Documentation of the Assistant module "Assistant Utilities"

FKE

November 5, 2018

Contents

1 Description

Introduction Assistant Utilities exists as a project inside the Assistant solution. The project contains classes used by multiple other projects in the same solution. These classes include

- BreanosLogger
- TimeBatcher
- SizeBatcher
- FullBatcher

BreanosLogger BreanosLogger fulfills the role of a centralized class for accessing the Nlog Logger so log messages share a more common layout across the Assistant.

Batchers The batcher classes are implementations of a solution to an issue in inter component communication particularly surfacing in distributed environments like the Service Fabric Applications. As components communicate, they send out "messages" to one another. Some are structured as imperatives, e.g. "Do this task", which would effectively be method calls in programming. Others are created as "messages", i.e. a set of data, the sender does not know what the recipient will do with, e.g. "I have this letter here that I'm giving to you". In both cases the "further apart" the two communicating participants are, the more overhead is generated with each passed message. With Service Fabric Applications' remoting technology (calling a method in another service, possibly in *some* other SFA, somewhere), the overhead might be substantial enough to be noticeable, consisting of, for example, establishing the connection and communication session between the services. Given a number of messages to be sent within a certain time frame from service A to service B, at some point it may become less taxing on the system, instead of sending each message separately, to send collections of messages in a batched manner. However, if you were to, say, collect a million messages and send them all at once, message transmission (of a million messages as one comparatively huge message) might fail due to size constraints for messages employed in the system. For the issue of too many single messages, TimeBatcher can be used to collect messages over a configurable period of time before forwarding them all. For the issue of one far too big message, comprised of many smaller ones, SizeBatcher can be used to break up a collection of messages in several sub-collections. FullBatcher combines these two so messages are first processed by a TimeBatcher and then sent through a SizeBatcher before being forwarded as a collection of collections of messages. These collections of messages, collected in collections of their own can then be forwarded separately which will cause far less overhead than sending each of the messages separately and be less likely to run into message size constraints.

2 Installation

Assistant Utilities is installed by deploying it within the Service Fabric Cluster Application “Assistant”.

3 Configuration

To configure the service’s logging, one can alter the Nlog.xml to switch to a different target / layout.[1]

4 Dependencies

Due to its unification of NLog usage, Assistant Utilitieshas a dependency on Nlog.

5 Requirements

Keine

[1] `https://github.com/nlog/NLog/wiki/Configuration-file`