Progress®

Database Diagnostic Data Collection

Richard Banville
Fellow, OpenEdge Development

November 17, 2017

Database Diagnostic Data Collection

- Capture database diagnostic data at time of “incident”
  - DBA generally cannot capture all data at the time of certain incidents
    - Lock table overflow example
  - Capture automatically (via an incident/event) or on-demand
- Data to capture
  - Incident applicable and configurable
  - Centralized and easily consumed
  - Disparate data correlated to triggering incident
- Alerting capability
- Fully configurable at startup and at runtime

Taking Action

- Triggering event (incident)
  - DBA selectable server side events
- Triggering events implemented
  - BI threshold exceeded
  - Lock table overflow
  - System Error
- Triggering options
  - Trigger data collection and actions anytime the event occurs
  - Trigger data collection and actions only if event is fatal to the database

Triggering Events & Event Level Startup Params

- -diagEvent
  - <Event>:<Level> – Comma separated list, no embedded spaces
  - LockTable:<#>,BiThold:<#>,SysErr:<#>
  - Uses –diagEvtLevel by default
- -diagEvtLevel (default: 0)
  - Default level for all events
  - Valid only at database startup
- Example:

proserve <*db*> -diagEvtLevel 1 –diagEvent LockTable:2,SysErr:0

  - Result
    - LockTable:2
    - BiThold:1
    - SysErr: Not Enabled **NOTE**: SysErr:0 overrides –diagEvtLevel 1

Event Levels - What are all these event levels?

- Each Triggering Event has its own data collection / action level
  - Uses a bitmap technique for maximum flexibility

0: Disabled

1: Summary

2: Summary & Detailed

3: Summary, Detailed & Protraces

4-7: Proc invocation in combination w/chosen option 0 thru 3

8-15: Report all enabled event collectors in combination w/chosen option 0 thru 4

Event Level	Action Taken
LockTable:3	Lock table summary, detailed & protraces
BiThold:1	BiThold summary data only
SysError:9	SysError summary data, BiThold actions and LockTable actions

Event Level Summary – Let's break it down

0-3: Basic diagnostic data collection levels

Level	Data
0	None
1	Summary
2	Summary & Detail
3	2 + protraces

Event Level Summary – Let's break it down

0-3: Basic diagnostic data collection levels

Level	Data
0	None
1	Summary
2	Summary & Detail
3	2 + protraces

- Summary and detailed data reported varies by event

Name	Triggering Event	Summary	Detailed
LockTable	Lock Table Overflow	_UserLock	_Lock, _Trans
BiThold	Bi Threshold Reached	_Logging	_ActBILog, _ActAILog, _ActIOFile, _Trans
SysErr	System Error	Same as detailed	User info via _UserLock

Event Level Summary – Let's break it down

0-3: Basic diagnostic data collection levels

Level	Data
0	None
1	Summary
2	Summary & Detail
3	2 + protraces

Event Level Summary – Let's break it down

4-7: Program execution in combination with chosen option 0 thru 3

Level	Data	Level	Data / Action
0	None	4	0 + proc exec
1	Summary	5	1 + proc exec
2	Summary & Detail	6	2 + proc exec
3	2 + protraces	7	3 + proc exec

Event Level Summary – Let's break it down

Level	Data / Action
0	None
1	Summary
2	Summary & Detail
3	2 + protraces
4	0 + proc exec
5	1 + proc exec
6	2 + proc exec
7	3 + proc exec

Event Level Summary – Let's break it down

8-15: Report diagnostic data requested for other events as well

Level	Data / Action	Level	Data/Action
0	None	8	0 + report other events' data too
1	Summary	9	1 + report other events' data too
2	Summary & Detail	10	2 + report other events' data too
3	2 + protraces	11	3 + report other events' data too
4	0 + proc exec	12	4 + report other events' data too
5	1 + proc exec	13	5 + report other events' data too
6	2 + proc exec	14	6 + report other events' data too
7	3 + proc exec	15	7 + report other events' data too

- Negative values (-1 thru -15) only perform action if fatal to database

Report On Fatal Database Errors Only

- Triggering events may be fatal to the database
  - Example: SIGBUS with locked buffer or latch OR
    Lock table overflow w/no more shared memory
    Either will force shutdown the database

Out of free shared memory. Use -Mxs to increase

SYSTEM ERROR: Releasing regular latch. latchId: 17

User 5 died holding 2 shared memory locks.

** Save file named core for analysis by Progress Software Corporation.

Begin ABNORMAL shutdown code 2

Report On Fatal Database Errors Only

- Triggering events may be fatal to the database
  - Example: SIGBUS w/locked buffer or latch OR lock table overflow w/no more shared memory - Either will force shutdown the database
- Negative event levels introduced
  - Data collection only triggered if event is fatal to the database

proserve <*db*> –diagEvent LockTable:-3,SysErr:11

  - Result: Trigger level 3 Lock Table diagnostics ONLY if fatal to the database
    Collect level 11 System Error diagnostics regardless if fatal to database
  - NOTE: A positive event level reports diagnostics for fatal and non-fatal events

Data Collection

- Flexible output location & naming
  - -diagDir (default: current db directory)
    - Directory name
    - Relative or absolute
    - Must already exists
  - -diagPrefix (default: diagEvent_)
    - Output filename prefix (16 byte maximum)

Data Collection

- Data collected on triggering event
- Data collected on demand
- Entry and table based output
  - BI Logging summary: One record summarizing BI configuration and activity
  - Lock Table summary: One record per user summarizing lock activity
  - Lock Table detail: Many records describing each entry in the lock table
  - Caution: Table based output can be very large
    - -L 1 000 000: up to 1 million entries reported in data file

Data Collection

- Example naming convention:
  - Diagnostic event “Tracking” file (One file per –diagDir directory – file can be shared amongst DBs)

    diagEvent_Tracking.csv

    Prefix File-type Suffix

  - Diagnostic event directory (One directory per event occurrence)

    diagEvent_2017-05-03T10:37:38.000-4:00_LockTable_1

    Prefix* Timestamp Event & occurrence

  - Diagnostic event data file

    diagEvent_locktable_detail.csv

    Prefix Event Level Suffix

Data

- Formats
  - -diagFormat CSV (default) Column separated values
    - Not comma separated value
  - -diagFS (default: “ “) CSV column/field separator

proserve <*db*> –diagFormat CSV –diagFS “tab”

    - Easily imports to excel
    - Easily loads into a database
  - -diagFormat JSON
    - Compressed format to save space
    - External tools can create “pretty” or “user readable” format

CSV Output

- Event Tracking (of course in quoted csv format)

Timestamp	PID	TID	Level	ProcType	Event	Dbname	EventDir	EventId	Status
2017…	P-29288	T1	3	SELF	LockTable	XYZ	…_LockTable_1	1	Start
2017…	P-29288	T1	3	SELF	LockTable	XYZ	…_LockTable_1	1	End
2017…	P-29288	T1	-11	SELF	SysErr	ABC	…_SysErr_2	1	Start
2017…	P-29288	T1	-11	SELF	SysErr	ABC	…_SysErr_2	1	End

- Event Data: (Lock table detail example) – One file per data collection type
  - diagEvent_locktable_detail.csv diagEvent_locktable_summary.csv diagEvent_transaction_detail.csv

Tracking ref	LockType	Rowid	HashChain	UserInfo	Flags	TransState	TransFlags	TransId
2017…	REC	32	5	#, name, tty	“X L”	ACTIVE	FWD	12
2017…	REC	64	6	#, name, tty	“X L”	ACTIVE	FWD	12
2017…	REC	72	7	#, name, tty	X L”	ACTIVE	WFD	12

JSON Output

- Event Tracking – one object per line

{"AnEvent_1_Start":{"Timestamp":"2017-05-15T12:07:34.000-4:00","PID":“P-28118","TID":"T1","Level":11,"ProcType":"SELF",
"Event":"LockTable","Dbname":"/usr1/richb/11/x","EventDir":"/db/diagEvent_2017-05-15T12:07:34.000-4:00_LockTable_1",
"EventId":1,"Status":"Start"}}
{"AnEvent_1_End":{"Timestamp":"2017-05-15T12:07:34.000-4:00","PID":"P-28118","TID":"T1","Level":11,"ProcType":"SELF",
"Event":"LockTable","Dbname":"/usr1/richb/11/x","EventDir":"/db/diagEvent_2017-05-15T12:07:34.000-4:00_LockTable_1",
"EventId":1,"Status":"End"}}

- Making it humanly readable

```
cat diagEvent_Tracking.json  | while read myLine
 do
    echo $myLine | python  -m json.tool
 done
```

JSON Output

- Diagnostic Data – One file per event (incident)


```
{
  "AnEvent_1": {
      "Dbname": "/usr1/richb/11/x",
      "Event": "LockTable",
      "EventDir": "/db/diagEvent_<TS>_LockTable_1",
      "EventId": 1,
      "Level": 11,
      "PID": "P-28118",
      "ProcType": "SELF",
      "Status": "N/A",
      "TID": "T-1",
      "Timestamp": "2017-05-15T12:07:34.000-4:00"
  },
  "file_detail": {
      "file_entry_1": {
          "Blksize": 8192,
          "BufReads": 7,
          "BufWrites": 1,
          "Extend": 512,
          "Extends": 0,
          "FileName": "/db/x.db",
          "IOMode": "BOTHIO",
          "InUse": 5,
          "Reads": 5,
          "Size": 640,
          "UnbufReads": 0,
          "UnbufWrites": 0,
          "Writes": 0
      },
  …
  "locktable_detail": {
      "locktable_entry_1": {
          "DomainID": 36,
          "HashChain": 0,
          "LockType": "/dev/pts/5",
          "Partition#": 0,
          "Rowid": 69184,
          "Table#": 2,
          "Tenant": "FWD",
          "UserNum": "ACTIVE"
      },
      "locktable_entry_10": {
          "DomainID": 36,
          "HashChain": 1,
          "LockType": "/dev/pts/5",
          "Partition#": 0,
          "Rowid": 69952,
          "Table#": 2,
          "Tenant": "FWD",
          "UserNum": "ACTIVE"
      },
  …
```

Taking Action

- Request protrace / prostack (Unix deployments only)
  - From all locally connected users
  - Protrace location reported in database .lg file

```
Protrace location: /usr1/richb/workdir/protrace.29288
```

```
Generating: /usr2/mikej/workdir/protrace.29290
```

  - Location reported once per connection
- Alerting capability / callout “hooks”
  - Program invocation
  - Trigger “Start” and “End” written to Event Tracking file
  - Ability to “Pause” between start and end
    - Useful for program invocation

Call Outs

- Program executable invocation
  - Executed with Event Levels 4-7, 12-15
  - File **<db-dir>/diagProc**
    - Parameters: {Event Directory, Event Name, database name}
  - Parent does NOT wait for diagProc to finish
    - Spawns executable script/program “diagProc” then pauses for –diagPause seconds
  - diagProc could create “pause end” file:diagCompleted when finished executing
- Security concerns
  - Program name and location hardcoded
  - The program is executed with the effective permissions of the caller
    - Non-servers downgrade setuid after initial connection
    - Server and non-servers retain setgid

Pausing Event Processing

- Pause time
  - Sleep for max of –diagPause seconds
  - -diagPause (0)
  - Up to 32 minutes (0 – 1920)
- Resources remain held during the pause
  - Allows a more consistent view
  - May affect OLTP
- Pause completion
  - -diagPause time exhausted
  - OR “diagCompleted” file created in the event output directory
    - Can be created by diagProc executable or manual procedure to stop the pause
  - OR spawned diagProc completed

Parameter Summary

For each _DbParams where _DbParams-Name = BEGINS “-diag”:

display _DbParams.

Name	Description	Default
-diagDir	Diagnostic directory	DB Directory
-diagEvent	Event level per event	-diagEvtLevel setting
-diagEvtLevel	Event level default	0 (disabled)
-diagFS	Field separator	“ “ (space)
-diagFormat	Data collection format	csv
-diagPause	Pause length	0 (none)
-diagPrefix	File prefix value	diagEvent_

Promon Configuration

- Promon R&D

  4. Admin Functions ...

  14. Diag Data Collection

```
04/12/17      OpenEdge Release 12 Monitor (R&D)
14:16:56      Diagnostic Data Collection

        Current Diagnostic Data Collection Settings:

        1. Diagnostic directory:         /usr1/richb/dbdiag
        2. Diagnostic field separator: ' '
        3. Diagnostic pause time:        5
        4. Diagnostic prefix value:      diagEvent_
        5. Diagnostic report format:     json

        6. Lock Table Overflow:  3: Summary, detailed & protrace data
        7. BI Threshold:         7: Summary, detailed & protrace data w/proc invocation
        8. System Error:        15: Summary, detailed & protrace data w/proc invocation
                                     and all other enabled collectors

        9. Collect enabled diagnostic data now

        Enter a number, P, T, or X (? for help):
```

Promon Configuration – All changeable online!

- *Data collection*
  - *On demand*
  - *Triggering event*
  - *Multiple levels*
  - *Tracking info and data detail*
  - *Formatting choices*
- *Multiple Actions*
  - *Protrace / prostack*
  - *Proc invocation*
  - *Pause time*
    - *Can change*
    - *Pause ignored on demand*

```
04/12/17      OpenEdge Release 12 Monitor (R&D)
14:16:56      Diagnostic Data Collection

        Current Diagnostic Data Collection Settings:

        1. Diagnostic directory:         /usr1/richb/dbdiag
        2. Diagnostic field separator: ' '
        3. Diagnostic pause time:        5
        4. Diagnostic prefix value:      diagEvent_
        5. Diagnostic report format:     json

        6. Lock Table Overflow:  3: Summary, detailed & protrace data
        7. BI Threshold:         7: Summary, detailed & protrace data w/proc invocation
        8. System Error:        15: Summary, detailed & protrace data w/proc invocation
                                     and all other enabled collectors

        9. Collect enabled diagnostic data now

        Enter a number, P, T, or X (? for help):
```

VST Configuration – All changeable online!

- Configuration

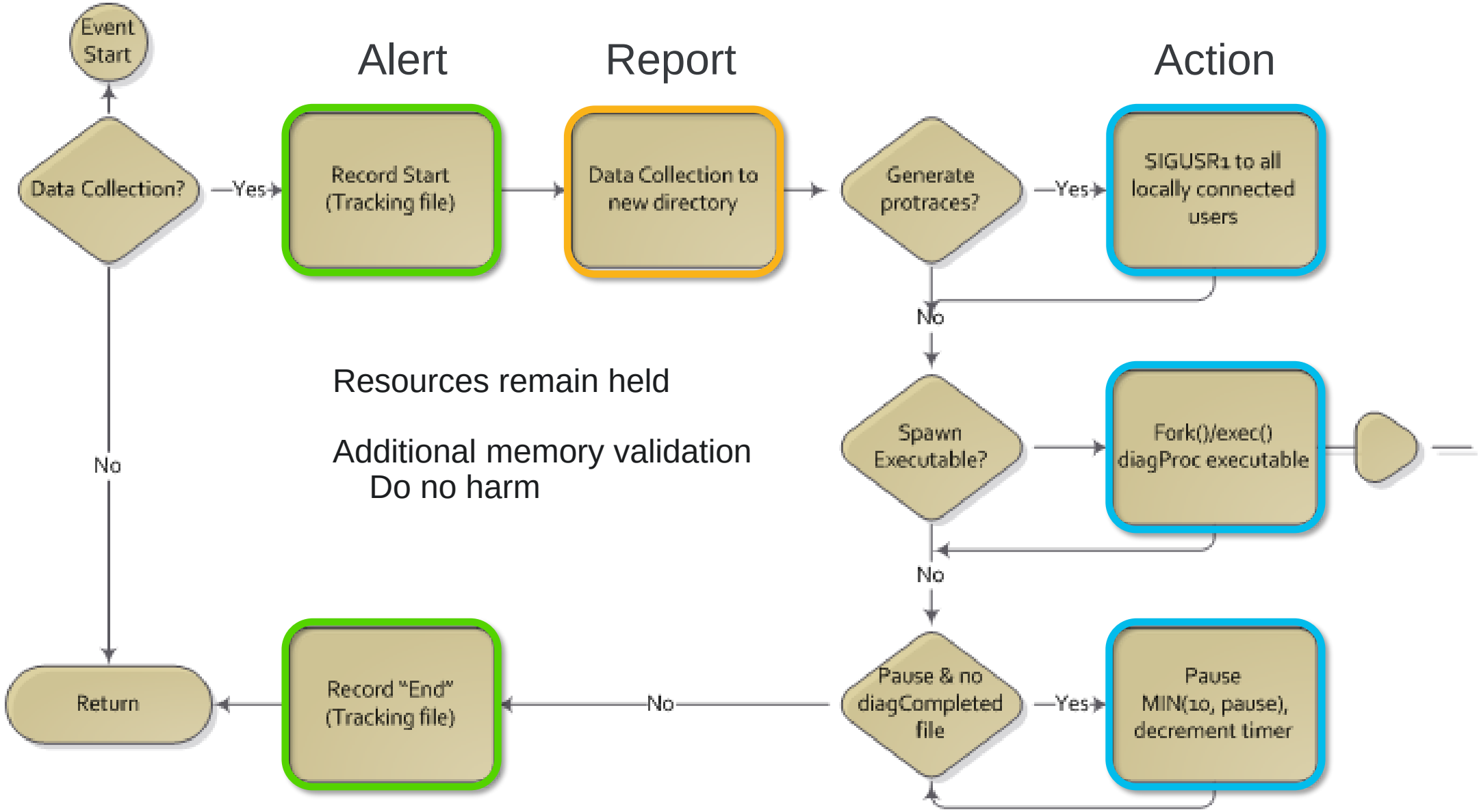
```
Find _DbParams where _DbParams-Name = "-diagPause":
    assign _DbParams-value = 10.      // pause for 10 seconds
```

- Dump now option

```
Find first _dbStatus.
    // Initiate diagnostics collection now.
    assign _dbStatus-InitiateDiag = TRUE.
```

  - Triggering "Event" name listed as "DbStatus"
  - Pause value is ignored
- Security concerns
  - Update permissions for the _dbStatus VST should be tightly controlled for both SQL and ABL users through the normal authorization / permission management mechanisms

Event Processing Review



Database Advanced Diagnostics Data Collection

Flexible

- *Multiple triggering events supported*
- *Multiple data collection levels*
- *Multiple output formats*

Configurable

- *Completely configurable online*
- *DB Startup, promon, VSTs*
- *Storage location*

Adaptable

- *Execute external program*
- *Delay continuation*
- *Delay is interruptable*


Progress®