

Appendix U

Data-Extraction Materials

Included items:

1. Data-Extraction Process
2. Sample Letter to Chief District Judges (ICMS Courts)
3. Letter to the Clerks of the United States District Courts (ICMS Courts)
4. Instructions for Executing the DCCWS_EXTRACT Program (ICMS Courts)
5. DCCWS_EXTRACT Program (ICMS Courts)
6. DCCWS_ECFXTR Program (CM/ECF Courts)
7. Example .SQL Program Generated by DCCWS_ECFXTR (CM/ECF Courts)

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

Data-Extraction Process

The design of the event-based case-weighting study called for frequency counts to be based on the information routinely recorded in the courts' case-management databases. Standard reporting integrated into the databases did not provide detailed information on all the events required for the study, so it was necessary for us to develop specialized case-extraction programs that would be executed by each court.¹ Event incidence for nontrial events (exclusive of revocation hearings) thus was based on case data pulled from the docketing systems of the courts.

Extraction from ICMS Courts

We reviewed documentation provided by the AO depicting the formal schema of the ICMS database to identify the data records and fields we would need to extract. The required data elements were those that provided (1) basic identification and characterization of each case or defendant; (2) an accounting of all events that were docketed in each case and associated relief and order-action information; and (3) the identification of parties and judges participating in the case with codes linking them to specific events. In each court, the data records were extracted directly from the individual database tables and written out to ten separate data files. These direct extractions placed less of a processing burden on the courts' systems, but they necessitated that the separate elements be merged later during processing at the FJC.

FJC project staff wrote the extraction program and styled it after model programs provided by the Administrative Office and members of the Technical Advisory Group. Responding to concerns the Technical Advisory Group raised about memory and space limitations in some courts, we incorporated into the program a mechanism to allow the courts to control what portion of their data would be ex-

1. We had little information from which to assess the burden associated with the extraction task. This made it difficult to predict the courts' response to the data-extraction request. Consequently, anticipating the possibility that court extractions might fail to produce a large, representative data sample, we sought to identify an alternative source for event incidence information so we could supplement the information provided through the data extractions, if necessary. Our investigations led us to construct an alternate database of docket events from electronic docket sheets available through PACER. We began downloading and processing the dockets for civil and criminal cases terminated in calendar 2002. These files had all the information needed for the study but the text format was not directly usable in a case-weighting analysis. To translate text information into usable data, we developed processing programs that we used to parse the information from PACER into defined data fields. We used analysis programs to perform rough categorizations of events through the use of keyword searches, but more refined interpretations required additional manual processing. Docket sheets from more than 40 district courts were processed before it became apparent from the courts' overwhelmingly positive response to the data-extraction request that sufficient information would be available from the primary data source. At that point, we suspended efforts to process the PACER docket sheets, but subsequently used the parsed docketing data for other purposes. We used the data to (1) investigate the feasibility of specific, alternative case categorizations we thought might be discussed at the National Consensus Meeting; (2) verify the correct processing of event sequences; and (3) help interpret unknown event codes.

tracted at one time. By defining date parameters processed at execution, a court could choose to do multiple smaller runs rather than one full run, thereby limiting the impact of the extraction on the system. Other processing parameters uniquely identified the output data files and allowed the court to direct the output to an appropriate location on disk.

Technical Advisory Group members reviewed the code and tested preliminary versions of the program. The Administrative Office provided access to a test ICMS database that allowed us to incorporate refinements into the program and conduct final testing. Two district courts, one with a database built on the Arizona Training Center model and the other built on the Texas Training Center model, performed a pilot test of the program and its execution and data-transmission instructions. The pilot test confirmed that the program worked as expected.

Once pilot testing was completed, we sent an e-mail request to the clerks of district courts that used an ICMS database for docketing cases; we identified the appropriate courts through the responses received to the docketing survey. The e-mail message provided background on the case-weighting project and explained the need for docketing data from the court's database. The request included a copy of the program and instructions for system managers on how to set the execution parameters and transfer the output files to the FJC (see copies of the request, program, and instructions included in this appendix). The e-mail also provided a link to an FJC web page that contained downloadable copies of the files (<http://cwn.fjc.dcn/newweb/jnetweb.nsf/pages/511>). We also sent a copy of the request to the court contact listed on the docketing survey if the contact was someone other than the clerk.

Shortly after we sent the e-mail request, we mailed a letter from the chair of the Statistics Subcommittee to chief judges of ICMS districts. The letter informed the chief judges of the data-extraction request and noted the critical role their courts played in the work being done for the Subcommittee. It ended with an expression of gratitude for the district's cooperation.

The initial request for the extraction was sent on December 12, 2003; it appealed to courts to respond by December 31, 2003. Execution of the program required minimal set up by the court. We recognized, however, that the time frame should allow courts to choose when to execute the program so that, if necessary, runs could be done at night or over a weekend to minimize the impact on daytime docketing to databases.

Caseload size and other environmental factors affecting the execution of the program varied greatly among the courts, making it difficult to anticipate how long the database extractions would take to run. Processing information captured during program execution confirmed the variability but also indicated that over 85% of the extractions took less than two hours to complete; many of the extractions took considerably less time.

Sealed cases were not explicitly excluded from the extraction. Responses to the docketing survey, however, indicated that sealed cases were not docketed in

the databases of some ICMS courts. Other courts specifically excluded sealed cases from the data they transmitted to the FJC.

Response to the data request was high. Almost 90% of the ICMS districts sent in data by early January 2004, and, following a quick e-mail reminder to the others, 99% responded by the end of the month. In all, sixty-nine courts sent in data from ICMS systems.

Extraction from Mixed-System Courts

We found it easy to determine which database system was in use for most districts, and thus knew which extraction program to send the court. There were some exceptions, however, that complicated the task. Conversion of ICMS databases to CM/ECF was ongoing during this time. The extraction request was made at the end of 2003, but the docketing data to be extracted was for cases disposed of during calendar 2002, and many of these cases had been filed years earlier. A few courts were in a “mixed” situation—the original docketing had been done using an ICMS database but the court had subsequently converted to CM/ECF.

We sent special letters in mid-December to courts whose database status was ambiguous, asking them to clarify which system the court would use when extracting data. Their answers primarily depended on (1) when the court’s conversion had taken place; (2) whether the ICMS system was still available to the system manager to perform the extractions; and (3) which system contained the most complete set of docketing information for the target cases.

Some of the districts receiving the letter chose to extract from the ICMS database and sent their data in immediately. Others preferred to extract from their CM/ECF database; these courts were then added to the CM/ECF extraction request that was sent in February 2004.

A few courts identified themselves as using both ICMS and CM/ECF. This circumstance represented courts in transition, where, for example, civil docketing had been converted to CM/ECF, but criminal docketing was still being done in ICMS. We sent such courts a special request later, when CM/ECF requests were mailed, and included extraction packages for both database systems. Two courts ultimately sent data from both systems.

Extraction from CM/ECF Courts

Development of the data-extraction package for the CM/ECF courts followed a sequence equivalent to the one described for the ICMS courts. Structural and execution differences between the systems, however, required technical modifications to the general approach. We again worked with database documentation from the AO to identify the data tables that needed to be accessed and the data fields to be extracted. The AO also gave the FJC access to a test database to facili-

tate the development and testing of the extraction program. We used available database utilities to assist us in writing the program code.

We again wrote the extraction program to allow courts control over the portion of their caseload to be processed in a single run and to direct the output of files. Sixteen output data files were created during the extraction, including files that provided the interpretation of the docketing codes used by the court. To accommodate a different approach to querying the database, the extraction had to be executed in two parts. Running the program provided by the FJC resulted in the creation of a database command file that incorporated the information from the execution parameters. Court staff could then use a standard database utility, *dbaccess*, to execute the generated command file and perform the data extraction. To increase the efficiency of the processing, the command file first created a temporary table of case identifiers corresponding to the cases to be processed and then used the table in subsequent extraction queries. Sealed cases—clearly identified in the CM/ECF database and protected with access restrictions—were explicitly excluded from the data extractions.

Members of the Technical Advisory Group again reviewed the code and tested preliminary versions of the program. This review resulted in identifying important modifications that enhanced the efficiency of the extractions. Two district courts, both represented on the Advisory Group, pilot tested the final version of the program and the instructions.

Once pilot testing was completed, we sent an e-mail request for extraction to the clerks of CM/ECF courts. The e-mail message provided background on the case-weighting project and explained the need for data from the court's database. The request included a copy of the program and instructions for system managers on how to set the execution parameters, run the generated command file, and transfer the output files to the FJC. The e-mail also provided a link to an FJC Web page that contained downloadable copies of the files (<http://cwn.fjc.dcn/newweb/jnetweb.nsf/pages/512>).

Those few courts that elected to extract from CM/ECF rather than ICMS when informed they had a choice received the extraction package with a letter tailored to their special situation. In addition, courts that needed to extract some data from an ICMS database and some from a CM/ECF database were sent applicable programs with a cover letter explaining how to perform the extractions.

We sent a copy of the request to the court contact listed on the docketing survey if it was someone other than the clerk, and we sent the informational letter from the chair of the Subcommittee to the chief judge if this had not been done previously.

We mailed requests on February 5, 2004, and asked courts to respond by February 23, 2004. Most courts responded by the deadline, and all of the CM/ECF courts sent in data by early March. Because of the two-phase execution it was more difficult to obtain processing times for the CM/ECF extraction runs. Information that was available, however, indicates that the extractions in most courts

took less than one hour. In all, nineteen courts sent in data from their CM/ECF databases, two of which also sent data from an ICMS database.

Extraction from Courts with Non-standard Databases

Four courts indicated on the docketing survey that they used neither the ICMS nor the CM/ECF system for docketing cases; they instead used specially developed local systems. We recognized that without substantial assistance from the courts, we would be unable to develop individualized data-extraction programs for these systems in the time available for project completion. With that in mind, we sent an e-mail to the clerk of each court describing the project and asking whether court staff might work with FJC staff to develop extraction programs. The letter described the kind of information that was required and included examples of the ICMS and CM/ECF extraction programs.

One of the four courts responded to this request. The court made several attempts to extract the information identified in the request, but ultimately was unable to provide some essential data elements that were not routinely captured in their database. Since the court, at that time, was in the process of converting its civil database to CM/ECF, they volunteered to try an extraction from their CM/ECF database even though it was not yet “live.” They were able to extract and send to the FJC complete information on the civil cases terminated in calendar 2002 that had already been converted.

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



Subcommittee Members

Honorable Wiley Y. Daniel, Chair
Honorable Harry F. Barnes
Honorable Susan Yvonne Illston
Honorable James R. Melinson
Honorable Ursula M. Ungaro-Benages
Honorable Karen J. Williams

Staff

Ellyn L. Vail

SUBCOMMITTEE ON JUDICIAL STATISTICS
OF THE
COMMITTEE ON JUDICIAL RESOURCES

December 15, 2003

MEMORANDUM TO SELECTED CHIEF DISTRICT JUDGES

SUBJECT: Data extractions performed for the district court case weight study.
(INFORMATION)

As you know, the Committee on Judicial Resources is working to revise the district court case weights before the next judgeship survey. At the request of the Committee, the Federal Judicial Center is conducting a new case weighting study.

The meetings of district judges recently held in each circuit are one component of the new study. Another component in determining the new weights is the extraction of information about case events that were docketed in each district's case management database for civil and criminal cases terminated during calendar year 2002. The Federal Judicial Center has developed data extraction programs that can be used with each court's database to obtain the information needed for the study. The Center recently sent a request to the clerks of court in districts that use the ICMS database system asking them to perform the required data extractions. Your district was included in the request. A similar request will be made to courts that use the CM/ECF database system early next year.

This message is being sent for your information only. Because database extractions are such a crucial part of the ongoing case weighting study, I wanted to make you aware of this aspect of the project, and express my appreciation for your district's continued assistance in this matter. If you have any questions about the data extractions specifically, or the case weighting project in general, please feel free to contact Pat Lombard of the Federal Judicial Center (202-502-4083 or plombard@fjc.gov).

Again, thank you for your support of this important project.

A handwritten signature in black ink that reads "Wiley Y. Daniel".

Wiley Y. Daniel
Chair, Subcommittee on
Judicial Statistics

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

Memorandum

To: Clerks of the United States District Courts
From: James B. Eaglin, Director Research Division, Federal Judicial Center
Subject: Data Extraction Request
Date: December 12, 2003

The Committee on Judicial Resources is working to revise the district court case weights before the next judgeship survey. The Federal Judicial Center has consequently been asked to conduct a new case weighting study, and is using an event-based approach to computing case weights. A critical element of the computations is the frequency with which different case events occur in different types of cases. This incidence data will be obtained by extracting docketed event information from each district court's case management database.

We sent a Survey of Court Docketing Practices, requesting information about a variety of docketing and database issues, to all district courts in August. Every court responded to the survey and we thank you for this extraordinary level of cooperation and assistance. The information we received helped us to prepare to extract and analyze the docketing data.

We have now moved to the next stage of the project, and must again call upon you for assistance to implement the data extraction. The Federal Judicial Center has developed a program that extracts the required data from an ICMS database. We ask that your ICMS systems manager run the data extraction program for your district and **return the extracted data to the FJC no later than December 31, 2003**. We recognize that this time frame may be ambitious. Please know that it is dictated by the very demanding project schedule that has been approved by the Statistics Subcommittee of the Judicial Resources Committee.

All cases, civil and criminal, that were terminated in your district court during calendar year 2002 should be included in the data extraction. A copy of the program and instructions for running the data extraction are attached to this message. These files can also be downloaded from the project's web page at <http://156.132.47.230:8081/newweb/jnetweb.nsf/pages/511>. The program can be run at any time that is convenient to the court during the next few weeks. The output files created during the execution should be electronically transferred to the FJC by following the instructions provided.

At this time, we are asking all ICMS courts to do the extraction; a similar request will be sent to courts using CM/ECF databases early next year. You were identified as an ICMS court based on your responses to the docketing survey; if you believe you received this message in error, please contact us as soon as possible. A copy of this message has also been sent to the court contact listed on the survey.

As this pivotal phase of the study gets underway, we greatly appreciate your continued cooperation. If you have any questions regarding the data extraction program or the case weighting study, please don't hesitate to contact Pat Lombard (202-502-4083, plombard@fjc.gov) or George Cort (202-502-4087, gcort@fjc.gov).

The Federal Judicial Center thanks you in advance for your assistance on this important project.

Sincerely yours,

James B. Eaglin
Director, Research Division
Federal Judicial Center
One Columbus Circle, NE
Washington, DC 20002-8003

202-502-4071
jeaglin@fjc.gov

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

**Federal Judicial Center
2003 - 2004 District Court Case-Weighting Study**

Instructions for Executing the DCCWS_EXTRACT Program

The purpose of this program is to extract case and event data from your ICMS database for all cases terminated during calendar year 2002.

To run the DCCWS_EXTRACT program:

- 1) Make the program file executable. Either of the following chmod commands will make the program fully accessible.

```
chmod ugo+rwx DCCWS_EXTRACT
or
chmod 777 DCCWS_EXTRACT
```

- 2) If the DBPATH, DBNAME and UNIFY environmental variables are not set, modify the first three lines of the program to set them and remove the "##" comment flag at the beginning of each of the lines.

```
## DBPATH=;                export DBPATH
## DBNAME=;                export DBNAME
## UNIFY=;                 export UNIFY
```

- 3) Invoke the program by entering the program name and execution parameters on the command line.

```
DCCWS_EXTRACT court_abbrev start_date end_date output_dir
```

where *court_abbrev* is your court's 2 or 3 letter abbreviation (ILN, CAC, MD, NE, etc.)
start_date is the earliest termination date to use to select cases (*usually 01/01/2002*)
end_date is the latest termination date to use to select cases (*usually 12/31/2002*)
output_dir is a directory on your system with enough space to hold the 13 output files.

The *court_abbrev* parameter is required, if it is missing from the command line the program will prompt you to enter a non-blank value. The value of any other parameter not included on the command line will default to the pre-defined value in the program.

Execution Parameter Defaults

```
STARTDATE=01/01/2002
ENDDATE=12/31/2002
OUTDIR=$DBPATH/../../rpt
```

NOTE: If you have problems with disk space or memory (for approximate disk space requirements, check the "Output File Descriptions" section of the program file) when running this program, you may need to modify the STARTDATE and ENDDATE

values and run the program more than once, each for a shorter span of time within the date range (e.g., run the program using STARTDATE and ENDDATE values that span the first 6 months of 2002, then re-run the program to capture the last half of the year). The names of the files generated by the program include an indication of the last month of data included in a particular run (e.g. DEC or JUN).

If your civil and criminal data reside in separate databases, you will need to run the program against each database. Please use the court_abbrev variable to make sure the output files are named differently (e.g., instead of just using CAC as the court_abbrev use CACcv and CACcr, otherwise the second set of files will overwrite the first when you send them to the FJC).

Since the size of the databases are so variable it is hard to estimate how long it will take to execute the program in any particular court, but in the various test and pilot runs the time ranged from about 15 minutes to about 1.5 hours.

Submitting OUTPUT to FJC

Once the program has successfully captured all the data for calendar year 2002, please send the files to the FJC via FTP using the instructions below.

cd to the output directory you specified or to the default, \$DBPATH/./rpt

ftp wind.fjc.dcn (or 156.132.47.249)

user: anonymous

password: <your email address>

cd incoming

prompt (turns off interactive mode)

mput FJC_* (this will upload multiple files at once)

quit

Unless there are severe disk space constraints, please keep a copy of the 13 output files on disk until you receive an acknowledgement back from the Federal Judicial Center that all of the files have been received (an email message will be sent to the court contact listed on the docketing practices survey we received from the court). This will avoid the need to rerun the program if a transmission error occurs.

If you have any questions, please contact George Cort at 202-502-4087 or gcort@fjc.gov. Thank you.

DCCWS_EXTRACT Program

Extraction Program for the ICMS Courts

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


```
## DBPATH=;                export DBPATH
## DBNAME=;                export DBNAME
## UNIFY=;                 export UNIFY

## Note:      The environmental variables listed above must be set for the proper
##            execution of this program.  Many courts have other scripts that set
##            these parameters, and so to avoid interference with already defined
##            values, these initialization statements are commented out.  If,
##            however, you want to use this program to set these variables, you
##            can edit in the appropriate value in the lines above.  BUT, DON'T FORGET
##            TO REMOVE THE "##" COMMENT FLAG AT THE BEGINNING OF THE LINE, IF YOU
##            WANT IT TO EXECUTE.

## Usage:  DCCWS_EXTRACT court_abbrev start_date end_date output_directory
##
##         or:  DCCWS_EXTRACT -h          <--- This will provide help on executing the program

## Execution Parameter Defaults

CRTABB=""
STARTDATE=01/01/2002
ENDDATE=12/31/2002
OUTDIR=$DBPATH/../../rpt

## The above parameters are required for the execution of this program.  Default
## values are set for all but the court abbreviation (CRTABB) which must either
## be entered on the command line or the program will prompt for a value
## during execution.  The values for the other parameters can be changed by
## editing this program or by including a new value on the command line.

## DCCWS_EXTRACT

## Federal Judicial Center
## District Court Case Weighting Study
## Court Data Base Extraction Program for ICMS Databases

## Contacts:  Pat Lombard (plombard@fjc.gov, 202-502-4083)
##            George Cort (gcort@fjc.gov, 202-502-4087)

## Created:  21-September-2003
## Version 2.0 Modified: 05-December-2003

## Description:
##
## This program extracts case and event data from an ICMS database for all cases
## (civil and criminal) terminated in the district court between two defined dates.
## By default the period is 01-January-2002 to 31-December-2002, and most courts will
## run the program once and get the data for the entire year.  However, these dates are
## defined through parameters that can be set when the program is run, so if a court
## cannot process a full year of data in a single pass - for example because of memory
## or disk space restrictions - they can change the start and end date to process only
## part of a year's terminated cases in each of several passes.

## To allow the data runs to be segmented to accommodate disk space limitations, the
## filenames include the 3 character abbreviation for the last month in calendar 2002
## that is included in the data (this is extracted from the $ENDDATE variable).

## The filenames also have a 2 or 3 character court abbreviation appended as the file
## extension for unique identification (e.g., CAN or LAW) when the files are forwarded
```

```

## to the Federal Judicial Center.  Instructions are provided for transferring the files
## to the FJC.

## These data are extracted into 10 files (Cases, ASCcases, Events, Reliefs,
## DPLink1, DPLink2, Who, Parties, JS2, and Judges).  There are also 3 housekeeping files
## generated (Log, SQLmsg, and Err).  All files are named beginning with "FJC" to avoid
## naming conflicts with other files on the system.  The output files are written
## to the directory area identified by the court, by default it is $DBPATH/./rpt.

## Output File Descriptions:
##

## File: FJC_CASES_endmth.crtabb (e.g., FJC_CASES_DEC.CAN)
## Layout:  cases.office, cases.year, cases.docket_type, cases.number,
##          cases.in_re_flag, cases.short1, cases.etal_flag1,
##          cases.short2, cases.etal_flag2, cases.type, cases.subtype,
##          cases.sstype, cases.nature_of_suit, cases.cause, cases.origin,
##          cases.jurisdiction, cases.class_action,
##          cases.us_plaintiff, cases.us_defendant,
##          cases.date_filed, cases.date_docketed, cases.date_terminated,
##          cases.js_reopen, cases.js_reterm,
##          cases.js_open_entry, cases.js_term_entry, cases.js_change,
##          cases.judge, cases.magistrate, cases.date_at_issue, cases.date_pretrial,
##          cases.date_trial_begin,
##          cases.date_trial_end, cases.progress_at_term, cases.disp_method,
##          cases.js_status, cases.js_cls_status, cases.js_ropn_status,
##          cases.js_reterm_status, cases.sealed,
##          cases.death_penalty, cases.case_dcid, cases.nxseq,
##          cases.nxpty, cases.next_document, cases.next_count, cases.last_update
## Number of Bytes (approx): 256 (data fields) + 46 (delimiters) = 302

## File: FJC_ASCCASES_endmth.crtabb (e.g., FJC_ASCCASES_DEC.CAN)
## Layout:  asccases.asc_id, asccases.asc_lead_cs_off, asccases.asc_lead_cs_year,
##          asccases.asc_lead_cs_dkt, asccases.asc_lead_cs_num,
##          asccases.asc_memb_cs_off, asccases.asc_memb_cs_year,
##          asccases.asc_memb_cs_dkt, asccases.asc_memb_cs_num, asccases.type,
##          asccases.begin_date, asccases.end_date
## Number of Bytes (approx): 50 (data fields) + 11 (delimiters) = 61

## File: FJC_EVENTS_endmth.crtabb (e.g., FJC_EVENTS_DEC.CAN)
## Layout:  cashist.histid, cashist.ch_csno_cs_off, cashist.ch_csno_cs_year,
##          cashist.ch_csno_cs_dkt, cashist.ch_csno_cs_num,
##          cashist.seq_no, cashist.seq_ext, cashist.apply_flag,
##          hist.event_group, hist.event_id, hist.odate, hist.service_date, hist.entry_date
## Number of Bytes (approx): 65 (data fields) + 12 (delimiters) = 77

## File: FJC_RELIEFS_endmth.crtabb (e.g., FJC_RELIEFS_DEC.CAN)
## Layout:  docproc.doc_id, docproc.doc_ext, docproc.doc_subext,
##          docproc.relief_group, docproc.relief_code, docproc.subtype, docproc.type,
##          docproc.dc_case_cs_off, docproc.dc_case_cs_year,
##          docproc.dc_case_cs_dkt, docproc.dc_case_cs_num,
##          docproc.terminator, docproc.filed, docproc.disposed,
##          docproc.req_date, docproc.flag, docproc.case_doc_no, docproc.case_doc_ext
## Number of Bytes (approx): 104 (data fields) + 17 (delimiters) = 121

## File: FJC_DPLINK1_endmth.crtabb (e.g., FJC_DPLINK1_DEC.CAN)
## Layout:  histdp.histid, histdp.hd_dc_dc_id, histdp.hd_dc_dc_xt, histdp.hd_dc_dc_sxt,
##          histdp.order_action_id

```

```

## Number of Bytes (approx): 31 (data fields) + 4 (delimiters) = 35

## File: FJC_DPLINK2_endmth.crtabb (e.g., FJC_DPLINK2_DEC.CAN)
## Layout: histdp.histid, histdp.hd_dc_dc_id, histdp.hd_dc_dc_xt, histdp.hd_dc_dc_sxt,
##          histdp.order_action_id
## Number of Bytes (approx): 31 (data fields) + 4 (delimiters) = 35

## File: FJC_WHO_endmth.crtabb (e.g., FJC_WHO_DEC.CAN)
## Layout: who.prid, who.histid, who.who_case_cs_off, who.who_case_cs_year,
##          who.who_case_cs_dkt, who.who_case_cs_num, who.involve,
##          person.prid, person.type, person.subtype, person.last, person.initials
## Number of Bytes (approx): 70 (data fields) + 11 (delimiters) = 81

## File: FJC_PARTY_endmth.crtabb (e.g., FJC_PARTY_DEC.CAN)
## Layout: ptycas.pc_csno_cs_off, ptycas.pc_csno_cs_year,
##          ptycas.pc_csno_cs_dkt, ptycas.pc_csno_cs_num,
##          ptycas.party_number, ptycas.party_id, ptycas.party_type, ptycas.party_prose
## Number of Bytes (approx): 29 (data fields) + 7 (delimiters) = 36

## File: FJC_JS2_endmth.crtabb (e.g., FJC_JS2_DEC.CAN)
## Layout: js2.j2_key, js2.office, js2.year, js2.docket_type, js2.number, js2.party,
##          js2.filing_date, js2.proceeding_dt, js2.proceeding_code, js2.dft_prid,
##          js2.cns1_cd, js2.citation_one, js2.status, js2.corp_code
## Number of Bytes (approx): 59 (data fields) + 13 (delimiters) = 72

## File: FJC_JUDGE_endmth.crtabb (e.g., FJC_JUDGE_DEC.CAN)
## Layout: judge.jdgid, judge.ao_code, judge.initials
## Number of Bytes (approx): 17 (data fields) + 2 (delimiters) = 19

## Space requirements are approximately 22 MB per 1000 cases (50 event average per case)
##
## Est_Bytes = (1000 * (302 + (10*61) + (50*77) + (50*121) + (50*35) + (50*35) + (100*81) +
##              (10*36) + (10*72))) + (50*19);

## Part 1
## Process Execution Parameters and Create Unique Run Variables

## Test parameters, if no arguments were passed, then prompt for court abbreviation.
## If arguments are present, assume first is the court abbreviation.

if [ $# -lt 1 ]
then
    echo "Enter an abbreviation for your court (e.g., CAN or TXW): \c"
    read CRTABB
    if [ -z $CRTABB ]
    then
        echo "\n\nThe court abbreviation cannot be blank, please re-enter"
        echo "\n\nEnter an abbreviation for your court (e.g., CAN or TXW): \c"
        read CRTABB
        if [ -z $CRTABB ]
        then
            echo "Improper court abbreviation. Exiting"
            exit
        fi
    fi
elif [ "$1" = "-h" ]

```

```

then
    echo "\nProgram Usage:"
    echo "\nDCCWS_EXTRACT court_abbrev start_date end_date output_directory"
    echo "\nwhere court_abbrev is a 2 or 3 character unique abbreviation"
    echo "      for your court (e.g., CAN or TXW)"
    echo "      start_date is the earliest termination date to use to select"
    echo "      cases (default: 01/01/2002)"
    echo "      end_date is the latest termination date to use to select"
    echo "      cases (default: 12/31/2002)"
    echo "      output_directory is the location for output files"
    echo "      (default: DBPATH/./rpt)"
    echo "\n"
    exit
else
    CRTABB=$1
fi

export CRTABB

## Check for additional arguments
## If more than 1, assume second is the start date.  If no argument, default start date is used.

if [ $# -gt 1 ]
then
    STARTDATE=$2
fi

export STARTDATE

## Test that the start date is valid

MTH1=`expr "$STARTDATE" : '\(.*)\./.*.*'`
DAY1=`expr "$STARTDATE" : '.*\/\(.*)\./.*'`
YEAR1=`expr "$STARTDATE" : '.*\/.*\/\(.*)'`

if [ -z "$MTH1" -o -z "$DAY1" -o -z "$YEAR1" ]
then
    echo "Invalid start date supplied. Exiting"
    exit
fi
if [ $YEAR1 -ne 2002 ]
then
    echo "Start date not in calendar 2002. Exiting"
    exit
fi

DATE1=`expr $YEAR1 \* 10000 + $MTH1 \* 100 + $DAY1`

## Check for additional arguments
## If more than 2, assume third is the end date.  If no argument, default end date is used.

if [ $# -gt 2 ]
then
    ENDDATE=$3
fi

export ENDDATE

## Test that the end date is valid

MTH2=`expr "$ENDDATE" : '\(.*)\./.*.*'` ; export MTH2
DAY2=`expr "$ENDDATE" : '.*\/\(.*)\./.*'` ; export DAY2

```

```

YEAR2=`expr "$ENDDATE" : '.*/*/\(.*\)'\` ; export YEAR2

if [ -z "$MTH2" -o -z "$DAY2" -o -z "$YEAR2" ]
then
    echo "Invalid end date supplied. Exiting"
    exit
fi
if [ $YEAR2 -ne 2002 ]
then
    echo "End date not in calendar 2002. Exiting"
    exit
fi

DATE2=`expr $YEAR2 \* 10000 + $MTH2 \* 100 + $DAY2`

if [ $DATE2 -lt $DATE1 ]
then
    echo "End date not later than start date. Exiting"
    exit
fi

## Check for additional arguments
## If more than 3, assume fourth is the output directory. If no argument, default directory is used.
## Verify output path area is present and writeable

if [ $# -gt 3 ]
then
    OUTDIR=$4
fi

if [ -d $OUTDIR -a -w $OUTDIR ]
then
    export OUTDIR
else
    echo "Directory $OUTDIR either does not exist or is not writeable. Exiting"
    exit
fi

## Setup Operational Parameters and Files

case $MTH2 in
01) ENDMTH=JAN;;
02) ENDMTH=FEB;;
03) ENDMTH=MAR;;
04) ENDMTH=APR;;
05) ENDMTH=MAY;;
06) ENDMTH=JUN;;
07) ENDMTH=JUL;;
08) ENDMTH=AUG;;
09) ENDMTH=SEP;;
10) ENDMTH=OCT;;
11) ENDMTH=NOV;;
12) ENDMTH=DEC;;
esac
export ENDMTH

SQLMSG=$OUTDIR/FJC_SQLMSG_${ENDMTH}.$CRTABB; export SQLMSG
LOGFIL=$OUTDIR/FJC_LOGFIL_${ENDMTH}.$CRTABB; export LOGFIL
ERRFIL=$OUTDIR/FJC_ERRFIL_${ENDMTH}.$CRTABB; export ERRFIL

```

```
## End of parameter processing.
```

```
echo "\nParameter processing has completed." | tee -a $LOGFIL
echo "\nThese parameters will be used for this run." | tee -a $LOGFIL
echo "\n Court Abbreviation: $CRTABB " | tee -a $LOGFIL
echo "\n Start Date: $STARTDATE " | tee -a $LOGFIL
echo "\n End Date: $ENDDATE " | tee -a $LOGFIL
echo "\n Output Directory: $OUTDIR " | tee -a $LOGFIL
echo "\n SQL Message File: $SQLMSG " | tee -a $LOGFIL
echo "\n Log File: $LOGFIL " | tee -a $LOGFIL
echo "\n Error File: $ERRFIL " | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
echo "\nData extraction begins:" | tee -a $LOGFIL
date | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
```

```
## Part 2
```

```
## Database Extractions
```

```
## Create Cases Filename and Extract Cases Data
```

```
CASESFILE=$OUTDIR/FJC_CASES_$(ENDMTH).$CRTABB; export CASESFILE
```

```
SQL << EOFsql >> $SQLMSG 2>>$ERRFIL
```

```
lines 0
```

```
select cases.office, cases.year, cases.docket_type, cases.number,
       cases.in_re_flag, cases.short1, cases.etal_flag1,
       cases.short2, cases.etal_flag2, cases.type, cases.subtype,
       cases.sstype, cases.nature_of_suit, cases.cause, cases.origin,
       cases.jurisdiction, cases.class_action,
       cases.us_plaintiff, cases.us_defendant,
       cases.date_filed, cases.date_docketed, cases.date_terminated,
       cases.js_reopen, cases.js_reterm,
       cases.js_open_entry, cases.js_term_entry, cases.js_change,
       cases.judge, cases.magistrate, cases.date_at_issue, cases.date_pretrial,
       cases.date_trial_begin,
       cases.date_trial_end, cases.progress_at_term, cases.disp_method,
       cases.js_status, cases.js_cls_status, cases.js_ropn_status,
       cases.js_reterm_status, cases.sealed,
       cases.death_penalty, cases.case_dcid, cases.nxseq,
       cases.nxpty, cases.next_document, cases.next_count, cases.last_update
from cases
where [cases.date_terminated >= $STARTDATE and cases.date_terminated <= $ENDDATE] or
      [cases.js_reterm >= $STARTDATE and cases.js_reterm <= $ENDDATE]
into '$CASESFILE' /
EOFsql
```

```
## Create Associated Cases Filename and Extract Associated Cases Data
```

```
ASCFILE=$OUTDIR/FJC_ASCCASES_$(ENDMTH).$CRTABB; export ASCFILE
```

```
SQL << EOFsql >> $SQLMSG 2>>$ERRFIL
```

```
lines 0
```

```
select asccases.asc_id, asccases.asc_lead_cs_off, asccases.asc_lead_cs_year,
       asccases.asc_lead_cs_dkt, asccases.asc_lead_cs_num,
       asccases.asc_memb_cs_off, asccases.asc_memb_cs_year,
       asccases.asc_memb_cs_dkt, asccases.asc_memb_cs_num, asccases.type,
       asccases.begin_date, asccases.end_date
from asccases
where <asccases.asc_lead_cs_off, asccases.asc_lead_cs_year,
      asccases.asc_lead_cs_dkt, asccases.asc_lead_cs_num> is in
      select cases.office, cases.year, cases.docket_type, cases.number
```

```

        from cases where
[[cases.date_terminated >= $STARTDATE and cases.date_terminated <= $ENDDATE] or
 [cases.js_reterm >= $STARTDATE and cases.js_reterm <= $ENDDATE]];
    or <ascases.asc_memb_cs_off, ascases.asc_memb_cs_year,
    ascases.asc_memb_cs_dkt, ascases.asc_memb_cs_num> is in
        select cases.office, cases.year, cases.docket_type, cases.number
        from cases where
[[cases.date_terminated >= $STARTDATE and cases.date_terminated <= $ENDDATE] or
 [cases.js_reterm >= $STARTDATE and cases.js_reterm <= $ENDDATE]]
into '$ASCFILE' /
EOFsql

```

Create Events Filename and Extract Events Data (HIST)

```
EVTFILE=$OUTDIR/FJC_EVENTS_$ENDMTH.$CRTABB; export EVTFILE
```

```

SQL << EOFsql >> $SQLMSG 2>>$ERRFIL
lines 0
select cashist.histid, cashist.ch_csno_cs_off, cashist.ch_csno_cs_year,
    cashist.ch_csno_cs_dkt, cashist.ch_csno_cs_num,
    cashist.seq_no, cashist.seq_ext, cashist.apply_flag,
    hist.event_group, hist.event_id, hist.odate, hist.service_date, hist.entry_date
from cashist, hist
where <cashist.ch_csno_cs_off, cashist.ch_csno_cs_year,
    cashist.ch_csno_cs_dkt, cashist.ch_csno_cs_num> is in
    select cases.office, cases.year, cases.docket_type, cases.number
    from cases where
[[cases.date_terminated >= $STARTDATE and cases.date_terminated <= $ENDDATE] or
 [cases.js_reterm >= $STARTDATE and cases.js_reterm <= $ENDDATE]];
    and [cashist.histid = hist.histid]
into '$EVTFILE' /
EOFsql

```

Create Reliefs Filename and Extract Reliefs Data (DOCPROC)

```
RLFFILE=$OUTDIR/FJC_RELIEFS_$ENDMTH.$CRTABB; export RLFFILE
```

```

SQL << EOFsql >> $SQLMSG 2>>$ERRFIL
lines 0
select docproc.doc_id, docproc.doc_ext, docproc.doc_subext,
docproc.relief_group, docproc.relief_code, docproc.subtype, docproc.type,
    docproc.dc_case_cs_off, docproc.dc_case_cs_year,
    docproc.dc_case_cs_dkt, docproc.dc_case_cs_num,
docproc.terminator, docproc.filed, docproc.disposed,
    docproc.req_date, docproc.flag, docproc.case_doc_no, docproc.case_doc_ext
from docproc
where <docproc.dc_case_cs_off, docproc.dc_case_cs_year,
    docproc.dc_case_cs_dkt, docproc.dc_case_cs_num> is in
    select cases.office, cases.year, cases.docket_type, cases.number
    from cases where
[[cases.date_terminated >= $STARTDATE and cases.date_terminated <= $ENDDATE] or
 [cases.js_reterm >= $STARTDATE and cases.js_reterm <= $ENDDATE]]
into '$RLFFILE' /
EOFsql

```

Create DPLink1 Filename and Extract Event / Relief / Action Data (HISTDP)

Selection based on HISTID

```
DPL1FILE=$OUTDIR/FJC_DPLINK1_$ENDMTH.$CRTABB; export DPL1FILE
```

```
SQL << EOFsql >> $SQLMSG 2>>$ERRFIL
```

```

lines 0
select histdp.histid, histdp.hd_dc_dc_id, histdp.hd_dc_dc_xt, histdp.hd_dc_dc_sxt,
       histdp.order_action_id
from histdp
where histdp.histid is in
select cashist.histid
from cashist
where <cashist.ch_csno_cs_off, cashist.ch_csno_cs_year,
       cashist.ch_csno_cs_dkt, cashist.ch_csno_cs_num> is in
       select cases.office, cases.year, cases.docket_type, cases.number
       from cases where
[[cases.date_terminated >= $STARTDATE and cases.date_terminated <= $ENDDATE] or
 [cases.js_reterm >= $STARTDATE and cases.js_reterm <= $ENDDATE]]
into '$DPL1FILE' /
EOFsql

```

```

## Create DPLink2 Filename and Extract Event / Relief / Action Data (HISTDP)
## Selection based on DOCPROC

```

```
DPL2FILE=$OUTDIR/FJC_DPLINK2_$ENDMTH.$CRTABB; export DPL2FILE
```

```

SQL << EOFsql >> $SQLMSG 2>>$ERRFIL
lines 0
select histdp.histid, histdp.hd_dc_dc_id, histdp.hd_dc_dc_xt, histdp.hd_dc_dc_sxt,
       histdp.order_action_id
from histdp
where <histdp.hd_dc_dc_id, histdp.hd_dc_dc_xt, histdp.hd_dc_dc_sxt> is in
select docproc.doc_id, docproc.doc_ext, docproc.doc_subext
from docproc
where <docproc.dc_case_cs_off, docproc.dc_case_cs_year,
       docproc.dc_case_cs_dkt, docproc.dc_case_cs_num> is in
       select cases.office, cases.year, cases.docket_type, cases.number
       from cases where
[[cases.date_terminated >= $STARTDATE and cases.date_terminated <= $ENDDATE] or
 [cases.js_reterm >= $STARTDATE and cases.js_reterm <= $ENDDATE]]
into '$DPL2FILE' /
EOFsql

```

```
## Create Who Filename and Extract Who Data (WHO)
```

```
WHOFILE=$OUTDIR/FJC_WHO_$ENDMTH.$CRTABB; export WHOFILE
```

```

SQL << EOFsql >> $SQLMSG 2>>$ERRFIL
lines 0
select who.prid, who.histid, who.who_case_cs_off, who.who_case_cs_year,
       who.who_case_cs_dkt, who.who_case_cs_num, who.involve,
       person.prid, person.type, person.subtype, person.last, person.initials
from who, person
where <who.who_case_cs_off, who.who_case_cs_year,
       who.who_case_cs_dkt, who.who_case_cs_num> is in
       select cases.office, cases.year, cases.docket_type, cases.number
       from cases where
[[cases.date_terminated >= $STARTDATE and cases.date_terminated <= $ENDDATE] or
 [cases.js_reterm >= $STARTDATE and cases.js_reterm <= $ENDDATE]];
and [who.prid = person.prid]
into '$WHOFILE' /
EOFsql

```

```
## Create Party Filename and Extract Party Data (PTYCAS)
```

```
PTYFILE=$OUTDIR/FJC_PARTY_$ENDMTH.$CRTABB; export PTYFILE
```



```

SQL << EOFsql >> $SQLMSG 2>>$ERRFIL
lines 0
select ptycas.pc_csno_cs_off, ptycas.pc_csno_cs_year,
       ptycas.pc_csno_cs_dkt, ptycas.pc_csno_cs_num,
       ptycas.party_number, ptycas.party_id, ptycas.party_type, ptycas.party_prose
from ptycas
where <ptycas.pc_csno_cs_off, ptycas.pc_csno_cs_year,
      ptycas.pc_csno_cs_dkt, ptycas.pc_csno_cs_num> is in
      select cases.office, cases.year, cases.docket_type, cases.number
      from cases where
[[cases.date_terminated >= $STARTDATE and cases.date_terminated <= $ENDDATE] or
 [cases.js_reterm >= $STARTDATE and cases.js_reterm <= $ENDDATE]]
into '$PTYFILE' /
EOFsql

```

```
## Create JS2 Filename and Extract JS2 Data (JS2)
```

```
JS2FILE=$OUTDIR/FJC_JS2_$ENDMTH.$CRTABB; export JS2FILE
```

```

SQL << EOFsql >> $SQLMSG 2>>$ERRFIL
lines 0
select js2.j2_key, js2.office, js2.year, js2.docket_type, js2.number, js2.party,
       js2.filing_date, js2.proceeding_dt, js2.proceeding_code, js2.dft_prid,
       js2.cns1_cd, js2.citation_one, js2.status, js2.corp_code
from js2
where <js2.office, js2.year, js2.docket_type, js2.number> is in
      select cases.office, cases.year, cases.docket_type, cases.number
      from cases where
[[cases.date_terminated >= $STARTDATE and cases.date_terminated <= $ENDDATE] or
 [cases.js_reterm >= $STARTDATE and cases.js_reterm <= $ENDDATE]]
into '$JS2FILE' /
EOFsql

```

```
## Create Judge Filename and Extract Judge Data (JUDGE)
```

```
JDGFILE=$OUTDIR/FJC_JUDGE_$ENDMTH.$CRTABB; export JDGFILE
```

```

SQL << EOFsql >> $SQLMSG 2>>$ERRFIL
lines 0
select judge.jdgid, judge.ao_code, judge.initials
from judge
into '$JDGFILE' /
EOFsql

```

```
## End of data extraction. Instruct users to send files to FJC
```

```

echo "\nThis data extraction run is complete." | tee -a $LOGFIL
date | tee -a $LOGFIL
echo "\nThese output files were generated by the program: " | tee -a $LOGFIL
echo " $CASESFILE " | tee -a $LOGFIL
echo " $ASCFILE " | tee -a $LOGFIL
echo " $EVTFILE " | tee -a $LOGFIL
echo " $RLLFFILE " | tee -a $LOGFIL
echo " $DPL1FILE " | tee -a $LOGFIL
echo " $DPL2FILE " | tee -a $LOGFIL
echo " $WHOFIL " | tee -a $LOGFIL
echo " $PTYFILE " | tee -a $LOGFIL
echo " $JS2FILE " | tee -a $LOGFIL
echo " $JDGFILE " | tee -a $LOGFIL

```

```
echo " $LOGFIL " | tee -a $LOGFIL
echo " $SQLMSG " | tee -a $LOGFIL
echo " $ERRFIL " | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
echo "Please FTP these output files to the Federal Judicial Center " | tee -a $LOGFIL
echo "using the instructions below: " | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
echo "ftp wind.fjc.dcn (or ftp 156.132.47.249)" | tee -a $LOGFIL
echo "user: anonymous" | tee -a $LOGFIL
echo "password: <your email address>" | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
echo "cd incoming" | tee -a $LOGFIL
echo "prompt (turns off interactive mode)" | tee -a $LOGFIL
echo "mput $OUTDIR/FJC_* (uploads multiple files at once)" | tee -a $LOGFIL
echo "quit" | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
echo "If you have any questions, please contact:" | tee -a $LOGFIL
echo "George Cort at 202-502-4087 or gcort@fjc.gov." | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
echo "Thank you for your help with this project." | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
```

```
##
## End of DCCWS_EXTRACT
##
```

DCCWS_ECFXTR Program

Extraction Program for the CM/ECF Courts

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

```
## Important Note: The .sql command file generated by this shell program creates
## a temporary table FJC_ECFXTR to hold the list of caseids to be extracted.
## Using a temporary table substantially simplifies the subsequent queries
## in the program and allows the program to run more quickly and efficiently.
## The temporary table is not logged and is dropped at the end of the session.
## If, however, you do not wish to allow the .sql command file to create a
## temporary table, please notify us (Pat Lombard, plombard@fjc.gov, 202-502-4083;
## or George Cort, gcort@fjc.gov, 202-502-4087) and we can send you a version
## of the program that does not use a temporary table.
```

```
## Usage: DCCWS_ECFXTR court_abbrev start_date end_date output_directory
##
## or: DCCWS_ECFXTR -h <--- This will provide help on executing the program
```

```
## Execution Parameter Defaults
```

```
CRTABB=""
STARTDATE=01/01/2002
ENDDATE=12/31/2002
OUTDIR=/tmp/FJC_ECFXTR
```

```
## The above parameters are required for the execution of this program. Default
## values are set for all but the court abbreviation (CRTABB) which must either
## be entered on the command line or the program will prompt for a value
## during execution. The values for the other parameters can be changed by
## editing this program or by including a new value on the command line.
```

```
## DCCWS_ECFXTR
```

```
## Federal Judicial Center
## District Court Case Weighting Study
## Court Data Base Extraction Program for CM/ECF Databases
```

```
## Contacts: Pat Lombard (plombard@fjc.gov, 202-502-4083)
##           George Cort (gcort@fjc.gov, 202-502-4087)
```

```
## Created: 08-January-2004
## Version 1.7 Modified: 04-February-2004
```

```
## Description:
```

```
##
## This program processes execution parameters and uses them to write out an SQL command
## file that can be run from dbaccess to extract case and event data from a CM/ECF
## database for all cases (civil and criminal) terminated in the district court between
## two defined dates. Sealed cases are excluded, however. To avoid extracting data
## from sealed cases we select cases where cs_restrict_view != 40. For efficiency, the
## extractions are done from the database tables (i.e., those preceded by "u_") rather than
## from the views that automatically exclude sealed cases.
```

```
## By default the extraction period is 01-January-2002 to 31-December-2002, and most courts
## will run the program once and get the data for the entire year. However, these dates are
## defined through parameters that can be set when the program is run, so if a court
## cannot process a full year of data in a single pass - for example because of memory
## or disk space restrictions - they can change the start and end date to process only
## part of a year's terminated cases in each of several passes.
```

```
## To allow the data runs to be segmented to accommodate disk space limitations, the
## filenames include the 3-character abbreviation for the last month in calendar 2002
## that is included in the data (this is extracted from the $ENDDATE variable).
```

```

## The filenames also have a 2 or 3 character court abbreviation appended as the file
## extension for unique identification (e.g., CAN or LAW) when the files are forwarded
## to the Federal Judicial Center.  Instructions are provided for transferring the files
## to the FJC.

## These data are extracted into 16 files (Case, ASCcase, ASClead, ASCmembr, CaseFlags,
## DktEntry, DktPart, DktPerson, Filer, JS23, JS56, Judge, Motion, Party, Codes, DocumentType).
##
## A Log file and an SQL command file are also generated.  All files are named beginning
## with "FJC" to avoid naming conflicts with other files on the system.  The output files
## are written to the directory area identified by the court, by default it is /tmp/FJC_ECFXTR.
## If the default directory is used and it does not already exist, the program creates it.

## Output File Descriptions:
##

## File: FJC_CASES_endmth.crtabb (e.g., FJC_CASES_DEC.CAN)
## Layout: cs_caseid, cs_date_filed, cs_office, cs_year,
##         cs_number, cs_def_num, cs_type, cs_sub_type,
##         cs_short_title, cs_short_name1, cs_no_plas, cs_short_name2,
##         cs_no_dfts, cs_date_merge, cs_date_term, cs_date_dismiss,
##         cs_date_last_filed, cs_date_reopen,
##         cs_reopen_code, cs_closed, cs_disp_method, cs_judprid_ptr,
##         cs_judge_code, cs_refjud_prid_ptr, cs_juris,
##         cs_restrict_view, cs_gencntr, cs_dncntr, cs_defcntr,
##         cs_jstype, cs_misd_class

## File: FJC_ASCCASES_endmth.crtabb (e.g., FJC_ASCCASES_DEC.CAN)
## Layout: as_caseid, as_seqno, as_lead_caseid, as_member_caseid,
##         as_group, as_type, as_date_start, as_date_end

## File: FJC_ASCLEAD_endmth.crtabb (e.g., FJC_ASCLEAD_DEC.CAN)
## Layout: as_caseid, as_seqno, as_lead_caseid, as_member_caseid,
##         as_group, as_type, as_date_start, as_date_end

## File: FJC_ASCMEMBR_endmth.crtabb (e.g., FJC_ASCMEMBR_DEC.CAN)
## Layout: as_caseid, as_seqno, as_lead_caseid, as_member_caseid,
##         as_group, as_type, as_date_start, as_date_end

## File: FJC_CASEFLGS_endmth.crtabb (e.g., FJC_CASEFLGS_DEC.CAN)
## Layout: cf_caseid, cf_deseqno_ptr, cf_value

## File: FJC_DKTENTRY_endmth.crtabb (e.g., FJC_DKTENTRY_DEC.CAN)
## Layout: de_caseid, de_seqno, de_type, de_sub_type,
##         de_document_num, de_date_filed, de_pubacc,
##         de_doc_id, de_replaced

## File: FJC_DKTPART_endmth.crtabb (e.g., FJC_DKTPART_DEC.CAN)
## Layout: dp_caseid, dp_seqno, dp_deseqno_ptr, dp_dpseqno_ptr,
##         dp_partno, dp_type, dp_sub_type, dp_action_type,
##         dp_dispositive, dp_date_filed, dp_date_term

## File: FJC_DKTPERSON_endmth.crtabb (e.g., FJC_DKTPERSON_DEC.CAN)
## Layout: dn_caseid, dn_dpseqno_ptr, dn_person_ptr, dn_seqno,

```

```

##          dn_ref_to, dn_ref_from, dn_date_start, dn_date_term,
##          dn_ref_type

## File: FJC_FILER_endmth.crtabb (e.g., FJC_FILER_DEC.CAN)
## Layout:  fi_caseid, fi_deseqno_ptr, fi_person_ptr

## File: FJC_JS23_endmth.crtabb (e.g., FJC_JS23_DEC.CAN)
## Layout:  j2_caseid, j2_seqno, j2_court_code, j2_magis_caseno,
##          j2_proceeding_date, j2_proceeding_code, j2_open_cit1,
##          j2_open_cnslcd, j2_corp_code, j2_date_disposed,
##          j2_date_term, j2_reopen,
##          cit_citation, cit_level, cit_severity, cit_ao_code

## File: FJC_JS56_endmth.crtabb (e.g., FJC_JS56_DEC.CAN)
## Layout:  j5_caseid, j5_seqno, j5_date_filed,
##          j5_date_term, j5_disp_method, j5_reopen_code, j5_nature_of_suit,
##          j5_cause, j5_origin, j5_class_action, j5_us_plaintiff,
##          j5_us_defendant, j5_date_at_issue, j5_date_pretrial,
##          j5_date_trial_begn, j5_date_trial_end, j5_progress_at_ter

## File: FJC_JUDGE_endmth.crtabb (e.g., FJC_JUDGE_DEC.CAN)
## Layout:  jd_caseid, jd_seqno, jd_jdprid_ptr, jd_ao_code,
##          jd_judge_role, jd_referral_type, jd_type, jd_last_name,
##          jd_initials, jd_date_start, jd_date_end

## File: FJC_MOTION_endmth.crtabb (e.g., FJC_MOTION_DEC.CAN)
## Layout:  mo_caseid, mo_dpseqno, mo_motion_text, mo_term_dpseqno

## File: FJC_PARTY_endmth.crtabb (e.g., FJC_PARTY_DEC.CAN)
## Layout:  py_caseid, py_seqno, py_aliasseq, py_def_num,
##          py_type, py_role, py_claim_no, py_prose,
##          py_corporation, py_last_name, py_initials, py_date_severance,
##          py_start_date, py_end_date, py_person_ptr

## File: FJC_CODES_endmth.crtabb (e.g., FJC_CODES_DEC.CAN)
## Layout:  co_type, co_code, co_translation, co_misc, co_other

## File: FJC_DOCTYPE_endmth.crtabb (e.g., FJC_DOCTYPE_DEC.CAN)
## Layout:  do_type, do_sub_type, do_summary_text,
##          do_ref_type, do_ref_sub_type, do_case_type

## Part 1
## Process Execution Parameters and Create Unique Run Variables

## Test parameters, if no arguments were passed, then prompt for court abbreviation.
## If arguments are present, assume first is the court abbreviation.

if [ $# -lt 1 ]
then
echo "Enter an abbreviation for your court (e.g., CAN or TXW): \c"
read CRTABB
if [ -z $CRTABB ]
then

```

```

    echo "\nThe court abbreviation cannot be blank, please re-enter"
    echo "\nEnter an abbreviation for your court (e.g., CAN or TXW): \c"
    read CRTABB
    if [ -z $CRTABB ]
    then
        echo "Improper court abbreviation.  Exiting"
        exit
    fi
fi
elif [ "$1" = "-h" ]
then
    echo "\nProgram Usage:"
    echo "\nDCCWS_ECFXTR court_abbrev start_date end_date output_directory"
    echo "\nwhere court_abbrev is a 2 or 3 character unique abbreviation"
    echo "          for your court (e.g., CAN or TXW)"
    echo "          start_date is the earliest termination date to use to select"
    echo "          cases (default: 01/01/2002)"
    echo "          end_date is the latest termination date to use to select"
    echo "          cases (default: 12/31/2002)"
    echo "          output_directory is the location for output files"
    echo "          (default: /tmp/FJC_ECFXTR)"
    echo "\n"
    exit
else
    CRTABB=$1
fi

export CRTABB

## Check for additional arguments
## If more than 1, assume second is the start date.  If no argument, default start date is used.

if [ $# -gt 1 ]
then
    STARTDATE=$2
fi

export STARTDATE

## Test that the start date is valid

MTH1=`expr "$STARTDATE" : '\(.*\)/.*.*'`
DAY1=`expr "$STARTDATE" : '.*\/\(.*\)/.*'`
YEAR1=`expr "$STARTDATE" : '.*\/.*\/\(.*)'`

if [ -z "$MTH1" -o -z "$DAY1" -o -z "$YEAR1" ]
then
    echo "Invalid start date supplied.  Exiting"
    exit
fi
if [ $YEAR1 -ne 2002 ]
then
    echo "Start date not in calendar 2002.  Exiting"
    exit
fi

DATE1=`expr $YEAR1 \* 10000 + $MTH1 \* 100 + $DAY1`

## Check for additional arguments
## If more than 2, assume third is the end date.  If no argument, default end date is used.

if [ $# -gt 2 ]

```



```

then
    ENDDATE=$3
fi

export ENDDATE

## Test that the end date is valid

MTH2=`expr "$ENDDATE" : '\(.*\)/.*/*.*'` ; export MTH2
DAY2=`expr "$ENDDATE" : '.*\/(.*)\/.*'` ; export DAY2
YEAR2=`expr "$ENDDATE" : '.*\/.*\/(.*)'` ; export YEAR2

if [ -z "$MTH2" -o -z "$DAY2" -o -z "$YEAR2" ]
then
    echo "Invalid end date supplied. Exiting"
    exit
fi
if [ $YEAR2 -ne 2002 ]
then
    echo "End date not in calendar 2002. Exiting"
    exit
fi

DATE2=`expr $YEAR2 \* 10000 + $MTH2 \* 100 + $DAY2`

if [ $DATE2 -lt $DATE1 ]
then
    echo "End date not later than start date. Exiting"
    exit
fi

## Check for additional arguments
## If more than 3, assume fourth is the output directory. If no argument, default directory is used.
## Verify output path area is present and writeable. If the default directory is used and it doesn't
## exist the program creates it.

if [ $# -gt 3 ]
then
    OUTDIR=$4
fi

if [ -d $OUTDIR -a -w $OUTDIR ]
then
    export OUTDIR
elif [ "$OUTDIR" = "/tmp/FJC_ECFXTR" ]
then
    mkdir $OUTDIR
    chmod 777 $OUTDIR
    if [ -d $OUTDIR -a -w $OUTDIR ]
    then
        export OUTDIR
    else
        echo "Directory $OUTDIR either does not exist or is not writeable. Exiting"
        exit
    fi
else
    echo "Directory $OUTDIR either does not exist or is not writeable. Exiting"
    exit
fi

```

```
## Setup Operational Parameters and Files
```

```
case $MTH2 in
01) ENDMTH=JAN;;
02) ENDMTH=FEB;;
03) ENDMTH=MAR;;
04) ENDMTH=APR;;
05) ENDMTH=MAY;;
06) ENDMTH=JUN;;
07) ENDMTH=JUL;;
08) ENDMTH=AUG;;
09) ENDMTH=SEP;;
10) ENDMTH=OCT;;
11) ENDMTH=NOV;;
12) ENDMTH=DEC;;
esac
export ENDMTH
```

```
SQLCMD=$OUTDIR/FJC_ECFXTR_${ENDMTH}_$CRTABB.sql; export SQLCMD
LOGFIL=$OUTDIR/FJC_LOGFIL_${ENDMTH}.${CRTABB}; export LOGFIL
```

```
## End of parameter processing.
```

```
echo "\nParameter processing has completed." | tee -a $LOGFIL
echo "\nThese parameters will be used for this run." | tee -a $LOGFIL
echo "\n Court Abbreviation: $CRTABB " | tee -a $LOGFIL
echo "\n Start Date: $STARTDATE " | tee -a $LOGFIL
echo "\n End Date: $ENDDATE " | tee -a $LOGFIL
echo "\n Output Directory: $OUTDIR " | tee -a $LOGFIL
echo "\n SQL Command File: $SQLCMD " | tee -a $LOGFIL
echo "\n Log File: $LOGFIL " | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
```

```
## Part 2
## Database Extractions
```

```
## *****
```

```
## Create temporary table to hold case IDs of cases to be extracted
```

```
## Cases to be extracted are those that are not sealed and were terminated
## during the defined time period, usually calendar 2002. Either the
## most recent termination of the case, logged in cs_date_term, or a
## previous life of the case, logged as either j2_date_term or j5_date_term,
## can trigger inclusion of a case into the sample.
```

```
##
## The restriction to cases that are not sealed is accomplished by using the
## clause cs_restrict_view != 40 while extracting from the data table u_case
## for efficiency purposes rather than from the case view that automatically
## excludes sealed cases.
```

```
echo "\n-- Set Lock Wait Command to Assist Creation of the Temporary Table " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "set lock mode to wait 300;" >> $SQLCMD
```

```
echo "\n-- Create Temporary Table FJC_ECFXTR " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "select unique cs_caseid as fjcxtr_caseid " >> $SQLCMD
echo "from u_case " >> $SQLCMD
echo "where (cs_restrict_view != 40 and " >> $SQLCMD
echo " (cs_date_term between '$STARTDATE' and '$ENDDATE' or " >> $SQLCMD
echo " cs_caseid in (select j2_caseid from js23 " >> $SQLCMD
```

```
echo "          where j2_date_term between '$STARTDATE' and '$ENDDATE') or      " >> $SQLCMD
echo "      cs_caseid in (select j5_caseid from js56      " >> $SQLCMD
echo "          where j5_date_term between '$STARTDATE' and '$ENDDATE'))      " >> $SQLCMD
echo "into temp FJC_ECFXTR with no log;      " >> $SQLCMD
```

```
echo "\n-- Create Index on Temporary Table FJC_ECFXTR      " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "create index fjc_ext on FJC_ECFXTR ( fjcxtr_caseid );      " >> $SQLCMD
```

```
## *****
```

```
## Create Cases Filename and Extract Cases Data
```

```
CASEFILE=$OUTDIR/FJC_CASES_$ENDMTH.$CRTABB; export CASEFILE
```

```
echo "\n-- Extract Cases Data      " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $CASEFILE delimiter '|' >> $SQLCMD
echo "select cs_caseid, cs_date_filed, cs_office, cs_year,      " >> $SQLCMD
echo " cs_number, cs_def_num, cs_type, cs_sub_type,      " >> $SQLCMD
echo " cs_short_title, cs_short_name1, cs_no_plas, cs_short_name2,      " >> $SQLCMD
echo " cs_no_dfts, cs_date_merge, cs_date_term, cs_date_dismiss,      " >> $SQLCMD
echo " cs_date_last_filed, cs_date_reopen,      " >> $SQLCMD
echo " cs_reopen_code, cs_closed, cs_disp_method, cs_judprid_ptr,      " >> $SQLCMD
echo " cs_judge_code, cs_refjud_prid_ptr, cs_juris,      " >> $SQLCMD
echo " cs_restrict_view, cs_gencntr, cs_dncntr, cs_defcntr,      " >> $SQLCMD
echo " cs_jstype, cs_misd_class      " >> $SQLCMD
echo "from u_case, FJC_ECFXTR      " >> $SQLCMD
echo "where (cs_caseid = fjcxtr_caseid);      " >> $SQLCMD
```

```
## Create Associated Cases Filename and Extract Associated Cases Data - On Caseid
```

```
ASCCFILE=$OUTDIR/FJC_ASCCASES_$ENDMTH.$CRTABB; export ASCCFILE
```

```
echo "\n-- Extract Associated Cases Data Based on Caseid      " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $ASCCFILE delimiter '|' >> $SQLCMD
echo "select as_caseid, as_seqno, as_lead_caseid,      " >> $SQLCMD
echo " as_member_caseid, as_group, as_type, as_date_start,      " >> $SQLCMD
echo " as_date_end      " >> $SQLCMD
echo "from asccase, FJC_ECFXTR      " >> $SQLCMD
echo "where (as_caseid = fjcxtr_caseid);      " >> $SQLCMD
```

```
## Create Associated Cases Filename and Extract Associated Cases Data - On Lead Caseid
```

```
ASCLFILE=$OUTDIR/FJC_ASCLEAD_$ENDMTH.$CRTABB; export ASCLFILE
```

```
echo "\n-- Extract Associated Cases Data Based on Lead Caseid      " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $ASCLFILE delimiter '|' >> $SQLCMD
echo "select as_caseid, as_seqno, as_lead_caseid,      " >> $SQLCMD
echo " as_member_caseid, as_group, as_type, as_date_start,      " >> $SQLCMD
echo " as_date_end      " >> $SQLCMD
echo "from asccase, FJC_ECFXTR      " >> $SQLCMD
echo "where (as_lead_caseid = fjcxtr_caseid);      " >> $SQLCMD
```

```
## Create Associated Cases Filename and Extract Associated Cases Data - On Member Caseid
```

```
ASCMFILE=$OUTDIR/FJC_ASCMEMBR_$ENDMTH.$CRTABB; export ASCMFILE
```

```
echo "\n-- Extract Associated Cases Data Based on Member Caseid " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $ASCMFILE delimiter '|' " >> $SQLCMD
echo "select as_caseid, as_seqno, as_lead_caseid, " >> $SQLCMD
echo " as_member_caseid, as_group, as_type, as_date_start, " >> $SQLCMD
echo " as_date_end " >> $SQLCMD
echo "from asccase, FJC_ECFXTR " >> $SQLCMD
echo "where (as_member_caseid = fjcxtr_caseid); " >> $SQLCMD
```

```
## Create Case Flags Filename and Extract Case Flags Data
```

```
FLGFILE=$OUTDIR/FJC_CASEFLGS_$ENDMTH.$CRTABB; export FLGFILE
```

```
echo "\n-- Extract Case Flags Data " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $FLGFILE delimiter '|' " >> $SQLCMD
echo "select cf_caseid, cf_deseqno_ptr, cf_value " >> $SQLCMD
echo "from case_flags, FJC_ECFXTR " >> $SQLCMD
echo "where (cf_caseid = fjcxtr_caseid); " >> $SQLCMD
```

```
## Create Docket Entry Filename and Extract Docket Entry Data
```

```
NTYFILE=$OUTDIR/FJC_DKTENTRY_$ENDMTH.$CRTABB; export NTYFILE
```

```
echo "\n-- Extract Docket Entry Data " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $NTYFILE delimiter '|' " >> $SQLCMD
echo "select de_caseid, de_seqno, de_type, de_sub_type, " >> $SQLCMD
echo " de_document_num, de_date_filed, de_pubacc, " >> $SQLCMD
echo " de_doc_id, de_replaced " >> $SQLCMD
echo "from u_dktentry, FJC_ECFXTR " >> $SQLCMD
echo "where (de_caseid = fjcxtr_caseid); " >> $SQLCMD
```

```
## Create Docket Part Filename and Extract Docket Part Data
```

```
PRTFILE=$OUTDIR/FJC_DKTPART_$ENDMTH.$CRTABB; export PRTFILE
```

```
echo "\n-- Extract Docket Part Data " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $PRTFILE delimiter '|' " >> $SQLCMD
echo "select dp_caseid, dp_seqno, dp_deseqno_ptr, dp_dpseqno_ptr, " >> $SQLCMD
echo " dp_partno, dp_type, dp_sub_type, dp_action_type, " >> $SQLCMD
echo " dp_dispositive, dp_date_filed, dp_date_term " >> $SQLCMD
echo "from u_dktpart, FJC_ECFXTR " >> $SQLCMD
echo "where (dp_caseid = fjcxtr_caseid); " >> $SQLCMD
```

```
## Create Docket Person Filename and Extract Docket Person Data
```

```
PERFILE=$OUTDIR/FJC_DKTPERSON_$ENDMTH.$CRTABB; export PERFILE
```

```
echo "\n-- Extract Docket Person Data " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $PERFILE delimiter '|' " >> $SQLCMD
echo "select dn_caseid, dn_dpseqno_ptr, dn_person_ptr, dn_seqno, " >> $SQLCMD
echo " dn_ref_to, dn_ref_from, dn_date_start, dn_date_term, " >> $SQLCMD
echo " dn_ref_type " >> $SQLCMD
echo "from u_dkt_person, FJC_ECFXTR " >> $SQLCMD
echo "where (dn_caseid = fjcxtr_caseid); " >> $SQLCMD
```

```
## Create Filer Filename and Extract Filer Data
```

```
FLRFILE=$OUTDIR/FJC_FILER_${ENDMTH}.${CRTABB}; export FLRFILE
```

```
echo "\n-- Extract Filer Data " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $FLRFILE delimiter '|' " >> $SQLCMD
echo "select fi_caseid, fi_deseqno_ptr, fi_person_ptr " >> $SQLCMD
echo "from u_filer, FJC_ECFXTR " >> $SQLCMD
echo "where (fi_caseid = fjcxtr_caseid); " >> $SQLCMD
```

```
## Create JS23 Filename and Extract JS23 Data
```

```
J23FILE=$OUTDIR/FJC_JS23_${ENDMTH}.${CRTABB}; export J23FILE
```

```
echo "\n-- Extract JS23 Data " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $J23FILE delimiter '|' " >> $SQLCMD
echo "select j2_caseid, j2_seqno, j2_court_code, j2_magis_caseno, " >> $SQLCMD
echo "      j2_proceeding_date, j2_proceeding_code, j2_open_cit1, j2_open_cnslcd, " >> $SQLCMD
echo "      j2_corp_code, j2_date_disposed, j2_date_term, j2_reopen, " >> $SQLCMD
echo "      cit_citation, cit_level, cit_severity, cit_ao_code " >> $SQLCMD
echo "from js23, citation, FJC_ECFXTR " >> $SQLCMD
echo "where (j2_caseid = fjcxtr_caseid) and " >> $SQLCMD
echo "      (j2_open_cit1 = cit_key); " >> $SQLCMD
```

```
## Create JS56 Filename and Extract JS56 Data
```

```
J56FILE=$OUTDIR/FJC_JS56_${ENDMTH}.${CRTABB}; export J56FILE
```

```
echo "\n-- Extract JS56 Data " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $J56FILE delimiter '|' " >> $SQLCMD
echo "select j5_caseid, j5_seqno, j5_date_filed, " >> $SQLCMD
echo "      j5_date_term, j5_disp_method, j5_reopen_code, j5_nature_of_suit, " >> $SQLCMD
echo "      j5_cause, j5_origin, j5_class_action, j5_us_plaintiff, " >> $SQLCMD
echo "      j5_us_defendant, j5_date_at_issue, j5_date_pretrial, " >> $SQLCMD
echo "      j5_date_trial_begn, j5_date_trial_end, j5_progress_at_ter " >> $SQLCMD
echo "from js56, FJC_ECFXTR " >> $SQLCMD
echo "where (j5_caseid = fjcxtr_caseid); " >> $SQLCMD
```

```
## Create Judge Filename and Extract Judge Data
```

```
JDGFILE=$OUTDIR/FJC_JUDGE_${ENDMTH}.${CRTABB}; export JDGFILE
```

```
echo "\n-- Extract Judge Data " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $JDGFILE delimiter '|' " >> $SQLCMD
echo "select jd_caseid, jd_seqno, jd_jdprid_ptr, jd_ao_code, " >> $SQLCMD
echo "      jd_judge_role, jd_referral_type, jd_type, jd_last_name, " >> $SQLCMD
echo "      jd_initials, jd_date_start, jd_date_end " >> $SQLCMD
echo "from u_judge, FJC_ECFXTR " >> $SQLCMD
echo "where (jd_caseid = fjcxtr_caseid); " >> $SQLCMD
```

```
## Create Motion Filename and Extract Motion Data
```

```
MTNFILE=$OUTDIR/FJC_MOTION_${ENDMTH}.${CRTABB}; export MTNFILE
```

```

echo "\n-- Extract Motion Data " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $MTNFILE delimiter '|' >> $SQLCMD
echo "select mo_caseid, mo_dpseqno, mo_motion_text, mo_term_dpseqno " >> $SQLCMD
echo "from u_motion, FJC_ECFXTR " >> $SQLCMD
echo "where (mo_caseid = fjcextr_caseid); " >> $SQLCMD

## Create Party Filename and Extract Party Data

PTYFILE=$OUTDIR/FJC_PARTY_$ENDMTH.$CRTABB; export PTYFILE

echo "\n-- Extract Party Data " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $PTYFILE delimiter '|' >> $SQLCMD
echo "select py_caseid, py_seqno, py_aliasseq, py_def_num, " >> $SQLCMD
echo " py_type, py_role, py_claim_no, py_prose, " >> $SQLCMD
echo " py_corporation, py_last_name, py_initials, py_date_severance, " >> $SQLCMD
echo " py_start_date, py_end_date, py_person_ptr " >> $SQLCMD
echo "from u_party, FJC_ECFXTR " >> $SQLCMD
echo "where (py_caseid = fjcextr_caseid); " >> $SQLCMD

## Create Codes Filename and Extract Codes Data

CODFILE=$OUTDIR/FJC_CODES_$ENDMTH.$CRTABB; export CODFILE

echo "\n-- Extract Codes Data " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $CODFILE delimiter '|' >> $SQLCMD
echo "select co_type, co_code, co_translation, co_misc, co_other " >> $SQLCMD
echo "from codes;" >> $SQLCMD

## Create Document Type Filename and Extract Document Type Data

DOCFILE=$OUTDIR/FJC_DOCTYPE_$ENDMTH.$CRTABB; export DOCFILE

echo "\n-- Extract Document Type Data " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Unload to $DOCFILE delimiter '|' >> $SQLCMD
echo "select do_type, do_sub_type, do_summary_text, " >> $SQLCMD
echo " do_ref_type, do_ref_sub_type, do_case_type " >> $SQLCMD
echo "from document_type;" >> $SQLCMD

## Remember to drop temporary table FJC_ECFXTR

echo "\n-- Drop temporary table FJC_ECFXTR " >> $SQLCMD
echo "\n" >> $SQLCMD
echo "Drop table FJC_ECFXTR; " >> $SQLCMD

## End of SQL file generation.
## Instruct users to execute the SQL file in dbaccess and send the output files to FJC

echo "\nThe data extraction command file $SQLCMD " | tee -a $LOGFIL
echo "has been generated." | tee -a $LOGFIL
date | tee -a $LOGFIL
echo "\nPlease use dbaccess to run this SQL program against your CM/ECF database: " | tee -a $LOGFIL
echo "\nThis extraction exercise will generate these output and processing files: " | tee -a $LOGFIL
echo " $CASEFILE " | tee -a $LOGFIL

```

```
echo " $ASCCFILE " | tee -a $LOGFIL
echo " $ASCLFILE " | tee -a $LOGFIL
echo " $ASCMFILE " | tee -a $LOGFIL
echo " $FLGFILE " | tee -a $LOGFIL
echo " $NTYFILE " | tee -a $LOGFIL
echo " $PRTFILE " | tee -a $LOGFIL
echo " $PERFILE " | tee -a $LOGFIL
echo " $FLRFILE " | tee -a $LOGFIL
echo " $J23FILE " | tee -a $LOGFIL
echo " $J56FILE " | tee -a $LOGFIL
echo " $JDGFILE " | tee -a $LOGFIL
echo " $MTNFILE " | tee -a $LOGFIL
echo " $PTYFILE " | tee -a $LOGFIL
echo " $CODFILE " | tee -a $LOGFIL
echo " $DOCFILE " | tee -a $LOGFIL
echo " $LOGFIL " | tee -a $LOGFIL
echo " $SQLCMD " | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
echo "Once you've executed the SQL program with dbaccess, " | tee -a $LOGFIL
echo "please FTP these files to the Federal Judicial Center " | tee -a $LOGFIL
echo "using the instructions below: " | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
echo "ftp wind.fjc.dcn (or ftp 156.132.47.249)" | tee -a $LOGFIL
echo "user: anonymous" | tee -a $LOGFIL
echo "password: <your email address>" | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
echo "cd incoming" | tee -a $LOGFIL
echo "prompt (turns off interactive mode)" | tee -a $LOGFIL
echo "mput $OUTDIR/FJC_* (uploads multiple files at once)" | tee -a $LOGFIL
echo "quit" | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
echo "If you have any questions, please contact:" | tee -a $LOGFIL
echo "George Cort at 202-502-4087 or gcort@fjc.gov." | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
echo "Thank you for your help with this project." | tee -a $LOGFIL
echo "\n" | tee -a $LOGFIL
```

```
##
## End of DCCWS_ECFXTR
##
```

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

Example .SQL Program
Generated by DCCWS_ECFXTR

Data Extractions in the CM/ECF Courts

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

```

-- Set Lock Wait Command to Assist Creation of the Temporary Table

set lock mode to wait 300;

-- Create Temporary Table FJC_ECFXTR

select unique cs_caseid as fjcctr_caseid
from u_case
where (cs_restrict_view != 40 and
      (cs_date_term between '01/01/2002' and '12/31/2002' or
       cs_caseid in (select j2_caseid from js23
                     where j2_date_term between '01/01/2002' and '12/31/2002')) or
       cs_caseid in (select j5_caseid from js56
                     where j5_date_term between '01/01/2002' and '12/31/2002'))))
into temp FJC_ECFXTR with no log;

-- Create Index on Temporary Table FJC_ECFXTR

create index fjc_ext on FJC_ECFXTR ( fjcctr_caseid );

-- Extract Cases Data

Unload to /tmp/FJC_ECFXTR/FJC_CASES_DEC.TST delimiter '|'
select cs_caseid, cs_date_filed, cs_office, cs_year,
       cs_number, cs_def_num, cs_type, cs_sub_type,
       cs_short_title, cs_short_name1, cs_no_plas, cs_short_name2,
       cs_no_dfts, cs_date_merge, cs_date_term, cs_date_dismiss,
       cs_date_last_filed, cs_date_reopen,
       cs_reopen_code, cs_closed, cs_disp_method, cs_judprid_ptr,
       cs_judge_code, cs_refjud_prid_ptr, cs_juris,
       cs_restrict_view, cs_gencntr, cs_dncntr, cs_defcntr,
       cs_jstype, cs_misd_class
from u_case, FJC_ECFXTR
where (cs_caseid = fjcctr_caseid);

-- Extract Associated Cases Data Based on Caseid

Unload to /tmp/FJC_ECFXTR/FJC_ASCCASES_DEC.TST delimiter '|'
select as_caseid, as_seqno, as_lead_caseid,
       as_member_caseid, as_group, as_type, as_date_start,
       as_date_end
from asccase, FJC_ECFXTR
where (as_caseid = fjcctr_caseid);

-- Extract Associated Cases Data Based on Lead Caseid

Unload to /tmp/FJC_ECFXTR/FJC_ASCLEAD_DEC.TST delimiter '|'
select as_caseid, as_seqno, as_lead_caseid,
       as_member_caseid, as_group, as_type, as_date_start,
       as_date_end

```

```
from asccase, FJC_ECFXTR
where (as_lead_caseid = fjcxtr_caseid);
```

```
-- Extract Associated Cases Data Based on Member Caseid
```

```
Unload to /tmp/FJC_ECFXTR/FJC_ASCMEMBR_DEC.TST delimiter '|'
select as_caseid, as_seqno, as_lead_caseid,
       as_member_caseid, as_group, as_type, as_date_start,
       as_date_end
from asccase, FJC_ECFXTR
where (as_member_caseid = fjcxtr_caseid);
```

```
-- Extract Case Flags Data
```

```
Unload to /tmp/FJC_ECFXTR/FJC_CASEFLGS_DEC.TST delimiter '|'
select cf_caseid, cf_deseqno_ptr, cf_value
from case_flags, FJC_ECFXTR
where (cf_caseid = fjcxtr_caseid);
```

```
-- Extract Docket Entry Data
```

```
Unload to /tmp/FJC_ECFXTR/FJC_DKTENTRY_DEC.TST delimiter '|'
select de_caseid, de_seqno, de_type, de_sub_type,
       de_document_num, de_date_filed, de_pubacc,
       de_doc_id, de_replaced
from u_dktentry, FJC_ECFXTR
where (de_caseid = fjcxtr_caseid);
```

```
-- Extract Docket Part Data
```

```
Unload to /tmp/FJC_ECFXTR/FJC_DKTPART_DEC.TST delimiter '|'
select dp_caseid, dp_seqno, dp_deseqno_ptr, dp_dpseqno_ptr,
       dp_partno, dp_type, dp_sub_type, dp_action_type,
       dp_dispositive, dp_date_filed, dp_date_term
from u_dktpart, FJC_ECFXTR
where (dp_caseid = fjcxtr_caseid);
```

```
-- Extract Docket Person Data
```

```
Unload to /tmp/FJC_ECFXTR/FJC_DKTPERSON_DEC.TST delimiter '|'
select dn_caseid, dn_dpseqno_ptr, dn_person_ptr, dn_seqno,
       dn_ref_to, dn_ref_from, dn_date_start, dn_date_term,
       dn_ref_type
from u_dkt_person, FJC_ECFXTR
where (dn_caseid = fjcxtr_caseid);
```

```
-- Extract Filer Data
```

```
Unload to /tmp/FJC_ECFXTR/FJC_FILER_DEC.TST delimiter '|'
select fi_caseid, fi_deseqno_ptr, fi_person_ptr
from u_filer, FJC_ECFXTR
```

```
where (fi_caseid = fjcxhr_caseid);
```

```
-- Extract JS23 Data
```

```
Unload to /tmp/FJC_ECFXTR/FJC_JS23_DEC.TST delimiter '|'
select j2_caseid, j2_seqno, j2_court_code, j2_magis_caseno,
       j2_proceeding_date, j2_proceeding_code, j2_open_cit1, j2_open_cnslcd,
       j2_corp_code, j2_date_disposed, j2_date_term, j2_reopen,
       cit_citation, cit_level, cit_severity, cit_ao_code
from js23, citation, FJC_ECFXTR
where (j2_caseid = fjcxhr_caseid) and
      (j2_open_cit1 = cit_key);
```

```
-- Extract JS56 Data
```

```
Unload to /tmp/FJC_ECFXTR/FJC_JS56_DEC.TST delimiter '|'
select j5_caseid, j5_seqno, j5_date_filed,
       j5_date_term, j5_disp_method, j5_reopen_code, j5_nature_of_suit,
       j5_cause, j5_origin, j5_class_action, j5_us_plaintiff,
       j5_us_defendant, j5_date_at_issue, j5_date_pretrial,
       j5_date_trial_begn, j5_date_trial_end, j5_progress_at_ter
from js56, FJC_ECFXTR
where (j5_caseid = fjcxhr_caseid);
```

```
-- Extract Judge Data
```

```
Unload to /tmp/FJC_ECFXTR/FJC_JUDGE_DEC.TST delimiter '|'
select jd_caseid, jd_seqno, jd_jdprid_ptr, jd_ao_code,
       jd_judge_role, jd_referral_type, jd_type, jd_last_name,
       jd_initials, jd_date_start, jd_date_end
from u_judge, FJC_ECFXTR
where (jd_caseid = fjcxhr_caseid);
```

```
-- Extract Motion Data
```

```
Unload to /tmp/FJC_ECFXTR/FJC_MOTION_DEC.TST delimiter '|'
select mo_caseid, mo_dpseqno, mo_motion_text, mo_term_dpseqno
from u_motion, FJC_ECFXTR
where (mo_caseid = fjcxhr_caseid);
```

```
-- Extract Party Data
```

```
Unload to /tmp/FJC_ECFXTR/FJC_PARTY_DEC.TST delimiter '|'
select py_caseid, py_seqno, py_aliasseq, py_def_num,
       py_type, py_role, py_claim_no, py_prose,
       py_corporation, py_last_name, py_initials, py_date_severance,
       py_start_date, py_end_date, py_person_ptr
from u_party, FJC_ECFXTR
where (py_caseid = fjcxhr_caseid);
```

```
-- Extract Codes Data
```

```
Unload to /tmp/FJC_ECFXTR/FJC_CODES_DEC.TST delimiter '|'
select co_type, co_code, co_translation, co_misc, co_other
from codes;
```

```
-- Extract Document Type Data
```

```
Unload to /tmp/FJC_ECFXTR/FJC_DOCTYPE_DEC.TST delimiter '|'
select do_type, do_sub_type, do_summary_text,
       do_ref_type, do_ref_sub_type, do_case_type
from document_type;
```

```
-- Drop temporary table FJC_ECFXTR
```

```
Drop table FJC_ECFXTR;
```