

# ADAMS

Advanced **D**ata mining **A**nd **M**achine learning **S**ystem

Module: adams-rats



Peter Reutemann

December 21, 2016

©2014-2016



THE UNIVERSITY OF  
**WAIKATO**  
*Te Whare Wānanga o Waikato*



Except where otherwise noted, this work is licensed under  
<http://creativecommons.org/licenses/by-sa/4.0/>

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Flow</b>	<b>7</b>
2.1	Actors . . . . .	7
	<b>Bibliography</b>	<b>9</b>



# Chapter 1

## Introduction

The *Reception And Transmission System*, or RATS for short, is aimed at scenarios where data is being received from various sources, processed and then transmitted to various destinations again. It simplifies the design of flows that handle these kind of scenarios, by providing off-the-shelf *receivers* and *transmitters*, e.g., for directory polling or FTPing files.

In contrast to regular ADAMS flows, the RATS sub-system is event-driven and not data-driven. The received data can then be processed by a data-driven flow before sending it somewhere else.



# Chapter 2

## Flow

### 2.1 Actors

The following standalone actors are available:

- *Rats* – This standalone encloses multiple RAT configurations.
- *Rat* – Definition of how to receive data and how to transmit it, based on the specified *RatInput* (= receiver) and *RatOutput* (= transmitter).
- *RatPlague* – creates copies of itself, one for each of the defined input queues, feeding into the same output queue.
- *LabRat* – the actual Rat setup gets generated at runtime using a generator.

The following control actors are available:

- *ChangeRatState* – changes the state (eg RUNNING, PAUSED) of a *Rat* actor. This allows the startup of Rat actors in a paused state before activating them later on.



# Bibliography

- [1] *ADAMS* – Advanced Data mining and Machine learning System  
<https://adams.cms.waikato.ac.nz/>