Programming with CICS TS Channels and Containers

CICS Users Group meeting - November 12, 2013

Mary Abdill
mabdill@CreativeDataMovers.com
www.CreativeDataMovers.com

Channels and Containers

A newer and better technique for passing large amounts of data between CICS programs.

Channels and Containers

- Enhanced Inter-Program Data Transfer
- Introduced in CICS Transaction Server 3.1
- Sort of like "Big COMMAREAs"
  - Not technically accurate, but you get the idea
  - Lets programs exchange over 32K of data with LINK, XCTL, START, RETURN TRANSID
- Not always the answer

Channels and Containers – Abstract

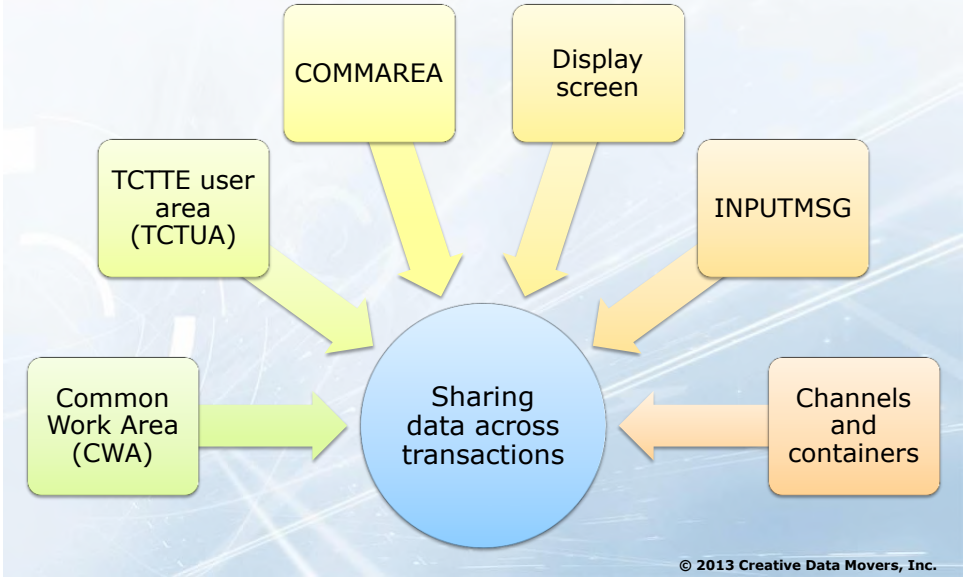
Newer and better technique for passing large amounts of data between programs

Generally better than using COMMAREAs

Require logic changes in the program code

Explore some best practices for channels and containers in CICS programs and CICS Web Services

Sharing Data Across Transactions-Review



CICS Needs to be More Flexible

- Many different languages and applications access and use CICS
  - Do *not* all use the same structure
  - Newer application design needs flexible data structures
- COMMAREA
  - Large, contiguous block of data containing all the date ever used, even if only part of this data is needed
- XML document as data
  - Different data format from that usually used in CICS
  - By design, XML is extensible, and can be BIG!
- CICS Web Services

Techniques to Pass Over 32K Data

Pass address of GETMAIN area
- For single region application

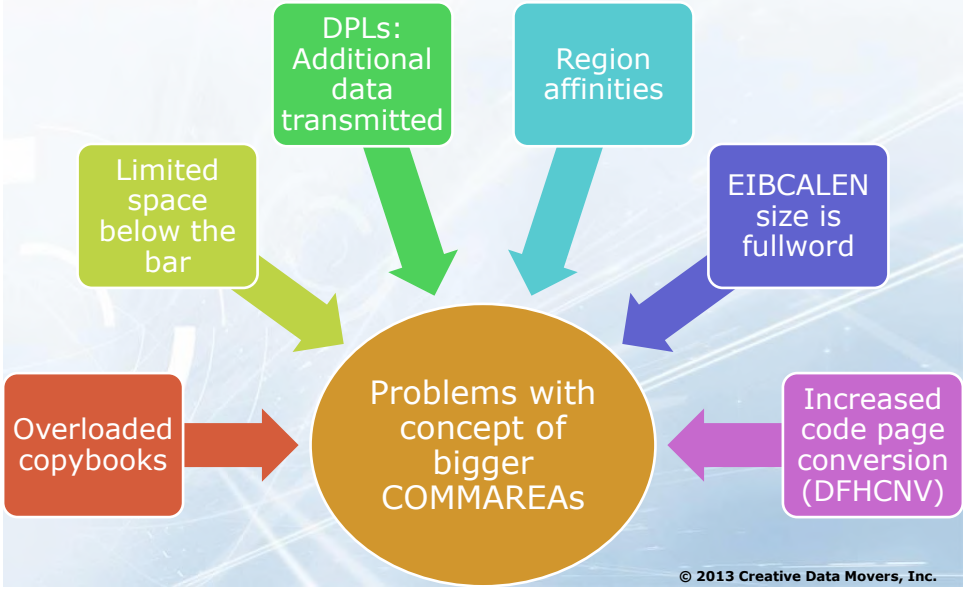
Use Temporary Storage Queue
- Pass queue name or use known queue name

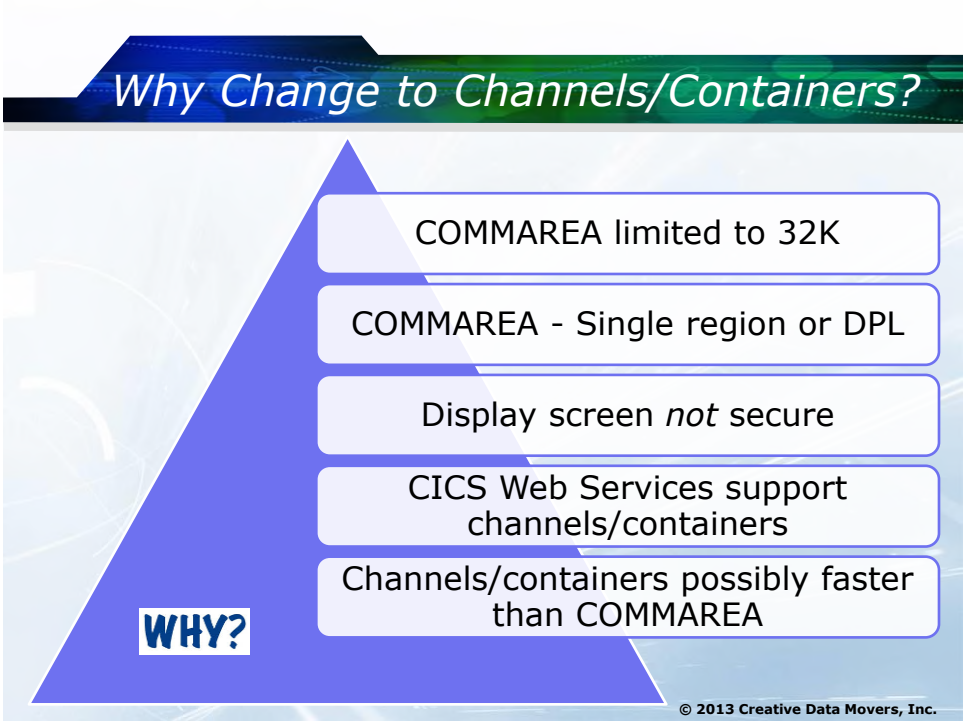
Pass as message in WebSphere MQ
- Or store in Database

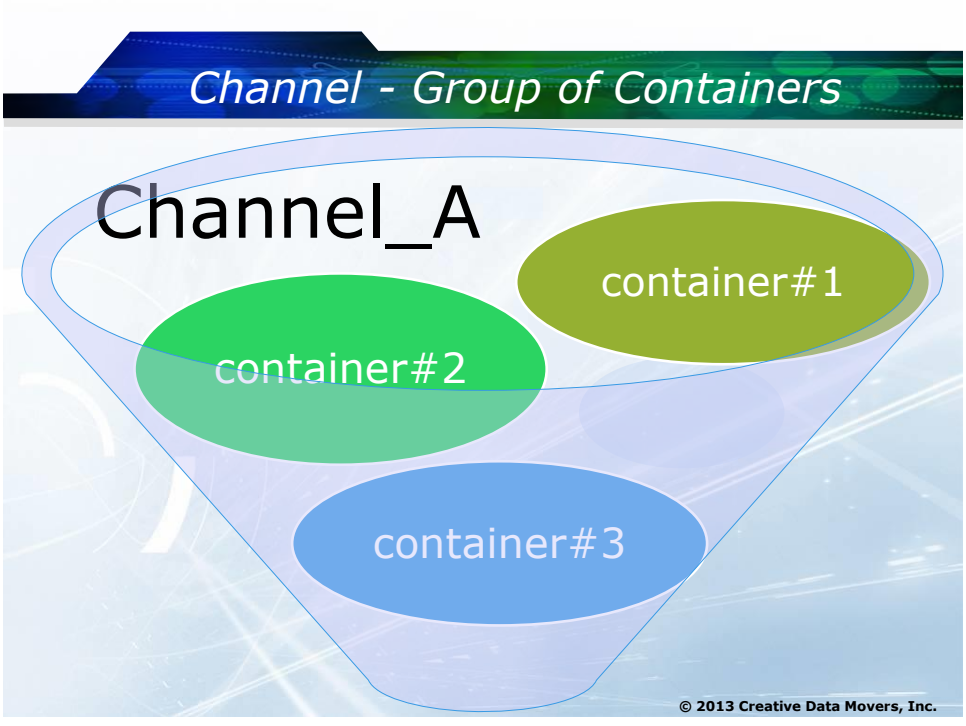
BTS Containers
- Anyone using BTS CICS?

Channels and containers
- Modern solution, and topic for today

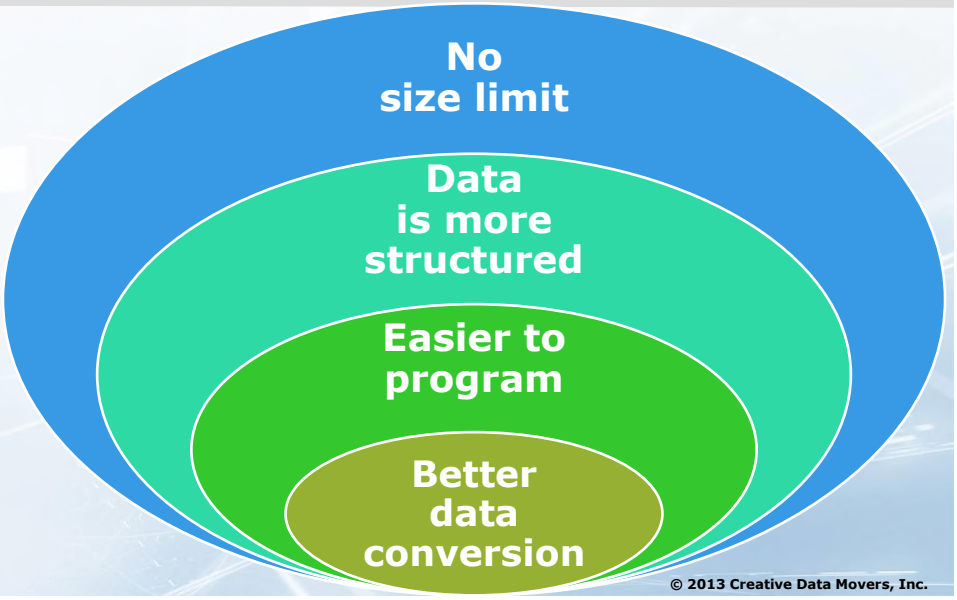
Why not just bigger COMMAREAs?




Why Change to Channels/Containers?
COMMAREA limited to 32K
COMMAREA - Single region or DPL
Display screen *not* secure
CICS Web Services support channels/containers
Channels/containers possibly faster than COMMAREA
WHY?
© 2013 Creative Data Movers, Inc.

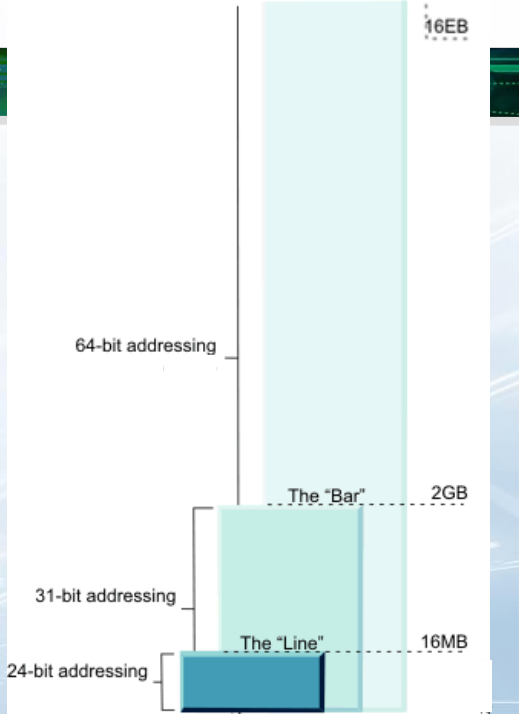

Channel - Group of Containers
Channel_A
container#1
container#2
container#3
© 2013 Creative Data Movers, Inc.

Benefits of Channels and Containers



AMODE/RMODE

- xMODE(64)
  - 16 exabytes
- ===BAR====
- xMODE(31)
  - 2 gigabytes
- ===LINE===
- xMODE(24)
  - 16 megabytes



Channel/container storage above "bar"

PUT CONTAINER

Copies data from working-storage/local-storage/ linkage section (below the bar) and directly writes it into a container above the bar

GET CONTAINER

Copies data from container, above the bar, into program data area, below the bar

There is no intermediate writing of the data

No intermediate storage area between your program's storage area and the 64-bit storage area CICS uses to hold the container data

Container/Channel Commands

Container	• PUT/PUT64 CONTAINER • GET/GET64 CONTAINER • MOVE CONTAINER • DELETE CONTAINER
Program transfer	• LINK PROGRAM • XCTL PROGRAM
Inquiry	• ASSIGN CHANNEL • STARTBROWSE CONTAINER • GETNEXT CONTAINER • ENDBROWSE CONTAINER
Transaction transfer	• RETURN TRANSID • START TRANSID

Channels and Containers – Basics

EXEC CICS PUT CONTAINER(*name*)
CHANNEL(*name*)

- Creates a container and assign to a channel

EXEC CICS LINK,XCTL,START,RETURN
with CHANNEL(*name*)

- Pass container/channel to second program

EXEC CICS GET CONTAINER(*name*)

- Second program reads container belonging to the channel that the program was invoked with.

Create Container and Channel; Pass It

- Create container and assign to a channel
  - Use PUT command

```
EXEC CICS PUT
          CONTAINER(container-name)
          CHANNEL(channel-name)
```

- Pass channel and its containers
  - Code **CHANNEL(channelName)** on link, xctl, start, or return

```
EXEC CICS LINK PROGRAM(PROG2)
          CHANNEL(channel-name)
```

Channel Example - Single Container

PROG1

```
EXEC CICS PUT
   CONTAINER(structName)
   CHANNEL(chanName)
   FROM(WS-structure)
END-EXEC

EXEC CICS LINK
   PROGRAM(PROG2)
   CHANNEL(channel)
END-EXEC

EXEC CICS GET
   CONTAINER(structName)
   CHANNEL(chanName)
   INTO(WS-structure)
END-EXEC
```

PROG2

```
EXEC CICS GET
   CONTAINER(structName)
   CHANNEL(chanName)
   INTO(WS-structure)
END-EXEC
***process***
EXEC CICS PUT
   CONTAINER(structName)
   FROM(WS-structure)
END-EXEC

EXEC CICS RETURN
END-EXEC
```

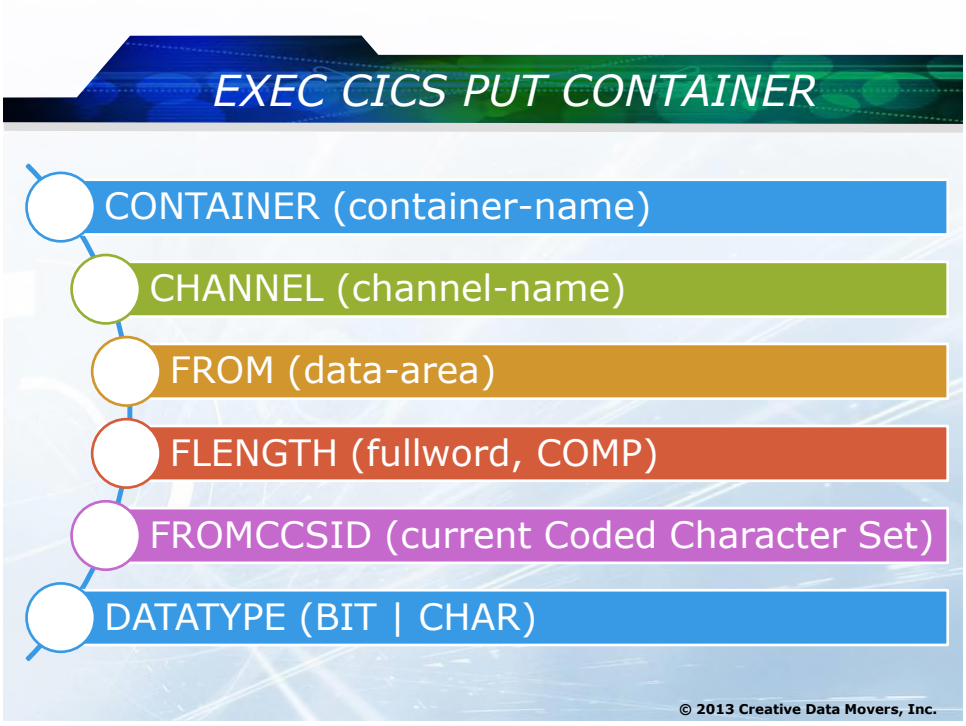

Naming Channels and Containers

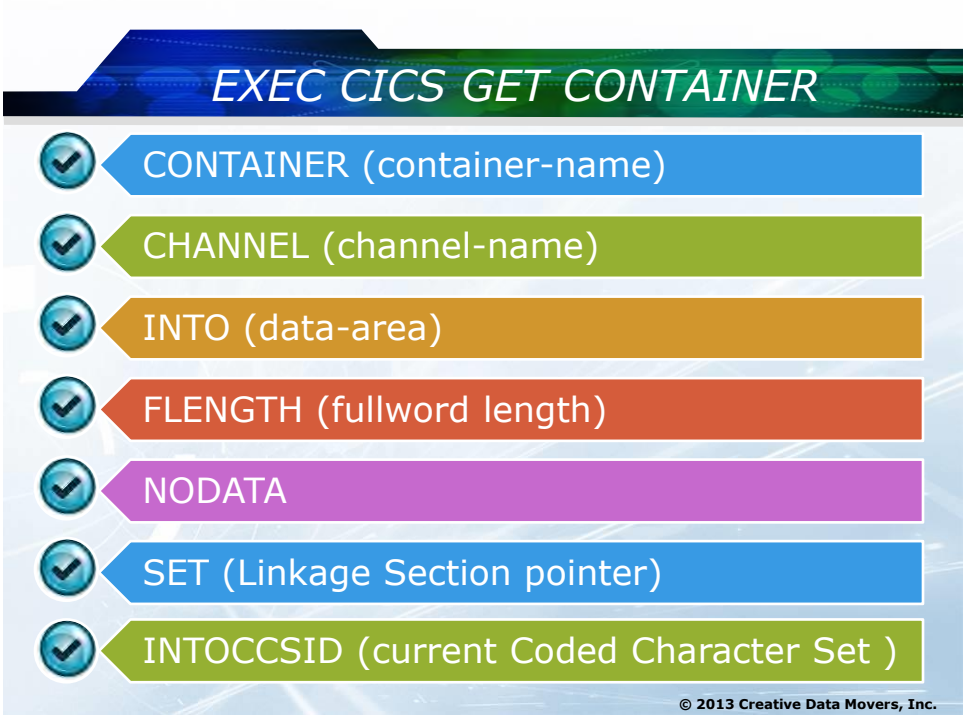
CHANNEL(*channel-name*)

- Name is 1–16 characters
- **A-Z a-z 0-9 & : = , ; < > . - _**
  | " $ @ # / % ? ! ***(avoid)***
- No leading or embedded spaces
- Names less than 16 characters are padded with trailing spaces
- Only the EBCDIC encoding of the characters
- If shipping channels between regions, just use:
  A-Z a-z 0-9 & : = , ; < > . - _

CONTAINER(*data-value*)

- Same rules as above
- Do *not* use container names beginning with 'DFH', unless requested to do so by CICS


EXEC CICS PUT CONTAINER
CONTAINER (container-name)
CHANNEL (channel-name)
FROM (data-area)
FLENGTH (fullword, COMP)
FROMCCSID (current Coded Character Set)
DATATYPE (BIT | CHAR)
© 2013 Creative Data Movers, Inc.


EXEC CICS GET CONTAINER
CONTAINER (container-name)
CHANNEL (channel-name)
INTO (data-area)
FLENGTH (fullword length)
NODATA
SET (Linkage Section pointer)
INTOCCSID (current Coded Character Set)
© 2013 Creative Data Movers, Inc.

START with Channel

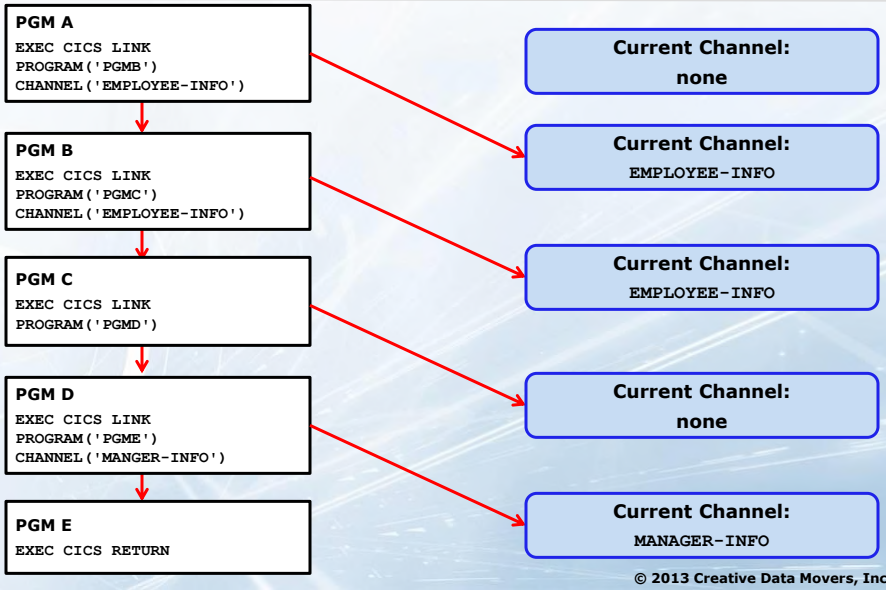
- EXEC CICS START CHANNEL
  - Start a task and pass it a channel
  - Only provides a channel to the started task
  - Uses temporary storage to provide data to new task
  - Started task uses RETRIEVE get the data
- RETURN
  - Can pass COMMAREA or CHANNEL
- LINK and XCTL
  - Transfer data to the second program with a COMMAREA or a CHANNEL

Current Channel

- Set by the calling program or transaction
- Current Channel is the channel coded on:
  - LINK, XCTL, START, or RETURN
- A program may create additional channels, but current channel does not change during the life of the program
- If a channel is *not* coded on an EXEC CICS, then the current channel is the default
  - On GET, PUT, etc.
  - *Except* for LINK, XCTL, START, or RETURN

cont.

Current Channel (cont.)



Channels and Containers – Basics (cont.)

Containers

- Named blocks of data for passing information between programs
- Programs can pass any number of containers between each other

Channels

- Containers grouped together in sets in a channel

Visibility / Life span

- Only to the program that created them and the programs they are passed to
- When these programs terminate, CICS automatically destroys containers and storage

cont.

Life of a Channel

- Created - by naming it on an EXEC CICS
  - **PUT CONTAINER CHANNEL**
  - **MOVE CONTAINER CHANNEL TOCHANNEL**
  - **LINK PROGRAM CHANNEL**
  - **XCTL PROGRAM CHANNEL**
  - **RETURN TRANSID CHANNEL**
  - **START TRANSID CHANNEL**

- Deleted - when it goes out of scope
  - When *no* application program is able to access the channel
- Temporary
  - When you create and use and immediately delete, with *no* intention of passing it

Assign Channel

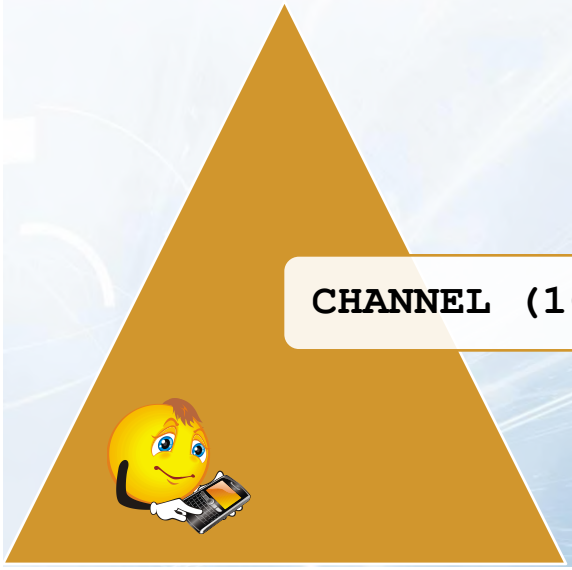
- Typically, both client and server programs know their channel name, and the names of all the containers in that channel

- However, a server program may handle multiple channels
- ASSIGN returns which channel was passed

```
EXEC CICS ASSIGN CHANNEL(channel-name)
```

- ASSIGN returns spaces if *no* current channel

EXEC CICS ASSIGN



Server Program - Multiple Channels

- PGMA

```
LINK PROGRAM('SERV1') CHANNEL('EMP')
```

- PGMB

```
LINK PROGRAM('SERV1') CHANNEL('DEPT')
```

- SERV1 - needs to determine current channel

```
ASSIGN CHANNEL
```

Sample Code for Migrating LINK or CALL

- PGMA with COMMAREA

```
link program('PGMB')
     commarea(WS-CA)
```

- PGMB with COMMAREA

```
address commarea(LS-CS)
*****process******
Return
```

- PGMA with channel

```
put container('contr')
     channel('chann1')
     from(WS-CA)
link progam('PGMB')
     channel('chann1')
get container('contr')
     into(WS-CA)
```

- PGMB with channel

```
get container('contr')
     into(WS-CS)
****process******
put container('contr')
     from(WS-CS)
return
```


Example - Multiple Containers

- PGMA

```
PUT CONTAINER('Invoice') CHANNEL('Bill')
                FROM(WS-INV)
PUT CONTAINER('Customer') CHANNEL('Bill')
                FROM(WS-CUST)
LINK PROGRAM('PGMB') CHANNEL('Bill')
GET CONTAINER('Receipt') CHANNEL('Bill')
                INTO(WS-RECEIPT)
```

- PGMB

```
GET CONTAINER('Invoice') INTO(WS-INV)
GET CONTAINER('Customer') INTO(WS-CUST)
**** PROCESS ****
PUT CONTAINER('Receipt') FROM(WS-RECEIPT)
RETURN
```

Data Conversion

- When is data conversion is necessary?
  - Pass data between EBCDIC and ASCII platforms
  - Change data encoding to a different CCSID
  - CHAR only; not BIT data
- Conversion models
  - Using COMMAREA
  - Using channels

```
PUT CHANNEL(temp) CONTAINER(temp)
    DATATYPE(CHAR) FROMCCSID(codepage1)
    FROM(input-data)
GET CHANNEL(temp) CONTAINER(temp)
    INTOCCSID(codepage2)
    SET(data-ptr) FLENGTH(data-len)
```

Make CICS Automatically Convert Data

- In client program
  - On PUT CONTAINER, code **DATATYPE(DFHVALUE(CHAR))**
  - Specifies that container holds character data and the data is eligible for conversion
  - FROMCCSID is *not* needed, unless data is *not* in the default CCSID of the client platform

- Put data in a container eligible for conversion

```
EXEC CICS PUT CONTAINER(cont_name)
     CHANNEL(channel_name) FROM(data1)
     DATATYPE(DFHVALUE(CHAR))
```

Best Practices

- Separate containers for input and output
- Separate containers for each structure and vice versa
- Separate Error messages container
- Do *not* create too many large containers
  - Limits storage available to other applications
- Copybook naming the channel and containers used and the data fields that map to the containers
  - Include copybook in both client and server program

cont.

Best Practices (cont.)

- Use separate containers for character data versus binary data
- Use separate containers for 'input-only' versus read/write data
- Server program creates the output containers
- If structure is optional, make it a separate container

Important References

- *CICS TS 4.2 Web Services Guide*, SC34-7191
- *CICS TS 4.1 Internet Guide*, SC34-7021
- *CICS TS for z/OS 4.2 Application Programming Reference*, SC34-7159
- *CICS TS for z/OS 4.2 Application Programming Guide*, SC34-7158
- *CICS TS 3.1 Channels and Containers Revealed Redbook*, SG24-7227
- *CICS TS 4.2 from Start to Finish Redbook*, SG24-7952

References and Resources

- CICS TS 4.2 - Information Center - all manuals
  `http://pic.dhe.ibm.com/infocenter/cicsts/v4r2/index.jsp`
- CICS TS 5.1 - Information Center - all manuals
  `http://pic.dhe.ibm.com/infocenter/cicsts/v5r1/index.jsp`
- Web Services @ IBM developerWorks
  `http://www.ibm.com/developerworks/webservices`
- CICS TS V3.1 Channels and Containers, Redbook
  - SG24-7227

Questions & Answers

Mary Abdill
mabdill@nyc.rr.com
www.CreativeDataMovers.com
Thanks to WebAge for recommending me!