

PDQ/PHP5 Analyzer v1.0

Based upon: "PDQ Analyzer v3.0 111904"

(04 April 2006)

PHP5 Translation by

Samuel Zallocco

University of L'Aquila
(Italy)

Original C-language version by

Dr. Neil J. Gunther

www.perfdynamics.com

Disclaimer:

This is a brief documentation for PDQ/PHP5 Analyzer v 1.0 "All In One". This is a C to PHP5 translation based upon: "PDQ Analyzer v3.0 111904" PDQ (Pretty Damn Quick) performance analyzer by Dr. Neil J. Gunther of www.perfdynamics.com. The PHP5 translation was made by Samuel Zallocco of University of L'Aquila (Italy) for research and study purpose.

All Copyright © and Rights are reserved by the original author of the C-language version.

CONSTANT:

Constants are been simply translated as it was defined in the original C-Language version except for some unused ones that results after the PHP5 translation.

Generic internal use constant:

```
define("MAXNODES",1000);      /* Max queueing nodes */
define("MAXBUF",256);        /* Biggest buffer */
define("MAXSTREAMS",30);     /* Max job streams */
define("MAXCHARS",24);      /* Max chars in param fields */
define("MAXVAL",21);
define("MAXSUFF",10);
define("MAXPOP1",1000);
define("MAXPOP2",1000);
define("MAXDEVS",10);
define("MAXCLASS",3);

define("TOL",0.0010);        /* Tollerance */

define("PDQ_COMMENTS_FILE","comments.pdq");
define("PDQ_OUT_FILE","PDQ.out");
```

Queueing Network Type:

```
define("VOID",-1);
define("OPEN",0);
define("CLOSED",1);
```

Nodes:

```
define("MEM",2);             /* Memory ?!?!?!? */
define("CEN",3              /* unspecified queueing center */
define("CENTER",3);        /* unspecified queueing center alias added by SZ */
define("SERVICECENTER",3 /* unspecified queueing center alias added by SZ */
define("SINGLESERVERQUEUE",3); /* unspecified queueing center alias added by SZ */
define("DLY",4             /* unspecified delay center */
define("DELAY",4);        /* unspecified delay center alias added by SZ */
define("DELAYCENTER",4);  /* unspecified delay center alias added by SZ */
define("MSQ",5);          /* unspecified multi-server queue */
define("MULTISERVERQUEUE",5); /* unspecified multi-server queue alias added by SZ */
```

Queueing Disciplines:

```
define("ISRV",6);          /* infinite server */
define("IS",6);           /* infinite server alias added by SZ */
define("FCFS",7);        /* first-come first-serve */
define("FIFO",7);       /* first-in first-out alias added by SZ */
define("PSHR",8);       /* processor sharing */
define("PS",8);         /* processor sharing alias added by SZ */
define("LCFS",9);      /* last-come first-serve */
define("LIFO",9);     /* last-in first-out alias added by SZ */
```

Job Types:

```
define("TERM",10);           /* Terminal workload */
define("TERMMINAL",10);     /* Terminal workload alias added by SZ*/
define("INTERACTIVE",10);   /* Terminal workload alias added by SZ*/
define("TRANS",11);         /* Transactional workload */
define("TRANSACTION",11);   /* Transactional workload alias added by SZ */
define("BATCH",12);         /* Batch workload */
```

Solution Methods:

```
define("EXACT",13);
define("APPROX",14);
define("CANON",15);
```

Service-demand Types:

```
define("VISITS",16);
define("DEMAND",17);
```

MP scalability:

```
define("PDQ_SP",18);        /* uniprocessor */
define("PDQ_MP",19);        /* multiprocessor */
```

VARIABLES

PHP assign variable types by variables actual value.

```
// This string gets read by GetVersion script amongst others
$PDQ_version = "PDQ Analyzer v3.0 111904";

$PDQ_model = ""; // char[MAXCHARS] /* Model name */
$PDQ_wUnit = ""; // char[10] /* Work unit string */
$PDQ_tUnit = ""; // char[10] /* Time unit string */

$PDQ_DEBUG = FALSE; // int
$PDQ_prevproc = ""; // char[MAXBUF];
$PDQ_prev_init = FALSE; //int
$PDQ_demand_ext = 0; //int
$PDQ_nodes = 0; // int
$PDQ_streams = 0; //int
$PDQ_iterations = 0; // int
$PDQ_method = 0; // int
//$PDQ_memdata = 0; // int from the original C-language Version but never used !!
$PDQ_sumD = 0.0; // double
$PDQ_tolerance = 0.0; // double
```

Temp buffers

```
$PDQ_k = 0; // static int
$PDQ_c = 0; // static int
```

```
$PDQ_syshdr = 0; // int;
$PDQ_jobhdr = 0; // int
$PDQ_nodhdr = 0; // int
$PDQ_devhdr = 0; // int
$PDQ_trmhdr = FALSE; // int
$PDQ_bathdr = FALSE; // int
$PDQ_trxhdr = FALSE; // int
```

```
$PDQ_node = NULL; // NODE_TYPE
$PDQ_job = NULL; // JOB_TYPE
```

```

$PDQ_typedtable = array(
    "VOID"=>VOID,          VOID=>"VOID",
    "OPEN"=>OPEN,          OPEN=>"OPEN",
    "CLOSED"=>CLOSED,      CLOSED=>"CLOSED",
    "MEM"=>MEM,             MEM=>"MEM",
    "CEN"=>CEN,            CEN=>"CEN",
    "DLY"=>DLY,            DLY=>"DLY",
    "MSQ"=>MSQ,            MSQ=>"MSQ",
    "ISRV"=>ISRV,          ISRV=>"ISRV",
    "FCFS"=>FCFS,          FCFS=>"FCFS",
    "PSHR"=>PSHR,          PSHR=>"PSHR",
    "LCFS"=>LCFS,          LCFS=>"LCFS",
    "TERM"=>TERM,          TERM=>"TERM",
    "TRANS"=>TRANS,        TRANS=>"TRANS",
    "BATCH"=>BATCH,        BATCH=>"BATCH",
    "EXACT"=>EXACT,        EXACT=>"EXACT",
    "APPROX"=>APPROX,      APPROX=>"APPROX",
    "CANON"=>CANON,        CANON=>"CANON",
    "VISITS"=>VISITS,      VISITS=>"VISITS",
    "DEMAND"=>DEMAND,      DEMAND=>"DEMAND",
    "SP"=>PDQ_SP,          PDQ_SP=>"SP",
    "MP"=>PDQ_MP,          PDQ_MP=>"MP"
); /* typedtable */

```

TYPES as CLASS

PHP can't define type structures so we have to use class!!

```

class SYSTAT_TYPE {
    var $response = 0.0; // double
    var $thruput = 0.0; // double
    var $residency = 0.0; // double
    var $physmem = 0.0; // double
    var $highwater = 0.0; // double
    var $malloc = 0.0; // double
    var $mpl = 0.0; // double
    var $maxN = 0.0; // double
    var $maxTP = 0.0; // double
    var $minRT = 0.0; // double
};

class TERMINAL_TYPE {
    var $name= ""; // char [MAXCHARS]
    var $pop = 0.0; // double
    var $think = 0.0; // double
    var $sys = NULL; // SYSTAT_TYPE
};

```

```

class BATCH_TYPE {
    var $name = ""; // char [MAXCHARS];
    var $pop = 0.0; // double
    var $sys = NULL; // SYSTAT_TYPE
};

class TRANSACTION_TYPE {
    var $name = ""; // char [MAXCHARS];
    var $arrival_rate = 0.0; // double
    var $saturation_rate = 0.0; //double
    var $sys = NULL; // SYSTAT_TYPE
};

class JOB_TYPE {
    var $should_be_class = 0; // int stream should_be_class
    var $network = 0; // int OPEN, CLOSED
    var $term = NULL; // TERMINAL_TYPE
    var $batch = NULL; // BATCH_TYPE
    var $trans = NULL; // TRANSACTION_TYPE
};

class NODE_TYPE {
    var $devtype = 0; // int CEN, ...
    var $sched = 0; // int FCFS, ...
    var $devname = ""; // char [MAXCHARS];
    var $visits = array(); // double [MAXSTREAMS];
    var $service = array(); // double [MAXSTREAMS];
    var $demand = array(); // double [MAXSTREAMS];
    var $resit = array(); // double [MAXSTREAMS];
    var $utiliz = array(); // double [MAXSTREAMS]; /* computed node utilization */
    var $qsize = array(); // double [MAXSTREAMS];
    var $vqsize = array(); // double [MAXSTREAMS];

    function __construct()
    {
        $NODE_TYPE->visits = array_fill(0,MAXSTREAMS-1,0.0); // double [MAXSTREAMS];
        $NODE_TYPE->service = array_fill(0,MAXSTREAMS-1,0.0); // double [MAXSTREAMS];
        $NODE_TYPE->demand = array_fill(0,MAXSTREAMS-1,0.0); // double [MAXSTREAMS];
        $NODE_TYPE->resit = array_fill(0,MAXSTREAMS-1,0.0); // double [MAXSTREAMS];
        $NODE_TYPE->utiliz = array_fill(0,MAXSTREAMS-1,0.0); /* computed node utilization */
        $NODE_TYPE->qsize = array_fill(0,MAXSTREAMS-1,0.0); // double [MAXSTREAMS];
        $NODE_TYPE->vqsize = array_fill(0,MAXSTREAMS-1,0.0); // double [MAXSTREAMS];
    }
};

```

FUNCTION PROTOTYPE

Public Function translated by the original version:

```
function PDQ_Init($name = "The Model Name")
```

```
function PDQ_CreateOpen($name, $lambda)
```

```
function PDQ_CreateClosed($name, $should_be_class, $pop, $think)
```

```
function PDQ_CreateNode($name, $device, $sched)
```

```
function PDQ_SetDemand($nodename, $workname, $time)
```

```
function PDQ_SetVisits($nodename, $workname, $visits, $service)
```

```
function PDQ_SetWorkloadUnit($unitName) // alias of PDQ_SetWUnit added by SZ
```

```
function PDQ_SetWUnit($unitName)
```

```
function PDQ_SetTimeUnit($unitName) // alias of PDQ_SetTUnit added by SZ
```

```
function PDQ_SetTUnit($unitName)
```

```
function PDQ_SetDebug($flag)
```

```
function PDQ_Solve($meth)
```

```
function PDQ_GetResponse($should_be_class, $wname)
```

```
function PDQ_GetThruput($should_be_class, $wname)
```

```
function PDQ_GetUtilization($device, $work, $should_be_class)
```

```
function PDQ_GetThruMax($should_be_class, $wname)
```

```
function PDQ_GetLoadOpt($should_be_class, $wname)
```

```
function PDQ_GetResidenceTime($device, $work, $should_be_class)
```

```
function PDQ_GetQueueLength($device, $work, $should_be_class)
```

```
function PDQ_Report()
```

Function added by SZ:

```
function gets(&$s)
```

```
function atol($s)
```

```
function atof($s)
```

```
function atoi($s)
```

```
function PDQ_Repair($m, $S, $N, $Z)
```

```
function PDQ_Erlang($servers, $traffic)
```

```
function PDQ_GetVersion($printit = FALSE)
```


Private “Internal” Function translated by the original version:

```
function PDQ_Debug($proc, $info)
function PDQ_TypeToStr(&$str, $type)
function PDQ_StrToType($str)
function PDQ_AllocateNodes($n)
function PDQ_AllocateJobs($jobs)
function PDQ_GetJobIndex($wname)
function PDQ_GetNodeIndex($name)
function &PDQ_GetNode($idx)
function PDQ_GetJobName(&$str, $c)
function PDQ_GetJobPop($c)
function &PDQ_GetJob($c)
function PDQ_RoundUp($f)
function PDQ_Resets(&$s)
function PDQ_ErrMsg($pname, $msg)
function PDQ_CreateTermStream($circuit, $label, $pop, $think)
function PDQ_CreateBatchStream($net, $label, $number)
function PDQ_CreateTransaction($net, $label, $lambda)
function PDQ_Approx()
function PDQ_SumQ($k, $skip)
function PDQ_Canonical()
function PDQ_SumU($k)
function PDQ_Exact()
function PDQ_ReportNull()
function PDQ_PrintNodeHead()
function PDQ_PrintNodes()
function PDQ_PrintJob($c, $should_be_class)
function PDQ_PrintSysHead()
function PDQ_PrintJobHead($should_be_class)
function PDQ_PrintDevHead()
function PDQ_PrintSystemStats($c,$should_be_class)
function PDQ_PrintNodeStats($c, $should_be_class)
function PDQ_BannerStars()
function PDQ_BannerChars($s)
```

Note:

For details see the source code!