Appendix V

Data-Cleaning Process

Included items:

1. Data-Cleaning Process
2. Brief Description of Main Data-Cleaning Programs

Blank pages inserted to preserve pagination when printing double-sided copies.

Data-Cleaning Process

Raw data records extracted from the district courts' docketing databases were transmitted electronically to the FJC. We assessed the integrity and completeness of the transmitted data and converted the individual pieces into a reconstructed set of docketed event records ready for final processing. The data-cleaning process and programs are described below. The event-processing programs are described in Appendix W.

Initial Processing of Raw Data

We developed a set of SAS programs to load and process the raw data received from each court. Because of substantial structural differences in the data extracted from the ICMS and CM/ECF databases, different programs—that were nonetheless functionally equivalent—were written to handle the files from each system. During the initial processing of raw data, we maintained and processed the data from each court separately. We used this approach both for data-management reasons and to increase processing efficiency. We developed database-driven macro programs to manage the required multiple program executions.

The SAS processing programs checked for a full range of data-integrity problems such as: (1) data-type errors in data fields (e.g., alpha characters in numeric fields); (2) unusual or out-of-range values; (3) adherence to the selection criteria (e.g., termination, or re-termination, date within calendar 2002); and (4) basic interrelationships among the case components (e.g., checking that party and event records matched to a case record).

We reviewed processing logs and field frequency reports to identify problem areas that might require additional review or data cleaning (e.g., deletion of duplicate case records), and then attended to the data-processing problems that arose. Data cleaning focused primarily on exception reporting. The information received was assumed to be correct and complete unless an unexpected or impossible outcome was detected. For example, a case record that had no matching event records is not necessarily an error and such cases were included in the analysis. Event records that could not be matched to a valid case record, however, were excluded. Because these two situations should occur rarely, incidence levels exceeding a nominal threshold in a court triggered a more detailed investigation of the data.

We additionally created reference data tables by pulling civil, criminal, and trial data records for each court from the FJC's Integrated Database.[1] We used

1. The Federal Judicial Center's Integrated Database contains basic descriptive and processing information for civil cases, criminal defendants, and trials and evidentiary hearings. The data are based on information received from the Administrative Office on case filings, terminations, and proceedings in the federal courts.

these tables to validate the population of cases received from the courts and as a source of additional case information.

As part of the preliminary data processing step, we created unique identifiers for each case and record. Identifiers permitted us to match case information from different data sources, link related case components together, establish a docketing sequence, and identify duplicate records created during processing. We also created a series of case flags and constructed specific data fields that assisted in the characterization of cases (e.g., the number of civil parties, the number of co-defendants). Flags also identified the set of "good cases"—that is, cases that met the defined case-selection criteria (e.g., "cv" or "cr" docket type, single-defendant ICMS criminal case, etc.)—that would be used in the final analyses.

Brief Description of Main Data-Cleaning Programs

Processing Task	SAS Programs
• Processed data extracted from ICMS courts. • Loaded raw court data and computed basic frequencies. • Created case ID codes for matching across data sets. • Created unique sequence IDs for identifying duplicate records created during joins. • Pulled comparison civil and criminal data from the IDB. • Pulled relevant trial data from the IDB. • Checked for basic status and error situations (e.g., checked that there were no duplicate records in the case data, that the termdates were within 2002 or, if not, that the reterm dates were). • Compared civil and criminal lists to the IDB to verify we had received the cases we expected. • Looked for internal consistencies (e.g., cases with no parties, parties with no cases, cases with no events, etc.). *Additional data-processing and analysis tasks also performed by this program are described in Appendix W.*	**xtract_processing_ICMS** version: 1.7 date: 17-Mar-2004 principal source data files: *asccases, cases, dplink1, dplink2, events, js2, judge, party, reliefs, who* principal output files used in further processing: *caseflgs* *evntrlfjn*
• Used macro shell to process all CM/ECF courts. • Retrieved parameters from dccws summary data set. • Processed data extracted from CM/ECF courts. • Loaded raw data and computed very basic frequencies.	**xtract_processing_CMECF** version: 1.3 date: 10-Mar-2004 principal source data files: *asccases, asclead, ascmember, ecfcaseflgs, cases, codes, dktntry, dktpart, dktperson, doctype, filer, js23, js56, judge, motion, party*

Brief Description of Main Data-Cleaning Programs

Processing Task	SAS Programs
• Created case ID codes for matching across data sets. • Created unique sequence IDs for identifying duplicate records created during joins. • Pulled comparison civil and criminal data from the IDB. • Pulled relevant trial data from the IDB. • Checked for basic status and error situations (e.g., checked that there were no duplicate records in the case data, that the termdates were within 2002 or, if not, that the reterm dates were). • Compared civil and criminal lists to the IDB to verify we had received the cases we expected. • Looked for internal consistencies (e.g., cases with no parties, parties with no cases, cases with no events, etc.).	principal output files used in further processing: all input files
• Created and set a new flag in the *caseflgs* table to identify cases that would be included in the final analysis (i.e., "good cases"). • Identified "good cases" as those with a "cr" or "cv" docket_type, and that could be matched to a case characteristic record that provided sufficient information to compute a DCCWS case type. • For ICMS criminal records, limited "good cases" to single-defendant cases. • For CM/ECF criminal records, processed all defendants individually, but excluded master case records.	**add_goodcase_flg (ICMS)** version: 1.1 date: 21-Apr-2004 **add_goodcase_flg_CMECF** version: 1.0 date: 27-Apr-2004 principal source data files: *caseflgs* principal output files used in further processing: *caseflgs*