

Precise Recovery

Precise recovery is the strongest recovery class and it completely masks failures and ensures that the output produced by an execution with failure and recovery is identical to the output produced by an failure-free execution (exactly-once). Achieving this guarantee requires input preservation as well as output preservation. (Reference: J.H. Hwang et al., High-Availability Algorithms for Distributed Stream Processing, 2005)

In Odysseus, the precise recovery uses an external application, [Backup of Data Streams \(BaDaSt\)](#), that preserves all incoming elements after the last [checkpoint](#). After a crash of Odysseus and a restart, Odysseus receives all preserved elements after the last checkpoint from BaDaSt and processes them before it processes the newer elements (which are again preserved by BaDaSt, so there is no data loss). Note that Odysseus will try to use the original data source again after some time, but for data streams with a high data rate and a long time span between crash and restart, it is possible that Odysseus can not switch back to the original source. In that case, Odysseus continues to process the elements from BaDaSt.

Typically this procedure results in duplicates that have to be eliminated for a precise recovery. This is done by special sender operators that automatically replace the common sender operators.

Requirements: To use the precise recovery, you have to (1) define a BaDaSt recorder for each data source (in the same file as the source definition) that elements shall be preserved (see [BaDaSt](#)), (2) choose the precise recovery and (3) manage the checkpoints (see [Checkpointing](#)):

Example

```
#PARSER PQL
[Definition of data source "bid"]
#BADASTRECORDER type=TCPStringRecorder sourceame=bid

#RECOVERYCONFIGURATION PreciseRecovery protectionpointunit=MINUTES protectionpointperiod=1
[Definition of queries to apply precise recovery]
```