

User Manual

Lipi Designer 4.0

Table of Contents

[1 Lipi Designer introduction 3](#_Toc225827829)

[2 Before you get started 4](#_Toc225827830)

[2-1 Supported platforms and environment 4](#_Toc225827831)

[2-2 Disk space requirements 4](#_Toc225827832)

[2-3 Software requirements 4](#_Toc225827833)

[3 Using Lipi Designer 4.0](#_Toc225827844) 5

[3-1 Creating a new project](#_Toc225827845) 7

[3-2 Loading an existing Project 9](#_Toc225827846)

[3-3 Packaging a project 1](#_Toc225827847)0

[3-4 Contents of the package 1](#_Toc225827848)1

[4 Building Lipi Designer Source code](#_Toc225827841) 12

[4-1 Building lipijniinterface](#_Toc225827842) 12

[4-2 Building javauserinterface](#_Toc225827843) 12

[5 Appendix 1](#_Toc225827849)3

[5-1 Setting environment variables in Windows 1](#_Toc225827850)3

[5-2 Setting environment variables in Linux 1](#_Toc225827851)3

Lipi Designer introduction

Lipi Designer is a Java application which provides a Graphical User Interface for creating, loading & training and packaging projects. It can be used for the rapid creation of recognizers for a set of characters or gestures.

Lipi Designer allows the user to provide a few samples of each shape and builds a recognizer using the Nearest-Neighbor shape recognizer from Lipi Core Toolkit 4.0 for shape recognition. This is especially convenient for actively adding new shapes or modifying old ones, as in gesture-based applications.

The configuration settings for the recognizer are stored in a configuration file, and can be changed at run-time.

If the user has a fixed set of shapes (such as a character set for a script) and is looking to create a highly accurate shape recognizer ,then better results can be obtained by formally collecting data from a large number (say 100) users using one of the [Data Collection Tools](http://lipitk.sourceforge.net/tools.htm), and using one of the shape recognition algorithms provided with the Lipi Core Toolkit.

Before you get started

Supported platforms and environment

Lipi Designer 4.0 has been tested on the following 32 and 64 bit platforms:

* Windows 7
* Ubuntu 10.10

Software requirements

|  |  |  |
| --- | --- | --- |
| Description | Windows XP | Linux |
| Building the C++ module | Visual Studio 2008 | GNU C++ compiler 4.4 |
| Building the java module | Java Development Kit  (JDK) jdk1.6.0\_26 or above | Java Development Kit  (JDK) jdk1.6.0\_26 or above |
| Running the application | Java Runtime Environment  (JRE) 1.6.0\_26 or above | Java Runtime Environment (JRE) 1.6.0\_26 or above |
| Executing utility scripts | ActivePerl and Archive::Zip | Perl 5.10 or above and Archive::Zip |

|  |  |
| --- | --- |
|  | NOTE: Archive::Zip perl module can be obtained from  <http://search.cpan.org/~adamk/Archive-Zip-1.30/lib/Archive/Zip.pm> |

# Using Lipi Designer 4.0

In $LIPI\_ROOT/src/apps/lipiDesigner/bin/ executing ‘lipiDesigner.bat’ in windows and executing ‘lipiDesigner.sh’ in Linux launches the Lipi Designer application, and the following user interface (Figure 1) is displayed.

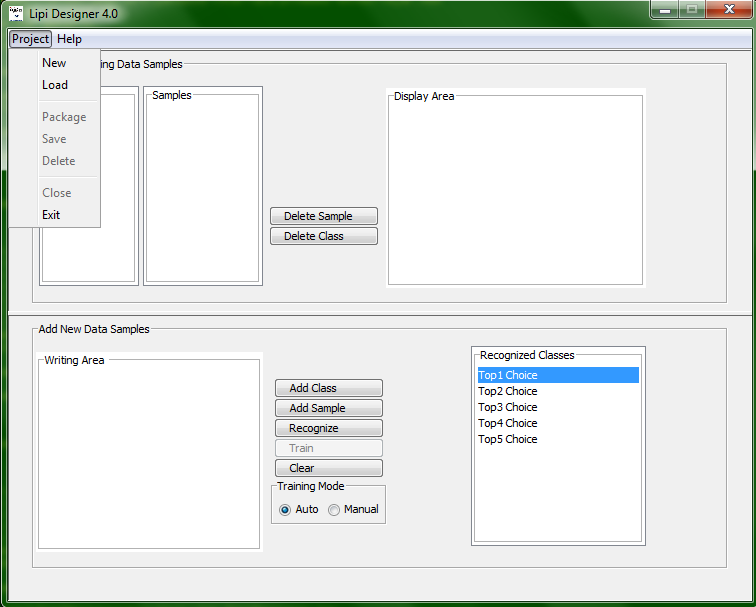


Figure : Lipi Designer user interface

The user interface contains the following elements

|  |  |
| --- | --- |
| Field | Description |
| Classes | Lists the names of all the shape classes in the current project. |
| Samples | Lists the files names of all the training samples of the shape class selected by the user in the Classes list box. |
| Display Area | Displays the ink sample corresponding to the sample selected from Samples list Box. |
| Writing Area | Allows the user to draw shapes for training and recognition. |
| Recognized Classes | Lists the class id and the confidence for top matching classes corresponding to the shape drawn in the Writing Area. |
| Training Mode | Lipi Designer provides two modes for training the recognizer   1. Auto Mode: If the training mode is set to Auto, the recognizer is automatically trained for every new shape added/deleted. 2. Manual Mode: In case of manual mode, the user is required to train the recognizer explicitly, by clicking on the Train button. Any new addition/deletion must be followed by training to update the mdt file. Training the recognizer results in the creation of mdt file. User can add multiple samples before training. |

1. The project menu provides the following items. A project is a collection of shape recognizer specific configuration files and the hand writing data collected using Lipi Designer.

|  |  |
| --- | --- |
| Menu item | Description |
| New | Create a new shape recognition project. |
| Load | Open an existing shape recognition project. |
| Package | Package a shape recognition project.   1. Only a trained project can be packaged. |
| Close | Close the current project. |
| Exit | Exit the application |

1. The items provided in the Help menu are as follows

|  |  |
| --- | --- |
| Menu item | Description |
| About | Displays the version and copyrights |
| Debug | Enables the Debugger to track debug information |

Creating a new shape recognizer project

Click New in the Project menu, and the Create new project dialog box (Figure 2) appears. The dialog box enables the user to enter the project name. For every new project created by the user, a new folder is created under $LIPI\_ROOT/projects.

|  |  |
| --- | --- |
| Warning | WARNING: The name of the new project must not clash with any of the already existing projects. A warning message is displayed in case of a clash. |

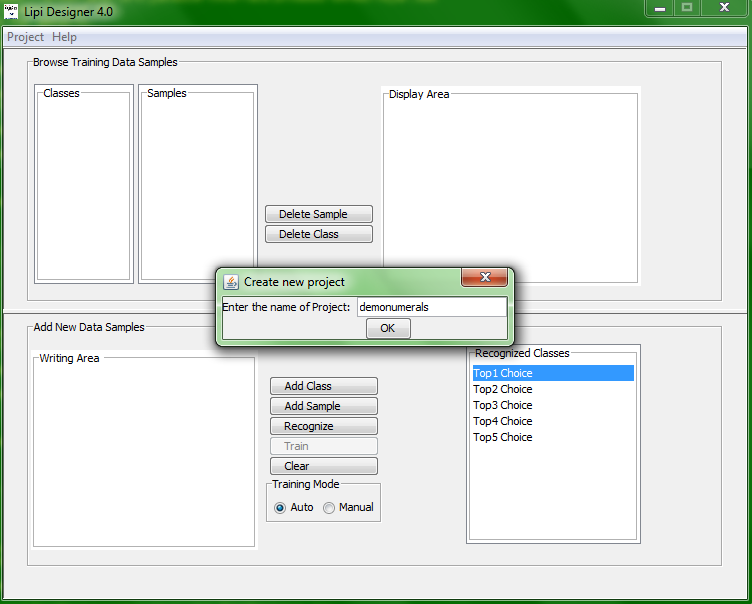


Figure : Create new project dialog box

1. To add a new class to the project, draw a shape in the Writing Area and click on Add Class. User can add multi-stroke shapes also.

|  |  |
| --- | --- |
| Warning | WARNING: A warning message pops up in case an empty stroke is added. |

1. To delete a class, select the class and click on Delete Class.
2. To add samples to an existing class, click on Add Sample. The new samples are added to the class selected the Classes list box. By default, the most recently added class is selected. To add samples to a different class, select the desired class in the Classes list box before clicking on Add Sample.
3. To delete a sample from the class, select the sample and click on Delete Sample.

|  |  |
| --- | --- |
|  | NOTE: Deleting the last sample of a class results in deletion of the class from the project. |

1. To train the recognizer, click on Train. There are two modes for training the recognizer:
   1. Auto Mode: If the training mode is set to Auto, the application automatically trains the recognizer after every addition/deletion of a sample.

* 1. Manual Mode: In this mode, the user is required to train the recognizer explicitly, by clicking on the Train button. The use can choose to invoke training after adding or deleting multiple classes/several samples.

1. The performance of a trained recognizer can be tested using the Recognize functionality of *Lipi Designer*
   1. Draw a shape in the writing area
   2. Click the Recognize button, and the five best matching classes are listed in the Recognized Classes panel (Figure 3), with the first choice indicating the highest confidence result. If the number of matching classes is less than five, the system will display only the available classes, with top choice indicating the highest confidence.

|  |  |
| --- | --- |
| Warning | WARNING: Recognize cannot be called for an empty stroke. |

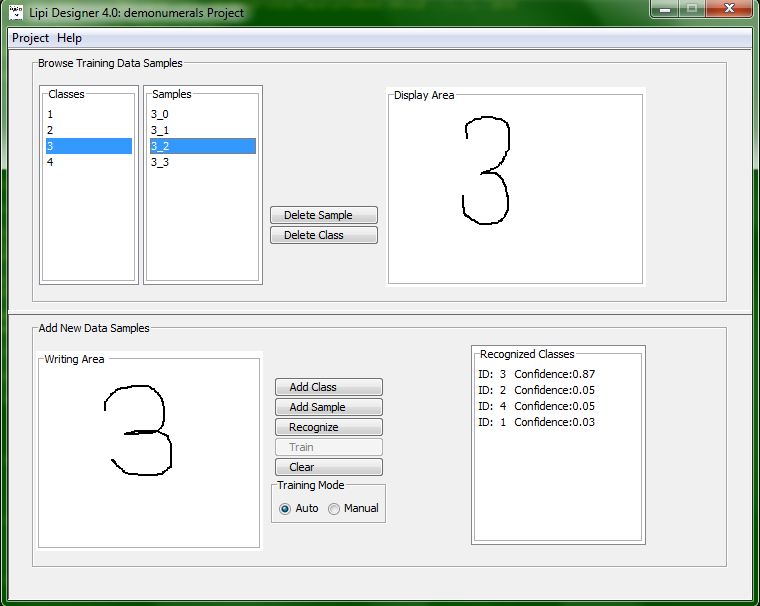


Figure : Recognition results

1. To clear the stroke(s) written in the Writing Area or the result shown in the Recognized Classes panel, click Clear.

Loading an existing Project

To load an existing project, click Load in the Project menu. A window (Figure 4) displaying all the existing projects appears. The user can load a project by opening the projects directory and selecting the <project name>, available under $LIPI\_ROOT/projects/

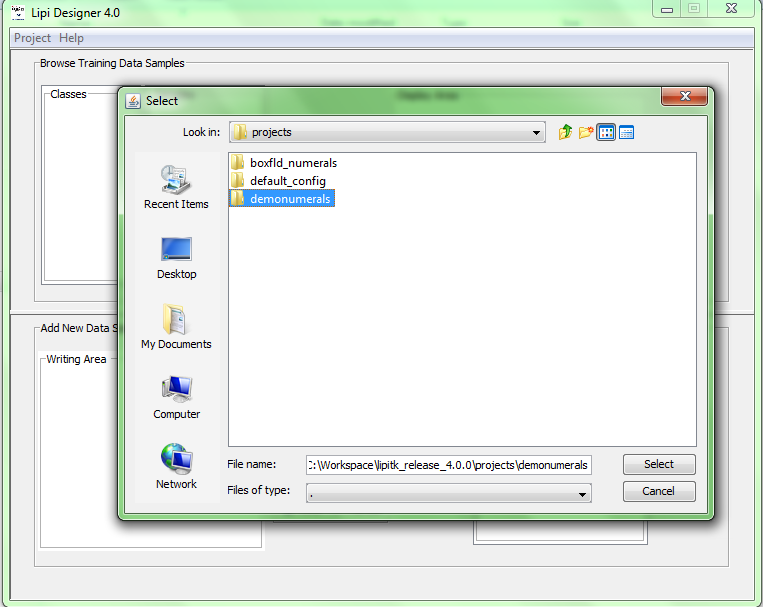


Figure : Load existing project

|  |  |
| --- | --- |
| Warning | WARNING: Before loading/opening a project, the current working project must be closed. |

Packaging shape recognizer project

Once the recognizer has been trained, tested, evaluated and found to be satisfactory, it may be packaged for integrating it into any recognition based applications on client systems. Packaging a project results in the creation of a zip file.

1. To create a package, click Package in the Project menu, and Enter Package Name dialog box (Figure 5) appears.

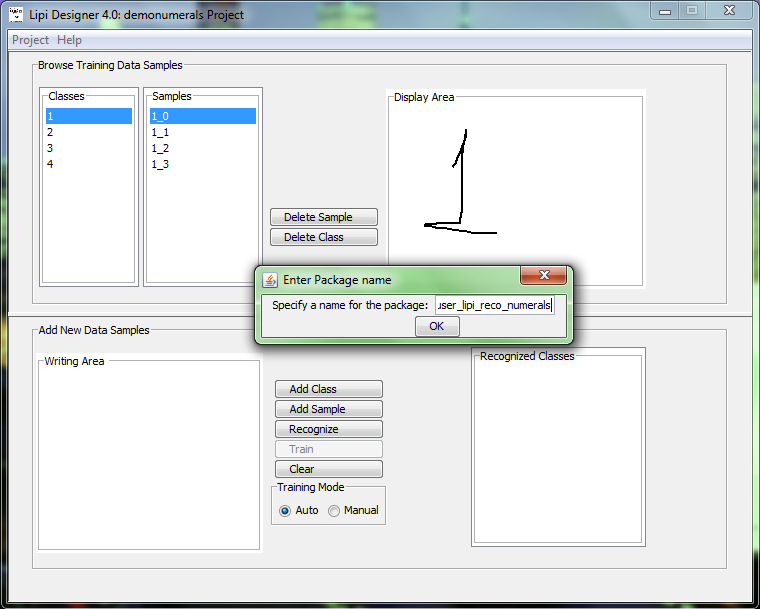


Figure 5: Enter package name dialog box

1. Enter package name and click Ok. Package is created under $LIPI\_ROOT/package directory. A prefix lipi-reco is, by default, added to every package name. However, user can remove the prefix from the final package name.

|  |  |
| --- | --- |
| Warning | WARNING: The package name entered must not clash with any already existing packages. To install this package in client machine please refer to section 10 in lipi toolkit user manual. |

# Building Lipi Designer Source code

|  |
| --- |
| NOTE: This section applies only if source package was downloaded. |

## Building lipijniinterface

a. To build lipijniinterface, on windows, execute the following command:

msbuild lipijniinterface.targets

in directory $LIPI\_ROOT/src/apps/lipiDesigner/src/lipijniinterface/windows.

b. To build lipijniinterface, on Linux, execute the following command:

make -f Makefile.linux

in directory $LIPI\_ROOT/ src/apps/lipiDesigner/src/lipijniinterface/linux.

## Building javauserinterface

a. To build javauserinterface, on windows, execute the following script:

buildjar.bat

in directory $LIPI\_ROOT/ src/apps/lipiDesigner/src/javauserinterface/windows.

b. To build javauserinterface, on Linux, execute the following script:

buildjar.sh

in directory $LIPI\_ROOT/ src/apps/lipiDesigner/src/javauserinterface /linux.

Appendix

Setting environment variables in Linux

In case of Linux, set the environment variable using the appropriate shell command:

export LIPI\_ROOT=/home/testusers/lipi-toolkit