EntireX MODERNIZES CICS WITH CHANNELS AND CONTAINERS

By Chris Pottinger, EntireX R&D, Software AG

CICS channels and containers is an architecture feature that shares data buffer resources without any content or size constraints. EntireX takes advantage of CICS’ emerging technology advancements for sizing and flexible sharing of data amongst diverse processes so you can increase the flexibility of your applications end to end.

BACKGROUND

In this day and age, the ability to handle data across a wide spectrum of workload platforms, programming environments, and diverse delivery environments sets the stage for discussing how EntireX and CICS are modernizing essential mainframe assets. The following information highlights how EntireX uniquely expands and extends the value of utilizing CICS channels and containers.

CICS channels and containers were introduced by IBM in Transaction Server V3.1 as a new architecture to share data buffer resources across CICS Regions without any content or size constraints. Historically pervasive as a programming technique, CICS programmers have used an embedded protocol to function ship contiguous data buffers across Regions called Dynamic Program Link (DPL) as shown in Figure 1. Although DPL has become the de-facto standard for function shipping application data across CICS programs and application owning regions (AORs), this technique has been limited to a 32K byte buffer size that is referred to as a COMMAREA (communication area).

The popularity of DPL is due to the programmatic simplicity of developing applications using the CICS EXEC LINK API, COMMAREA processor efficiency, and the built-in scalability of data routing within a complex of CICS images running on the System z platform (simple Multi Region Operation or multi-LPAR’d CICSPLEX coupling). The simple CICS COBOL program reference for receiving the DPL message is the DFHCOMMAREA keyword structure in the LINKAGE Section which provisions in-bound data directly to the Procedure Division logic. The CICS API for sending DPL messages to other programs is the EXEC CICS LINK (with specified data parameters) which returns by default.

For applications that require the sharing of data greater than the 32KB DPL maximum, other programming techniques such as shared memory (see DFHCOMMAREA Large Buffer Interface below) or persisted data queueing (VSAM, DB2, TSQ, MQ, or Reliable RPC) have been used but are either more complex or limited to local resources.

WHAT ARE CHANNELS AND CONTAINERS?

A container is a named reference to a CICS-managed storage area that can hold any form or size of application data. A container may be any size and can hold data in any format that the application requires. An application can reference any number of containers. CICS provides EXEC API verbs to create, delete, reference, access, and manipulate a container as well as to associate it with a channel.1 The size of a container is limited only by the amount of storage available to CICS.

A channel is a uniquely named reference to a collection of application parameter data held in containers. A channel is analogous to a COMMAREA, but it does not have the constraints of a COMMAREA. CICS provides EXEC API, which associates a named channel with a collection of one or more containers. This is an easy way of grouping parameter data structures which may pass to a called application. CICS destroys a channel when it can no longer be referenced, which means, when a channel becomes ‘out of scope’.2

BENEFITS OF USING ENTIREX

Unbound by traditional size or content constraints, the channels and containers approach improves data sharing flexibility but still maintains the workload isolation, scalability, and routing features of a multi-instance CICS environment.

The channels and containers approach does not affect how existing COMMAREA applications work but provides an enhanced technique for sharing data between applications. This will allow for the evolution of mainframe application design and usage which progressively may require larger amounts of data to pass between applications. This enhanced design also provides the possibility to dynamically add or remove parts of the passed data.

2 Ibid.
Simplicity. Like the DPL/COMMAREA method, channels and containers is a well-documented technique to develop program interfaces for sharing data within CICS Regions or with off-board programs. Channels and containers are configured through the standard CSD tools or the new Explorer. Application interfaces require programming the proper API commands to use the channels and containers as a data sharing method as demonstrated in the examples in figures 2 and 3.

```
EXEC CICS PUT CONTAINER(containerA)
   CHANNEL(channelA)
   FROM(dataA)
EXEC CICS LINK PROGRAM(program2)
   CHANNEL(channelA)
```

**FIGURE 2:** Passing a channel from program1 on a link to program2

```
EXEC CICS GET CONTAINER(containerA)
   CHANNEL(channelA)
   INTO(dataA)
```

**FIGURE 3:** Program2 receiving the container (will return container to program1)

The EntireX Workbench simplifies this approach. It generates IDLs from source artifacts (CICS server programs) and also generates source programs (CICS client programs) to make development of channel and container interfaces easy. Essentially, EntireX makes the channels and containers access method transparent to application developers through its RPC messaging architecture.

Flexibility. The COMMAREA is a limited, inflexible block of application data that contains all of the data to be passed to the called program, even if only part of this data is required within the program.

```
EXEC CICS PUT CONTAINER(containerA)
   CHANNEL(channelA)
   FROM(dataA)
EXEC CICS LINK PROGRAM(program2)
   CHANNEL(channelA)
```

Routability. Channels and containers take advantage of CICS Multi-Region Operations for the purpose of availability, reliability, and scalability. You can define channel and container resources (ownership) as local or remote for optimizing resource run-time and for making administration more proficient.

**CHANNELS AND CONTAINERS VS. DFHCOMMAREA LARGE BUFFER INTERFACE**

The DFHCOMMAREA Large Buffer Interface was originally implemented by the since retired webMethods MIS component and supported now by EntireX. Using the standard CICS Socket Listener (CSKL), there was a CICS program started (WMTLSRV) to manage the TCP/IP session and handle the movement of data within CICS.

```
EXEC CICS GET CONTAINER(containerA)
   CHANNEL(channelA)
   INTO(dataA)
```

**FIGURE 4:** DFHCOMMAREA Large Buffer Interface

Based on the premise of allocating shared memory, this method uses the GETMAIN which is constrained by dynamic storage areas below the 2GB boundary (known as below-the-bar). The dynamic storage areas are essential for CICS operation and over utilizing these limited storage subpools results in significant performance degradation or short-on-storage conditions during which it curtails system activity until it can recover enough storage to resume normal operations.

Storage for channel container segments on the other hand, are performed above the bar in a 64-bit CICS dynamic storage subpool (known as GCDSA) thus dramatically extending the buffer size and flexibility for running concurrent transaction invocations from on-demand workload requests as shown in Figure 4.

**BOTTOM-LINE**

Using channels and containers to increase the flexibility of your end-to-end application and infrastructure investments is a wise approach. EntireX easily supports channels and containers through its CICS RPC solutions and Eclipse-based tooling. EntireX provides unique support for multiple containers from a single call. This offers customers multiple paths and flexibility to program-enabled data within CICS server programs.

Visit the Software AG Technology Community to learn more about how EntireX leverages channels and containers in my upcoming forum post at techcommunity.softwareag.com/webmethods/products/applmod.

---

**ABOUT THE AUTHOR**

Chris Pottinger is a member of the EntireX R&D team at Software AG Company. His mainframe experience spans more than 30 years and includes COBOL Application Developer for CICS, IMS/TM, and Batch. He is a pundit for supporting process diversity and data virtualization solutions in enterprise organizations.

To find the Software AG office nearest you, please visit: www.softwareag.com
© 2012 Software AG. All rights reserved. Software AG and all Software AG products are either trademarks or registered trademarks of Software AG. Other product and company names mentioned herein may be the trademarks of their respective owners.