Progress Legos-
Scoping, Blocks, and other Critical Concepts

Michael Lonski

President - Allegro

AllegroConsultants.com

Who I Am…

- ❖Started working with Progress® v3 (1986)
- ❖Founded Allegro in 1993
- ❖Internationally recognized speaker
- ❖Author of “Coding Smart” book on ADM2
- ❖Punster and Frequent Talker
- ❖Ask me about Capoferro or Giganti

…And Why I Am Here

- A block by any other name
- What scope through yonder window
- Who started it, who controlled it
- Proving it with examples
- Are you sure?
- Nested buckets

What’s Next…

- A block by any other name

Blocks

- Definition: a sequence of one or more statements, including any nested blocks, that share a single context
- Typically closed with the “END.” statement
- Differing default properties depending on type of block
- Most blocks can be nested

Control Blocks

- FOR [EACH | FIRST | LAST]
- REPEAT
- DO
- EDITING

Procedure Blocks

- External Procedure
- TRIGGER
- PROCDURE (internal)
- FUNCTION (user defined)

Block Properties

- Some examples are:
  - Transaction
  - Record scoping
  - Iteration
  - Undo
- Some blocks control some properties implicitly
- Control blocks can be modified to explicitly control properties

Block Properties (Details)

	REPEAT		FOR		DO		Proc/Trigger	
Property	Imp	Exp	Imp	Exp	Imp	Exp	Imp	Exp
Record Reading	No	No	Yes	Record Phrase	No	No	No	No
Frame Scoping	Yes	WITH FRAME	Yes	WITH FRAME	No	WITH FRAME	Yes	No
Record Scoping	Yes	FOR	Yes	No	No	FOR	Yes	No
UNDO	Yes	TRANSACTION	Yes	TRANSACTION	No	TRANSACTION	Yes	No
Looping	Yes	WHILE TO/BY	Yes	WHILE TO/BY	No	WHILE TO/BY	No	No
Transaction	Yes	TRANSACTION	Yes	TRANSACTION	No	TRANSACTION	Yes	No

Some Examples

- REPEAT:
  - Implicitly scopes transactions, records and frames
  - Scoping is weak
- REPEAT FOR customer WITH FRAME x:
  - Explicitly scopes record and frame
  - Record scoping is strong
  - Transaction scoping still implicit (weak)

Working With Blocks

- Gives structure to execution flow
  - LEAVE
  - NEXT
  - RETURN
- Label blocks to remove doubt
  - Same rules as variable name
  - Immediately precedes block statement
  - Self-documentation

Impacting Transactions

- Explicitly manipulate transactions
  - UNDO, <action>
    - LEAVE
    - NEXT
    - RETRY
    - RETURN
    - THROW
- Use block labels to remove doubt

What’s Next…

- A block by any other name
- What scope through yonder window

Scope

- Definition: the duration that a resource is available to an application
- Prime component when discussing buffers and transactions
- Variables/Objects scoped where defined

Scope Types

- Record
- Transaction
- Frame
- Object

Seeing The Scope

- TRANSACTION function

```
MESSAGE "Trans Active?:" TRANSACTION.
```

- COMPILE ... LISTING

```
COMPILE demo1a.p LISTING demo1a.lst.
```

Listing Contents

```
File Name     Line Blk. Type Tran Blk. Label
------------- ---- --------- ---- ----------
.\demo1a.p    0    Procedure No
    Buffers: sports.Customer
    Frames:  Unnamed

.\demo1a.p    19   Repeat    Yes  order-rpt
    Buffers: sports.Order
    Frames:  a-frm
```

What’s Next…

- A block by any other name
- What scope through yonder window
- Who started it, who controlled it

Cause vs. Control

- Specific statements cause record or transaction scope
- Blocks control (or not) the scope of items started within them
- Scope can be explicitly increased and sometimes decreased programmatically
- Can both cause and control

Record Scope

- Smallest enclosing block that encompasses all references to the record
- Determines when…
  - The record buffer is active/available
  - Progress clears the record from the buffer
  - Writes the record to the database
  - How long a record lock is in effect
  - When to validate the record

Types of Record References

- *Strong-scoped reference* - buffers *explicitly* referenced in REPEAT FOR or DO FOR
- *Weak-scoped reference* - buffers *implicitly* referenced in FOR EACH or PRESELECT EACH block
- *Free reference* - All other references to records are free references

Record Reference Causes

- FIND
- FOR EACH
- REPEAT FOR
- INSERT/CREATE
- DEFINE QUERY

Transaction Scope

- Set of changes to the database that is either executed completely or leaves no modification to the database
- Transaction scope does ***not*** equal Record scope

Transaction Causes

- ❖ EXCLUSIVE-LOCK
- ❖ CREATE/INSERT
- ❖ DELETE
- ❖ ASSIGN
- ❖ UPDATE
- ❖ TRANSACTION keyword

Transaction GT Record

- Transaction started
- Reference to record is invalid

- If before record retrieval, record is NOT AVAILABLE
- Record has EXCLUSIVE-LOCK
- Record available until end of record scope

- Record returns to SHARE-LOCK
- Reference to record is invalid but lock *IS* held
- Transaction prevents full release of record

Record GT Transaction

- Record is NOT AVAILABLE until reference.
- Record is retrieved with SHARE-LOCK and now AVAILABLE

  - Transaction started
  - SHARE-LOCK upgraded to EXCLUSIVE-LOCK
  - Record updates occur
  - Nothing written out to database until end of transaction

- Record returns to SHARE-LOCK
- Transaction effectively bleeds out to record scope

Record EQ Transaction

What’s Next…

- A block by any other name
- What scope through yonder window
- Who started it, who controlled it
- Proving it with examples

What’s Next…

- A block by any other name
- What scope through yonder window
- Who started it, who controlled it
- Proving it with examples
- Are you sure?

The Prime Directives

- All scopes are defined in terms of their containing blocks
- Transaction scope and record scope are not automatically identical but do affect each other.
- Beware of bleeding locks

RELEASE Statement

- Probably one of the most misunderstood and misused Progress 4GL statements
- Too often used as a band-aid to “fix” scoping issues

Actual Effects

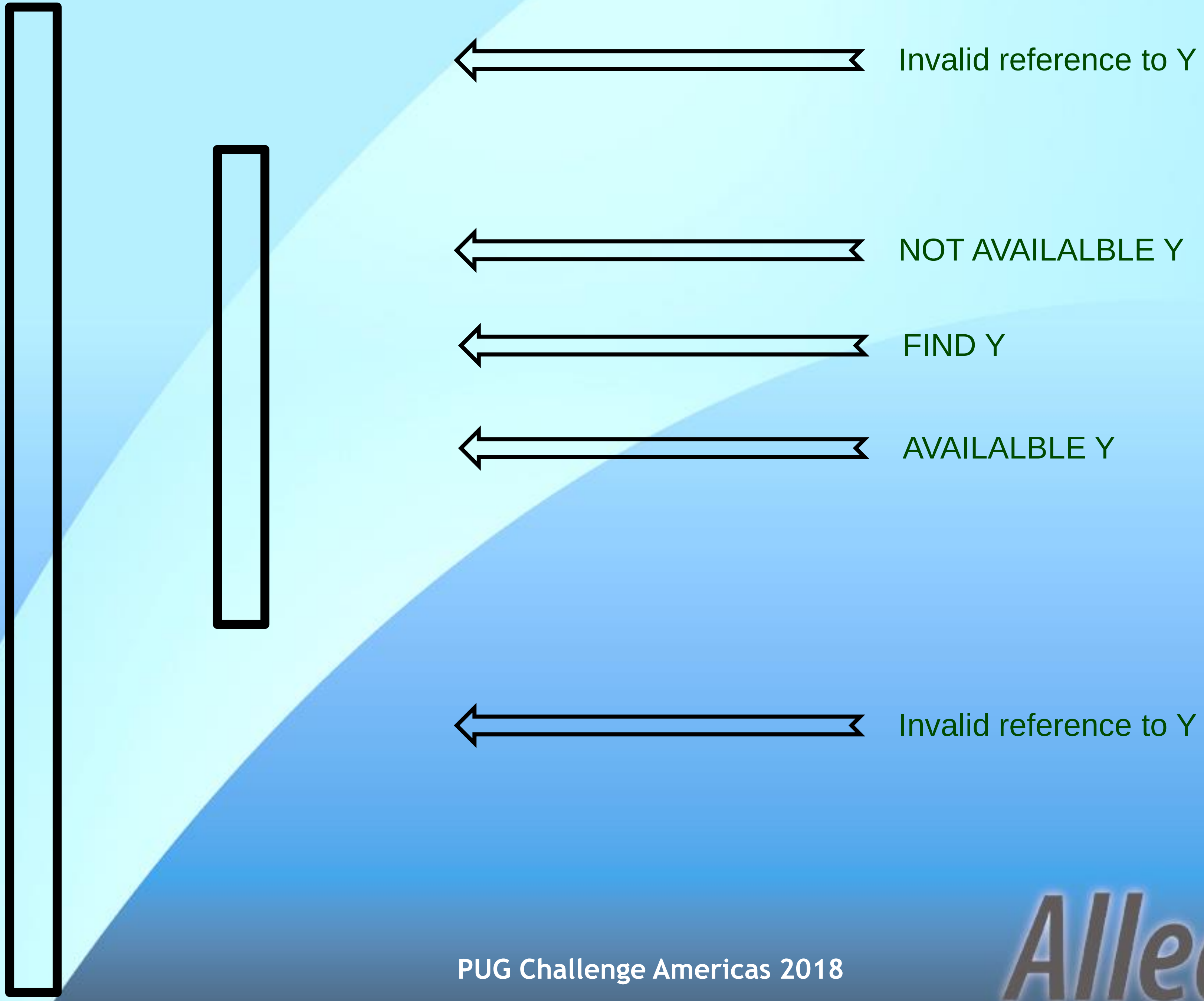
- Validates mandatory fields
- Validates unique index constraints
- Clears the record from the buffer
- If changed, writes back to the database

Misunderstandings

- Not a tool for releasing record locks
- Does *not* magically fix transaction or scoping issues
- Will not release a SHARE-LOCK if still inside of a transaction

What’s Next…

- A block by any other name
- What scope through yonder window
- Who started it, who controlled it
- Proving it with examples
- Are you sure?
- Nested buckets



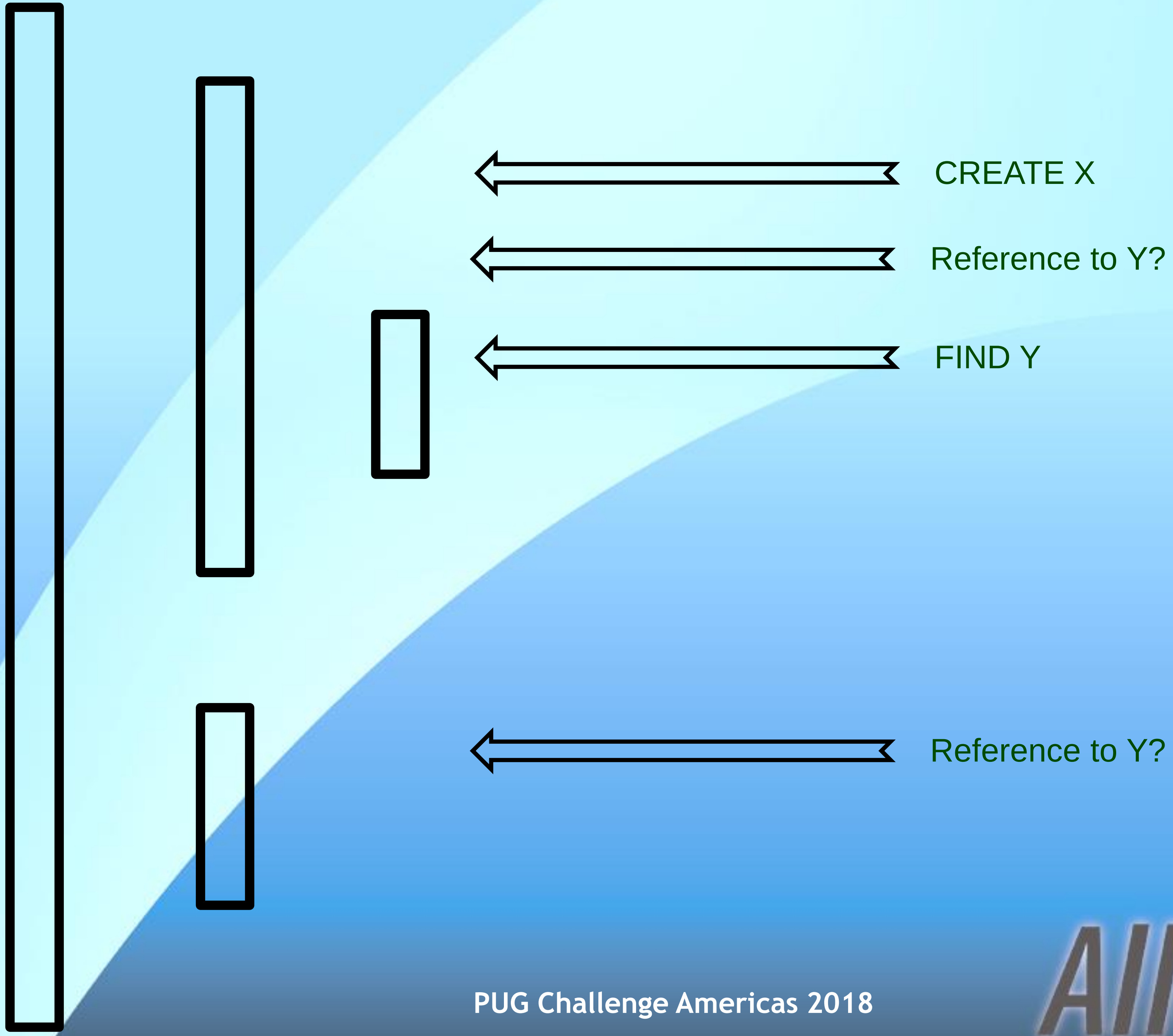
Invalid reference to Y

NOT AVAILALBLE Y

FIND Y

AVAILALBLE Y

Invalid reference to Y



CREATE X

Reference to Y?

FIND Y

Reference to Y?

Now that I've rambled on, are there any questions?

Final Notes

- ❖ Examples bundled with the presentation.
  - AllegroConsultants.com/about/downloads

- ❖ Thanks for attending!

- ❖ Please fill out your evaluations!