## Flex Toolkit

This document describes the Flex expert system toolkit, an expressive and flexible rule-based development system for building and delivering scalable and flexible expert systems and business rules applications. Flex provides a comprehensive and versatile set of facilities for both programmers and non-programmers to construct reliable and maintainable applications.

### Hybrid Expert Systems

Flex is an expressive and powerful expert system toolkit which supports frame-based reasoning with multiple inheritance, rule-based programming and data-driven procedures fully integrated within a logic programming environment. To make these constructs accessible in an intuitive way, Flex contains its own dedicated English-like Knowledge Specification Language (KSL).

### Rules, rules and rules ...

Flex includes support for different types of rule-based inferencing. The main two are forward-chaining production rules, ideal for data-driven reasoning and backward-chaining rules best suited for goal-based deduction. In addition, Flex is integrated with [Flint](http://www.lpa.co.uk/fln.htm) which offers ways of handling inexact reasoning namely Fuzzy Logic, Bayesian Updating and Certainty Factors. This means you can describe your business rules and processes, even when you do not have a complete functional description.

### Delivery options

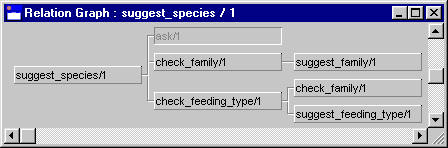
Run-time delivery options include: compiling to a self-contained Windows application using the proprietary LPA run-time generator. Alternatively, the rule-based component can be embedded within Java or C# or .NET using the [LPA Intelligence Server toolkit](http://www.lpa.co.uk/int.htm). In addition, Flex programs can be delivered straight on to the internet using [WebFlex](http://www.lpa.co.uk/wfs.htm).

### Knowledge Specification Language

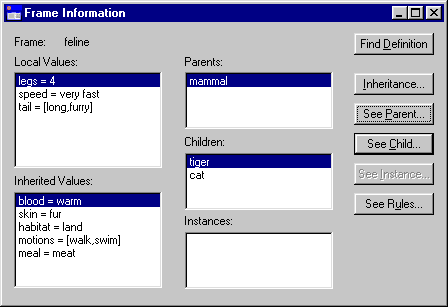
Flex has its own expressive English-like Knowledge Specification Language (KSL) for defining rules, frames and procedures. The KSL enables developers to write simple and concise statements about the expert's world and produce virtually self-documenting knowledge-bases which can be understood and maintained by non-programmers. The KSL supports mathematical, boolean and conditional expressions and functions along with set abstractions; furthermore, the KSL is extendable through synonyms and templates. By supporting both logical and global variables in rules, Flex avoids unnecessary rule duplication and requires fewer rules than most other expert systems.

### Integrated Development Tools

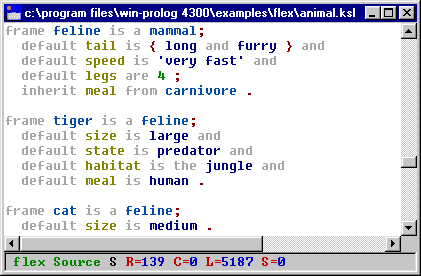
Flex contains a highly interactive development environment with integrated editor, frame browser, debugger, etc. Relationships and the connections between frames can be viewed graphically, and printed for reference.



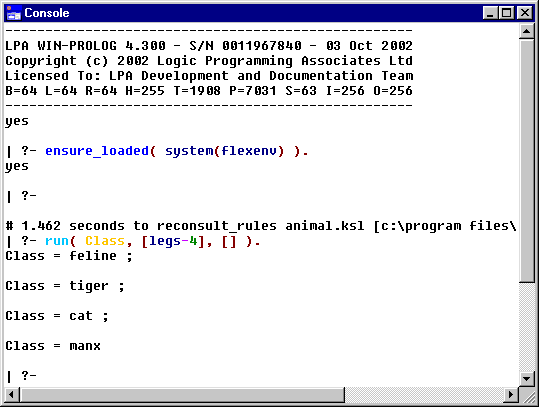
More detailed information about frames, classes and other objects can be viewed through browser dialogs.



The Flex development environment uses dynamic Rich Syntax Colouring to identify class names, actions, numbers, predicates, etc., in real time during editing and query entry.



Meanwhile, queries can be entered directly into the console window.



### Extendibility

Flex goes beyond most expert system shells in that it employs an open architecture and allows you to access, augment and modify its behaviour through a layer of access functions. Because of this, Flex is referred to as a toolkit rather than a shell. The combination of Flex and Prolog provides a hybrid expert system toolkit with a powerful general-purpose AI programming language, and results in a functionally rich and versatile development environment where you can fine tune and enhance the built-in control mechanisms to suit your own requirements.

### Integration with other Toolkits

LPA provides a lot of AI toolkits, and with Flex and its Prolog layer, you can use any of them. So, say, if you want to do some data-mining or case-based reasoning, you would simply plug-in the relevant toolkit.

### Rule-based Systems

Rule-based or expert systems allow the scarce and expensive knowledge of experts to be explicitly stored into computer programs and made available to others who may be less experienced. They range in scale from simple rule-based systems with flat data to very large scale, integrated developments taking many person-years to develop. They typically have a set of if-then rules which forms the knowledge base, and a dedicated inference engine, which provides the execution mechanism. This contrasts to conventional programs where domain knowledge and execution control are closely intertwined such that the knowledge is implicitly stored in the program. This explicit separation of the knowledge from the control mechanism makes it easier to examine knowledge, incorporate new knowledge and modify existing knowledge.

### VisiRule

The latest development for Flex is the arrival of [VisiRule](http://www.lpa.co.uk/vsr.htm). VisiRule lets you simply draw your rule-based applications as charts using a dedicated graphical tool and generates the executable code for you.

[Find out more about Flex](http://www.lpa.co.uk/flx_det.htm)