



Software Release Notes

Lipi Core Toolkit

lipitk-core 2.0

Table of Contents

Lipi Core Toolkit.....	1
1-1 Media & Configuration	3
1-1-1 System/Deployment Configuration	3
1-2 Release Components.....	3
1-2-1 Package Name	3
1-3 Documentation	3
1-4 New features in this release.....	4
1-5 Bugs fixed	6
1-6 Limitations/Known Issues.....	7
1-6-1 Spaces in ink file paths in training/test list files causes error	7
1-7 Backward compatibility.....	8

1-1 Media & Configuration

1-1-1 System/Deployment Configuration

lipitk-core 2.0 can be installed on any of the following platforms:

- Windows XP Professional edition
- Redhat Enterprise Linux Edition 4.0
- Ubuntu Gutsy Gibbon 7.10

1-1-1-1 Software requirements

Description	Windows XP Profession edition	Linux
Extracting the contents of the package	Cabarc sdk (Refer to User Manual for download instructions)	tar
Building lipitk-core code	Microsoft Visual Studio 6.0 with SP6 or Microsoft Visual C++ 2005	GCC 3.4.6-9 or above
Executing scripts	Perl 5.0 or above (Refer to User Manual for download instructions for Active Perl)	Perl 5.0 or above

1-2 Release Components

1-2-1 Package Name

Component Name	Version No.	Description
lipitk-core2.0.0-winv6.0.cab	2.0	Package for Windows, for VC6.0
lipitk-core2.0.0-winv2005.cab	2.0	Package for Windows, for VC2005
lipitk-core2.0.0-linux.tar.gz	2.0	Linux package

1-3 Documentation

The following documents are released with core toolkit:

- User Manual – describes the toolkit usage
- Release Notes – describes release information

1-4 New features in this release

1-4-1-1 Generic Nearest-Neighbor Shape Recognizer: NNShapeRecognizer

NNShapeRecognizer is a generic, prototype-based shape recognizer. It allows the run-time selection of different feature extractors (e.g. Point features) and distance measures (such as Euclidean and DTW) for nearest-neighbor classification. It also allows Euclidean distance to be used to short-list prototypes for the more expensive DTW matching.

1-4-1-2 DTWShapeRecognizer and PCAShapeRecognizer deprecated

The **DTWShapeRecognizer** and **PCAShapeRecognizer** shape recognizers have been deprecated as of version 2.0 of the core toolkit. The functionality previously offered by DTWShapeRecognizer is now part of NNShapeRecognizer.

1-4-1-3 Support for VC2005

lipitk-core 2.0 supports the VC2005 also. The vcproj files for every module and are available under the `windows\vc2005` directory.

1-4-1-4 Feature extractor design

The new flexible object-oriented design makes the feature extractor (the component responsible for extracting features) a runtime configurable parameter specified in the shape recognizer config file. A feature extractor, implemented as a dynamic library, is loaded at runtime.

1-4-1-5 PointFloatShapeFeatureExtractor

PointFloodShapeFeatureExtractor is the default feature extractor bundled with *lipitk-core 2.0*,

PointFloodShapeFeatureExtractor extracts the following features from each point along the stroke trajectory:

- X dimension
- Y dimension
- Sine theta
- Cosine theta

1-4-1-6 Setting LIPI_ROOT programmatically

LTKLipiEngineInterface exposes a function, *setLipiRootPath*, which allows applications to set the environment variable LIPI_ROOT programmatically. Applications must call this function for setting the LIPI_ROOT. If an empty string is passed as an argument, the value of LIPI_ROOT is retrieved from the environment.

1-4-1-7 Logging

1-4-1-7-1 Description

The improved *lipitk-core* logger eliminates the drudgery of starting and stopping the *lipitk-core* logging facility.

1-4-1-7-2 Impact on applications using Lipi core toolkit

Applications need not start the *lipitk-core* logger explicitly. Every module takes care of starting and stopping the logger. The following functions have been removed as of version 2.0 of the core toolkit

- a. `initModuleLogger`
- b. `startLogging`
- c. `stopLogging`

Two new key-value pairs have been added to the `lipiengine.cfg`, which allow the application to specify the log level and the log file name. Following keys hold the information pertaining to the *lipitk-core* logger in `lipiengine.cfg`

Key Name	Default value	Description
LogFile	lipi.log	Allows the user to specify the log file name
LogLevel	ERR	Allows the user to specify the log level. Possible values are <ul style="list-style-type: none">• DEBUG• ERR• INFO• ALL• NONE

Applications can set the log level and log file name programmatically using the following functions exported by Lipiengine:

- a. `setLTKLogLevel`
- b. `setLTKLogFileName`

1-4-1-8 Versioning

The LipiTk versioning mechanism has been changed to store all version information in `version.h`, available under the `lipiengine` module. Every module specifies the minimum version supported. The model data file version must lie between toolkit's current version and shape recognizer's minimum supported version.

1-4-1-9 Class/Interface names changed

The names of pre-existing classes and interfaces not in compliance with the *lipitk-core* C++ coding guidelines have been changed. For example, `ChecksumGenerate` has been changed to `LTKChecksumGenerate`.

1-4-1-10 LTKConfigFileReader changed

LTKConfigFileReader class has been redesigned to encapsulate the reading of the config file, retrieve values of various keys, and data member to cache the config file read.

1-4-1-11 Lipi exceptions LTKException

LTKException design has been changed to take error code for object instantiation as opposed to error message in previous versions.

1-5 Bugs fixed

Note: Bug-Ids below are for internal tracking purposes only.

S. No	Bug Id	Brief description
1	LIPTKT-280	Recognize engine fails for one point data sample file
2	LIPTKT-282	The train and test list files should allow the user to specify the path w.r.t. LIPI_ROOT Example: the path of ink file can specified as follows \$LIPI_ROOT/projects/numerals/data/usr0/000000t01.txt 0
3	LIPTKT-293	The result of testing using runshaperec outputs classes that are not present during training.
4	LIPTKT-294	The Perl script,eval.pl, not able to find the LipiTk style sheet, lipitk.css
5	LIPTKT-316	The Euclidean distance between two feature vectors can be computed only if the two vectors are of same length. The function computeEuclideanDistance should return error, if feature vectors of different length are passed as parameters.
6	LIPTKT-317	The application crashes if wrong preprocessing sequence is mentioned in the shape recognizer config file.
7	LIPTKT-271	The function normalizeSize should not modify capture device
8	LIPTKT-338	The LTKShapeRecognizer::recognize accepts vector of integers for subset of classes. In the previous versions vector of Boolean variable was accepted.
9	LIPTKT-295	The Perl script, benchmark.pl, should take project and profile as command line arguments.
10	LIPTKT-296	The names of the dynamic libraries must be all lowercase
11	LIPTKT-297	MDT files created on Windows is not compatible with Linux
12	LIPTKT-331	The Perl script, package.pl, picks up only the first two directories mentioned for the key "data" in package.cfg
13	LIPTKT-349	The user should be allowed to specify non contiguous shape ids listed in ascending order in the train list file.
14	LIPTKT-361	loadModelData should validate parameters common between shape recognizer config file and the mdt header
15	LIPTKT-364	While creating an instance of shape recognizer, lipiengine should pass control information, LTKControlInfo

		<pre> struct LTKControlInfo { string projectName; string profileName; string cfgFileName; string cfgFilePath; string lipiRoot; LTKLogger::EDebugLevel logLevel; string logFile; string toolkitVersion; } </pre> <p>Shape recognizer passes this control information to preprocessing and feature extractor module, while creating an instance.</p>
16	LIPTKT-374	Runwordrec aborts if the UNIPEN ink file is not proper
17	LIPTKT-412	Invalid output when mdv is run with -recver as command line argument
18	LIPTKT-413	<p>The mdv application crashes if user does not specify the input file for the -input option</p> <p>\$>mdv -input</p>
19	LIPTKT-418	The default log file for runwordrec and runshaperec should be lipi.log
20	LIPTKT-432	The application should log the error message to both stderr and the log file and exit if any of the config file (project.cfg, profile.cfg or shape recognizer's config file) are missing,
21	LIPTKT-435	The Perl script, imagewriter.pl, should take file extension and image size as command line arguments

1-6 Limitations/Known Issues

1-6-1 Spaces in ink file paths in training/test list files causes error

1-6-1-1 Problem statement

The runshaperec.exe reports error if the file paths mentioned in the list files for training or testing contain spaces.

1-6-1-2 Workaround

Do not use directory or files names with spaces in them, or (for Windows platforms) use the DOS path instead (e.g. for C:\program files, use "C:\progra~1").

1-7 Backward compatibility

Version 2.0 of core toolkit does not maintain backward compatibility with previous releases of core toolkit.