



## CA Gen Asynchronous Communications

*The Best Kept Secret*

Rony Van Hove, Sr. Principal Consultant, EMEA Mainframe  
CA

Session #  
12 October

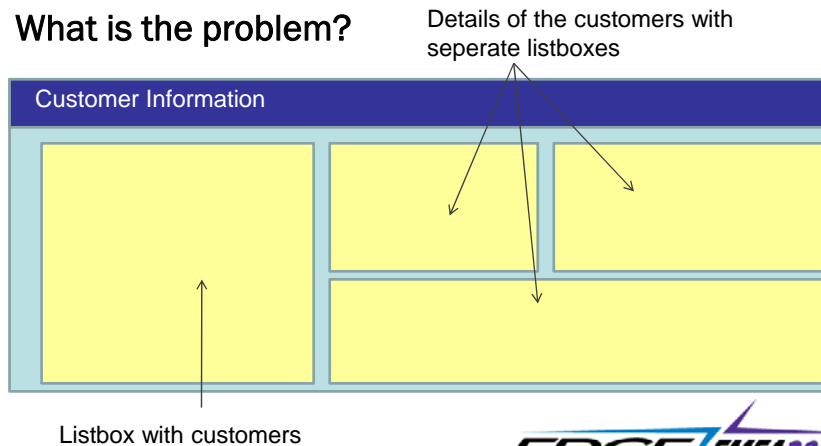
## Agenda

- Introduction
- Synchronous Server Calls
- How to resolve?
- Asynchronous Server Calls
- CA Gen Statements
- Asynchronous Calls without Events
- Asynchronous Calls with Events
- Conclusions
- More information?



## Introduction

### What is the problem?



**EDGE/EMEA 2009**  
October 11-13, 2009  
Amsterdam

## Traditional Approach

### Problem 1:

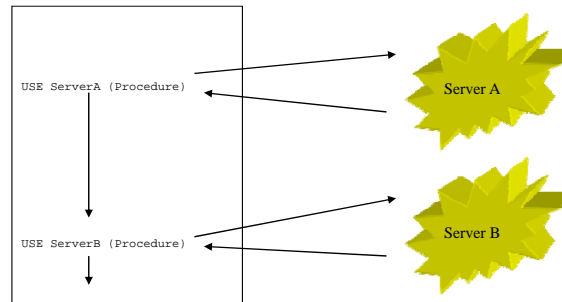
- Two or more server flows or USEs are initiated to retrieve the data.
- The user is waiting until all data is filled in and displayed.

### Problem 2:

- If the listbox is quite large (e.g. 1000 occurrences) and the READ EACH is complex and the server is rather busy then the End-User has to wait considerable amount of time.

**EDGE/EMEA 2009**  
October 11-13, 2009  
Amsterdam

## Synchronous Server Calls



Synchronous calls = “blocking” calls

Perceived “slowness” when many server calls are required

**EDGE/EMEA 2009**  
October 11-13, 2009  
Amsterdam

## How to resolve?

As from COOL:Gen 5.x, a concept of Asynchronous server procedure calls were introduced.

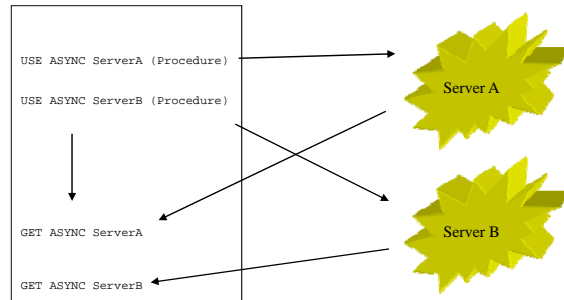
This server procedure call technique is widely underused and not exploited.

The asynchronous server calls can improve performance, end-user experience of the application and increase productivity.

CA Gen has specific extensions to the language to allow to use ASYNC calls.

**EDGE/EMEA 2009**  
October 11-13, 2009  
Amsterdam

## Asynchronous Calls



Asynchronous = non-'Blocking' calls

Increases the Client responsiveness when multiple server calls are required.

**EDGE EMEA 2009**  
October 11-13, 2009  
Amsterdam

7

## CA Gen statements

USE ASYNC	Call procedure step in an asynchronous method
GET ASYNC	Get the results from the asynchronous called procedure step
CHECK ASYNC	Check the status of the asynchronous called procedure step
IGNORE ASYNC	Ignore any response from the asynchronous called procedure step

**EDGE EMEA 2009**  
October 11-13, 2009  
Amsterdam

## USE ASYNC Statement

Syntax :

```
+--USE ASYNC target_server_pstep
|   IDENTIFIED BY: viewname async_request
|   RESPONSE HANDLING: POLL | NOTIFY EVENT | NO RESPONSE
|   RESPONSE EVENT: event_name
|   RESPONSE SCOPE: PSTEP | GLOBAL
|   WHICH IMPORTS...
+--WHEN request accepted
+--WHEN request not accepted
+--
```



## USE ASYNC Statement

*Response handling options:*

POLL	The response of the server will later be processed.
NOTIFY EVENT	If the response is available from the server then an notify event <i>&lt;event_name&gt;</i> will be triggered and executed.
NO RESPONSE	Ignore the response of the server ( <i>fire-and-forget</i> ).



## Synchronous call without event usage:

```

CLIENT PROCEDURE
IMPORTS:
EXPORTS:
LOCALS:
ENTITY ACTIONS: ...

  USE ASYNC search_server_procedure (procedure step)
  IDENTIFIED BY: loc_async_request
  RESPONSE HANDLING: POLL
  RESPONSE SCOPE: PSTEP
  WHICH IMPORTS: Entity View customer TO Entity View imp customer
  WHEN request accepted
  WHEN request not accepted
  EXIT STATE IS problem_to_call_server
  ESCAPE

  NOTE *****
  Logic continues immediately after the USE ASYNC
  It doesn't wait until server is finished.
  *****

  NOTE *****
  All logic processed, check now if there is a response.
  *****

  SET loc_ief_supplied flag TO "R"
  REPEAT
  SET loc_ief_supplied flag TO "E"
  GET ASYNC RESPONSE search_server_procedure
  IDENTIFIED BY: loc_async_request
  WHICH EXPORTS: Group View loc_rgv FROM Group View out_rgv
  WHEN successful
  SET loc_ief_supplied flag TO "A"
  WHEN pending
  SET loc_ief_supplied flag TO "R"
  WHEN invalid ASYNC_REQUEST ID
  WHEN server error
  WHEN communications error
  UNTIL loc_ief_supplied flag IS NOT EQUAL TO "R"
  IF loc_ief_supplied flag IS EQUAL TO "E"
  EXIT STATE IS problem_to_call_server
  ESCAPE

```

Ensure that the workset for  
IDENTIFIED BY is always the  
same as you have used for the  
USE ASYNC statement.

**EDGE EMEA 2009**  
October 11-13, 2009  
Amsterdam

## Asynchronous calls with Events:

```

NOTIFY EXAMPLE
IMPORTS:
EXPORTS:
LOCALS:
ENTITY ACTIONS: ...

  SET loc_async_request id TO 0
  USE ASYNC search_server_procedure (procedure step)
  IDENTIFIED BY: loc_async_request
  RESPONSE HANDLING: POLL
  RESPONSE SCOPE: PSTEP
  WHICH IMPORTS: Entity View customer TO Entity View imp customer
  WHEN request accepted
  WHEN request not accepted
  EXIT STATE IS problem_to_call_server
  ESCAPE

  EVENT ACTION data_found_event
  GET ASYNC RESPONSE search_server_procedure
  IDENTIFIED BY: loc_async_request
  WHICH EXPORTS: Group View loc_rgv FROM Group View out_rgv
  WHEN successful
  SET loc_async_request id TO 0
  NOTE *****
  Process response now
  *****

  WHEN pending
  WHEN invalid ASYNC_REQUEST ID
  EXIT STATE IS problem_to_call_server
  ESCAPE
  WHEN server error
  EXIT STATE IS problem_to_call_server
  ESCAPE
  WHEN communications error
  EXIT STATE IS problem_to_call_server
  ESCAPE

```

Event is required to  
handle the response  
The response can be  
retrieved with GET  
ASYNC RESPONSE.

**EDGE EMEA 2009**  
October 11-13, 2009  
Amsterdam

## Example – Fire and Forget principle

```

FIRE_AND_FORGET_EXAMPLE
IMPORTS:
EXPORTS:
LOCALS: ...
ENTITY ACTIONS: ...

SET loc async_request id TO 0
USE ASYNC search_server_procedure (procedure step)
  IDENTIFIED BY: loc async_request
  RESPONSE HANDLING: NO RESPONSE
  WHICH IMPORTS: Entity View customer TO Entity View imp customer
  WHEN request accepted
  WHEN request not accepted
  EXIT STATE IS problem_to_call_server
  ESCAPE
  
```

NO RESPONSE: "Forget" the response of the server



13

## Conclusions

### Asynchronous Calls

#### Benefits:

- Increases visibly the responsiveness of clients
  - User does not have to wait until all data is read.
  - Allows multiple server transactions to be executed at once from a client. Increases the utilization of the available server capacity (z/OS, UNIX,...)
  - For large RGVs, the user doesn't have to wait.
  - Easier to program then in C, C++ or Java.



## More information?

For more information:

- Read the *CA Gen Asynchronous Communication Guide*

