisken-DeNew, Essen, Germany
Not responsible for any links outside this site.

5. Installation and Maintenance

The PanelWhiz system comprises of two basic parts: (1) the PanelWhiz distribution and (2) the data pre-loaded with the item-correspondence.

The first step is to install the PanelWhiz system for the data packages that the user has subscribed to. Installation is carried out over the internet with the Stata Add-on `pwnetinstall.ado`, such that all required files are copied to the user's hard disk from the PanelWhiz homepage. As such, the installation of PanelWhiz requires access to the internet. As updates of PanelWhiz become available online, the user can simply install the newer files by clicking on a link when prompted. Additional "Plugin" files will become available as users contribute to the ever growing knowledge base.

PanelWhiz is the largest user written package for Stata, whether based on number of Add-ons or volume or number of files. As such it has been imperative to minimize the update volume required over the internet and at the same time, the update time required. PanelWhiz uses a strategy of breaking PanelWhiz into its broader component parts, and archiving them separately. Then when an update is required, a component part is downloaded (in archived form to reduce bandwidth usage) and then unpacked on the user's computer, thus maximizing update speed. This is done transparently for the user in the background. See Hahn (2006) for details.

6. Summary

PanelWhiz is a data retrieval tool that eases data extractions from the many large scale data sets such as the German SOEP or the Australian HILDA. PanelWhiz is directly combined into Stata SE, allowing a seamless interaction between the micro data and the statistics package. Vectors of variables, called item-correspondences can be selected all at once. Special cleaning programs written in Stata called "plugins" can clean a particular item-correspondence and make it time and/or content consistent. Groups of item-correspondences can be stored as projects. Groups of projects can be stored as libraries. This method of organizing the projects and plugins allows for a modular administration, facilitating knowledge transfer and group work. Data can be retrieved by mouse-click, providing rectangularized data in wide and long format. As new releases of the micro data become available, the user can "automatically" update his projects to include the latest wave of information. All programs used are available in source Stata code which allows complete transparency of content. All commands used in the generated retrieval are documented in a full functional retrieval DO file, capable of recreating the identical retrieval at any time.

References

Hahn, Markus (2006) "Optimal Large Package Administration for Stata", Stata Users Conference 2006, Mannheim, Germany.

Haisken-DeNew, John P. (2001) "A Hitchhiker's Guide to the World's Household Panel Data Sets." *The Australian Economic Review*. 34(3), 356-366.

Haisken-DeNew, John P. (2005) "SOEP Menu: A Menu-Driven Stata/SE Interface to the German Socio-Economic Panel", http://www.soepmenu.de, mimeo.

Haisken-DeNew, John P. and Joachim R. Frick (2005) "The Desktop Companion to the German Socio-Economic Panel Study", DIW Berlin, Germany

Technical Appendix 1: Legal Considerations

This product is provided only "as is". The user alone must decide whether use of this product is appropriate. No warrantees or guarantees implied or intended. The author acts alone and on his own behalf. The author is not responsible for loss of data, and any resulting damages due to use of this product. Current or past employers of the author are not responsible for this product, loss of data, and any resulting damages.

PanelWhiz was written by the author and sole copyright holder Dr. John Haisken-DeNew. Markus Hahn has also been a significant contributor of subroutines to this project.

Stata, SPSS, SAS, Excel, DBMSCOPY, StatTransfer, UNIX, LIMDEP, are trademarks of their respective owners and are mentioned here for informational purposes only. There is no business relationship between the author of PanelWhiz and these trademark holders.

The micro data sets used in PanelWhiz are owned by their respective data providers, who alone control access to the micro data. The author of PanelWhiz never distributes protected micro data. Please beware that in your country of residence that data misuse offences may be punishable with severe criminal and civil penalties. PanelWhiz users are required to make sure they comply with all applicable laws and regulations while using or storing the micro data.