Database Diagnostic Data Collection

- Capture database diagnostic data at time of “incident”
  - DBA generally cannot capture all needed data at the time of certain incidents
    - Lock table overflow example
  - Capture automatically or on-demand

- Data to capture
  - Incident applicable and configurable
  - Centralized and easily consumed
  - Disparate data correlated to triggering incident

- Fully configurable at startup and at runtime
- Alerting capability
Taking Action

- Triggering event (incident)
  - User 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 Parameters

- **-diagEvent**
  - LockTable:<#>,BiThold:<#>,SysErr:<#>
  - Event:Level – Comma separated list, no embedded spaces
  - Uses –diagEvtLevel by default

- **-diagEvtLevel (default: 0)**
  - Default level for all events
  - Valid only at database startup

- **Example:**
  ```bash
  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

<table>
<thead>
<tr>
<th>Event Level</th>
<th>Action Taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>LockTable:3</td>
<td>Lock table summary, detailed &amp; protraces</td>
</tr>
<tr>
<td>BiThold:1</td>
<td>BiThold summary data only</td>
</tr>
<tr>
<td>SysError:9</td>
<td>SysError summary data, BiThold actions and LockTable actions</td>
</tr>
</tbody>
</table>
Event Level Summary – Let’s break it down

0-3: Basic diagnostic data collection levels

<table>
<thead>
<tr>
<th>Level</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>1</td>
<td>Summary</td>
</tr>
<tr>
<td>2</td>
<td>Summary &amp; Detail</td>
</tr>
<tr>
<td>3</td>
<td>2 + protraces</td>
</tr>
</tbody>
</table>
Event Level Summary – Let’s break it down

0-3: Basic diagnostic **data collection** levels

<table>
<thead>
<tr>
<th>Level</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>1</td>
<td>Summary</td>
</tr>
<tr>
<td>2</td>
<td>Summary &amp; Detail</td>
</tr>
<tr>
<td>3</td>
<td>2 + protraces</td>
</tr>
</tbody>
</table>

- Summary and detailed data reported varies by event

<table>
<thead>
<tr>
<th>Name</th>
<th>Triggering Event</th>
<th>Summary</th>
<th>Detailed</th>
</tr>
</thead>
<tbody>
<tr>
<td>LockTable</td>
<td>Lock Table Overflow</td>
<td>_UserLock</td>
<td>_Lock, _Trans</td>
</tr>
<tr>
<td>BiThold</td>
<td>Bi Threshold Reached</td>
<td>_Logging</td>
<td>_ActBILog, _ActAILog, _ActIOFile, _Trans</td>
</tr>
<tr>
<td>SysErr</td>
<td>System Error</td>
<td>Same as detailed</td>
<td>User info via _UserLock</td>
</tr>
</tbody>
</table>
Event Level Summary – Let’s break it down

0-3: Basic diagnostic **data collection** levels

<table>
<thead>
<tr>
<th>Level</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>1</td>
<td>Summary</td>
</tr>
<tr>
<td>2</td>
<td>Summary &amp; Detail</td>
</tr>
<tr>
<td>3</td>
<td>2 + protraces</td>
</tr>
</tbody>
</table>
Event Level Summary – Let’s break it down

<table>
<thead>
<tr>
<th>Level</th>
<th>Data</th>
<th>Level</th>
<th>Data / Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None</td>
<td>4</td>
<td>0 + proc exec</td>
</tr>
<tr>
<td>1</td>
<td>Summary</td>
<td>5</td>
<td>1 + proc exec</td>
</tr>
<tr>
<td>2</td>
<td>Summary &amp; Detail</td>
<td>6</td>
<td>2 + proc exec</td>
</tr>
<tr>
<td>3</td>
<td>2 + protraces</td>
<td>7</td>
<td>3 + proc exec</td>
</tr>
</tbody>
</table>

4-7: Program execution in combination with chosen option 0 thru 3
Event Level Summary – Let’s break it down

<table>
<thead>
<tr>
<th>Level</th>
<th>Data / Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>1</td>
<td>Summary</td>
</tr>
<tr>
<td>2</td>
<td>Summary &amp; Detail</td>
</tr>
<tr>
<td>3</td>
<td>2 + protraces</td>
</tr>
<tr>
<td>4</td>
<td>0 + proc exec</td>
</tr>
<tr>
<td>5</td>
<td>1 + proc exec</td>
</tr>
<tr>
<td>6</td>
<td>2 + proc exec</td>
</tr>
<tr>
<td>7</td>
<td>3 + proc exec</td>
</tr>
</tbody>
</table>
Event Level Summary – Let’s break it down

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

<table>
<thead>
<tr>
<th>Level</th>
<th>Data / Action</th>
<th>Level</th>
<th>Data / Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None</td>
<td>8</td>
<td>0 + report other events’ data too</td>
</tr>
<tr>
<td>1</td>
<td>Summary</td>
<td>9</td>
<td>1 + report other events’ data too</td>
</tr>
<tr>
<td>2</td>
<td>Summary &amp; Detail</td>
<td>10</td>
<td>2 + report other events’ data too</td>
</tr>
<tr>
<td>3</td>
<td>2 + protraces</td>
<td>11</td>
<td>3 + report other events’ data too</td>
</tr>
<tr>
<td>4</td>
<td>0 + proc exec</td>
<td>12</td>
<td>4 + report other events’ data too</td>
</tr>
<tr>
<td>5</td>
<td>1 + proc exec</td>
<td>13</td>
<td>5 + report other events’ data too</td>
</tr>
<tr>
<td>6</td>
<td>2 + proc exec</td>
<td>14</td>
<td>6 + report other events’ data too</td>
</tr>
<tr>
<td>7</td>
<td>3 + proc exec</td>
<td>15</td>
<td>7 + report other events’ data too</td>
</tr>
</tbody>
</table>

- 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 with 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
  - 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

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

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>PID</th>
<th>TID</th>
<th>Level</th>
<th>ProcType</th>
<th>Event</th>
<th>Dbname</th>
<th>EventDir</th>
<th>EventId</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017…</td>
<td>P-29288</td>
<td>T1</td>
<td>3</td>
<td>SELF</td>
<td>LockTable</td>
<td>XYZ</td>
<td>…_LockTable_1</td>
<td>1</td>
<td>Start</td>
</tr>
<tr>
<td>2017…</td>
<td>P-29288</td>
<td>T1</td>
<td>3</td>
<td>SELF</td>
<td>LockTable</td>
<td>XYZ</td>
<td>…_LockTable_1</td>
<td>1</td>
<td>End</td>
</tr>
<tr>
<td>2017…</td>
<td>P-29288</td>
<td>T1</td>
<td>-11</td>
<td>SELF</td>
<td>SysErr</td>
<td>ABC</td>
<td>…_SysErr_2</td>
<td>1</td>
<td>Start</td>
</tr>
<tr>
<td>2017…</td>
<td>P-29288</td>
<td>T1</td>
<td>-11</td>
<td>SELF</td>
<td>SysErr</td>
<td>ABC</td>
<td>…_SysErr_2</td>
<td>1</td>
<td>End</td>
</tr>
</tbody>
</table>

**Event Data** (Lock table detail example) – One file per data collection type

- diagEvent_locktable_detail.csv
- diagEvent_locktable_summary.csv
- diagEvent_transaction_detail.csv

<table>
<thead>
<tr>
<th>Tracking ref</th>
<th>LockType</th>
<th>Rowid</th>
<th>HashChain</th>
<th>UserInfo</th>
<th>Flags</th>
<th>TransState</th>
<th>TransFlags</th>
<th>TransId</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017…</td>
<td>REC</td>
<td>32</td>
<td>5</td>
<td>#, name, tty</td>
<td>“X  L”</td>
<td>ACTIVE</td>
<td>FWD</td>
<td>12</td>
</tr>
<tr>
<td>2017…</td>
<td>REC</td>
<td>64</td>
<td>6</td>
<td>#, name, tty</td>
<td>“X  L”</td>
<td>ACTIVE</td>
<td>FWD</td>
<td>12</td>
</tr>
<tr>
<td>2017…</td>
<td>REC</td>
<td>72</td>
<td>7</td>
<td>#, name, tty</td>
<td>X  L”</td>
<td>ACTIVE</td>
<td>WFD</td>
<td>12</td>
</tr>
</tbody>
</table>
JSON Output

- **Event Tracking** – one object per line

  ```json
  {"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**

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

- Diagnostic Data – One file per event (incident)

```json
{  
  "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"
  },  
  "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"
    }
  },  
  "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
    },  
    "file_entry_2": {  
      "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
    }
  }
}
```
Taking Action

- Request protrace / prostack (Unix deployments only)
  - From all locally connected users
  - Protrace location reported in database .lg file
  
  \[
  \text{Protrace location: /usr1/richb/workdir/protrace.29288} \\
  \text{Generating: /usr2/mikej/workdir/protrace.29290}
  \]
  - 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 program “diagProc” then pauses for –diagPause seconds
  - diagProc could create “pause end” file:diagCompleted when finished executing

- Security concerns
  - Program name and location hardcoded to help with security concerns
  - 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 \(-\text{diagPause}\) seconds
  - \(-\text{diagPause} \) (0)
  - Up to 32 minutes (0 – 1920)

- **Resources remain held during the pause**
  - Allows a more consistent view
  - May affect OLTP

- **Pause completion**
  - \(-\text{diagPause}\) time exhausted
  - OR “\text{diagCompleted}” file created in the event output directory
    - Can be created by diagProc executable or manual procedure to stop the pause
Parameter Summary

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>-diagDir</td>
<td>Diagnostic directory</td>
<td>DB Directory</td>
</tr>
<tr>
<td>-diagEvent</td>
<td>Event level per event</td>
<td>-diagEvtLevel setting</td>
</tr>
<tr>
<td>-diagEvtLevel</td>
<td>Event level default</td>
<td>0 (disabled)</td>
</tr>
<tr>
<td>-diagFS</td>
<td>Field separator</td>
<td>“ “ (space)</td>
</tr>
<tr>
<td>-diagFormat</td>
<td>Data collection format</td>
<td>csv</td>
</tr>
<tr>
<td>-diagPause</td>
<td>Pause length</td>
<td>0 (none)</td>
</tr>
<tr>
<td>-diagPrefix</td>
<td>File prefix value</td>
<td>diagEvent_</td>
</tr>
</tbody>
</table>
## Promon Configuration

- **Promon R&D**
  4. Admin Functions ...
  14. Diag Data Collection

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Description</th>
<th>Current Diagnostic Data Collection Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/12/17</td>
<td>14:16:56</td>
<td>Diagnostic Data Collection</td>
<td>1. Diagnostic directory: /usr1/richb/dbdiag</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Diagnostic field separator: ' '</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3. Diagnostic pause time: 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4. Diagnostic prefix value: diagEvent_</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5. Diagnostic report format: json</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6. Lock Table Overflow: 3: Summary, detailed &amp; protrace data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7. BI Threshold: 7: Summary, detailed &amp; protrace data w/proc invocation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8. System Error: 15: Summary, detailed &amp; protrace data w/proc invocation and all other enabled collectors</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9. Collect enabled diagnostic data now</td>
</tr>
</tbody>
</table>

Enter a number, P, T, or X (?) for help:
Promon Configuration – Everything is 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 – Everything is changeable online!

- Configuration

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

- Dump now option

  ```plaintext
  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

Alert
- Data Collection? Yes: Record Start (Tracking file)
- Data Collection to new directory
- Generate protraces?
  - Yes: SIGUSR1 to all locally connected users
  - No: Spawn Executable?
    - Yes: Fork()/exec() diagProc executable
    - No: Pause & no diagCompleted file
- No: Return

Report
- Data Collection to new directory
- Generate protraces?
  - Yes: SIGUSR1 to all locally connected users
  - No: Fork()/exec() diagProc executable
- No: Pause & no diagCompleted file
- No: Pause MIN(10, pause), decrement timer

Resources remain held
Additional memory validation
Do no harm
Database Advanced Diagnostics Data Collection (OE 11.7.1)

**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