

# Jejejeke Prolog Beta Testing

Date: June 8<sup>th</sup>, 2010

## 1 Introduction

Some years ago we have create the Jejejeke brand to provide an umbrella for our products in the area of logic programming. The first product has now left our development laboratory and a beta-version can be obtained by interested parties. It consists of a Prolog interpreter.

In the following we will give a overview of some aspects of the beta version. For the Prolog interpreter we provide now a development environment and a runtime library variant. For both variants there exists documentation which covers the programming language Prolog, the API and the user interface of the development environment.

## 2 Product Variants

The runtime library provides an embeddable Prolog interpreter. The Prolog interpreter permits the dynamic creation of interpreter objects. Multiple interpreter objects can be concurrently used in an application. The access to the Prolog interpreter happens through an API which hides many of the details of the Prolog interpreter. In the same time it gives more convenience to the application programmer.

The development environment provides a simple console interface whereby the developer can interact with the Prolog interpreter. It also adds a debug and trace facility to the interpreter. In debug mode exceptions will show a stack trace. In trace mode the box model can be explored. Multiple windows can be used to run multiple threads that share one knowledge base. The API is also available inside the development environment and support to access foreign libraries is provided.

- Development Environment
- Runtime Library

## 3 Product Documentation

Many of the documents are applicable to both the development environment and the runtime library. The only exception is the console manual which exclusively applies the user interface of the development environment. The runtime library does not include any user interface. The development environment is based on Java 1.6, the runtime library is based on Java 1.5. Details on the deployment of both can be found in the installation guide.

The delivered language and predicate set is documented in the language reference. We have tried to stick to the ISO standard and provide compatibility information. Currently we only support a subset of the ISO Prolog core and not yet modules. The Prolog API provides the invocation of the Prolog interpreter and the definition of Java foreign predicates. The Prolog API does not replicate the predicate set, since any predicate can be called via the invocation mechanism. The Prolog API is document in the programming reference.

- Console Manual
- Language Reference
- Programming Interface
- Installation Guide

## **4 Test Scope**

The goal of the current public tests is to determine whether products can be obtained via the sales system and whether the installation of the products works. Further general experiences with the integration scenarios standalone, applet and servlets are in the focus, as well as the interplay between the development environment and the runtime library.

Further tests that will deepen the details of the programming language, the programming interface and the user interface are currently not planned. Corresponding tests are planned for the alfa version. This version will come with an extended language- and function set. The development is internally done in parallel to the beta testing.

Jan Burse  
Managing Director