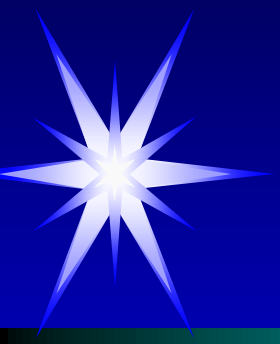
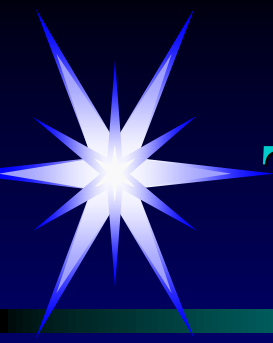


Transaction Support

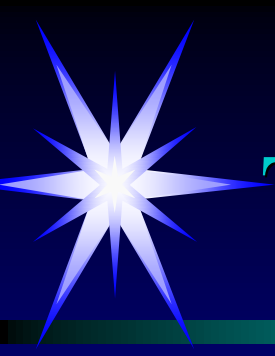
Bill Keech

Sunbelt Computer Systems



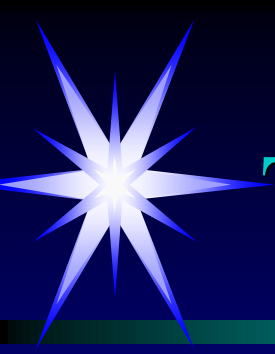
Transactions OVERVIEW

- Features
- Limitations
- TRANSACTION Statement
- Locking
- Considerations
- Questions



Transaction Features

- Single new statement
  - TRANSACTION
- Supports both local files and multiple file managers in one transaction
- Supports both pessimistic and optimistic locking

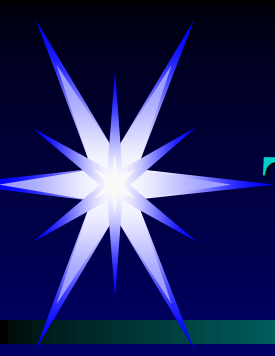


Transaction Features

- Supports optional pessimistic transaction verification
- Supports common memory pool on file manager and application server
- Supports record locking

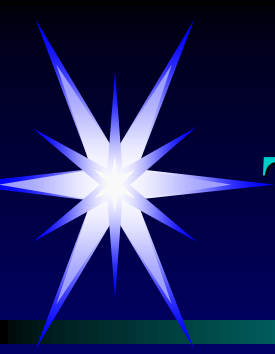
Transaction Features

- Compatible with FILEPI
- New OPEN/PREP mode
  - CMP_NO_OPT_TRAN
  - Prevent optimistic inclusion in transaction



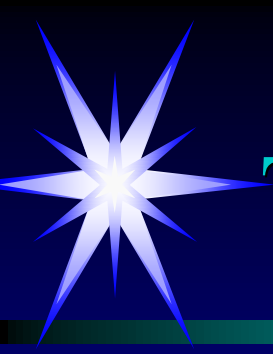
Transaction Features

- Supports ACID acronym
  - (Atomicity Consistency Isolation Durability):
  - Atomicity
    - A transaction represents an atomic unit of work.
    - Either all modifications within a transaction are performed, or none of the modifications are performed.



Transaction Features

- Consistency
  - When committed, a transaction must preserve the integrity of the data within the system. If one part of the transaction fails, all of the pending changes are rolled back, leaving the database in its' original state.



Tranaction Features

- Isolation
  - Modifications made by a transaction are isolated from the modifications made by other transactions.
- Durability
  - After a transaction has committed, all modifications are permanently in place in the system.

Transaction Limitations

- Only 250 concurrent save points allowed
- Statements not allowed under any transaction:

OPEN	CLOSE
PREP	WEOF
COPYFILE	ERASE
RENAME	PAUSE
AAMDEX	INDEX
SORT	PAGESETUP
EVENTWAIT	ALERT

Transaction Limitations

- Statements not allowed under any TRANSACTION:

SNDOPEN	GETFNAME
EXECUTE	ROLLOUT
PRTPAGE	PRTOPEN
PRTPLAY	EXTCALL
SPLOPEN	PRINT
SPLCLOSE	RELEASE

Transaction Limitations

- Statements not allowed under a pessimistic transaction:

FILEPI	DISPLAY
KEYIN	EVENTCHECK
ACTIVATE	SETPROP
SHUTDOWN	

Object methods

Rollback performed on CHAIN

PL/B TRANSACTION

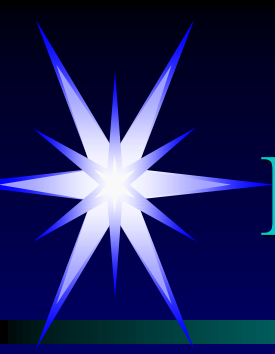
- TRANSACTION {action}[,{options}]
- action
  - One of the required actions from the table below.
- Options
  - A list of options based upon the action parameter.

PL/B TRANSACTION

- The {action} operand defines the action to be taken as follows:

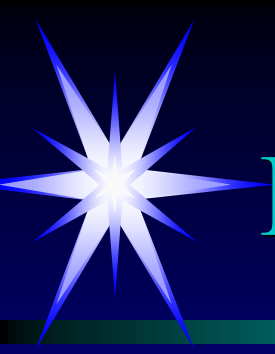
- START
  - Starts a transaction

- COMMIT
  - Ends the transaction and commits the data



PL/B TRANSACTION

- ROLLBACK
  - Ends the transaction without committing the data
- INFO
  - Provides information on the state of the transaction



PL/B TRANSACTION

- SAVE
  - Create a new save point
- RESTORE
  - Restores back to a save point

PL/B TRANSACTION

- START {action}
- {options} parameter can be;
  - FILE, IFILE, AFILE, and FILELIST
- Each file locked for transaction duration
- This is known as pessimistic locked files and a pessimistic transaction.

PL/B TRANSACTION

- COMMIT {action}
- {options} parameter can be:
- VERIFY keyword
  - If specified, all files specified on the TRANSACTION START statement, have integrity verification performed before the commit proceeds.

PL/B TRANSACTION

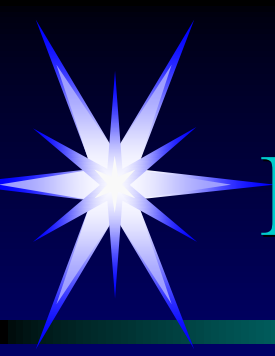
- SAVE {action}
- {options} parameter can be:
- LEVEL= nvar
  - The new current transaction level will be placed in the specified variable.

PL/B TRANSACTION

- RESTORE {action}
- {options} parameter can be:
- LEVEL= dnumnvar
  - If not specified, the RESTORE {action} will restore to the previous level. Otherwise the RESTORE {action}will restore to the specified level.

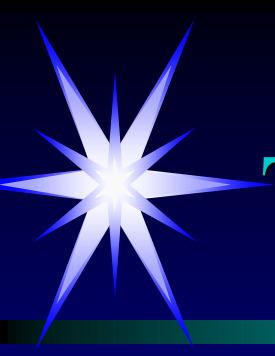
PL/B TRANSACTION

- INFO {action}
- {options} parameter can be:
- LEVEL= nvar
  - The current transaction level will be placed in the specified variable. A level of 0 means no transaction is taking place.



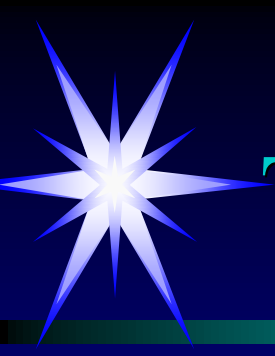
PL/B TRANSACTION

- ROLLBACK {action}
  - Has no {options} parameter.



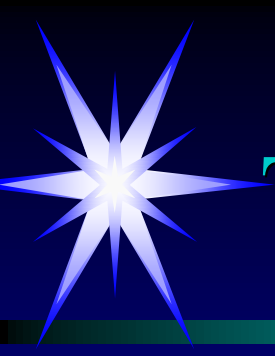
Transaction Locking

- Two types:
  - Pessimistic
  - Optimistic
- A transaction can contain both types



Transaction Locking

- Pessimistic features:
  - Like FILEPI, so no record locking
  - Files specified on TRANSACTION START statement
  - Verified at commit if VERIFY option used



Transaction Locking

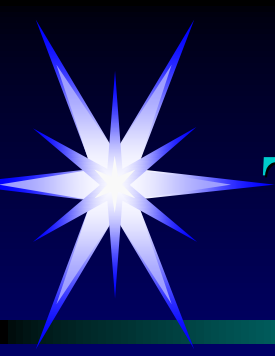
- Optimistic Features
  - Like record locking or implicit FILEPI
  - Files are added to transaction when used
  - Verified on every read/write operation after the first write operation

Transaction Locking

- Supports record locking
- Best option for files that are only read

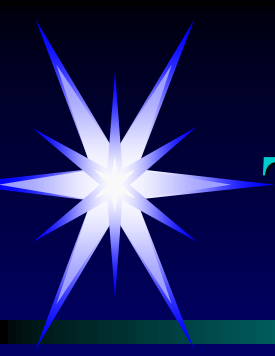
Transaction Considerations

- U48 error given for programmatic error
  - Sub-codes of U48 are from 200 to 299
- I27 error given for transaction error
  - Sub-codes of I27 are from 100 to 199



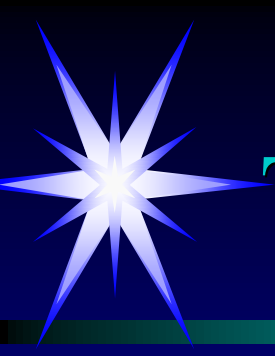
Transaction Considerations

- I27 can occur during any file I/O statement under a transaction
- Files only read are ignored by transactions
- Writes are optimized on commits



Transaction Considerations

- COMMIT is in two phases:
- Phase 1
  - Lock all optimistic files and indexes
  - Lock all records area in record locking files
  - Verify all write data
  - Update files



Transaction Considerations

- Phase 2
  - If error during Phase 1 then back out changes
  - Unlock all files

QUESTIONS?