



---

## Automated Collection of Real-Time Production Data

---

This paper describes the functions, benefits, and architecture of a robust automatic data collection system for machines used in discrete manufacturing

---

Wintriss Controls Group

---

## Introduction

Every manufacturer wants to reduce machine downtime. With limited resources available to devote to downtime reduction, perhaps the most important ongoing process in any downtime reduction program is to accurately identify where your resources can do the most good. The first step in any downtime reduction program should be measurement, because you can't fix what you don't measure. The most efficient and reliable method of downtime measurement is through automatic shop floor data collection.

Automatically collected data can also improve your business in other ways. The software systems that you use to run your business - including Enterprise Resource Planning (ERP) and Manufacturing Execution Systems (MES) - rely on production data collected from the shop floor. In most cases, this data is collected manually and entered into the system by a human operator. The three common problems with manually collected data are untimeliness, inaccuracy, and bias. Considering that this raw data forms the basis for all subsequent production reports - and that important decisions are made based on those reports - any problems with the initial data collection can start a ripple effect that results in a negative impact on your business. Automatic shop floor data collection allows you to make better business decisions.

This paper will detail the shortfalls of manual data collection, explain how automatic data collection can eliminate these errors, and describe the function and architecture of a robust automatic data collection system.

### Manually collected data is not timely

In a typical scenario, manually collected production data is entered into the system only at predetermined times such as at the end of a shift or job. The data is then made available in the form of reports and used for analysis. For long-term analysis, viewing the data "after the fact" is usually sufficient.

However, the factory floor is a dynamic environment, and only knowing what happened after the fact can be a detriment to productivity. Consider the following:

- When a machine goes down, how long does it take maintenance personnel to respond?
- Do the people responsible for material handling know when a machine is starved for raw material?
- If a machine is down for an extended period of time, is the scheduler able to quickly respond and modify workflow to keep jobs moving through the queue?

Another problem with manually entered data is that it is often incomplete. This is due to the fact that the actual data entry task usually falls to someone with many other responsibilities. Generally speaking, data entry is a tedious, boring task that is often put off for as long as possible. It is not uncommon for raw data to sit around for hours, if not days, before being entered into the system. This results in reports that are incomplete insofar as the latest data is concerned.

### Manually collected data is inaccurate

There are many opportunities for inaccuracies to creep into a manual data collection system. Often the data must be first written down, and later entered into the system – sometimes by a different person than the one that recorded it in the first place. Typographical and transcription errors are common. Human error is inevitable, and once these errors become part of the data set, they become difficult to detect and eradicate, and make all the resulting reports suspect.

### **Manually collected data is biased**

Whenever a human operator enters data, he or she has influence over exactly what information is entered and when it is entered. Consider the following example:

Let's say that the controller on Machine Number 1 detects an easily correctable problem and stops the machine. Let's also say that the operator responsible for Machine Number 1 is busy elsewhere in the plant, and Machine Number 1 sits idle for an hour while the operator is otherwise occupied. When the operator is finally free, he spends 5 minutes fixing the problem, and then restarts Machine Number 1. Even though he was busy with another machine, the operator is worried that he'll be disciplined for not attending to Machine Number 1 quickly enough, so he logs "Tooling problem – 10 minutes" as the reason for downtime.

The resulting 'data' is wrong. The machine was down for 1 hour and five minutes, not 10 minutes as the operator indicated. Unfortunately, the "real" reason for most of this downtime - "No Operator Available" – was not even captured. Perhaps because the operator did not want to appear inattentive, or simply because it is easier to blame downtime on a mechanical problem rather than personnel issue, the operator's bias played a part in how the downtime was reported. Even more unfortunate is the fact that the real problem (the plant is understaffed) goes completely undetected, and as a result will more than likely continue to happen again and again. Bias can be expensive. In the example above, a legitimate problem- "We need an additional operator" - ends up being ignored.

### **Automated Data Collection**

The problems outlined above can be eliminated when production data is collected automatically. When production data is collected automatically as it happens, you can be assured that it is timely, accurate, and unbiased. Until recently, automatically collecting production data was a costly and unreliable proposition. You had the choice of writing your own custom system (at great expense), or using a series of machine-specific commercially available packages.

Commercially available data collection software tended to be vendor-specific – especially those that collected data from proprietary machine controllers. If you had several different types of machinery/controllers, you'd need several different data collection systems that would result in fragmented data unless you hired a third-party to integrate them into an expensive custom solution. Often these proprietary collection systems required an antiquated serial network, as well as client software that had to be installed on any workstation that needed access to the production data. These features made them quite maintenance intensive.

Fortunately, a number of factors have combined to make automatic data collection technology much more reliable and accessible:

- The emergence of high speed Ethernet as the de facto standard for local area networks has resulted in a tremendous decrease in the cost of network cards, adapters, and other hardware, thus allowing Ethernet to replace the outdated, slow, and expensive dedicated serial networks.
- It turns out that the Internet and email were not passing fads after all. Virtually all computers are now shipped with a web browser and an email client. Data collection software can now use a web browser to display and manipulate data and email to distribute reports. This has eliminated the need to install maintenance-intensive client software on workstations.

- Database manufacturers adopted Structured Query Language (SQL) as a means to get data in and out of various databases. This makes it much easier for different software packages to share data, and allows an SQL-based data collection system to provide data to MES and ERP software.
- It is now easier to communicate with the equipment. Microsoft and the automation suppliers got together and developed a set of standards (called OPC – described below) that allows for interoperability of different devices.

Turnkey data collection software packages that use these technologies are easy to install, maintain, and upgrade. These products are scalable and have the ability to share with and receive data from your existing software systems.

### **Communicating with the equipment**

The biggest challenge in automated data collection is finding a way to communicate with a variety of different types of machinery. There are two ways to collect data from production machinery:

Option 1 - Communicate directly with the existing controllers on the machines

Option 2 - Add a dedicated data collection device to each machine

In recent years, several factors have made Option 1 much easier.

The ubiquity of Ethernet, along with its inherent reliability and speed has all but eliminated the need for separate slow and costly serial networks dedicated to machine data collection. New controllers feature built-in Ethernet capability; older controllers can usually be fitted with Ethernet cards or serial-to-Ethernet converters.

The maturation and widespread use of communication protocols such as Modbus TCP have provided the means to move raw data in and out of many proprietary controllers, PLCs, and other devices.

Finally, the adoption of the OPC Data Access Specification by controller manufacturers has enabled interoperability in multi-vendor systems. OPC is a communication standard that resulted from the collaboration of over 150 leading worldwide automation suppliers working in conjunction with Microsoft. To be certified, OPC devices must pass a series of compliance tests.

The often-cited analogy for the impact of OPC on the data collection world is the change undergone by the handling of printer drivers when DOS became Windows. Under DOS, the developer of every application had to also write a printer driver for every printer. So Lotus wrote the Lotus123 application and the Lotus123 printer drivers; WordPerfect wrote the WordPerfect application and the printer drivers, and so on. In addition, developers had to write a separate printer driver for every brand or type of printer they wanted to support.

Windows solved the printer driver problem by incorporating printer support into the operating system - now one printer driver serves all applications. The printer drivers had to meet certain requirements in order to be approved, and were written by the printer manufacturer - not the application developer.

In the industrial automation world before OPC, every manufacturer had to write their own proprietary “drivers” to enable their controllers to communicate with Windows. Any data collection system that needed to communicate with

a variety of controllers had to have a driver for every controller. The problem was that the manufacturers were not coordinated. Many of the controller drivers conflicted with one another, making the data collection system inoperative. OPC fixed this by enabling automation manufacturers to create non-conflicting Windows-compatible drivers that standardized the method of data communication.

While OPC made it possible for a data collection server to communicate with a variety of controllers, it does not mean that the software can make sense of the raw data that it receives. Even with OPC-compatible controllers, an interface must be written for each device to arrange the raw data into a format that can be used by the data collection software.

For Option 1 to be cost-effective, a large number of machine controllers (a dozen or more) must be identical, so that the cost of the project to write the interface can be amortized over a number of duplicate machines.

Option 2 – Adding a dedicated data collection terminal – is the best solution when you need to collect data from a variety of different machines. These terminals have built-in Ethernet capability, are easy to install (often requiring only two or three simple connections to each machine) and feature an intuitive and unobtrusive user interface, that requires operator attention only at the start of each job, or when the machine goes down for an extended period of time.



*Figure 1 – Dedicated Data Collection Terminal*

The dedicated terminal is typically inexpensive, and allows the operator to enter job and part information, as well as to specify the reasons for downtime events when they occur. The devices can be set up with downtime reason menus that are custom-tailored to each machine.

### **Requirements of a good data collection system**

A modern data collection system is composed of a data logger/transaction manager, a database, a report generator, and a factory viewer real-time interface. These programs typically operate together on a single server

The two “client” portions are made up of an email client (such as Microsoft Outlook) to receive reports, and a web browser (Internet Explorer, Firefox, Navigator, etc) to display the factory viewer portion. The web browser acts as the “front end” to the system. It is used to view real time data as well as for setup and maintenance, security access, etc.

The heart of the data collection system is the data logger. The data logger is a program that gathers the raw production data, organizes it, and puts it in a database. There are two different data logging methods that can be employed by data collection systems. These are distributed data loggers and centralized data loggers.

In a distributed system, a data logger is installed on each machine. The data logger collects production data directly from the machine, stores it in memory, and periodically sends it off to the data repository (usually a database). The advantage of a distributed system is that in the event of a network outage, the on-board data logger can continue to collect production information from the machine while the network is down, and then “backfill” the database when the network comes back up.

However, the disadvantages of a distributed system outweigh this one advantage:

- Since distributed data loggers do not continuously update the database, you cannot view the production environment in real time.
- Data logging capability cannot easily be retrofitted into existing controllers. The hardware requirements for a data logger typically include multiple processors and significant on-board memory. Most existing machine controllers do not have the spare “horsepower” to run a data logger. In almost all cases, a separate dedicated (usually expensive) data logging device must be installed on each production machine.
- Maintenance is more difficult with distributed systems. Inevitably, there will come a time when it will become necessary to upgrade/update the data loggers. This is usually triggered by changes in the Windows operating system. Instead of simply updating a single data logger (as you would in a centralized system) you must update every individual machine.
- When it's time for the distributed data logging devices to upload their information to the database, the resulting spike in network traffic has the potential to adversely affect your LAN.

In a centralized system, there is one data logger that constantly polls each individual production machine. A centralized system has many advantages:

- Since the system operates at Ethernet speeds, the data coming from the production machinery can be displayed within a few seconds of real time.

- The control requirements at the machine are greatly simplified when compared to a distributed system. A centralized system requires the machine to pass only a few of the most recent data items (called “tags”) during each polling cycle. Since the controller on the production machine does not need to store and organize data, it is a simple matter to modify existing controls to provide the tags. Many proprietary, PLC, and PC-based controllers employ a communication standard called OPC (described above) that allows them to connect directly to the centralized data logger without modifications. If the production machine is older or has a control that cannot be modified for communications - for example, a relay-based control – a simple Ethernet compatible HMI terminal can be installed on the machine to provide the needed data tags.
- Maintenance is easier because only one data logger needs to be cared for.
- Since the machine constantly updates the data logger, only a few bytes of data need to be sent on each polling cycle. As a result, the network traffic is minimal and constant – there are no spikes caused by large amounts of information being transferred all at once.

The one drawback of a centralized system – data loss in the event of a network outage – is mitigated by the fact that properly designed Ethernet networks are now extremely reliable. By far, the most common cause of network outage is power failure and you’re unlikely to be running your machines during a power failure.

The database stores all of the production information and “feeds” the report generator and web browser front end. An SQL database (such as Microsoft’s SQL Server) makes it easy to transfer data to and from other software. The database for a data collection system should have the ability to simultaneously handle multiple users and should be scalable to grow with your needs.

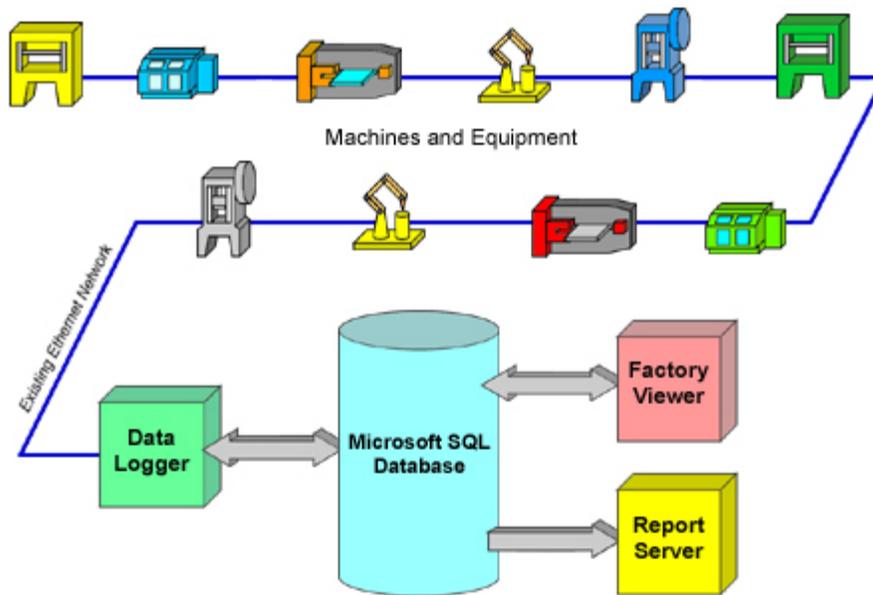


Figure 2 – Data Collection System Components

The report generator allows you to create tabular and graphic reports from the information in the database. Reports can typically be configured to display data according to the date range, shift, machine, tool/part, and/or operator. For example, you could specify that you want a report showing production data for four of your machines, while they were making ten specific parts, being run by five selected operators, on third shift. Once configured, you can save the settings as a new report.

The report generator also controls report distribution. Reports can typically be sent to a printer, sent via email to any recipients that you chose, and/or posted on your network as HTML documents that can be viewed in a web browser by anyone with appropriate access.

In addition, the report generator schedules when reports are distributed. For example, you could set the report generator to post the previous day's Production Report onto your network each morning. Once you've set up the distribution schedule, the report generator will automatically generate and distribute the reports.

The factory viewer is the main user interface for the data collection software. In addition to displaying production data in real time, it allows users to configure reports and deliveries, and control the settings behind the system. Ideally, the factory viewer will not require the user to install any special software other than the existing web browser.

### **Which data are collected?**

Believe it or not, you can learn a lot about your operation by collecting just three pieces of data from each machine or line. These are part (or cycles) count (if applicable), machine "state" (running, idle, setup, and planned or unplanned downtime), and an error code that corresponds to a downtime reason.

### **Parts Counts**

Collecting an accurate part count is not as simple as it might appear. When the data logger receives a count from a machine, it must compare it to the counter value that was collected during the previous polling cycle. Advanced systems also compare both the current and previous counter values to a third "hidden" non-resettable counter. By applying a series of logical steps, these advanced data loggers can look at the current counter value and determine:

- Did the counter simply increment or decrement?
- If so, by how many counts?
- Was the counter manually reset or manually changed by the machine operator?
- Was the counter reset because a new job was started?
- Did the counter run up against a maximum value and roll over?

By running the three counters through various algorithms, the transaction manager can correctly decide how to place this information into the database, thereby avoiding "nonsense" data such as indicating that a negative number of parts was produced (caused by an undetected counter reset or job change) or showing that a ridiculously high number of parts were made in a very short time (caused by a manual change to the parts counter).

## Machine States

The “machine state” refers to the current condition of the machine. These are the typical machine states:

- Running – The machine or line is in production and producing parts
- Idle – The machine is stopped for reasons unknown
- Unplanned Downtime – The machine is stopped and the data collection software knows the reason for the unplanned stoppage.
- Planned Downtime – The machine is stopped and it was planned in advance (lunch, break, scheduled maintenance, etc)
- Changeover/Setup – The machine is being set up or changed to make a different part. It may be stopped, or run intermittently during this time. Some data collection packages treat changeover/setup as if it were running or downtime. More advanced systems track it separately, enabling the creation of reports detailing the differences in setup time from any part to every other part on every machine. By tracking changeover separately, advanced systems can accurately predict the completion time for each job in a machine’s queue, and can assist in scheduling jobs to minimize the total changeover time (described below).
- Offline – While not a true “state”, Offline indicates that the power was off or the network was down.

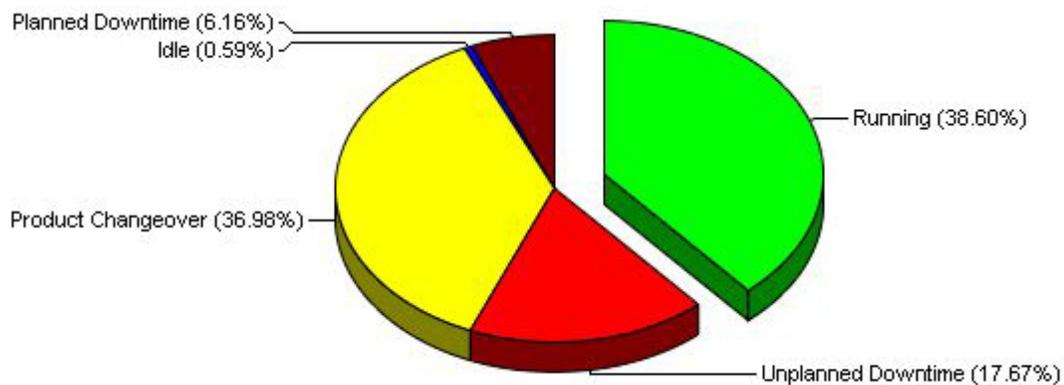


Figure 3 – Pie chart showing time spent in the various machine states

The data logger constantly watches the machine state and logs the elapsed time accordingly. The data logger logs every second of time to one of the above states.

## Error Codes

The most efficient way to track downtime is for the data logger to collect an error code any time a machine is stopped for any reason. These error codes can be automatically generated by the controller on the machine, or can correspond to a list of downtime reasons that are manually selected by the machine operator (see the section below on Operator Interfaces). The error code is placed into the database by the Data logger and compared to a lookup

table specifically created for each machine in the database. By using an error code instead of an error message, network traffic is kept to a minimum.

Another benefit obtained by comparing the error code to a lookup table unique to each machine is that you can create a different set of downtime reasons for each machine type. This way, you can create a small set of specific downtime reasons that make sense for each machine. The alternatives are to either use a huge “master list” of reasons (most of which will not apply to any given machine) or a list of generic reasons too vague to allow you to easily identify specific reasons for downtime.

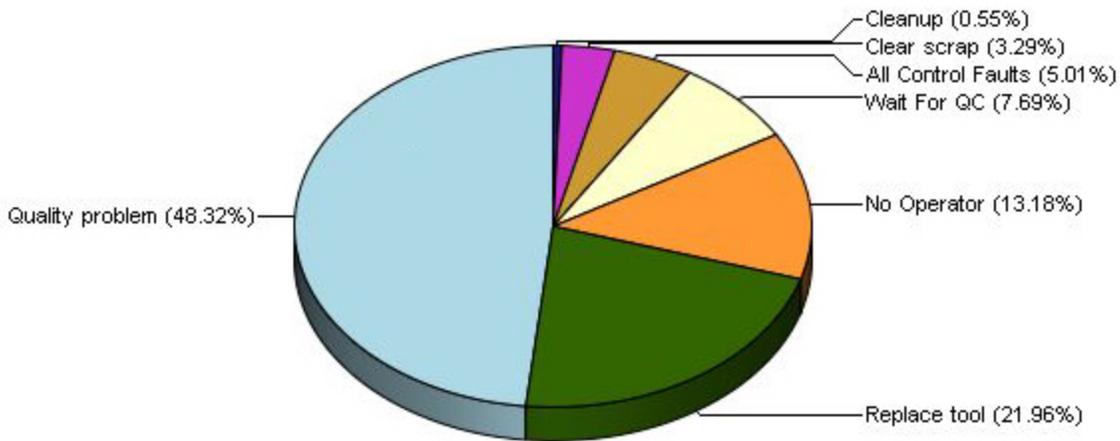


Figure 4 - Downtime Summary Chart with error codes.

Depending on the machine or line, data such as temperature, pressure, force, or other process parameters may also be collected. Typically, these types of data are stored along with a time and date stamp, and can be put into a report as a graph for a specified time period.

### Operator Interfaces and Downtime Reasons

Ideally, all data collected from a production machine would be collected automatically with no input required from the machine operator. Unfortunately, this is not possible. While parts count and machine state can usually be collected without operator input, the exact reasons for downtime cannot.

It may be possible to collect some downtime reasons automatically. If a machine is equipped with an “intelligent” controller or monitoring device, and the controller initiates the machine stop, the reason – in the form of an error code - can be reported back to the data logger automatically.

However, if the machine is stopped by the operator or by a piece of ancillary equipment, the data collection software will have no visibility to the reason for stoppage. When this is the case, the data logger is forced to log this time as “Idle” time. Since one of the main benefits of a data collection system is to identify reasons for productivity loss, simply logging idle time is unhelpful. There is no such thing as idle time – there is always a reason for a machine not running.

The most effective way to enable an operator to specify the cause of a machine stoppage is to provide a menu of choices from which the operator selects the appropriate downtime reason. If the machine is equipped with a dedicated data collection terminal, the downtime reason can be simply selected from a menu. If the machine has a proprietary control, or a PLC or PC-based control with a menu-driven Human Machine Interface (HMI), then this ability will have to be added to the control system.

Even with the ability to specify the downtime reason, it is not enough to simply create the menus or install the terminal and instruct the operator to select a downtime reason. Unless the operator is reminded, he or she will not always select a downtime reason each time the machine stops. Remember, if the machine is restarted without selecting a reason, the time will be logged as “idle”, and the real reason for the stoppage is lost.

One way to remind the operator to select a downtime reason is to set the controller or data terminal to prevent machine restart until a reason is selected. There is usually a short grace period to allow for brief stoppages, but if the machine is stopped for a significant amount of time, the operator will be forced to select a downtime reason.

Data collection cannot come at the expense of productivity. There are a few design considerations to adopt that will minimize the negative impact on production.

First, you should carefully select a small number of reasons (a dozen or so) for each machine. It helps if you standardize the reasons for each machine type so that operators are not forced to learn a new menu when they move from machine to machine. The most frequently selected downtime reasons should be put near the top of the menu and the least frequently selected reasons should go near the bottom. Finally, if the controller or terminal features multiple menus, it should be set to display the downtime reason menu automatically when it is looking for a downtime reason. This helps to minimize unnecessary navigation on the operator’s part.

### **What can be done with the data?**

Once you’ve collected the data, the sky is the limit on what you can do with it. Just the initial downtime analysis can help you to quickly identify the “low hanging fruit” that will often pay for the entire project. Production information can be passed on to ERP and MES systems. Schedulers can use the system to accurately predict when jobs will finish and machines will be available. Production counts and efficiencies can be put into report form and distributed. An important metric that can be created with the data described above is Overall Equipment Effectiveness or OEE.

Overall Equipment Effectiveness (OEE) is a simple percentage that shows the ratio of actual equipment output to its theoretical maximum. OEE factors in equipment availability, speed performance, and quality, and is based on the premise that all production losses on machines and processes can be measured and quantified.

OEE is calculated using a simple formula:

$$\text{OEE} = \text{Availability} \times \text{Performance} \times \text{Quality}$$

The *Availability* accounts for unplanned downtime losses. It is equal to the actual machine/process running time divided by the total available time. Planned downtime events such as lunch breaks are not part of, and don’t affect, the OEE calculation.

The Performance accounts for speed loss. It is equal to the ratio of the number of parts produced over the measurement period (shift, day, etc.) to the theoretical maximum number of parts that could be produced if the machine or process ran at its highest possible speed. The performance calculation is simple when the machine or process produces one part per cycle. However, the performance calculation gets complicated when a machine produces multiple parts per cycle, and even more complicated when a machine makes varying numbers of parts depending on the job being run.

Advanced systems have the ability to calculate an accurate OEE for operations where multiple parts are made during each machine cycle - even when multiple jobs making differing numbers of parts-per-cycle were run during the same period. This is important for metal stamping and injection molding operations.

The Quality is the ratio of good parts to total parts produced.

**Here's an example:**

During an 8-hour shift with 1/2 hour for lunch and two 15-minute breaks, a machine has a maximum availability of 7 hours (420 minutes). If there were 82 minutes of unplanned downtime during the shift, then the machine would have actually run for 338 minutes. The availability would be calculated as follows:

$$\text{Availability} = 338 \text{ minutes} / 420 \text{ minutes} = 80\%$$

Running at full speed, the machine is capable of producing 6000 parts/hour (or 100 parts per minute). During the 338 minutes of running time in our example, the machine made 25,000 parts. The performance percentage is:

$$\text{Performance} = 25,000 \text{ parts} / 338 \text{ mins.} / 100 \text{ parts/min.} = 74\%$$

Out of the 25,000 parts produced, 500 had to be scrapped. The quality percentage is the ratio of good parts to total parts, and is calculated as follows:

$$\text{Quality} = 24,500 \text{ Good Parts} / 25,000 \text{ Total Parts} = 98\%$$

The OEE for this example is:

$$\text{OEE} = \text{Availability (80\%)} \times \text{Performance (74\%)} \times \text{Quality (98\%)} = 58\%$$

You can see that although the component measurements - 80% uptime at 74% of maximum speed with 98% quality - indicate an efficient process, when taken together as OEE, the process is really only 58% effective.

Tracking OEE has many benefits such as:

- Quickly highlighting the greatest potential areas of improvement to provide the greatest return on assets. This will allow you to focus your resources where they can do the most good
- Making the most out of a limited amount of investment capital
- Avoiding making inappropriate equipment purchases
- Freeing up capacity to better compete for new business
- Prioritizing lean initiatives

- Decreasing costs through waste elimination
- Shortening equipment ROI through increased utilization
- Reducing investigation time for root cause analysis
- Directly tying production efficiencies to fiscal reporting

### Real Time Data

In addition to tracking OEE, an automatic data collection system can provide you the ability display production data in real time. This can have a huge positive affect on downtime reduction in ways that you might not expect.

Any person responsible for keeping a machine running has instant visibility to its current state. When a machine goes down, the responsible party can more quickly respond to the issue, thus reducing downtime.

Real time data enables the ‘visual factory’. Large-format displays can be configured and placed around the factory to display production information in real time. These displays can be configured to display such metrics as current OEE, efficiency by machine/operator/shift, parts counts, and estimated time until job completion.

The visual factory alone has been shown to increase machine uptime. When used in conjunction with a real time automatic data collection system, users have seen an average downtime reduction of 9%, after implementing visual factory display – before taking any other action to reduce downtime!

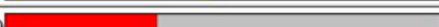
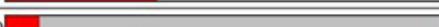
Machine	Efficiency vs. Standard Rate	Percent	Duration
Press 10	0  100%	93.3%	07:07:36
Press Brake 1	0  100%	80.3%	01:43:34
Lathe	0  100%	74.0%	02:14:30
Tube Laser	0  100%	72.0%	07:20:40
Shear	0  100%	69.6%	01:11:29
Main Turrett Press	0  100%	47.5%	01:51:11
Molder #1	0  100%	35.9%	00:16:27
Packaging Line	0  100%	8.2%	00:51:18
Robot Welder (A)	0  100%	4.8%	00:50:59
3-Axis Mill	0  100%	0.0%	00:20:18

Figure 4 – “Visual Factory” display showing efficiency by machine for the currently running job

Real time data can also help to eliminate ‘hidden downtime’. Hidden downtime is waste that will not necessarily show up on a typical report. In any production environment, there will be non-production personnel such as maintenance technicians or engineers that need periodic access to the machinery. Such access, while important in the long term, cannot get in the way of short-term production requirements. Real time data displays that indicate estimated

complete times for current jobs can allow technicians and others to adjust their schedules to time access to the machines so that it has the minimum impact on the production schedule.

### **Benefits for Scheduling**

It is often said that production scheduling is more an art than a science. In order to effectively execute a manufacturing plan, the person doing the scheduling in a production environment must have a keen insight into any hidden inefficiencies, as well as an intimate knowledge of how well each machine runs each part that must be produced. Given enough time, a well-designed data collection system can provide you with an unbiased version of this set of “tribal knowledge”.

Over time, your data collection software will measure the production rate for every machine as it produces every part. In addition, it can track the changeover time for each machine as it changes from one part to the next. This data becomes more accurate the longer the system tracks it.

Given this information, the data collection system can accurately predict when jobs will be finished. By adding in changeover time, as well as applying the measured efficiency to the quantity of parts that are required for a run, the software can indicate the time and date of completion for any job in the schedule queue. The software can indicate when exceptions occur by simply comparing the estimated complete time/date to the due date for the job.

The data collection system can also determine which machines run which jobs most efficiently. When more than one machine is capable of running a particular job, the scheduler can use this information to select the available machine that will run the job most efficiently.

### **Event Logs**

Have you ever wondered what exactly led up to a catastrophic event in a machine on 3<sup>rd</sup> shift? An automatic data collection system can tell you. Many systems create “event logs”. An event log is a time- and date-stamped list of every event that occurred on a machine. These are invaluable when you’re trying to troubleshoot or find the root cause of a problem on a machine.

### **Conclusion**

Turnkey automatic data collection systems can reliably collect timely, accurate, and unbiased data. These systems take advantage of your existing network infrastructure and have the ability to gather valuable production data from virtually every machine in your shop. Once collected, information is stored in a scalable database and is available for use by your other business software.

Initially, an automated data collection system provides huge benefits by identifying the leading causes of waste; allowing you to focus your resources where they can have the biggest impact on efficiency. After a time, when efficiency is improved to the point where your gains start to flatten out, the real-time data display capabilities become more important in maintaining your newfound levels of efficiency by allowing the appropriate personnel to more quickly respond to downtime events. The visual factory also gives managers the opportunity to more quickly detect areas that are starting to lag behind, allowing them to make more informed personnel deployment and scheduling decisions.