

New and improved features in ErgoAI 2.0 (Myia)

This release addresses a number of stability issues and also introduces additional features and extensions -- primarily in the packages (see the [Guide to ErgoAI Packages](#)) for more information. Updated manuals and tutorials can be accessed from the [Ergo Suite Documentation](#) page on the [Coherent Knowledge](#) website. The new features are listed below. ErgoAI was formerly known as Ergo Suite. ErgoAI includes the Ergo reasoner (also known as ErgoAI Reasoner), the ErgoAI Studio, and the various connectors and interfaces to external software and formats.

1. A Python to Ergo interface. Allows Python programs to query and update Ergo knowledge bases. Works with Python 2.7.x and 3.x. Details are in the Guide to ErgoAI Packages, Chapter 3. A [runnable example](#) is given in the [ErgoAI Example Bank](#).
2. A new *HTTP and Web Services* package (`\http`), which simplifies the job of accessing the Web and talking to Web services. See Chapter 4 of the Guide to ErgoAI Packages.
3. Enhancements to the Ergo-to-Java package (`\e2j`), which lets one invoke methods on Java classes from within Ergo Reasoner. In this way, it is possible to use Ergo to *script* the invocation of various Java methods. Details are in the Guide to ErgoAI Packages, Chapter 2.
4. Improvements in the display of and control over the floating point (doubles) numbers. See Section 50.1.3 of the [ErgoAI Reasoner User's Manual](#).
5. Enhancements in the package for obtaining explanations (`\why`), including a simplified JSON representation. See sections 52.3 and 52.6 of the ErgoAI Reasoner User's Manual.
6. New syntax for combining binary operators like `<`, `>`, `!=`, `~`, and others with frames (like it was previously possible with `:` and `::`). For instance, `?p:person[age->?A>30]` is a shorthand query for `?p:person[age->?A] \and ?A>30`.
7. A new `system{?prop=?value}` primitive, which allows KBs to get information about the ErgoAI installation and the host OS.
8. The `once{query}` and `random{query}` primitives. The first ensures that the query will return only one of the answers (always the same on each call); the second is similar, but it will return a randomly chosen answer each time. This can be useful in implementing intelligent chatbots. See ErgoAI Reasoner User's Manual, Section 24.3.
9. Typed variables are now allowed in quantifiers, e.g., `forall(?X^^person)^(...)`. This added flexibility simplifies translation of certain sentences from natural language to Ergo.
10. The `random{...}` primitive for generation of random numbers. See Section 33 in ErgoAI Reasoner User's Manual.