

---

# **CALEN9: A Calendaring and Assignment System for Courts of Appeals**

---

## **COURTRAN II System Description**

---

68571

## THE FEDERAL JUDICIAL CENTER

### Board

The Chief Justice of the United States  
*Chairman*

Judge Ruggero J. Aldisert  
*United States Court of Appeals  
for the Third Circuit*

Judge Frank J. McGarr  
*United States District Court  
Northern District of Illinois*

Judge Robert H. Schnacke  
*United States District Court  
Northern District of California*

Judge Aubrey E. Robinson, Jr.  
*United States District Court  
District of Columbia*

Judge John C. Godbold  
*United States Court of Appeals  
for the Fifth Circuit*

William E. Foley  
*Director of the Administrative  
Office of the United States Courts*

---

### Director

A. Leo Levin

### Deputy Director

Joseph L. Ebersole

### Division Directors

Kenneth C. Crawford  
*Continuing Education  
and Training*

William B. Eldridge  
*Research*

Charles W. Nihan  
*Innovations  
and Systems Development*

Alice L. O'Donnell  
*Inter-Judicial Affairs  
and Information Services*

1520 H Street, N.W.  
Washington, D.C. 20005  
Telephone 202/633-6011



NCJRS

JUN 16 1980

ACQUISITIONS

CALEN9: A CALENDARING AND ASSIGNMENT  
SYSTEM FOR COURTS OF APPEALS

By Michael R. Leavitt

Federal Judicial Center  
October, 1978

## CALEN9: A CALENDARING AND ASSIGNMENT SYSTEM FOR COURTS OF APPEALS

The purpose of this report is to describe "Calen9," a calendaring and assignment computer software system designed for the Ninth Circuit Court of Appeals,<sup>1</sup> and to suggest some possible future applications of this software. To summarize, Calen9 is a computer program that the staff attorneys' office in the Ninth Circuit can use as an aid in creating calendars of cases for hearing. Information about cases to be heard is stored in the computer, and used by the program to determine, according to court-specified procedures, the priorities of unheard cases, and how they should be organized for most efficient hearing.

The only important difference between the present version of the system and its original

---

1. This program has been prepared by the Research Division. Once AIMS (Appellate Information Management System) development is complete, Calen9 will be a part of that system and will be available for use in other circuits.

design (appendix A) is that use of the ability to empanel judges has been deferred by the courts. The principal reason for the deferral is that the criteria for selection remain unsettled. Since there is a desire on the part of the court to move into longer-range judge assignments, resolution of the ambiguities for a period of only a month or two, was deemed unnecessary.

This report begins by discussing how the program works, including a description of the input requirements and the calendaring procedure (the algorithm). The physical operating procedures are discussed next, followed by more detail on the program's operation. Finally, some potential future applications are presented. A brief description of modifications to the system since it began to be used as an operational tool is presented as appendix B.

## How the Program Works

### Input Requirements

The program that has been created, called Calen9, works interactively. This means that an

operator, sitting at a typewriter-like terminal, is asked by the computer to provide certain information that then becomes the basis for the program's decisions. The most important pieces of information are:

1. The location of the sitting
2. The date of the sitting
3. The number of calendars to be selected
4. Whether or not "difficult" cases are to be chosen
5. Additions or subtractions from the program-assigned priority for selected cases.

The location of the sitting is used to let the program conform to the court's rules requiring that cases from particular districts may only be heard (or may not be heard) in certain cities. The date of the sitting is used in the determination of the cases' priorities, as will be discussed below. The number of calendars determines, in part, the number of cases to be finally included.

The court has adopted the practice that most regular sittings will hear only "difficult" cases with the "simple" ones being disposed of in panels created especially for that purpose. (The difficulty of a case affects the time that it takes a

three-judge panel to hear it.) The program needs to know whether a particular case is difficult or not. Case difficulty is determined by specially trained staff attorneys who assign from one to ten points to each case. (Actually, only the numbers one, three, five, seven, and ten are currently in use.) One-point cases are sufficiently simple usually not to require oral argument, while ten-pointers can dominate a particular sitting's activity. The court has determined that a given panel can hear fifteen<sup>2</sup> points' worth of cases, no matter how the points are divided among the cases (although no more than six to eight cases per morning or afternoon session are usually held, and judges rarely sit more often than one session per day).

The procedure for determining the priority of a given case is of primary importance. The

---

2. Since the beginning of program operation, the Court has changed to sixteen-point calendars, and the program has been modified accordingly. For the remainder of this report, the initial characteristics will be described, with changes detailed in appendix B.

program gives criminal cases a higher priority than any other type of case: civil or administrative. Certain civil cases have a statutory priority, but the court has interpreted this to be a relative weighting: an old nonpriority case might be heard before a new one with priority.

The following procedure is used for setting priorities. All cases are given 1 point for every month that they are in the inventory for the first year. For the second and third years, cases receive  $1 \frac{1}{4}$  points per month, and after three years, cases receive  $1 \frac{1}{2}$  points per month. Priority civil cases are given an additional 20 points, and criminal cases 1,000 points. It must be emphasized that these point values are arbitrary, and are used to get the effect desired by the court. Their acceptability is based on the fact that they appear, to the court, to have the "right" effect. (Modifying the program to change either the points or the procedure for applying the points is straightforward if needed). Finally, certain cases are occasionally "ordered on" to



a particular calendar even though their normal priority would exclude them. The program can accept modifications to the usual calculations to have this effect.

### Calendaring

Once the case data are read and priorities calculated, a pool of cases, based on the number of calendars needed, is selected from the inventory. The size of the pool is increased by a factor that permits the program some flexibility in putting cases together. The main reason for a larger pool than necessary is that when point counts of only three, five, seven and ten are used to create fifteen-point calendars, it is possible to have certain cases omitted for nonsubstantive reasons. For example, only five-point cases can go with ten-point cases, and if there are more ten-pointers than five-pointers (an unlikely event, but a clear example of the kind of problem that must be faced), then some ten-point cases will not get assigned, and thus there must be some extra capacity to fill the number of calendars

requested. A pool that is 115 percent of the required size will usually permit the proper number of calendars to be formed.

After the pool is created, the cases are sorted by their difficulty in such a way as to minimize the chance of the above kind of mismatch. Within each point category, higher priority cases are ranked first. A calendar is created by finding the highest priority case not already assigned to a calendar and matching other cases to it.

The matching process first attempts to group cases that are on the same general subject<sup>3</sup> and that come from the same district court. When a case is inventoried, a trained staff attorney classifies it by its subject category, like civil rights, habeas corpus, labor, etc. The court has determined that similarity in subject matter is an important grouping criterion. The preference for cases from the same district being together was intended to make it easier to have designated dis-

---

3. Similarity in subject has subsequently been eliminated by the court as a matching characteristic.

trict court judges sit on panels, since they are barred from hearing any case from their own districts.

After all such cases have been grouped, the program next relaxes the same-district criterion and again searches the file to make the calendars. If not all calendars have been assigned, the same-subject rule is relaxed, and only the same-district cases are considered. Finally, if any calendars remain to be filled, no criteria other than fitting into fifteen-point groups are used.

When all calendars have been filled, they may be printed at the terminal, and the operator is given the opportunity to modify priorities according to court requirements. For example, the priority of cases that the court has ordered on can be increased. If any changes are made, the entire calendaring process must be redone. When no more changes are needed, the program stops.

## Operating Procedures

Before the program can be run, certain other files must have already been prepared. The most important of these is the case inventory. This is a regularly updated file of information (called CASES.INV in the computer) on all cases that have been docketed in the clerk of the court's office, but have not yet been calendared. The following information is maintained for all such cases:

1. The case name (anything as long as it is sixty characters or less)
2. The docket number, in the form 77-xxxx
3. The type of case (CRIM, CIV or ADMIN)
4. The difficulty of the case (1-A, 10-E, etc.)
5. The district (or administrative agency) from which the case was appealed
6. The judge (or district, if an administrative agency), from whom the case was appealed
7. The subject of the case.

In the program's present version the operator must be very careful to put the information in the proper columns, although it is expected that this restriction will be relaxed in some future implementation of the system.

One additional file, that says which districts' cases may be heard at which hearing sites, must be present. The rows of the table are the districts, the columns are the hearing locations (cities), and the entries are the letters T and F. T(true) means that a case from that district (row) may be heard at that site (column). F(false) means that the district may not have its cases heard at that site. This file is presented in Table 1.

TABLE 1

## THE DISTRICT-HEARING SITE FILE

DISTRICTS	HEARING SITES (CITIES)									
	SF	LA	PO	SE	PH	HO	GU	SD	AL	OT
D. ARIZONA	F	T	F	F	T	F	F	F	F	T
D. HAWAII	T	F	F	F	F	T	F	F	F	T
D. GUAM	T	F	F	F	F	T	T	F	F	T
D. MARIANNAS	T	F	F	F	F	T	F	F	F	T
S. D. CAL.	F	T	F	F	F	F	F	T	F	T
C. D. CAL.	F	T	F	F	F	F	F	F	F	T
N. D. CAL.	T	F	F	F	F	F	F	F	F	T
E. D. CAL.	T	F	F	F	F	F	F	F	F	T
D. NEVADA	T	F	F	F	F	F	F	F	F	T
D. ALASKA	T	F	F	F	F	F	F	F	T	T
D. IDAHO	T	F	T	T	F	F	F	F	F	T
D. OREGON	T	F	T	F	F	F	F	F	F	T
D. MONTANA	T	F	T	T	F	F	F	F	F	T
W. D. WASH.	T	F	F	T	F	F	F	F	F	T
E. D. WASH.	T	F	F	T	F	F	F	F	F	T

## Description of the Program

Calen9 was written in as structured a manner as the DEC FORTRAN-10 language will allow, without resorting to excessively awkward formalisms. The main program is simply a "driver" that calls the following four functional subroutines in order:

<u>Subroutine</u>	<u>Function</u>
INIT	Initializes all variables and parameters and asks the terminal operator about the location, date, case types and number of calendars for the sitting to be calendared.
ACDTIN	Reads data from the case inventory file, checks the data for consistency and correctness, calculates values of some new variables from the input, and, if everything is correct, tells the operator how many cases have been read, as well as the number of criminal, priority civil, and non-priority civil cases.
CALEND	Does the main work of the program. Sorts cases by priority, searches for highest priority cases, combines other cases according to the rules described above, and types out the calendars after all the work has been completed.

ACDTOT      Writes out non-calendared cases into a new version of the inventory file, and appends calendared cases to a new version of the "calendared cases" file.

"Correctness" is used in a very limited sense here and in the remainder of this report. The program has no way of knowing, of course, that the operator was wrong in typing "LA" when "SF" was intended. The program does, however, know that if "SG" is typed it is wrong, since there is no city with that abbreviation. The program checks correctness by looking at the range of possibilities and seeing if a given answer, or variable value, is within that range.

#### INIT

INIT begins by asking the terminal operator when and where the sitting will take place, after typing out some heading information, including the current date, time, and date of the last major program revision. The date and place are checked for correctness: the month must be between "1" and "12", the year must be between 1978 and 1985, and

the place must be one of the following two-letter abbreviations:

<u>Initials</u>	<u>Sitting City</u>
SF	San Francisco
LA	Los Angeles
PO	Portland
SE	Seattle
PH	Phoenix
HO	Honolulu
GU	Guam
SD	San Diego
AL	Alaska
OT	Other

If any of the information is not correct, the operator is asked to retype it. The special "truth table" that specifies districts from which cases can be heard at each "sitting city," is then read. The information in this file will filter the cases as they are read in ACDTIN. The operator is then asked whether 1-A cases, non-1-A cases, or all cases will be heard in the current sitting, and this answer, too, will filter cases as they are read later. Finally, the operator is asked how many calendars are to be formed. The program calculates the number of points implied by that number of calendars (fifteen points per calendar) and so informs the operator by way of confirming the



entry. The subroutine is then over, and control is returned to the driver, which calls ACDTIN.

#### ACDTIN

ACDTIN reads CASES.INV, the case inventory file, determines eligibility, checks the data for correctness, calculates priorities for each case and tells the operator the overall result of the effort. Specifically, the program reads every case in the inventory file. The eligibility of a case for calendaring, based on its district (compared to the location of the sitting) and its difficulty, is then determined. The case is kept in the memory, even if it is not eligible, so it can be written to a new inventory file in ACDTOT. If the case is eligible, its age is calculated from the difference between the date it entered the inventory and the date of the sitting.

After all cases have been read (and no errors encountered, at which point the program would simply stop), the priority for each case is calculated. Ineligible cases are given a priority of 0 points. Of the eligible cases, criminal cases

are given a very high priority (currently 1,000 points); priority civil cases are given a moderate initial priority (corresponding, at present, to 20 points, or nineteen months in the inventory). A variable priority calculated from the age of the case is then added to the base priority as described above.

The operator is then told how many cases were read and how many were eligible. The number of criminal, priority civil and nonpriority civil (out of the eligible) are also displayed, after which the subroutine returns control to the driver, which, in turn, yields to CALEND.

#### CALEND

CALEND does its work by establishing a linkage system for each calendar it puts together. Each case has a variable that "points to" the next case in the same calendar. Each calendar's basic information indicates only its first case. The last case in a calendar is identified by having a "pointer" with a value of zero. Further, each

case has a variable called IN, whose value is as follows:

<u>Value</u>	<u>Meaning</u>
IN = -1	Case is ineligible for calendaring
IN = 0	Case is eligible for calendaring, but has not been calendared
IN > 0	Case has been calendared, value of IN is the number of the calendar.

At the beginning of CALEND, after initialization, the cases are sorted by their priority. (All sorting is done by simple exchanging of pointers.) The next step is determining a "pool" of eligible cases from which the calendar will be created. Creating a pool is an important step, since later stages re-sort cases on other criteria in such a way as to permit otherwise low-priority calendared before their more deserving, but perhaps harder to fit, cousins. The pool is simply a fraction (currently 115 percent) of the needed number of points to fill the number of calendars specified by the operator. That number is determined by simply summing across all cases, in order of their priority, until the cutoff point is

reached. The operator is then told the priority of the last case in the pool.

The cases in the pool are then re-sorted so that the most difficult cases are "fitted" first. This was found to be a needed modification of the priority system, since otherwise those cases had an inordinately difficult time getting calendared.

The next step is actually forming the calendars. All cases in the pool are examined up to four times, each time with more relaxed criteria for grouping, until the requisite number of calendars are formed. Every cycle begins by selecting the first yet-uncalendared case in the pool as the base for the calendar. Other cases are then compared to the base according to the criteria.

The first time through permits a case to be linked to a calendar if it comes from the same district as the base, and is about the same subject. (The reasons for these criteria are discussed above in the "Calendaring" section of "How the Program Works.") When enough cases have been added to the calendar to produce fifteen points

(or whatever value is set by the court), the cycle begins again by selecting the next highest priority case still uncalendared. The second pass relaxes the "same district" criterion, and only requires that cases be about the same subject as the base case. The third time, the district criterion is reintroduced, and the subject criterion is removed. The final time, if necessary, removes all criteria, and requires only that cases add up to fifteen points to be included.

Whenever fifteen points are reached, all cases in the calendar have their "IN" variable set equal to the "calendar number," and the next calendar's number is set equal to the previous calendar's, plus one. This is the point at which the program determines whether enough calendars have been formed. The process terminates either when the desired number of calendars have been put together, or when the last search through the pool has been completed without obtaining the desired number of calendars. This will happen occasionally, particularly when there are a large number

of difficult cases in a pool. These cases are difficult to fit when 1-A cases are not eligible for inclusion.

The operator is then asked whether the calendars should be displayed at the terminal. If not, they are stored in a file for later retrieval. At this point, the operator is given the chance to selectively modify individual case priorities to move low-priority cases into the pool, or high-priority ones out. This would presumably be done only under special circumstances, such as a particular case being ordered on a calendar by the court. If changes are made to the priorities, the entire subroutine is rerun from the beginning. This implies that although a case can be ordered on, it cannot be placed on a particular calendar with any degree of ease. The CALEND subroutine then returns control to the driver, that calls the ACDTOT subroutine.

#### ACDTOT

ACDTOT simply writes two files that represent the effect of the run on the inventory. The

first file is an updated version of the case inventory that could be used directly as input to another run of Calen9. The difference, as one might expect, is that it excludes those cases that have been calendared. It names the new file NCASES.INV, so that if the calendar is unsatisfactory for any reason, making the new inventory has not destroyed the contents of the old inventory. Immediately after a successful calendar has been run, the old CASES.INV can be deleted, renamed, or archived (for historical purposes, perhaps), and the NCASES.INV file can be renamed as CASES.INV.

The other file that is output is a cumulative list of all cases that have been calendared. This, called NCASES.CAL is created by reading CASES.CAL, immediately writing it to NCASES.CAL, and then appending to NCASES.CAL those cases that were calendared. When ACDTOT completes these activities, it returns control to the driver, which stops.

### Future Applications

Several additional applications of the data have already been begun by court personnel. One of them, an aid to report processing, is a simple use of our Statistical Package for the Social Sciences (SPSS) to summarize and tabulate the frequency of different kinds of cases in the inventory. A single statement is typed for each new report desired, and each report has saved several hours of staff attorney time. Additional reports are fairly easy to prepare, and court personnel can be taught to set them up with little difficulty.

The court has indicated that it would like to keep track of some additional information associated with each case, in particular, information concerning the judge to whom the case is assigned after it is heard. This information might be used to help court administrators improve the balance of judges' workloads. Reports, similar to



those mentioned above, could be generated with little additional difficulty.

The method used to store the data in the computer facilitates its modification. For example, the court may wish to change the labels of the subject headings. The standard system editing facilities (and data base management software) can be used to make the modifications to individual cases, as well as to groups of cases. Until the court becomes completely comfortable with its categories, making such changes can be expected, and the ability to do so with some facility should be welcomed.

Although modifications to Calen9 to fit other courts' needs is not an insuperably difficult technical task, it is not anticipated that this will be done at present. The Federal Judicial Center's Courtran project will be implementing its Appellate Information Management System (AIMS) within the next year. This system will provide a much more sophisticated and complete method of managing data generated by the courts of

appeals. Further major development of Calen9 should take place within the AIMS framework, and thus will be suspended until that framework is more fully defined.

### Summary

The Ninth Circuit calendaring project was designed to create and deliver a working computer program to assist the Court of Appeals for the Ninth Circuit in managing its case backlog. Specifically, the project was designed to group cases into calendars based on their difficulty, their subject matter, and secondarily, the district from which they originated. In addition, a system for grouping judges into panels to hear the cases so calendared was to be developed. Both of these capabilities have been completed and delivered, although the court has chosen to make use of only the case calendaring facility at this time. Although modifications to the system can be made, it is expected that any extensive changes will await

## the implementation of the Courtran Appellate Information Management System.

1. The Courtran Appellate Information Management System (CAIMS) is a computerized system designed to manage appellate information. It is a part of the Courtran Appellate Information Management System (CAIMS) project.

2. The CAIMS project is a part of the Courtran Appellate Information Management System (CAIMS) project.

3. The CAIMS project is a part of the Courtran Appellate Information Management System (CAIMS) project.

4. The CAIMS project is a part of the Courtran Appellate Information Management System (CAIMS) project.

5. The CAIMS project is a part of the Courtran Appellate Information Management System (CAIMS) project.

6. The CAIMS project is a part of the Courtran Appellate Information Management System (CAIMS) project.

7. The CAIMS project is a part of the Courtran Appellate Information Management System (CAIMS) project.

8. The CAIMS project is a part of the Courtran Appellate Information Management System (CAIMS) project.

9. The CAIMS project is a part of the Courtran Appellate Information Management System (CAIMS) project.

10. The CAIMS project is a part of the Courtran Appellate Information Management System (CAIMS) project.

UNITED STATES GOVERNMENT

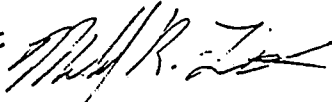
# Memorandum

APPENDIX A

TO : Judge Hufstedler

DATE: March 2, 1977

FROM : Michael R. Leavitt



SUBJECT: Progress Report on Ninth Circuit Case Assignment Problem

At our meeting in San Francisco on January 31, we agreed that I would provide a progress report to you on the Judicial Center's possible contribution to the Case Management Project. We have been actively considering two alternative tracks on which we might proceed: a fairly speedy "specific solution" to your immediate problem or a general process for handling the judge-case assignment problem. My recommendation is that, with the Judicial Center's approval, we go ahead with the "specific solution," reserving generalization for later. To that end I am sending this memorandum to Judge Hoffman. He will want to know the response of your court before taking final action.

## Case-Judge Assignments

In order to set a context for my suggestions, I would like to review the situation as I understand it. It is possible to think of the process by which groups of cases get assigned to groups of appeals court judges as consisting of three steps:

- (1) Organizing judges into panels;
- (2) Organizing cases into calendars;
- (3) Assigning calendars to panels (or vice-versa).



March 2, 1977

Every circuit may well have different rules governing each of the three steps, but all circuits accomplish the steps in some manner.

Considerations affecting judge-panel (1) organization include:

- \* equalizing the number of times any two active judges sit together;
- \* minimizing the number of times district judges are needed to complete a panel;
- \* maintaining a consistent rate of sittings per month (excluding vacation and conference times);
- \* taking geography into account by compressing sitting times for panels with distant judges on them;
- \* maintaining given proportions of sitting sites (e.g., equality between Los Angeles and San Francisco).

Any solution -- "quick and specific" or "general and comprehensive" -- must account for some or all of these factors.

"Case-into-calendar" organizations must recognize other factors:

- \* combining cases with similar subject matter to conserve preparation time;
- \* combining cases whose differing degrees of complexity permit them to be heard at the same sitting;
- \* statutory and local priorities;
- \* ability of court personnel to override any "machine-generated" combination.

The Ninth Circuit has a highly developed technique for grouping cases that is closely integrated into its inventory management system. Another circuit might permit staff to make the combinations less systematically. In either situation, deliberate efforts are made to "balance" the load for a given sitting.

March 2, 1977

Finally, the process by which panels (already-determined groups of judges) are linked to calendars (already-determined groups of cases) must recognize the following:

- \* Random assignments should be the rule,
- \* A panel with a district judge on it should not get a calendar with a case from that judge's district,
- \* Equalization of sitting time among active judges should be achieved within every twelve-month period.

#### General "Solution" to Appellate Case Assignments

A moderate expenditure of time and money could result in the creation of a set of computer programs and associated clerical procedures that would substantially aid the processes outlined above. The programs could combine some "data-base management" techniques with fairly straightforward computational and reporting procedures. The results would theoretically be usable in many different courts by modifying specific programs but leaving the general design the same.

The system would consist of at least the following six components:

- (1) Case (inventory) management--in which cases would be logged into the system with whatever coding information is required; the information about each case could be modified as needed, and the case would be "logged out" as it was disposed of.
- (2) Judge information--in which salient characteristics of each judge (active, senior, visitor) would be maintained--including home location, backlog status, number of cases, number of opinions written, number of joint sittings with other judges, etc.
- (3) Panel formation--whenever it is necessary to group judges into new panels, this program would interact with a court staff member to be certain that the necessary conditions are met. The program would also "update" the judge information bank.

March 2, 1977

- (4) Calendar formation--in which appropriate (as defined by the court) cases are selected from the inventory to be heard, and the inventory information is updated.
- (5) Assignment of panels to calendars--in which the collection of panels and the group of calendars are brought together.
- (6) Reports--a collection of programs with which the administrative personnel could "query" the various sets of data in the system.

A very rough estimate is that perhaps nine weeks would be needed to design such a collection of programs, followed by seven weeks of programming and testing. Including time for design approvals, changes and "unknown" delays, a good six months elapsed time would seem called for.

#### Specific "Solution"

The immediate problem facing the Ninth Circuit, however, can be managed with substantially less effort, if "just" managing the caseload inventory is desired. A program could be written without great difficulty (or time) that would permit:

- \* retrieval of cases' docket numbers based on assigned descriptors (vocabulary, keywords),
- \* entry and deletion of cases into and from the inventory,
- \* calculation of characteristics of all or sections of the inventory (average docketing points, length of time in inventory, etc.),
- \* tentative grouping of cases into calendars with a given number and distribution of points.

This program could be run by a person with relatively little training in computers, and could be maintained (fixed, improved) by Research staff personnel. A rough estimate of time would be one week of design and two weeks of programming and testing. If the project could be given 40% of a person's time, this implies about two months elapsed time.

Recommendation

Given the not insubstantial expense and the relatively uncertain benefits of the "general solution," the following is recommended:

- \* A version of the "specific solution" be developed after an additional two days of consultation with Ninth Circuit staff.
- \* No major additions be planned for this program; rather, if additional capabilities are needed, they should be considered within the framework of a "general solution."
- \* An explicit evaluation of the program be made by Ninth Circuit staff no later than six months after it is implemented.



## APPENDIX B

### PROGRAM MODIFICATIONS SINCE IMPLEMENTATION

Since its implementation, Calen9 has been modified in a number of important ways, as a direct result of the courts's operational experiences, and at the court's request. The changes, listed below, will be described briefly in the remainder of this appendix:

1. Efficient selection of sixteen-point panels
2. An output file containing case priorities
3. More flexibility in selection of 1-A cases
4. Conversion of data formats
5. Court-specific pooling factors.

#### Efficient Selection of Sixteen-Point Panels

The program was designed to facilitate changing the number of "difficulty" points per calendar. What was not anticipated was the difference in efficiency that results from such a modification. When only cases with three, five, seven, and ten points are considered, there are

just three combinations of cases that can produce sixteen-point panels: two three-points and one ten-point case; two three-points and two five-point cases; or three three-point cases and a seven-point case. If the standard "free search" procedure were followed, it would take much longer, and some matches would be missed. As a result, special programming was added for the special case where sixteen-point panels were needed, and one-point cases were excluded.

#### Putting Calculated Priorities in an Output File

As an aid to the court's manual case management procedures, a new output file was created containing all cases that were "eligible for calendaring," that included the calculated (or assigned) priorities. This improved the staff's ability to evaluate a calendar's inclusion and exclusion of cases. If a case is subsequently ordered on to a calendar, the staff can better judge the number of additional priority points to give it to assure its inclusion.

### Additional Flexibility in 1-A Cases

A design goal of the program was to give the staff attorneys the ability to include or exclude the simplest cases (1-A) from a given calendar. An additional level of flexibility proved to be needed, in that it became desirable to include simple civil cases with statutory priority while excluding nonpriority cases. This feature was added.

### Conversion of Data Formats

The largest related set of changes to Calen9 resulted from a substantial redesign of the data formats in the inventory file, as well as an addition of several data fields. The original data structure was solely for the purpose of facilitating the calendaring program. It proved useful to the staff attorneys' office in other ways as well, including the preparation of various reports. It quickly became clear that if additional data elements, such as a more elaborated taxonomy of issue codes, were included, the reporting aspects would be greatly improved. The data files were rede-

signed to include the required fields, and the program was modified accordingly.

One result was that the program did not need to write new versions of files to account for deleted cases, since a case's status was explicitly recorded as a field in the new format. This permitted a substantial modification of the ACDTOT subroutine to eliminate unneeded output. Further, a large reduction in program storage was permitted, since noneligible cases did not need to be stored for later output into an updated file.

Additional results included the abbreviation of districts' names, requiring modification of the "truth table" of legitimate sittings. For example, C.D.Cal. became CC, E.D.Wash became EW, etc. Judges' names also were abbreviated to save file space; descriptive subject names became numerical codes (in the more elaborated taxonomic framework), and more categories of civil cases were added. As a result of these and other, more minor changes, the appearance of the reports is substantially different from what it had been.

### Different Pooling Factors for Different Districts

One of the recurring problems in operating the calendaring program was adjusting the size of the pool of eligible cases from which the calendars were to be formed. The problem is that if there are too few cases in the pool, it may be impossible to simultaneously meet all restrictions; while if there are too many cases, it becomes too easy for relatively low priority cases to "bump" higher priority cases.

It became clear that because of the different distributions of cases in different districts, different pooling factors were required. The 115 percent discussed in the text quickly became inadequate for any district. The current assignment is to have 225 percent pooling in Los Angeles sittings, and 175 percent elsewhere, but it is anticipated that these figures are subject to change.

## THE FEDERAL JUDICIAL CENTER

The Federal Judicial Center is the research, development, and training arm of the federal judicial system. It was established by Congress in 1967 (28 U.S.C. §§ 620-629), on the recommendation of the Judicial Conference of the United States.

By statute, the Chief Justice of the United States is chairman of the Center's Board, which also includes the Director of the Administrative Office of the United States Courts and five judges elected by the Judicial Conference.

The Center's Continuing Education and Training Division conducts seminars, workshops, and short courses for all third-branch personnel. These programs range from orientation seminars for judges to on-site management training for supporting personnel.

The Research Division undertakes empirical and exploratory research on federal judicial processes, court management, and sentencing and its consequences, usually at the request of the Judicial Conference and its committees, the courts themselves, or other groups in the federal court system.

The Innovations and Systems Development Division designs and helps the courts implement new technologies, generally under the mantle of Courtran II—a multipurpose, computerized court and case management system developed by the division.

The Inter-Judicial Affairs and Information Services Division maintains liaison with state and foreign judges and judicial organizations. The Center's library, which specializes in judicial administration, is located within this division.

The Center's main facility is the historic Dolley Madison House, located on Lafayette Square in Washington, D.C.

Copies of Center publications can be obtained from the Center's Information Services office, 1520 H Street, N.W., Washington, D.C. 20005; the telephone number is 202/633-6365.

**END**