

# Introduction to NaturalX

This chapter contains a short introduction to component-based programming involving the use of the NaturalX interface and a dedicated set of Natural statements.

The following topics are covered:

- Why NaturalX?
- 

## Why NaturalX?

Software applications that are based on component architecture offer many advantages over traditional designs. These include the following:

- Faster development. Programmers can build applications faster by assembling software from prebuilt components.
- Reduced development costs. Having a common set of interfaces for programs means less work integrating the components into complete solutions.
- Improved flexibility. It is easier to customize software for different departments within a company by just changing some of the components that constitute the application.
- Reduced maintenance costs. In the case of an upgrade, it is often sufficient to change some of the components instead of having to modify the entire application.

Using NaturalX you can create component-based applications.

You can use NaturalX to apply a component-based programming style. However, on this platform the components cannot be distributed and can only run in a local Natural session.