What is Publish/Subscribe?

Message producers:
- Publish or send messages to abstract subjects (strings)
- Publish to any subject at any time, independent of others
- No knowledge of consumers and their subscriptions necessary
- May publish to a subject no consumer subscribes to
- No prior registration of subjects required
- Subjects can be created dynamically, at will
- No connection or “coupling” of a subject to a producer process
- Publish messages at will in a “publish-and-forget” mode

Message consumers:
- Subscribe to subjects (strings), wildcards often supported
- No knowledge of producers and the subjects they publish to required
- May subscribe to a subject that no producer ever publishes to
- Operate in a “subscribe-and-forget” mode

Asynchronous communication:
- Producers do not block when a message is published
- Producers do not have to wait for some process to receive it
- Consumers receive messages via an asynchronous callback mechanism, running in a separate thread
- Consumers do not block when the subscription is made

What is cMsg and how does it decouple?

Software implementing a sophisticated version of the publish/subscribe model and thus automatically decouples users

Some cMsg Features:
- 2 subject fields (subject & type) used in publishing & subscribing
- Messages hold all fundamental data types, their arrays, as well as cMsg messages and their arrays – as many components as desired
- Binary conversions are handled automatically
- Message routing is performed by high-performance background servers

The API is narrow in that only basic messaging functionality is provided, i.e. there is only one type of message and one way to fill, publish, subscribe, and receive messages

Additional useful synchronous capabilities are provided for convenience.

Conclusions

The asynchronous publish/subscribe model is ideal for implementing a decoupled interprocess communication system. Producers can publish messages to any subject with no regard for the existence of other producers or consumers. Consumers can subscribe to any subject with no regard for the existence of other consumers or producers. New consumers can be added to implement additional functionality with no change needed to the existing system.

The cmsg package implements a narrow interface that has changed hardly at all over five years. It provides basic messaging functionality, and all additional customization must be done by developers via conventions in the control system. We have often created simple systems to implement basic functionality, then added new functionality via new processes that listen in on the existing messaging and perform some new task (e.g., archiving or display), with no change to the original system needed. In this way functionality can be built up incrementally and transparently, and modified as needed with no effect on existing systems.