Please read this file thoroughly to build and run Indaba.

**WHAT IS INDABA**

Indaba is a collaborative content creation and publishing platform. It consists of the following systems:

Field Manager (a.k.a. Builder) – this system is used by field workers to collaboratively create content based on pre-defined project configuration.

Indaba Admin – this system is used by system admins to manage the Indaba Platform. Main functions include: manage user accounts, manage platform resources, create/configure projects, and manage project executions.

Publisher – this system makes content created on Indaba Platform available to end users. It allows users with proper rights to view and download content and aggregation results. It also offers web widgets that can be incorporated into 3rd-party websites to display content created on Indaba Platform.

Workflow Engine – content creation collaboration is coordinated by predefined workflows. This system executes the workflows for all projects.

Data Aggregator – This system periodically performs aggregation computation against raw data created on Indaba Platform. The results are available through the Publisher system.

Control Panel – This system will make some functions of Indaba Admin available to 3rd-party project admins. This system is currently under development.

**TECHNICAL REQUIREMENTS**

Java 5 or above

Tomcat5.5 or above

PHP 5 or above

IDE. We recommend NetBeans 6.8 or above

**BUILD INDABA SYSTEMS**

Publisher and Builder are developed with Java and run on Tomcat. You need an IDE (e.g. Netbeans) to build these two systems.

Indaba\_admin is developed with PHP. You simply deploy it to your web server.

1. Unpack the tar file. You should see 4 folders extracted: Builder, Publisher, indaba\_admin, and schema.
2. To build the Builder and Publisher, simply load them into your IDE and click “build”.
3. Publisher depends on Builder. Depending on how you organize your project directory structure, you may need to modify “custom.properties” in “nbproject” and make sure to set “indaba.builder.dir” properly. It must point to the Builder’s home directory.

**CONFIGURE INDABA SYSTEMS**

Database

Indaba requires the use of a RDBMS. It’s been fully tested with MySQL.

Create a database called “indaba”. Load the latest indaba data model SQL file in the “schema” folder. Then load test data and i18n text SQL file.

Create a database called “indaba\_publisher”. Load the latest indaba\_publisher data model SQL file in the “schema” folder. Then load test data SQL file.

Configure Builder

You need to modify the applicationContext.xml file to use the correct database user name and password. You can also configure the connection pool parameters in this file.

You also need to modify indaba\_config.properties in “web/WEB-INF”, and set the parameters properly, especially the mail-related parameters.

Configure Publisher

You need to modify the applicationContext.xml file to use the correct database user name and password. You can also configure the connection pool parameters in this file.

You also need to modify indaba\_config.properties in “web/WEB-INF”, and set the parameters properly. Publisher’s PDF export function requires the use of open-source solutions wkhtmltopdf and pdftk. Make sure to set the parameters to point to your installations of these two packages.

The system has been fully tested with pdftk.1.44 and wkhtmltopdf.0.9.9. Other versions of wkhtmltopdf don’t seem to work.

**DEPLOY INDABA**

All systems share the same database.

The Builder and the Workflow Engine use the same build. But you should deploy them on separate Tomcat instances. The Workflow Engine is driven by the URL “runWorkflow.do”. You should configure a cron job to fire this URL to the appropriate installation periodically (e.g. every few minutes).

Similarly, the Builder and the Aggregator use the same build. But you should deploy them on separate Tomcat instances. The Aggregator is driven by the URL “dataLoader.do”. You should configure a cron job to fire this URL to the appropriate installation periodically (e.g. every few hours).

Indaba saves data files in predefined directories. You must create /data/indaba directory and give write access to your Indaba instances.

Note that both the Aggregator and the Publisher need to access data files in /data/indaba/aggregation. Data aggregation computation could take long time if you have a lot of content to be processed. You should make sure that the data aggregation process does not interfere with the Publisher. A typical way is to configure the Aggregator to use a temporary working directory, and symlink it to /data/indaba/aggregation after completion.