

DEIS

Demographic and Economic Information System

Installation and Maintenance Guide

Version 1.2.6

Table of Contents

About DEIS Version 1.0	3
Hardware and Software Requirements	4
Installing DEIS and its Components	5
Installing Oracle Client 8.1.7 (Optional)	5
Installing the Java Development Kit (JDK) 1.4	5
Installing Apache Tomcat 4.0	5
Installing Utilities	6
Installing DEIS	7
Frequently Asked Questions Regarding Installation	10
Uninstalling DEIS	11
SVG Maps	12
Converting .shp Files to .svg Format	12
Adding a Basemap Using any ODBC-Compliant Database Server	13
DEIS Metadata Tables	15
Adding a Data Source	16

About DEIS Version 1.0

The Demographic and Economic Information System (DEIS) is a web-based application that provides demographers and economists with an easily accessible, single source of national geographic and economic data. Users need no prior database knowledge in order to retrieve information from the DEIS repository.

DEIS was created in 2002-3 by members of the iDataMap Software Engineering Team at the University of Massachusetts at Boston. The team members are:

Jacob K. Asiedu
Honglei Dai
Shengli Fu
Tiegang Han
Abbie T. Lyons
Marilyn A. Moucharite

Hardware and Software Requirements

In order to install and use DEIS, your operating system must include the following components:

- **Application server:** Tomcat 4.0 or higher Binary Distribution.
- **Database server:** Any client-server, JDBC-compliant database.
- **Utilities:** JDOM Version b08 only, ANT Version 1.5 or higher, and Java Development Kit (JDK) 1.4 or higher.
- **DEIS client:** Client must contain Internet Explorer 6.0 or higher or Netscape 6.0 or higher, Adobe SVGView plugin 3.0 and you need to have cookies enabled in your browser.

Installing DEIS and its Components

Installing Oracle Client 8.1.7 (Optional)

Note: Due to conflicts with Oracle's JVM, it is recommended that you install Oracle Client 8.1.7 before installing the Java Development Kit (JDK).

- **Get Oracle Client 8.1.7.** Oracle Client can be found at: <http://otn.oracle.com/software/content.html>
- **Unpack and install Oracle.** Unpack and install Oracle Client according to the instructions included with the release.

Installing the Java Development Kit (JDK) 1.4

- **Get the JDK.** The JDK can be found on the Java Sun web site at: <http://java.sun.com/j2se>
- **Unpack and install the JDK.** Unpack and install the JDK according to the instructions included with the release.
- **Set an environment variable.** Set an environment variable named JAVA_HOME to the pathname of the directory into which you have installed the JDK.

Installing Apache Tomcat 4.0 or above Binary Distribution

- **Get Tomcat.** DEIS is designed to operate with Tomcat 4.0 Binary Distribution and higher. The Tomcat for Windows download page can be found on the Apache Jakarta Project web site at:
<http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0.6/bin/>
- **Unzip and install Tomcat.** Unzip and place the contents of **jakarta-tomcat-4.0.6.zip** in a convenient location in Windows so that the distribution resides in a conventionally named directory, such as C:\Tomcat.
- **Set an environment variable.** Set an environment variable named CATALINA_HOME to the pathname of the directory into which you have saved Tomcat.
- **Start Tomcat.** Tomcat may be started using one of two procedures, either by using an environment variable or by modifying your current working directory. Perform one of the following four procedures:

To start Tomcat by using an environment variable in Windows:

1. At the DOS command prompt, type:

```
%CATALINA_HOME%\bin\startup
```

Or, start Tomcat by using an environment variable in UNIX:

1. At the UNIX command prompt, type:

```
$CATALINA_HOME/bin/startup.sh
```

Or, start Tomcat by modifying the current working directory in Windows:

1. At the DOS command prompt, type:

```
cd %CATALINA_HOME%\bin
```

```
startup.bat
```

Or, start Tomcat by modifying the current working directory in UNIX:

1. At the UNIX command prompt, type:

```
cd $CATALINA_HOME/bin
```

```
./startup.sh
```

- **Verify Tomcat connection.** In your web browser, view Tomcat at:

<http://localhost:8080/> (for local access)

http://your_url:port (for remote access)

After startup, you can also browse the default web applications included with Tomcat at:

<http://localhost:8080>

Installing Utilities

- **Get JDOM Version b08.** The JDOM binaries can be found at:
<http://www.jdom.org/downloads/index.html>
- **Unpack and install JDOM.** Unpack and install JDOM according to the instructions included with the release.
- **Set an environment variable.** Set an environment variable named JDOM_HOME to the pathname of the directory into which you have saved JDOM.
- **Copy jdom.jar file.** Locate **jdom.jar** in JDOM_HOME/build and place a copy of this file in CATALINA_HOME/common/lib.
- **Get JDBC drivers.** Locate and download the JDBC drivers specific to your database server. For example, JDBC drivers for MySQL can be found here:

<http://www.mysql.com/downloads/api-jdbc.html>

- **Copy .jar file.** Locate and place a copy of your JDBC driver **.jar** file in CATALINA_HOME/common/lib.
- **Get ANT.** Download and install ANT from:
<http://ant.apache.org/bindownload.cgi>
- **Unpack and install ANT.** Unpack and install ANT according to the instructions included with the release.
- **Set an environment variable.** Set an environment variable named ANT_HOME to the pathname of the directory into which you have saved ANT.

Installing DEIS

Note: Before installing DEIS, you must have already installed Tomcat, the JDK and the utilities described in the section entitled "Installing Utilities."

- **Establish a program directory.** Create a directory named DEIS wherever you intend to store the source files. For example, C:\DEIS in Windows or your_home/DEIS in UNIX.
- **Get DEIS.** The DEIS package is available for Windows and UNIX platforms through SourceForge.net.

To obtain DEIS using Windows:

1. Download **DEIS1.0.zip** at: <http://idatamap.sourceforge.net/download.html>
2. Unzip **DEIS1.0.zip** into your newly created DEIS program directory.
3. Read the **README.txt** file included with the package.

To obtain DEIS using UNIX:

1. Download **DEIS1.0.tar.gz** at: <http://idatamap.sourceforge.net/download.html>
2. Gunzip **DEIS1.0.tar.gz** into your newly created DEIS program directory.
3. Read the **README.txt** file included with the package.

- **Edit server.xml file.** Configure your Tomcat server's Database Connection Pooling (DBCP) attribute. Locate the **server.xml** file in the /tomcat/conf/ directory. Copy the information in **(Figure 1)** and paste it into **server.xml**, placing the information between the </context> and </host> tags. Edit the driverClassName, url, username, and password name values according to your settings.

```

<Context path="/DEIS1.0" docBase="DEIS1.0" debug="5" reloadable="true"
crossContext="true">

  <Logger className="org.apache.catalina.logger.FileLogger"
    prefix="localhost_DEIS1.0_log." suffix=".txt"
    timestamp="true"/>

  <Resource name="jdbc/Database" auth="Container" type="javax.sql.DataSource"/>
    <ResourceParams name="jdbc/myDatabase">
      <parameter>
        <name>factory</name>
        <value>org.apache.commons.dbcp.BasicDataSourceFactory</value>
      </parameter>
      <parameter>
        <name>driverClassName</name>
        <value>oracle.jdbc.driver.OracleDriver</value>
      </parameter>
      <parameter>
        <name>url</name>
        <value>jdbc:oracle:thin:@jupiter.geog.umb.edu:1521:or817</value>
      </parameter>
      <parameter>
        <name>username</name>
        <value>your user name</value>
      </parameter>
      <parameter>
        <name>password</name>
        <value>your passwr</value>
      </parameter>
      <parameter>
        <name>maxActive</name>
        <value>20</value>
      </parameter>
      <parameter>
        <name>maxIdle</name>
        <value>10</value>
      </parameter>
      <parameter>
        <name>maxWait</name>
        <value>-1</value>
      </parameter>
    </ResourceParams>
  </Context>

```

(Figure 1)

Additional information regarding Tomcat's DBCP can be found at:

<http://jakarta.apache.org/tomcat/tomcat-4.1-doc/jndi-datasource-examples-howto.html>

- **Edit web.xml file.** Locate **web.xml** in the DEIS directory. Modify the **<env-entry>** values in **web.xml** according to your settings. Change only the parameters within the **<env-entry-value>** **</env-entry-value>** tags. An example is shown here:

```
<env-entry>
```

```
<description>Database table that stores the names of all the U.S.
States. Edit metatb1 to match your own table name (if different)
</description>
```

```
<env-entry-name>getStates</env-entry-name>
```

```
<env-entry-value>metatb1</env-entry-value>
```



```
<env-entry-type>java.lang.String</env-entry-type>
```

```
</env-entry>
```

- **Locate and run the deploy script.** The deploy script is located in the DEIS directory. In Windows, the deploy script is named **deploy.bat**. In UNIX, it is named **deploy.sh**. The deploy script will first verify the accuracy of all class paths, and will then compile the application.

To run the deploy script using Windows, type the following command:

```
deploy.bat
```

To run the deploy script using UNIX, type the following command:

```
deploy.sh
```

To compile DEIS without running the deploy script, place ANT in your class path and type the following command (Windows/UNIX):

```
ant deploy
```

- **Restart Tomcat.** If Tomcat is running, begin with step 1 below. If it is stopped, skip step 1 and proceed to step 2.

1. Stop Tomcat. Tomcat may be stopped using one of two procedures: using an environment variable or by modifying your current working directory.

To stop Tomcat by using an environment variable in Windows:

a. At the DOS command prompt, type:

```
%CATALINA_HOME%\bin\shutdown
```

Or, to stop Tomcat by using an environment variable in UNIX:

a. At the UNIX command prompt, type:

```
$CATALINA_HOME/bin/shutdown.sh
```

Or, to stop Tomcat by modifying the current working directory in Windows:

a. At the DOS command prompt, type:

```
cd %CATALINA_HOME%\bin
```

```
shutdown
```

Or, to stop Tomcat by modifying the current working directory in UNIX:

a. At the UNIX command prompt, type:

```
cd $CATALINA_HOME/bin
```

```
./shutdown.sh
```

2. Re-start Tomcat. Tomcat may be started using one of two procedures, either by using an environment variable or by modifying your current working directory.

To start Tomcat by using an environment variable in Windows:

a. At the DOS command prompt, type:

```
%CATALINA_HOME%\bin\startup
```

Or, to start Tomcat by using an environment variable in UNIX:

a. At the UNIX command prompt, type:

```
$CATALINA_HOME/bin/startup.sh
```

Or, to start Tomcat by modifying the current working directory in Windows:

a. At the DOS command prompt, type:

```
cd %CATALINA_HOME%\bin
```

```
startup
```

Or, to start Tomcat by modifying the current working directory in UNIX:

a. At the UNIX command prompt, type:

```
cd $CATALINA_HOME/bin
```

```
./startup.sh
```

- **Redirect your web browser.** In your web browser, type the address listed below and begin using DEIS. Note that *tomcat_host* is your Tomcat host name.

```
http://tomcat_host/DEIS1.0/
```

Frequently Asked Questions Regarding Installation

I restarted Tomcat but I'm getting DBCP errors.

Be sure that your **server.xml** file has been properly configured to work with DEIS. Re-read the step "Configuring Tomcat to use DEIS" in the *Installing DEIS* section.

Uninstalling DEIS

- **Locate the undeploy script.** The undeploy script is located in the DEIS directory. In Windows, the undeploy script is named **undeploy.bat**. In UNIX, it is named **undeploy.sh**.
- **Run the undeploy script.** In DOS, type:

```
undeploy.bat
```

Or, in UNIX, type:

```
undeploy.sh
```

The deployed DEIS application will be removed from your Tomcat engine.

SVG Maps

Converting .shp Files to .svg Format

The following information includes instructions that pertain to the conversion of **.shp** files to **.svg** format, and the exporting of **.svg** files to the DEIS database server. Accordingly, we assume that the reader is a familiar user of ArcView 3.2 or higher.

- **Obtain map.** Locate and download the appropriate **.shp** file.
- **Load .shp file into ArcView.** Import the **.shp** file into ArcView 3.2 or higher. Open the file.
- **Get Avenue script.** Locate the **toascii.ave** script in the `/SVGgen/Avenue_Script/` directory.
- **Load Avenue script.** Import **toascii.ave** into ArcView using the Load Text File utility.
- **Generate .asc file.** Compile and run **toascii.ave** on the **.shp** file currently open in ArcView. An **.asc** file will result.
- **Get SVG Generator.** Locate the SVG Generator source files in the `/SVGgen/SVG_Generator/` directory.
- **Compile and run the SVG Generator.** In DOS or UNIX, compile the SVG Generator source files by typing the following command within the `/SVGgen/SVG_Generator/` directory:

```
javac *.java
```

- **Generate .svg file.** To convert the **.asc** file to a file of type **.svg**, type one of the following commands below, where *pathname* is the relative path of the **.asc** file and *filename* is the name of the previously generated **.asc** file.

To generate an .svg file in DOS:

```
java SVGgen pathname\filename.asc
```

To generate an .svg file in UNIX:

```
java SVGgen pathname/filename.asc
```

The corresponding **.svg** basemap will result.

- **Move the .svg file.** Copy the newly generated **.svg** file (the basemap) to the `/SVGBasemaps` directory. As your collection of basemaps grows, you may decide to organize them by placing them in various subdirectories. For example, a map of California might be placed into a subdirectory named CASVG: `/SVGBasemaps/CASVG/abc.svg`.

Adding Basemaps to DEIS Using any ODBC-Compliant Database Server

If you are using a database such as MySQL, SQL_SERVER or Oracle, follow the steps below in order to add a basemap (.svg) to DEIS.

- **Write SQL statements that will create an empty metadata table.** Using any text editor, create a document named **CreateTable1.sql** and save it in a convenient location. Copy the following SQL statements into the document, in order to create one new metadata table named 'META_SVG'.

```
create table META_SVG(  
STATE char(2) not null,  
DATASOURCE varchar(50) not null,  
GEO_AREA varchar(20) not null,  
SVGFILENAME varchar (50) not null);
```

- **Create empty table.** Open the SQL client associated with DEIS. Type the following command at the prompt, replacing *absolutepath* with the location of **CreateTable1.sql**:

```
> start absolutepath/CreateTable2.sql
```

This command will execute the statements contained in **CreateTable1.sql**, thereby creating an empty table named META_SVG.

- **Write SQL statements that will populate metadata table.** Using a text editor, create a document named **InsertData.sql** and save it in a convenient location. Copy the following SQL statements into that file, substituting the *stateval*, *dataSourceval*, *geoval*, and *basemapval* parameters with the appropriate values¹. (**Figure 1**) depicts the populated META_SVG metadata table.

```
insert into META_SVG(STATE, DataSource, GEO_AREA, SVGFILENAME)  
values ('stateval', 'dataSourceval', 'geoval', 'basemapval');
```

¹The appropriate values are as follows:

- *stateval* is the U.S. state represented by the .svg file (e.g. CA). This value must be the same as that in metatb1 (see section **DEIS Metadata Tables**).
- *dataSourceval* is the data source associated with the .svg file(e.g. PUMS 1990). This value must be the same as that in metatb1 (see section **DEIS Metadata Tables**).
- *geoval* is the geographic area associated with the .svg file (e.g. PUMA). This value must be the same as that in metatb1 (see section **DEIS Metadata Tables**).
- *basemapval* is the name of the relative path to the .svg file (e.g. /SVGBasemaps/CASVG/abc.svg). This value must be the same as that in metatb1 (see section **DEIS Metadata Tables**).

META_SVG

Fields	STATE	DATASOURCE	GEO_AREA	SVGFILENAME
Type	char(2)	varchar(100)	Varchar(50)	varchar(100)
Data Entry	CA	PUMS 1990	PUMA	/SVGBasemaps/CASVG/ca_1990_puma5.svg
	CA	PUMS 1990	MSA	/SVGBasemaps/CASVG/ca_1990_msa.svg
	CA	PUMS 2000	PUMA	/SVGBasemaps/CASVG/ca_2000_puma5.svg
	CA	CPS March 2000	MSA	/SVGBasemaps/CASVG/ca_2000_msa.svg
	CA	CA boundary	Boundary	/SVGBasemaps/CASVG/ca_boundary.svg

(Figure 1)

- **Verify web.xml data.** Locate the **web.xml** file located in the top level of DEIS1.0 directory. Find the following line in the file:

```
SVGTablename=META_SVG
```

If different, replace META_SVG with the name of the metadata table you created above.

DEIS Metadata Tables

DEIS uses metadata tables to populate and maintain information displayed in its state, data source, geographical level, topic, and variable boxes.

For each data source, the DEIS metadata tables preserve the following information:

1. for which states the data source is available,
2. the available levels of geographical boundaries, and
3. the available topics and variables for each data source.

Some data sources may not be available for every state and topics may be unique to a given data source. Variables are unique to a given topic. Because of these dependencies, the front-end controls that make selections available to the user will be filled dynamically as the user makes his selections.

When the user selects a specific state, DEIS queries the **metatb1** metadata table to determine the available data sources for that state. Similarly, when the user selects a data source, DEIS again queries the **metatb1** table to determine the available levels of geographical boundaries. When the user chooses a data source, DEIS will query the **metatb2** metadata table to determine the available topics and variables for that data source. Finally, when the user selects one or more topic and variable and clicks "submit," DEIS will send the query to database.

If metadata pertaining to a particular data source is not contained within tables **metatb1** and **metatb2**, that data source will not be available to the user for mapping. When a new data source is added to the database, the metadata tables must be updated so that the data source will be available for mapping.

The information contained in metadata tables **metatb1** and **metatb2** is depicted in **(Figure 1)**.

metatb1

Fields	State	DataSource	Geo
Type	char(2)	varchar(50)	varchar(20)
Data Entry	CA	PUMS 1990	MSA
	CA	CPS March 2000	MSA
	CA	PUMS 1990	PUMA

metatb2

Fields	DataSource	Topic	Variable	FieldNum	State	TableName
Type	varchar(50)	varchar(200)	varchar(200)	varchar(20)	char(2)	Varchar(30)
Data Entry	PUMS 1990	sex	male	1	CA	pums90ca_vars
	PUMS 1990	sex	female	0	CA	pums90ca_vars
	--	--	--	--	--	--
	--	--	--	--	--	--

(Figure 1)

Adding a Data Source

To add a new data source to the database, follow these steps:

1. Create an empty table to store the contents of the new data source (reference [Procedure A](#)).
2. Populate the empty table with the contents of the new data source by using the SQL*Loader utility (reference [Procedure B](#)).
3. If metadata tables *metatb1* and *metatb2* are already contained in your database, skip Step 3 and proceed to Step 4. Otherwise, if this is the first data source you intend to add, create two metadata tables named *metatb1* and *metatb2* (reference [Procedure C](#)).
4. Append one row to table *metatb1*, inserting the appropriate 'State,' 'DataSource' and 'Geo' information associated with the new data source (reference [Procedure D](#)).
5. Append *n* rows to table *metatb2*, where *n* is the number of data source variables you wish to add. In each row, insert the appropriate *DataSource*, *Topic*, *Variable*, *FieldNum*, *State*, and *TableName* values associated with the new data source (reference [Procedure D](#)).

Procedure A

- **Write SQL statements that will create empty tables.** Using any text editor, create a document named **CreateTable1.sql** and save it in a convenient location. Follow the format below to create SQL statements within the document, in order to create a new table named 'tb1'. Note that the *field*, *type* and *constraint* values are obtained from the raw data file. Include as many lines containing *field*, *type* and *constraint* values as the raw data file contains.

```
create table tb1(  
  field type constraint,  
  field type constraint,  
  ..  
  field type constraint);
```

- **Create empty tables.** Open the SQL client for the database you are using. If using Oracle, type the following command at the prompt, replacing *absolutepath* with the location of **CreateTable1.sql**:

```
> start absolutepathCreateTable1.sql
```

This command will execute the statements contained in **CreateTable1.sql**, thereby creating an empty table named 'tb1' with fields.

Procedure B

- **Write SQL statements that will import data.** The following steps describe the import of comma-delimited data to the table.

Using a text editor, create a control file named **AppendData.ctl** and save it in a convenient location. Copy the following SQL statements into that file. Note that *absolutepath* is the

location of the raw data file (example: C:\ca00012.dat), and *field1*, *field2*, and *field3* are the names of the fields contained in table 'tb1'. **Note:** The fields must be listed in the same order as they appear in the raw data file.

```
load data
infile 'absolutepath'
append
into table dataTable1
fields terminated by ","
(field1, field2, field3..)
```

- **Import data using the SQL*Loader utility.** Open the DOS window. At the prompt, type the following command. You must specify the *username*, *password*, *host*, and *absolutepath*:

```
sqlldr userid=username/password@host control="absolutepathAppendData.ctl"
```

Procedure C

- **Write SQL statements that will create empty metadata tables.** Using any text editor, create a document named **CreateTable2.sql** and save it in a convenient location. Copy the following SQL statements into the document, in order to create two new tables named 'metatb1' and 'metatb2'.

Note: Be sure to duplicate the table names (metatb1 and metatb2) and parameter names (DataSource, Geo, Topic, Variable, TableNum, FieldNum, State, TableName) exactly as indicated below. The parameter values will vary according to the data source.

```
create table metatb1(
STATE char(2) not null,
DataSource varchar(50) not null,
Geo varchar(20) not null);
```

```
create table metatb2(
DataSource varchar(50) not null,
Topic varchar(200) not null,
Variable varchar(200) not null,
FieldNum varchar(20) not null,
State char(2) not null,
Tablename varchar(30) not null);
```

- **Create empty tables.** Open the SQL client associated with DEIS. Type the following command at the prompt, replacing *absolutepath* with the location of **CreateTable2.sql**:

```
> start absolutepathCreateTable2.sql
```

This command will execute the statements contained in **CreateTable2.sql**, thereby creating two empty tables with fields.

Procedure D

- **Write SQL statements that will populate metadata tables.** Using a text editor, create a document named **InsertData.sql** and save it in a convenient location. Copy the following SQL statements into that file, substituting the *stateVal*, *dataSourceVal* and *geoVal* parameters with the appropriate values.

```
insert into metatb1 (STATE, DataSource, Geo)
values ('stateVal', 'dataSourceVal', 'geoVal');
```

Example: Suppose that our new data source is Massachusetts CPS April 2000 data for the geographic area MSA. We then create the following statement:

```
insert into metatb1 (STATE, DataSource, Geo)
values ('MA', 'CPS April 2000', 'MSA');

insert into metatb2 (DataSource, Topic, Variable, FieldNum,
State, TableName)
values ('dataSourceVal', 'topicVal', 'variableVal', 'fieldNumVal',
'stateVal', 'tableNameVal');
```

Example: In reading a supporting document of our new data source, Massachusetts CPS April 2000, we choose a topic **sex** (a field in the data source) that contains two values, 0 and 1, representing female and male respectively. Let us assume that the data source table name is *cpsmaApril2000*. We create the following statements in order to populate the topic **sex** into our application:

```
insert into metatb2 (DataSource,Topic,Variable, FieldNum, State, TableName)
values ('CPS April 2000', 'sex', 'female', '0', 'MA', ' cpsmaApril2000');

insert into metatb2 (DataSource, Topic, Variable, FieldNum, State, TableName)
values ('CPS April 2000', 'sex', 'male', '1', 'MA', ' cpsmaApril2000');
```

Recall the structure of tables metatb1 and metatb2 depicted in **(Figure 1)**. After these insertions, our tables will resemble those in **(Figure 2)**.

metatb1

Fields	State	DataSource	Geo
Type	char(2)	varchar(50)	varchar(20)
Data Entry	CA	PUMS 1990	MSA
	CA	CPS March 2000	MSA
	CA	PUMS 1990	PUMA
	MA	CPS April 2000	MSA

metatb2

Fields	DataSource	Topic	Variable	FieldNum	State	TableName
Type	varchar(50)	varchar(200)	varchar(200)	Varchar(20)	char(2)	varchar(30)
Data Entry	PUMS 1990	sex	male	1	CA	pums90ca_vars
	PUMS 1990	sex	female	0	CA	pums90ca_vars
	--	--	--	--	--	--
	--	--	--	--	--	--
	CPS April 2000	sex	male	1	MA	cpsmaApril2000
	CPS April 2000	sex	female	0	MA	cpsmaApril2000

(Figure 2)