

# National Database for Autism Research

## **Global Unique Identifier (GUID) Web Service User Manual**

Version 3.0.1  
October 20, 2009

## Revision

<b>Version</b>	<b>Date</b>	<b>Description</b>
1.5.3	7/7/2008	The document provides an overview of how the GUIDWS system works and how to obtain GUIDs from the system
1.5.4	8/1/2008	<ul style="list-style-type: none"><li>• Updated “Special Instructions” to reflect that name suffixes</li><li>• Added requirement of Java 1.5 to run the client</li><li>• Added information on the Batch functionality</li><li>• Removed reference to an obsolete document</li><li>• Updated document format</li></ul>
1.5.5	10/10/2008	<ul style="list-style-type: none"><li>• Section on “Running the Client Using the Graphical User Interface (GUI), For MS Windows” was updated</li></ul>
2.0.2	12/18/2008	<ul style="list-style-type: none"><li>• Section on “Running the Client Using the Graphical User Interface (GUI), For MS Windows” was updated to reflect the latest changes in GUID interface</li></ul>
3.0.1	10/20/2009	<ul style="list-style-type: none"><li>• Added information on the Convert Pseudo-GUID functionality and changed nomenclature to Pseudo-GUID from Invalid GUID</li></ul>

## Table of Contents

Introduction .....	4
GUID Process: How it works .....	4
GUIDWS Security .....	5
GUIDWS Functions .....	6
PII Fields Used to Generate a GUID .....	8
Special Instructions .....	10
Valid versus Pseudo-GUID .....	10
GUID Composition.....	11
Valid GUID Example .....	11
Pseudo-GUID Example.....	12
Obtaining the GUIDWS Client.....	12
Running the GUID Process.....	12
For UNIX Systems.....	13
For MS Windows .....	13
Program Input—GET.....	14
Program Input—GETBATCH .....	15
Program Output .....	16
output_guid_timestamp.txt .....	16
output_log_timestamp.txt.....	16
output_error_timestamp.txt .....	17
Running the Client Using the Graphical User Interface (GUI) .....	17
For UNIX systems .....	18
For MS Windows .....	18
Running the GUIDWS Client Programmatically.....	27
Troubleshooting .....	27
Timestamp Error .....	27
Transport Out Error .....	28

## Introduction

The generation of a GUID (Global Unique Identifier) is fundamental to the implementation of the National Database for Autism Research (NDAR). The GUID in NDAR enables data to be associated with a research participant without exposing or transferring the research participant's personally identifiable information (PII). This capability provides two primary benefits. First, it allows data about a research participant to be accumulated across projects over time, regardless of where and when that data were collected. Secondly, it enables a researcher to define a study population within NDAR supporting a research hypothesis, even if the data were not collected by the researcher.

This document defines the technical implementation and use of the NDAR GUID Web Service (GUIDWS). For general information on the use of the GUID, refer to <http://ndar.nih.gov/ndarpublicweb/standards.go#GUIDs>. For NDAR Policy related to the NDAR GUID, refer to <http://ndar.nih.gov/ndarpublicweb/policies.go>.

## GUID Process: How it works

Researchers collect PII (sometimes referred to as Private Health Information or PHI) from their participants and store that data in a local database that is not made available outside the research institution. Typically, that data is only available to a limited number of individuals. NDAR uses that data available at the investigator site to generate GUIDs. This is made possible by issuing special software that runs at the research site on an investigator computer. This software performs a one-way encryption, often called a one-way hash<sup>1</sup>, which is sent to the GUIDWS to determine if the research subject hash codes have been seen by NDAR before. The encrypted hash codes do not have information to recreate the PII. However, they do have enough information to determine if a research participant GUID already exists in NDAR.

---

<sup>1</sup> A one-way hash is a reproducible method of turning some kind of data into a (relatively) small number that may serve as a digital "fingerprint" of the data. The algorithm "chops and mixes" (i.e., substitutes or transposes) the data to create such fingerprints. The hash represents the "fingerprint". However, it cannot be used to recreate the original.

The GUIDWS consists of two components:

**GUIDWS Server** – A single web service application used to create GUIDs, update and merge duplicate participants, and notify client sites of a change to the GUID of a research participant.

**GUIDWS Client** – A web service client application, implemented as a Java application, which encrypts the PII collected by the researcher at the research site to produce hash codes for each participant.

When the client site, using the GUIDWS client application, requests a GUID for the chosen participant, the client application sends the hash codes associated with the participant to the NDAR GUIDWS server. The GUIDWS server processes the hash codes and returns a GUID to the client application. PII data is never transmitted or stored in the GUIDWS server application. Communication between the GUIDWS web server occurs via HTTP & SOAP (Refer to *Figure 1. GUIDWS System Architecture*).

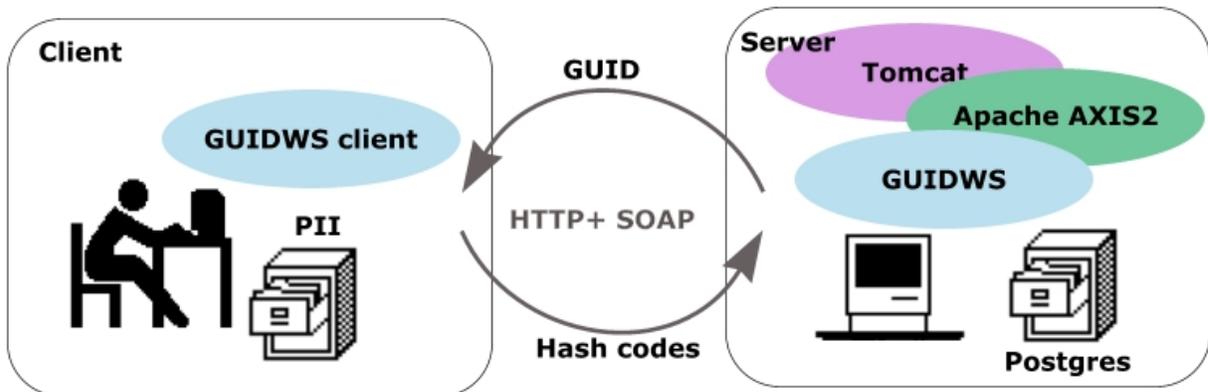


Figure 1. GUIDWS System Architecture

## GUIDWS Security

The GUIDWS is protected by a number of security measures. There are two specific layers to the GUIDWS security: application and network (IP restriction).

Security for GUIDWS is implemented within the application using Web Services Security (WS-Security), which allows a secure connection between a client and server by providing encryption and authentication support. Each client site is configured with the client's public-private key pairs that are used to identify a user to the server. This reduces the ability for an unauthorized user to gain access.

The second security layer is the client IP restriction access. This layer maximizes security by preventing unauthorized client machines to access the Web Service even if a valid client application with valid credentials exists. All sites requesting access must submit the fully qualified domain name of the machine (*e.g. somehost.example.com*) and static IP address of each client machine that will run the GUID Client application. A unique GUID Client application will be specifically configured for each client machine. If the client machine has a dynamic IP address instead of a static IP address, then a range of dynamic IP addresses with subnet mask is required. Each GUID Client will only work on the client machine for which it was created.

The above security measures, along with all security controls, are independently audited to ensure that no vulnerabilities exist in both the technical and procedural implementation of this process (Refer to NDAR SOP-03 NDAR Certification and Assurance to Operate at <http://ndar.nih.gov/ndarpublicweb/policies.go#sop3>).

## **GUIDWS Functions**

The following are the service functions that compose the NDAR GUIDWS:

**LOGIN** – This function allows users to login to the GUIDWS. Users need to log in first in order to perform any service functions within the application. Only authorized users can gain access the GUIDWS server.

**GET** – This function generates a Valid GUID in a format that is used in the NDAR system provided the minimum required PII information is available. This module uses the hash function to generate a unique GUID for each participant. (Refer to the *Valid versus Pseudo-GUID* section for more information on Valid GUIDs. For more information about GUID format, refer to the *GUID Composition* section.)

- **Subject Comparison** – This sub-module of the GUID Generator module compares the incoming participant hash codes to the existing participant hash codes to determine if there are any matches in the database.
- **Subject Merging** – This module activates when the hash codes from more than one GUID match indicating that they are the same participant and need to be merged into one participant. The GUID of the participant that has the most complete PII information is then mapped to be the representative GUID for merged participant GUIDs.

**GETBATCH** – This function allows for the request of multiple participant GUIDs from a single call to the server. There are no limits to the number of GUIDs in a request. However, the batch functionality will process 50 participants at a time with a 5 minute delay between each 50 participants processed. The functionality takes as an input a CSV file that includes PII information (Refer to Table 1. PII Fields) and creates as an output with the list of GUID(s). The following functions are supported:

**GETINVALIDGUID** – Generates a Pseudo-GUID with a different format from a Valid GUID. A Pseudo-GUID is a unique ID that is not based on PII. PII is not required to get a Pseudo-GUID. This function should only be used if a user is unable to get a Valid GUID using the GET function due to insufficient PII for a participant. (Refer to the *Valid versus Pseudo-GUID* section for more information on Valid GUIDs. For more information about GUID format, refer to the *GUID Composition* section.)

**TESTCONNECTION** – Tests the client connectivity to the GUID server. This function does not store any information to the GUID database and no input is needed.

**DOESGUIDEXIST** – Allows users to check if a GUID exists in the database. Users must provide a Valid GUID or a Pseudo-GUID to use this function. If the submitted GUID exists in the database, the function will return true. If the GUID does not exist in the database, the function will return false.

**CONVERTPSEUDOGUID** – Allows users to convert Pseudo-GUIDs to Valid GUIDs. Users need to provide the minimum required PII fields and the Pseudo-GUIDs. Once a Pseudo-GUID has been converted, it cannot be used to submit data to NDAR.

**UPDATE** (future feature) – Will be used to update the hash codes based on PII of a participant that has already been entered to the GUIDWS application. **This module is not yet implemented in the system.**

### **PII Fields Used to Generate a GUID**

The PII fields that are used to generate a GUID are listed in *Table 1. PII Fields* below. PII fields are data that does not change over the lifetime of the participant and are uniquely specific to the participant. Each PII field has an associated probability of a match in the general population. By combining full legal name, date of birth, gender, and municipality of birth, the probability that two individuals share the same information and thus the same hash codes (i.e. a false positive) becomes negligible and is the minimum required information to generate a Valid GUID. Additional data that is provided beyond the required minimum further decreases the probability of a false positive.

<b>PII Field Name</b>	<b>Abbreviation used in the NDAR GUIDWS client</b>	<b>Required minimum for Valid GUID</b>
Complete legal given name of subject at birth	FIRSTNAME	Yes
Complete legal family name/surname of subject at birth	LASTNAME	Yes
Complete additional legal name or names at birth	MIDDLENAME	Yes
Day of month of birth	DOB	Yes
Month of birth	MOB	Yes
Year of Birth	YOB	Yes
Physical sex of subject at birth [M/F]	SEX	Yes
City or municipality of birth	COB	Yes
Government Issued / National ID (For the United States, this is the Social Security Number)	GIID	Recommended
ISO 3166 STANDARD 2 letter abbreviation of the COUNTRY of origin of THE GIID. See APPENDIX 1 for a Listing of the ISO 3661 Country Abbreviations	GIIDCOUNTRY	Recommended
Mother's complete legal given name at her birth	MFIRSTNAME	
Mother's complete legal family name at her birth	MLASTNAME	
Father's complete legal given name at his birth	FFIRSTNAME	
Father's complete legal family name at his birth	FLASTNAME	
Mother's day of month of birth	MDOB	
Mother's month of birth	MMOB	
Father's day of month of birth	FDOB	
Father's month of birth	FMOB	
Indicator if subject has NO Middle Name AT BIRTH	SUBJECTHASNOMIDDLENAME	Yes

*Table 1. PII Fields*

## Special Instructions

- 1 The "Last Name" field must contain the family name given at birth, prior to legal name change, or marriage. If there is any doubt as to the original legal name at birth, refer to the information on the birth certificate.\* Name suffixes such as "Jr.", "Sr.", "III", etc. should be ignored.
- 2 If the participant's "First Name" is a compound name, such as Anne Marie, or Jose-Luis, it may be unclear whether the second part of the compound is a first name or a middle name. In such cases, use the first name as you would report it on other records, such as school transcripts, or credit card billing statements. If in doubt, refer to the birth certificate.\*
- 3 If the participant does not have a "Middle Name" (known not to have a middle name at birth), enter "NOTAPPLICABLE". If in doubt, refer to the birth certificate.\* The GUID Software has a selection to accommodate this possibility.
- 4 If the "City or Municipality of Birth" has undergone a name change during a participant's lifetime, use the name of the city at the time of the participant's birth. Examples of this are Peking / Beijing, or Bombay / Mumbai. Again, if there is any doubt, refer to the birth certificate.\*

*\* NDAR recommends that the participant's birth certificate and any copies of the participant's birth certificate be returned to the participant and not retained by the site.*

## Valid versus Pseudo-GUID

A Valid GUID is a GUID that has been successfully issued using PII. For retrospective data, it is understood that the minimum amount of PII may not be available or sufficient to generate a Valid GUID. For this reason, NDAR provides the capability to issue a Pseudo-GUID. A Pseudo-GUID contains no – or less than – the minimum amount of PII, essentially making the GUID a random identifier. These GUIDs are identified as a Pseudo-GUID.

The GUID server, using the GET function, will return a Valid GUID only. A Valid GUID is returned when all the required PII information is supplied to distinctly identify the participant. If the minimum required PII information is not supplied to create the hash codes, then an error

message is returned instead of a Valid GUID. A Valid GUID can be merged, and possibly changed, through mapping to another Valid GUID based on the amount of PII supplied (refer to the information on *Subject Merging* for the GET method in the *GUIDWS Functions* section). Therefore, while all fields are not required for a Valid GUID, the entry of as much PII as possible will help reduce the possibility of the Valid GUID changing due to a merge. *Table 1. PII Fields*, above, denotes the required minimum PII inputs to ensure a Valid GUID will be obtained.

A Pseudo-GUID can be generated using the GUID server function, GETINVALIDGUID. A Pseudo-GUID will always be unique. Pseudo-GUIDs are not compared to other participants (including Valid GUIDs) and are never merged since they have no PII associated with them (Refer to information on *Subject Comparison* and *Subject Merging* for the GET method in *GUIDWS Functions* section). Pseudo-GUIDs support the need for participants to be uniquely identified in the system within a project when sufficient PII for a Valid GUID is either not available or not possible. The advantage of using a Pseudo-GUID as an ID within a study is that it can later be updated to get a new Valid GUID while maintaining a mapping in NDAR for the Pseudo-GUID to the Valid GUID.

## GUID Composition

The GUID consists of a prefix, GUID pattern, and check character. The prefix chosen for this system is “NDAR”. The GUID a pattern is AANNNA, where A represents alphabetic and N represents numerical characters. The check character can be numerical or alphabetic.

## Valid GUID Example

An example of a Valid GUID is: **NDARCJ743PV3**

<u>GUID Section</u>	<u>Example</u>	<u>Description</u>
Prefix	NDAR	
Pattern	CJ743PV	<ul style="list-style-type: none"> <li>• The pattern is AANNNA.</li> <li>• A represents an alphabetic character.               <ul style="list-style-type: none"> <li>• The letters I, O, Q, S are not used because they letters are easily confused with certain numbers.</li> </ul> </li> <li>• N represents a number both A and N are generated</li> </ul>

		using a nondeterministic random number generator <sup>2</sup>
Check Character	3	

### Pseudo-GUID Example

An example of a Pseudo-GUID is: **NDAR\_INVZG542YHV**

The format of the Pseudo-GUID is identical to that of a Valid GUID with the exception that the prefix NDAR\_INV is used instead of NDAR.

### Obtaining the GUIDWS Client

The NDAR SOP #8 GUID Generation Permission Request provides instructions on how to obtain the GUID Client (refer to <http://ndar.nih.gov/ndarpublicweb/policies.go#sop8>). A GUID Client for the NDAR test/demonstration site will be issued first to ensure that the software is operating appropriately. This version generates GUIDs that are not associated with an individual (i.e. test data) in our production site. Once the demonstration software is issued and is working correctly, the user should send a request to [ndar@mail.nih.gov](mailto:ndar@mail.nih.gov) to receive a production version of the GUID Client.

### Running the GUID Process

The GUID Client software is a Java application allowing it to be run on many different operating systems and in a variety of ways.

#### Running the GUIDWS Client Standalone Application

The GUIDWS client can be run using either the command line or the Graphical User Interface (GUI) on most operating systems. The GUIDWS client requires the installation of Java JRE 1.5.

---

<sup>2</sup> A variant of ISO7064 Mod 37, only difference is that I, O, Q, S are removed.

## Running the GUIDWS Client Using the Command Line

To run the GUIDWS client using the command line on UNIX run `guid_client.sh`. For Windows run `guid_client.bat`. These programs accept a file of participant information allowing multiple GUIDs to be generated in one batch program. The example below illustrates the process to request a GUID for a new participant by using a participant file called `text_pii.txt`, as an input for the GUIDWS client for the GET command:

### **For UNIX Systems**

```
GUID_CLIENT_DIR% ./GUID_client.sh -a get -u username -p password -d  
'http://host:port/path/to/GuidService' -f test_pii.txt
```

### **For MS Windows**

```
GUID_CLIENT_DIR:\ GUID_client.bat -a get -u username -p password -d  
"http://host:port/path/to/GuidService" -f test_pii.txt
```

## Program options

```
-a <action> -- The action which the GUID Client should perform: get, getInvalidGuid,  
testConnection, doesGuidExist or convertPseudoGuid.  
-u <username> -- The username to use when logging into the GUID web service.  
-p <password> -- The password to use when logging into the GUID web service.  
-d <url> -- The URL to the GUID web service, including the hostname, port number and service  
path.  
-c <charset> -- The charset for the input, default is UTF-16.  
-b <batch_file>-- The comma separated value file containing the ordering of the PII fields on the  
first line and PII data for one subject on each subsequent line, without limit on the total number of  
subjects.  
-f <filename> -- The file is used to store the PIIs for a single subject. Each piece of PII should be on  
its own line, in the format '<FIELD><=><VALUE>  
-g <guid> -- The GUID that is submitted to check if a GUID exists or not.  
-h -- Display this message.  
-i <pseudo-GUID> -- The Pseudo-GUID that is to be converted into a Valid GUID.
```

The table below shows the options that are required for each command:

Commands	Options required
GET	-a, -u, -p, -d, -f
GETINVALIDGUID	-a, -u, -p, -d
TESTCONNECTION	-a, -u, -p, -d
DOESGUIDEXIST	-a, -u, -p, -d, -g
GET(BATCH)	-a, -u, -p, -d, -b
CONVERTPSEUDOGUID	-a, -u, -p, -d, -f, -i

### Program Input–GET

GET accepts an input file, as shown below, with the PII field key-value pairs (Refer to *Table 1. PII Fields*). All required fields must be entered in order to receive a GUID. The value for SUBJECTHASNOMIDDLENAME must be either YES or NO. If the participant was not given a middle name at birth, enter YES; otherwise, enter NO.

```

COB=Bethesda
DOB=20
MOB=10
YOB=2001
FDOB=17
FMOB=3
MDOB=8
MMOB=6
FIRSTNAME=Alexandra
MIDDLENAME=
LASTNAME=Smith
FFIRSTNAME=Philips
FLASTNAME=Smith
MFIRSTNAME=Danna
MLASTNAME=White
SEX=M
GIID=000-00-0000
GIIDCOUNTRY=US
SUBJECTHASNOMIDDLENAME=yes

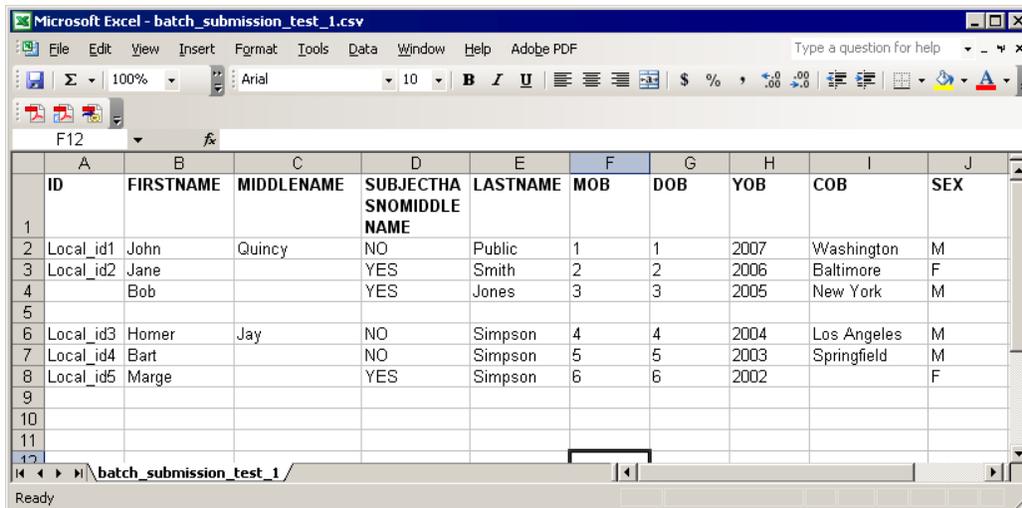
```

## Program Input–GETBATCH

GETBATCH accepts an input file in comma separated format. The first line of the file should contain the PII field names as defined in *Table 1. PII Fields*. An additional “ID” field is required for the GUID output to map from the original participant to each generated GUID. The content of this field is left to the user’s discretion for output mapping, but could be a site’s local ID for the participant. If no such ID exists, then a simple ascending integer should be used for each participant. Below is an excerpt from an example input file format. The complete example file, viewed in MS Excel, is displayed in *Figure 2. CSV Input File*.

### Example Batch CSV with Select PII Input Data

```
ID,FIRSTNAME,MIDDLENAME,SUBJECTHASNOMIDDLENAME,LASTNAME,
MOB,DOB,YOB,COB,SEX
Local_id1,John,Quincy,NO,Public,1,1,2007,Washington,M
Local_id2,Jane,,YES,Smith,2,2,2006,Baltimore,F
```



ID	FIRSTNAME	MIDDLENAME	SUBJECTHASNOMIDDLENAME	LASTNAME	MOB	DOB	YOB	COB	SEX	
1										
2	Local_id1	John	Quincy	NO	Public	1	1	2007	Washington	M
3	Local_id2	Jane		YES	Smith	2	2	2006	Baltimore	F
4		Bob		YES	Jones	3	3	2005	New York	M
5										
6	Local_id3	Homer	Jay	NO	Simpson	4	4	2004	Los Angeles	M
7	Local_id4	Bart		NO	Simpson	5	5	2003	Springfield	M
8	Local_id5	Marge		YES	Simpson	6	6	2002		F
9										
10										
11										
12										

Figure 2. CSV Input File

If subsequent requests for multiple GUIDs are sent, the required wait time between sending each GUID batch request containing 50 participants is 30 seconds.

## Program Output

The output from the various GUID functions appears in three output files each time the functions are run. The “timestamp” is a number indicating the actual date and time the client was executed. Limited output will also be generated at the command line; however, these three files in combination will have the complete information of an execution.

- output\_guid\_timestamp.txt
- output\_log\_timestamp.txt
- output\_error\_timestamp.txt

### output\_guid\_timestamp.txt

This file contains the normal/expected return values of the GUID functions. For example, for the GET function, the output is:

```
NDARTL435KVE
```

For GETBATCH, each line of output is prefaced by the “ID” field (Refer to the

*Program Input*—GETBATCH section):

```
Local_id1 - NDARTL435KVE  
Local_id2 - NDARNX545XZE  
Local_id3 - NDARVR604PPX
```

### output\_log\_timestamp.txt

This is a log file that contains the details of an execution and is useful for debugging. A sample of the file’s contents is shown below:

```
Jul 31, 2008 3:45:08 PM org.apache.axis2.deployment.DeploymentEngine prepareRepository  
INFO: No services directory was found under C:\projects\ndar-  
guid_client\client\Java\trunk\classes\repository.  
Jul 31, 2008 3:45:08 PM org.apache.axis2.deployment.ModuleDeployer deploy  
INFO: Deploying module: addressing-1.1  
Jul 31, 2008 3:45:08 PM org.apache.axis2.deployment.ModuleDeployer deploy  
INFO: Deploying module: addressing-1.2  
Jul 31, 2008 3:45:08 PM org.apache.axis2.deployment.ModuleDeployer deploy  
INFO: Deploying module: rampart-1.2  
Reading batch file column headers.  
Reading subject number 1 from batch file line 2  
Reading subject number 2 from batch file line 3  
Reading subject number 3 from batch file line 4  
Reading subject number 4 from batch file line 6  
Reading subject number 5 from batch file line 7
```

```
ERROR - One or more required fields are missing/need modification, and therefore a Valid GUID can not be generated. Please enter and/or modify values for the following fields and resubmit: - Middle name is missing
Reading subject number 6 from batch file line 8
ERROR - One or more required fields are missing/need modification, and therefore a Valid GUID can not be generated. Please enter and/or modify values for the following fields and resubmit: - City of Birth is missing
Completed reading batch file subjects.
Number of subjects read from batch file: 6
Number of subjects read from batch file without errors: 3
Number of subjects read from batch file with errors: 3
Sending getBatch request number 1
Sending getBatch request of size: 3
Logging into GUID webservice.
Jul 31, 2008 3:45:12 PM org.apache.xml.security.signature.Reference verify
INFO: Verification successful for URI "#Id-12082199"
Jul 31, 2008 3:45:12 PM org.apache.xml.security.signature.Reference verify
INFO: Verification successful for URI "#Timestamp-28653851"
Jul 31, 2008 3:45:13 PM org.apache.xml.security.signature.Reference verify
INFO: Verification successful for URI "#Id-11729694"
Jul 31, 2008 3:45:13 PM org.apache.xml.security.signature.Reference verify
INFO: Verification successful for URI "#Timestamp-19002963"
Completed getBatch request number 1
Logging out of GUID webservice.
Outputting errored subject information.
Finished outputting errored subject information.
Finished GUID batch request.
```

### **output\_error\_timestamp.txt**

This file contains a list of errors that appeared while the program is running. A sample file is shown below:

```
ERROR - No ID field found in batch submission file (line: 4).
Local_id4 - ERROR - One or more required fields are missing/need modification, and therefore a Valid GUID can not be generated. Please enter and/or modify values for the following fields and resubmit: - Middle name is missing
Local_id5 - ERROR - One or more required fields are missing/need modification, and therefore a Valid GUID can not be generated. Please enter and/or modify values for the following fields and resubmit: - City of Birth is missing
```

## **Running the Client Using the Graphical User Interface (GUI)**

The Graphical User Interface (GUI) for the GUIDWS can be used with both UNIX and Windows.

To run the GUI, copy the `guid_client_gui.bat` file on the desktop or run the `guid_client_gui.bat` file from the command line. In Unix, run the shell script from the command line. Command line syntax for both Unix and Windows is provided below:

## For UNIX systems

```
[username@domain]~/temp/guid_java_client_DIR% ./GUID_client_gui.sh
```

## For MS Windows

```
guid_java_client_DIR:\ GUID_client_gui.bat
```

The NDAR GUID Client dialog box appears, refer to *Figure 3. NDAR GUID Client Dialog Box*.

NDAR GUID Client version 2.0.1.287 [DEV]

Functions Settings

Please enter subject's Personally Identifiable Information (PII) twice:  
\* indicates required fields

1. Government Issued or National ID	1.		
2. Country of Government Issued or National ID	2.		
3. Complete legal given (first) name of subject at birth *	3.		
4. Complete legal family (last) name of subject at birth *	4.		
5. Select yes if the subject does NOT have a Middle Name, no otherwise *	5.	<input type="radio"/> Yes	<input type="radio"/> No
6. Complete middle name *	6.		
7. Day of birth [1-31] *	7.		
8. Month of birth [1-12] *	8.		
9. Year of birth [####] *	9.		
10. Physical sex of subject at birth [MF] *	10.		
11. Name of city/municipality in which subject was born *	11.		
12. Mother's complete legal given (first) name at birth	12.		
13. Mother's complete legal family name at birth	13.		
14. Father's complete legal given (first) name at birth	14.		
15. Father's complete legal family name at birth	15.		
16. Mother's day of birth [1-31]	16.		
17. Mother's month of birth [1-12]	17.		
18. Father's day of birth [1-31]	18.		
19. Father's month of birth [1-12]	19.		

GUID

Generate GUID Copy GUID to clipboard New Exit

Figure 3. NDAR GUID Client Dialog Box

## Connection Settings

Log in to the GUIDWS with the username and password supplied with the client package. From the menu, select Settings > Connection Information. The dialog box shown in *Figure 4. Connection Settings Dialog Box* will appear.

Connection settings

User name client\_dev

Password \*\*\*\*\*

Server URL http://w-apps.nbirn.net:8081/axis2/services/GuidService/

OK

Figure 4. Connection Settings Dialog Box

## Testing the Connection

To verify that the user's computer is still connected to the NDAR GUID server, select Settings > Test Connection from the menu. The dialog box shown in *Figure 4. Connection Settings Dialog Box* will appear. Click the OK button. If the computer is connected to the NDAR GUID server, a message indicating a successful connection will appear as shown in *Figure 5. Test Connection Window*.



Figure 5. Test Connection Window

## Obtaining a GUID

To obtain a GUID for a participant, enter the participant information into the dialog box (refer to *Figure 7. Completing the NDAR GUID Client Dialog Box*). The information should be entered twice – once in each column of the dialog box. There are a total of 19 fields that should be completed. Fields 3–11 (marked with asterisks) are **required** to obtain a GUID. The required field names are listed below:

- Complete legal given (first) name of subject at birth\*
- Complete legal family (last) name of subject at birth\*
- Select YES if the subject DOES NOT HAVE a Middle Name, NO otherwise\*
- Complete Middle Name\*
- Day of birth [1–31]\*
- Month of birth [1–12]\*
- Year of birth [####]\*
- Physical sex of subject at birth [M/F]\*
- Name of city/municipality in which subject was born\*

To complete the NDAR GUID dialog box and obtain the GUID, complete the following steps, starting in the left column of the dialog box:

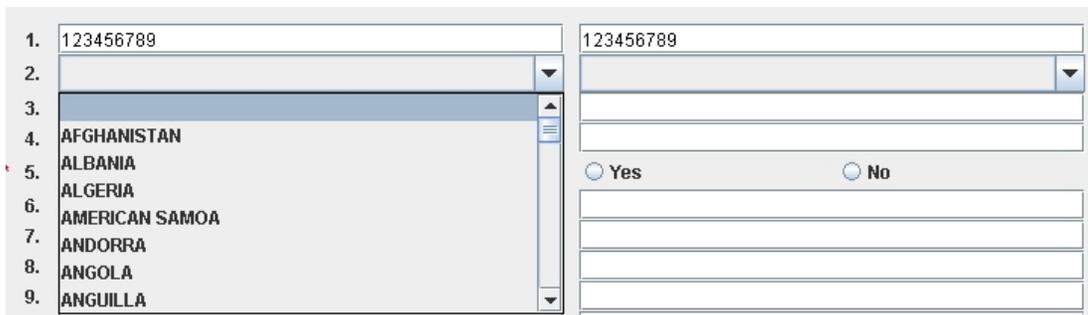
1. Enter the government issued or national ID number into the first field (note that the list box in the second field becomes available);
2. Select the country of issued ID from the list (refer to *Figure 6. List of Countries*);
3. Complete the required fields 3–11;
4. Complete the optional fields 12–19 (refer to *Figure 7. Completing the NDAR GUID Client Dialog Box*);
5. Enter the same information into the right column of the dialog box (refer to *Figure 7. Completing the NDAR GUID Client Dialog Box*); and
6. Click the Generate GUID button.

The GUID appears in the GUID field at the bottom of the dialog box (refer to *Figure 8. Obtaining a GUID*). The PII fields and the Generate GUID button will be grayed out only if enough PII is provided and the newly generated GUID is valid.

*Note that spaces, hyphens and apostrophes can be used in the non-numeric GUID fields in the dialog box and are stripped silently by the application.*

**To Create another GUID,**

1. Click the New button;
2. Complete the dialog box for another participant; and
3. Click the Generate GUID button.



*Figure 6. List of Countries*

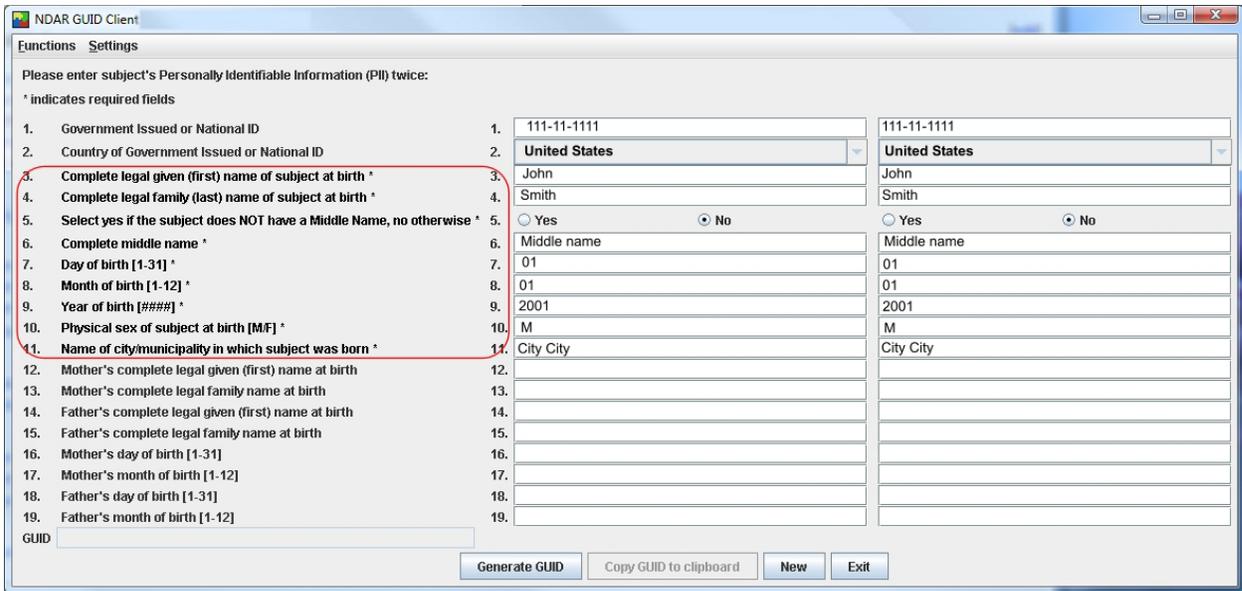


Figure 7. Completing the NDAR GUID Client Dialog Box

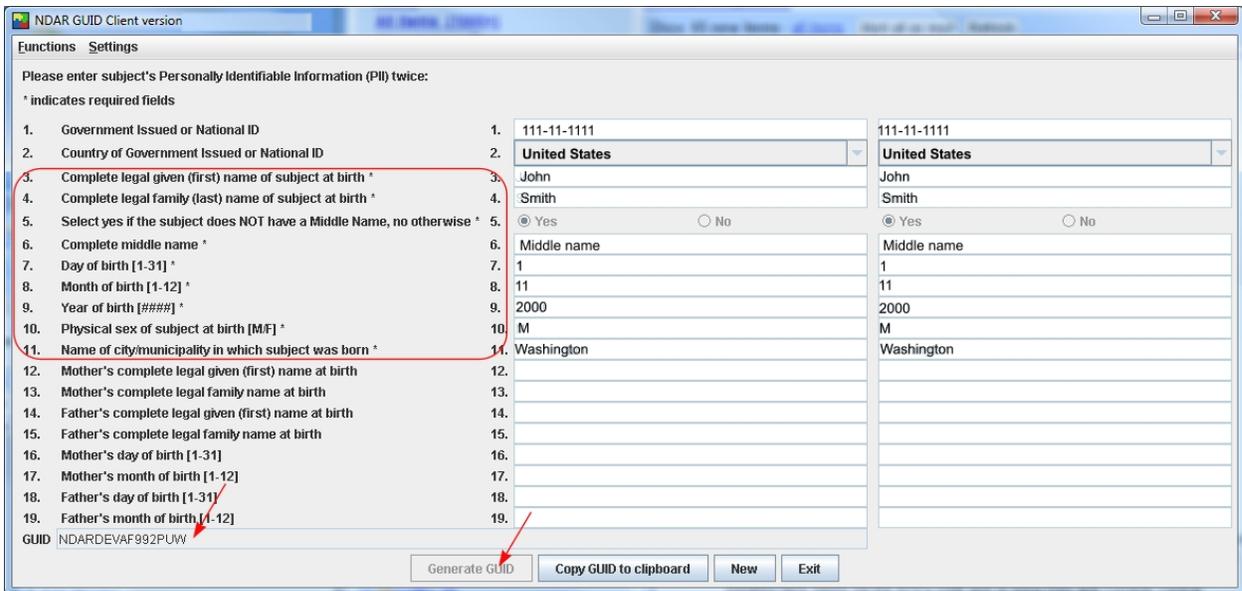


Figure 8. Obtaining a GUID

## To Copy the GUID to the Clipboard

Once the Valid GUID appears in the GUID field (refer to *Figure 8. Obtaining a GUID*), it can be copied to the Clipboard by clicking the Copy GUID to Clipboard button. This allows the GUID to be used in other applications.

## What happens if the user did not provide enough information?

A GUID will not be issued if the required participant information is not provided. In this case, an Error window will appear listing the missing fields. The missing fields will also be highlighted in red in the dialog box (refer to *Figure 9. GUID Error Message Window*). To obtain the GUID, the user should provide information for all required fields, and then click the Generate GUID button again.

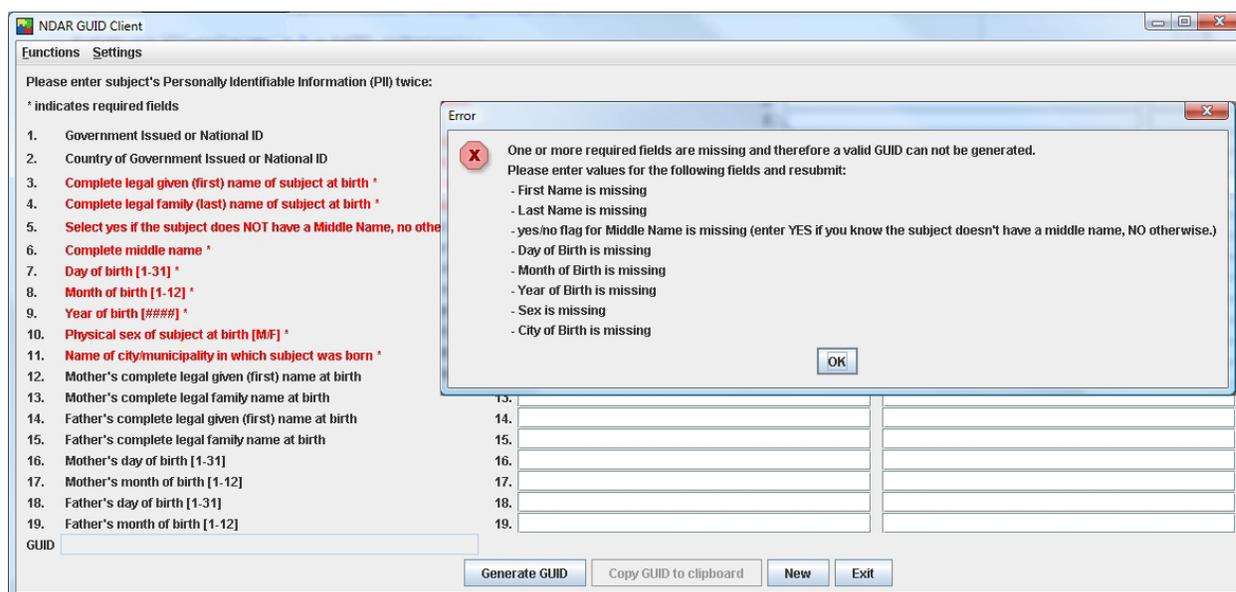


Figure 9. GUID Error Message Window

## Getting GUIDs for Multiple Participants

To obtain GUIDs for multiple participants:

1. Enter the participants' information in MS Excel;
2. Save the file in CSV format (refer to *Figure 2. CSV Input File*);
3. Run the NDAR GUID client for MS Windows;
4. Select Functions > Get GUIDS for Multiple Subjects from the menu (refer to *Figure 10. Functions Menu*);



Figure 10. Functions Menu

5. Browse for and select the CSV file from the dialog box that appears; and
6. Click the OK button.

The Multiple Subject GUID Request Output dialog box appears displaying the following information:

- Name of the CSV file that has been processed;
- Name of the file where obtained GUIDS are stored (note that the GUID file is stored in the same catalog as the source file);
- List of “subject Local ID–GUID” for participants that obtained NDAR GUIDs; and
- List of “subject Local ID–Error(s)” for those participant records for which information was processed with errors, and therefore, did not obtain NDAR GUIDs (refer to *Figure 11. Multiple Subject GUID Request Dialog Box*).

Click the Close button to close the dialog box (refer to *Figure 11. Multiple Subject GUID Request Output Dialog Box*).

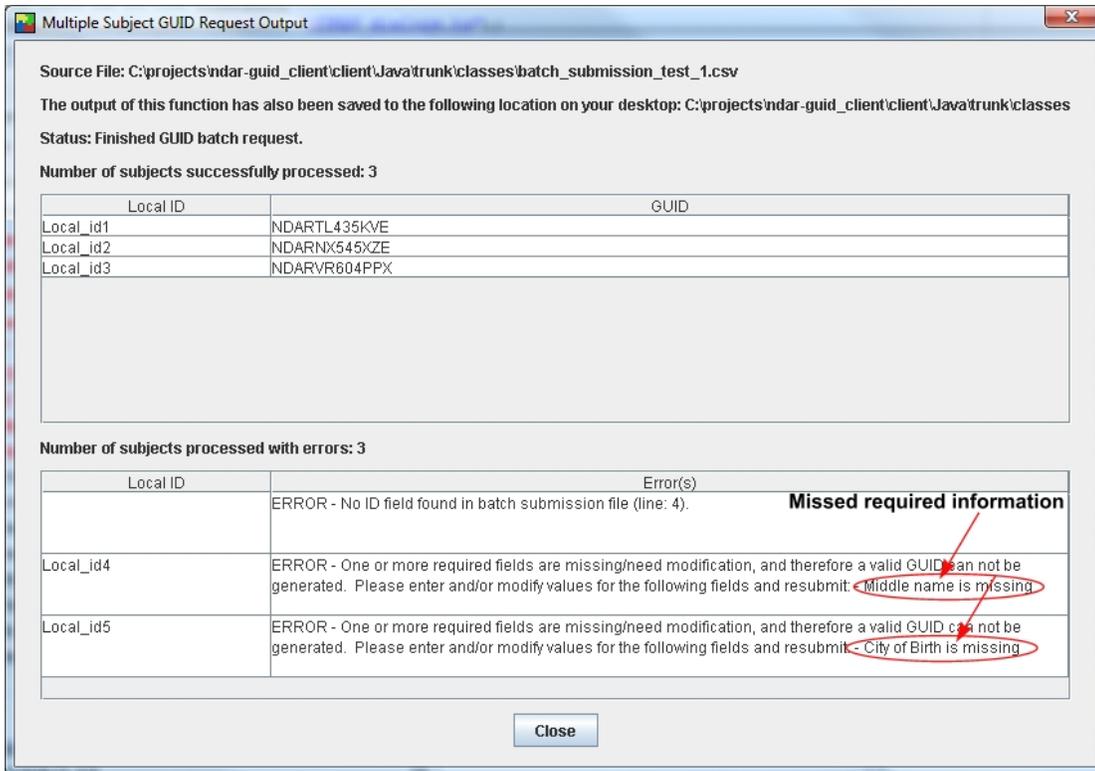


Figure 11. Multiple Subject GUID Request Output Dialog Box (note that the error messages contain missed field names)

Note that if subsequent requests for multiple GUIDs are sent, the required wait time between sending each GUID request containing 50 participants is 30 seconds.

### Checking if GUID Exists in NDAR

To check if a GUID already exists in NDAR, select Functions > Check if GUID Exists from the menu (refer to *Figure 10. Functions Menu*). The Check if GUID Exists in NDAR dialog box appears. Enter the GUID in the GUID field, and click the Check GUID button. The result appears in the Result field (refer to *Figure 12. Check if GUID Exists in NDAR Dialog Box*).

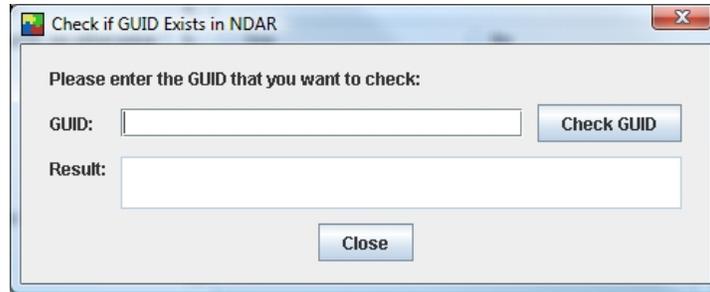


Figure 12. Check if GUID Exists in NDAR Dialog Box

## Getting a Pseudo-GUID

To get a Pseudo-GUID, select Functions > Get Invalid GUID from the menu (refer to *Figure 13. Getting a Pseudo-GUID*). The Get Invalid GUID dialog box appears. Click the OK button. The software will generate a Pseudo-GUID and display it in the dialog box.

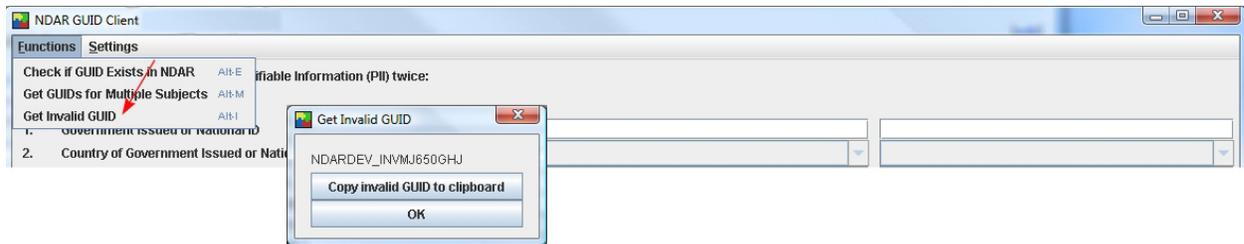


Figure 13. Getting a Pseudo-GUID

To copy the Pseudo-GUID to the Clipboard, click the Copy Invalid GUID to Clipboard button.

## Converting Pseudo-GUIDs to Valid GUIDs

To convert a Pseudo-GUID to a Valid GUID, select Functions > Convert Pseudo GUID from the menu (refer to *Figure 14. Selecting the Convert Pseudo-GUID Menu*).



Figure 14. Selecting the Convert Pseudo-GUID Menu

Once the menu item is clicked, the convert Pseudo-GUID dialog box appears (refer to *Figure 15. Convert Pseudo-GUID Dialog Box*). To convert a Pseudo-GUID to a Valid GUID, enter the Pseudo-GUID and the participant information. Then click the Convert Invalid button. This process is similar to the process of entering participant information to generate a new GUID. If any of the required fields are missing, the error window appears listing the missed fields.

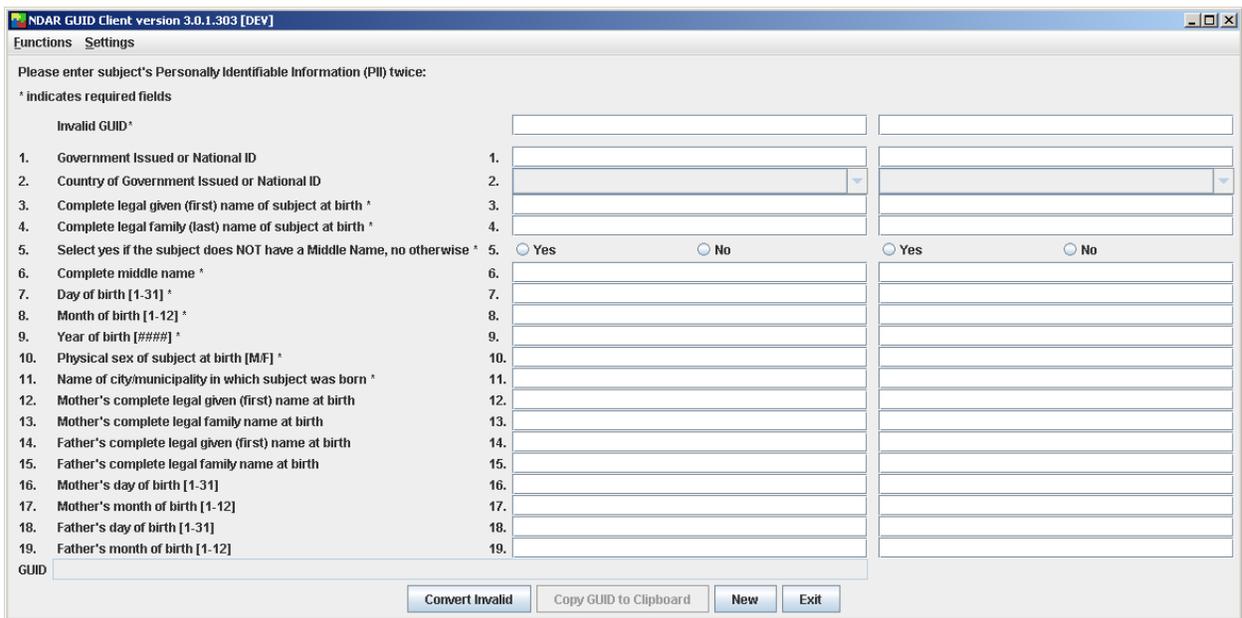


Figure 15. Convert Pseudo-GUID Dialog Box

Select Functions > Get GUID from the menu to return to the main screen (refer to *Figure 16. Returning to the Main GUID Screen*).

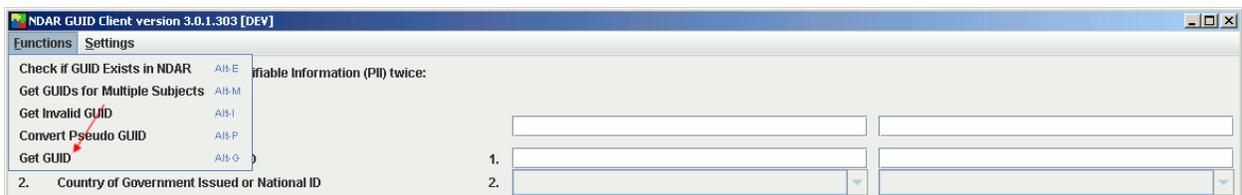


Figure 16. Returning to the Main GUID Screen

## Running the GUIDWS Client Programmatically

An Application Programming Interface (API) for GUIDWS client functions is available (refer to the provided Javadocs). The API shows the functions that are used by the GUID client to communicate with the GUID Web Server. To run these functions programmatically, extend the `AbstractGuidClient.java` class (`gov.nih.ndar.ws.guid.client.AbstractGuidClient`). This class and other supporting classes are available from the `GUID_client.jar` library file supplied as part of the GUID client distribution.

## Troubleshooting

### Timestamp Error

The time between the GUID server and a client computer should be synchronized within five minutes difference in order to access the server. Users in the United States should set their local computer clocks to match the time in <http://www.time.gov/> to avoid an error that can be caused by a time difference between the user's computer and the GUID server. In order to ensure that the user's clock remains correct, it is recommended that users utilize a clock synchronization tool, such as Network Time Protocol (NTP), <http://www.ntp.org/>, or some other local synchronization tool.

If the time is not synchronized within five minutes, an error similar to the example below will be generated.

```
Nov 26, 2007 9:40:15 AM org.apache.axis2.deployment.DeploymentEngine
prepareRepository
INFO: No services directory was found under
C:\projects\guid_client\Java\trunk\classes\repository.
Nov 26, 2007 9:40:15 AM org.apache.axis2.deployment.ModuleDeployerdeploy
INFO: Deploying module: addressing-1.1
Nov 26, 2007 9:40:15 AM org.apache.axis2.deployment.ModuleDeployerdeploy
INFO: Deploying module: addressing-1.2
Nov 26, 2007 9:40:16 AM org.apache.axis2.deployment.ModuleDeployerdeploy
INFO: Deploying module: rampart-1.2
Nov 26, 2007 9:40:16 AM gov.nih.ndar.ws.guid.client.PWCBHandler getPasswordFile
INFO: File Path for pwcbHandler properties is repository\conf\pwcbhandler.properties
org.apache.axis2.AxisFault: An error was discovered processing the <wsse:Security> header.
(WSSecurityEngine: Invalid timestamp The security semantics of message have expired)
    at org.apache.axis2.util.Utils.getInboundFaultFromMessageContext(Utils.java:434)
    at
org.apache.axis2.description.OutInAxisOperationClient.send(OutInAxisOperation.java:373)
    at
org.apache.axis2.description.OutInAxisOperationClient.execute(OutInAxisOperation.java:294)
```

```
at org.apache.axis2.client.ServiceClient.sendReceive(ServiceClient.java:520)
at org.apache.axis2.client.ServiceClient.sendReceive(ServiceClient.java:500)
at gov.nih.ndar.ws.guid.client.AbstractGuidClient.login(AbstractGuidClient.java:200)
at gov.nih.ndar.ws.guid.client.CmdLineGuidClient.main(CmdLineGuidClient.java:152)
```

## Transport Out Error

A transport error will be observed if the entered command line command is incorrect. This error is often encountered when the entered command appears to be correct, but contains special characters. If the error/exception below is encountered, verify that the command has been transcribed correctly.

```
Feb 7, 2008 11:21:09 AM org.apache.axis2.deployment.DeploymentEngine prepareRepository
INFO: No services directory was found under D:\NDAR\GUID_client\repository.
Feb 7, 2008 11:21:09 AM org.apache.axis2.deployment.ModuleDeployer deploy
INFO: Deploying module: addressing-1.1
Feb 7, 2008 11:21:09 AM org.apache.axis2.deployment.ModuleDeployer deploy
INFO: Deploying module: addressing-1.2
Feb 7, 2008 11:21:12 AM org.apache.axis2.deployment.ModuleDeployer deploy
INFO: Deploying module: rampart-1.2
org.apache.axis2.AxisFault: Transport out has not been set
at org.apache.axis2.engine.AxisEngine.send(AxisEngine.java:439)
at org.apache.axis2.description.OutInAxisOperationClient.send(OutInAxisOperation.java:330)
at
org.apache.axis2.description.OutInAxisOperationClient.execute(OutInAxisOperation.java:294)
at org.apache.axis2.client.ServiceClient.sendReceive(ServiceClient.java:520)
at org.apache.axis2.client.ServiceClient.sendReceive(ServiceClient.java:500)
at gov.nih.ndar.ws.guid.client.AbstractGuidClient.login(AbstractGuidClient.java:365)
at gov.nih.ndar.ws.guid.client.CmdLineGuidClient.main(CmdLineGuidClient.java:174)
```