

**IN THE THIRTEENTH JUDICIAL CIRCUIT
HILLSBOROUGH COUNTY, FLORIDA**

**ADMINISTRATIVE ORDER S-2008-110
(Supersedes Second Amendment to Local Rule 3)**

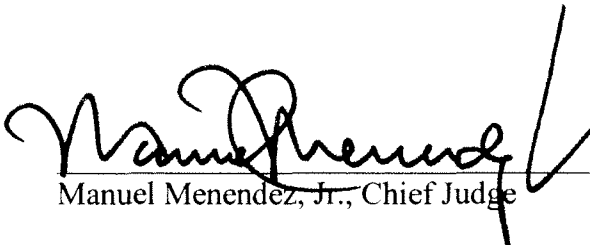
COMPUTERIZED SELECTION OF JURY VENIRES

Section 40.225, Florida Statutes, establishes an alternative method for obtaining jury venires by random lot using mechanical, electronic or electrical devices. This alternative method for obtaining jury venires requires the consent of a majority of the judges authorized to conduct jury trials and the approval of the Florida Supreme Court. The Thirteenth Judicial Circuit has been using some form of computerized selection of jury venires since 1979. Due to the Clerk of the Circuit Court's transition to a new computer system it is necessary to update procedures for selection of juror venires.

Therefore, by the power vested in me under Florida Rule of Judicial Administration 2.215(b)(2) and with the consent of a majority of the judges in the Thirteenth Judicial Circuit authorized to conduct jury trials, it is **ORDERED**:

The attached Hillsborough County Juror Selection Plan will be effective the date on which the plan is approved by the Florida Supreme Court. When the Hillsborough County Juror Selection Plan is approved by the Florida Supreme Court, the Second Amendment to Local Rule 3 will be superseded.

It is **ORDERED** in Tampa, Hillsborough County, Florida, on this 29th day of July, 2008.


Manuel Menendez, Jr., Chief Judge

Attachment: Hillsborough County Juror Selection Plan
Original to: Pat Frank, Clerk of the Court
Copies to: All Judges
Jury Services

Hillsborough County Juror Selection Plan

Creation of the Initial Candidate Selection List

Candidates for Hillsborough County petit and grand jury venires are drawn from a consolidated juror pool. This pool is created from the list of licensed Hillsborough County Florida residents that is received from the Florida Department of Highway Safety and Motor Vehicles (DHSMV). In accordance with section 40.011, Florida Statutes, the pool is further supplemented by Hillsborough County residents that have formerly indicated a desire, in the form of an affidavit, to serve as jurors.

On a weekly basis, updates to the licensed drivers list for Hillsborough County are obtained by the Clerk of the Circuit Court for Hillsborough County (“the Clerk”) from the DHSMV. These weekly update files are used to append, purge, or otherwise update the consolidated juror pool for Hillsborough County. If a person is in the weekly update list, and is also in the current database, update information is applied to the existing record while retaining any historical information. If the person is in the DHSMV list, but is not in the current database list, the new person is added to the database. If the weekly file indicates the person is no longer eligible for jury service within Hillsborough County, any existing juror record is marked as ineligible and precluded for use during the venire selection processes.

The Clerk, or a designate, will enter into the Hillsborough County Jury System (HCJS) all requests that have been received for residents indicating a desire to serve as a juror in accordance with section 40.011, Florida Statutes. The HCJS provides functionality to enter these names and other relevant data directly into the HCJS database on an as-occurring basis. These persons then become part of the Hillsborough County Juror current list. The current list represents all of the county residents for Hillsborough County that are in the HCJS regardless of their eligibility to serve on jury duty. In essence, the current list is a comprehensive list of current and past residents of Hillsborough and their eligibility with respect to serving on a jury. The Clerk shall be responsible for preserving the security of the jury lists as specified in section 40.02, Florida Statutes.

The Clerk shall purge or permanently excuse from the jury selection list all those persons statutorily excluded from juror service as specified in sections 40.013 and 40.022, Florida Statutes. Additionally, the Clerk shall at least monthly purge or permanently excuse the jury selection lists of names of those persons:

1. Adjudicated mentally incompetent;
2. Convicted of a felony; or
3. Deceased.
4. Other persons as the Chief Judge directs.

The HCJS provides functionality for the purging, or marking as permanently excused, a person described above. A person can be purged individually, or in bulk when Hillsborough County receives the deceased persons’ data from the Department of Health, Office of Vital Statistics in an electronic format. All

persons marked as disqualified become ineligible for future selection for jury service until the disqualification is reversed. Functionality exists within the system to support the reversal of disqualifications. When a juror record is updated to change eligibility, the date, time and id of the user making the change is recorded for audit purposes.

Records are also maintained within the HCJS for persons who are not able to serve for other reasons. All persons who have previously completed their jury service obligation are marked with a date served value and do not become eligible for re-selection until the eligibility period for selection has expired. The exemption period is a user-defined value based on section 40.013(7), Florida Statutes and is maintained in a system configuration file. Other persons that are excused or have had their jury service postponed, are also maintained in the HCJS as a part of the current list. Detailed historical information is created to record when and by whom excusals or postponements have been granted. This historical information is maintained in a history table that is related to a summons that has been issued. This history log tracks a myriad of events related to a jury summons that is issued.

Everyone in the HCJS database who does not have a current summons pending, has not served within the configured time period, has not been either excused, disqualified, or deferred, and is 18 years of age or older is considered during the randomization process that selects venire recipients as further explained in the Juror Randomization Process below.

The Jury staff has the ability to create a jury venire and select jurors via the Hillsborough County Juror System from the source list stored within the system and will print and mail summons to those persons selected six weeks prior to the jury service. Typically, venires are scheduled twice weekly and summonses are generated individually for each venire. The summons provides reporting instructions for each person during their jury service.

The equipment used in jury selection consists of a database server, application server, and hosted services server. The database server is an HP BL45 blade server (or another server of equal or greater efficiency) running Windows 2003 Enterprise Server and SQL Server 2005 Enterprise Edition. The application and services servers are virtual servers running Windows 2003 Server under VMWare Infrastructure 3 on an HP BL25 blade server (or another server of equal or greater efficiency). All data resides on an EMC CX3-20 SAN. The Hillsborough County Juror System (application and database) will be securely housed at all times on site in the Hillsborough County Clerk's Office.

Juror Randomization Process

The rules outlined below explain the Hillsborough County Juror Randomization Process. These rules will provide consistent randomization necessary for pool generations that will meet the standards mandated by the Florida Supreme Court.

- A. The Mersenne Twister Random Number Generator (RNG) is used to randomly select eligible juror candidates for service.

The implementation of the Mersenne Twister algorithm used was created by CenterSpace Software (www.centerspace.net).

C# Version Copyright (c) 2003 CenterSpace Software, LLC

This code is free software under the Artistic license.

CenterSpace Software

2098 NW Myrtlewood Way

Corvallis, Oregon, 97330

USA

<http://www.centerspace.net>

Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura, all rights reserved.

- B. A list of all potential juror candidates is selected from the database in no particular order. The criteria for the list is:
 - a. Not currently summoned for jury duty
 - b. Have not served within a pre-defined period (configurable within the system)
 - c. Are not permanently excused or otherwise disqualified
 - d. Are over the age of 18
- C. All eligible juror candidates meeting the above criteria are then assigned a sequential index value, beginning at zero. (C# uses zero based arrays.) This value will be used in the following step to match to a randomly generated number. Each number generated will correspond to a name output to the final candidate list.
- D. The RNG is seeded with a single 32-bit number that is created based on user input. The selection process for a random seed involves a user creating a ten-digit number by making a selection from a random page and line from the book, A Million Random Digits with 100,000 Deviates (Rand Corporation). To select a ten-digit seed value, the user will follow the procedure outlined in the Introduction of this book. The user will record the starting line, block and number, final line, column and ten-digit number. The ten-digit number selected is entered on a screen in the HCJS. The software converts this ten-digit number into an integer by dividing it by 10,000,000,000, multiplying the result by 2^{32} , and taking the greatest integer less than or equal to the floating point result. The ten-digit seed value is also stored in a table in the database for future verifiability.
- E. The total number of required summons is entered by the user through a graphical user interface. The system will create a loop to select venire recipients. In each pass of the loop, the RNG will return a number between 0 and the total number of qualified candidates minus one. This random number is matched to an index value of the list that was created in step C above. The Juror candidate to whom that index was assigned is then removed from the candidate list

and added to a list of selected summons recipients. The loop continues until the required total number of summons has been selected. Per Florida Statutes 40.02, a minimum of 250 summonses are randomly selected for each request. Once the list is completed, each selected juror candidate record is updated to indicate that a summons has been issued. This update precludes the Hillsborough County resident from being issued an additional summons until they become re-eligible for another summons.

Summons recipients that fail to appear become eligible for a new summons as soon as the venire to which they were summoned has concluded. Summons recipients that are granted an excusal become eligible for a new summons as soon as the excusal request is processed. Hillsborough County residents that successfully complete their jury service become eligible for a new summons when the requisite period of time has expired from the completion of their jury service.

- F. The following code sample shows the step-by-step logic for obtaining randomly selected resident indexes using the Mersenne Twister RNG.

```
//This method creates a list of integers whose size is equal to the
//count parameter. The integers will be between the specified min and
//max and will be randomly selected using the Mersenne Twister
//algorithm. No number can be added to the list twice.
public List<int> RandomlySelectListInRange(int min, int max, int count)
{
    List<int> result = new List<int>();

    //Check for null
    if (count <= 0)
        return result;

    //Loop until result is full
    while(result.Count < count)
    {
        //Get a random number between min and max
        int number = Next(min, max);

        //Prevent duplicates
        if (!result.Contains(number))
        {
            //Add number to list
            result.Add(number);
        }
    }

    return result;
}

//This method instantiates an instance of the MersenneTwister class
//using the default constructor and calls its Next method passing on
//the min and max values.
public int Next(int minValue, int maxValue)
{
```

```

        MersenneTwister _randomNumberGenerator = new MersenneTwister();

        return _randomNumberGenerator.Next(minValue, maxValue);
    }

    //This is the default constructor for the MersenneTwister class. The
    //seed value that is used is derived from the milliseconds value of
    //the DateTime.Now property.
    public MersenneTwister()
    {
        init_genrand((uint)DateTime.Now.Millisecond);
    }

    //This method initializes the MersenneTwister class using the seed
    //value that is passed in.
    private void init_genrand(uint s)
    {
        mt[0] = s & 0xffffffffU;

        for (mti = 1; mti < N; mti++)
        {
            mt[mti] = (uint)(1812433253U * (mt[mti - 1] ^ (mt[mti - 1]
                >> 30)) + mti);

            //For >32 bit machines
            mt[mti] &= 0xffffffffU;
        }
    }

    //This method gets a random number between the min and max
    public int Next(int minValue, int maxValue)
    {
        //Validate min and max
        if (minValue > maxValue)
        {
            int tmp = maxValue;
            maxValue = minValue;
            minValue = tmp;
        }

        return (int)(Math.Floor((maxValue - minValue + 1) *
            genrand_reall() + minValue));
    }

    //This method generates a random number on [0,1]-real-interval
    private double genrand_reall()
    {
        return genrand_int32() * (1.0 / 4294967295.0);
    }

    //This method generates a random number on [0,0xffffffff]-interval
    private uint genrand_int32()
    {
        uint y;

        if (mti >= N)
        {

```

```

//Generate N words at one time
int kk;

//Verify that init_genrand() has been called
if (mti == N + 1)
    //If not use a default initial seed
    init_genrand(5489U);

for (kk = 0; kk < N - M; kk++)
{
    y = (mt[kk] & UPPER_MASK) | (mt[kk + 1] &
        LOWER_MASK);
    mt[kk] = mt[kk + M] ^ (y >> 1) ^ mag01[y & 0x1U];
}

for (; kk < N - 1; kk++)
{
    y = (mt[kk] & UPPER_MASK) | (mt[kk + 1] &
        LOWER_MASK);
    mt[kk] = mt[kk + (M - N)] ^ (y >> 1) ^ mag01[y &
        0x1U];
}

y = (mt[N - 1] & UPPER_MASK) | (mt[0] & LOWER_MASK);
mt[N - 1] = mt[M - 1] ^ (y >> 1) ^ mag01[y & 0x1U];

mti = 0;
}

y = mt[mti++];

// Tempering
y ^= (y >> 11);
y ^= (y << 7) & 0x9d2c5680U;
y ^= (y << 15) & 0xefc60000U;
y ^= (y >> 18);

return y;
}

```

- G. The HCJS, C# implementation of the Mersenne Twister algorithm was tested and the output was successfully compared to the control file posted by the algorithm's author at <http://www.math.sci.hiroshima-u.ac.jp/%7Em-mat/MT/MT2002/CODES/mt19937ar.out>. Additionally, the randomizer class exposes a public static method that can be used to regenerate the file that was compared against this control file. This static method uses the hard-coded seed values that the author used to create the control file.