Tutorial Table of Contents

A Brief Note:

This document provides a general outline for the tutorials that are provided with the Clarion Library. While it is by no means exact (or necessarily complete for that matter), the order in which these tutorials are presented herein represent an approximate ordering in which they should be viewed. As a general rule of thumb, by following our recommended order, the material presented within the tutorials should gradually move from easy to advanced.

Setting Up & Using the ACS (in Basics Tutorials)

Walkthrough of the Simple Hello World Task

The using Clause
Declaring the SimpleHelloWorld Class
The Main Method
Initializing the World
Agent Initialization
Tweaking Parameters

Running a Simulation (& Continued Walkthrough)

Initializing the Sensory Information
Perceiving and Acting
Processing Outcomes and Delivering Feedback
Killing an Agent

Intermediate ACS Setup (in Intermediate Tutorials)

Optimizing Task Performance via "Tuning" Parameters

Making Global Parameter Changes
Making Local Parameter Changes

Setting Up the Working Memory

Manually Setting a Chunk in Working Memory
Using Action Chunks

Setting Up & Using the Goal Structure (in Basics Tutorials)

Setting Up the Goal Structure

Manually Setting a Goal

Using Action Chunks

Intermediate MS and MCS Setup (in Intermediate Tutorials)

Setting Up and Using Drives and Meta-Cognitive Modules

Initializing a Drive

The Drive Equation

Stimulating a Drive

Accessing Agent Meta Info

Correlating Drives and Meta-Cognitive Modules

Meta-Cognitive Module Integration

Basic Customization (in Customizations)

Customized Methods (Using Delegates)

Specifying Delegates as Parameters during Initialization

Creating Custom Rules

Using the `SupportCalculator` Delegate to Set Up an IRL Rule

Initializing the IRL Rule

Using the `SupportCalculator` Delegate to Set Up a Fixed Rule

A Note on the Generically Typed `DimensionValuePair<DType,VType>` Class

Initializing the Fixed Rule

Generic Equations

Useful Features (in Features & Plugins)

Viewing an Agent's "Internals"

Logging (using `Trace`)

The Implicit Component Initializer

Pre-Training
Auto-Encoding
Distributed Dimension-Value Pairs
Populating the Input and Output Layers of an Implicit Component

Timing

Response Time
"Real-time" Mode

Asynchronous Operation

Setting Up & Using the NACS (in Advanced Tutorials)

Setting Up & Performing Reasoning

A Walk-through of the "Simple Reasoner" Task

Distributed Dimension-Value Pairs
Adding Knowledge to the GKS
Initializing Associative Memory Networks
Initializing Associative Rules
Performing Reasoning

Setting Up & Using Episodic Memory

Creating Episodes
Initializing Associative Episodic Memory Networks
Generating New Knowledge and Associative Rules
Performing "Offline" Learning

Advanced ACS Setup (in Advanced Tutorials)

Interacting with the NACS

Making Reasoning Requests

Specifying Alternative Dimensions

Filtering Input/Conclusions

Retrieving Chunks from the GKS

Interacting with Episodic Memory

Retrieving Episodes

“Offline” Learning

Generative Actions

An Example: Using Generative Actions to Change Local Parameters

Using Plugins (in Features & Plugins)

The Serialization Plugin

Serializing (or Saving) Various Aspects of a Simulating Environment

De-serializing (or Loading) Various Aspects of a Simulating Environment

Interacting with Front-Ends

Remote Simulating Environments

Communicating via XML

Communicating via JSON

The Keyboard and Mouse Plugins

Using the “Built-In” Plugin Actions

Advanced Customization (in Customizations)

Getting Started
ACS Structure
NACS Structure
MS Structure
MCS Structure
Interfaces and Templates

How to Implement a Custom Component
Requirements for Implementing a Custom Component
Implementing a "Factory"
Implementing a "Parameters" class
Local (per instance) Parameters
Global (`static`) Parameters
Factor # 1
Factor # 2
Factor # 3
Factor # 4
Factor # 5
Factor # 6
Factor # 7
Commiting and Retracting
Using the InitializeOnCommit Property

How to Implement a Custom (Secondary) Drive
Implementing the Nested "Factory" Class
Implementing the Nested "Parameters" Class

Serializing a Custom Component (or Drive)
Specifying the System.Runtime.Serialization Resource
The `DataContract` Attribute
The `DataMember` Attribute
Pre/Post Serialization and Deserialization Attributes
Serializing the Global (`static`) Parameters