

Topic Sheet:
Using SPC in CPC-NT

“This topic sheet shows how you can use CPC-NT for limited SPC and Database reporting and trend analysis operations.”

Copyright © 2002, ASC Process Systems, Inc.

Revised: April 14, 2002

<u>1.</u>	<u>Introduction</u>	<u>1-1</u>
	What is SPC?	1-1
	Why use SPC analysis in CPC-controlled processes?	1-1
<u>2.</u>	<u>Run Database</u>	<u>2-1</u>
	Run Database defined	2-1
	Where is the database?	2-1
	Database type	2-1
	Database contents	2-1
	Tables.....	2-1
	No raw data	2-2
	Using database tables.....	2-2
	Relational structure	2-2
	Linking tables with key fields	2-2
	Database fields	2-2
	Attachments	2-2
	Equipment	2-4
	Messages	2-4
	Notes	2-4
	Parts	2-5
	QCStatus.....	2-5
	Runs	2-6
	Recipes	2-6
	SPCVariables.....	2-7
	Tools.....	2-7
	Variables	2-8
<u>3.</u>	<u>Using Queries</u>	<u>3-1</u>
	SQL language	3-1
	Field identification.....	3-1
	SELECT and FROM statement.....	3-1
	WHERE statement	3-1
	INNER JOIN and ON statements	3-2
	Where are queries saved?	3-3
	Entering queries in CPC.....	Error! Bookmark not defined.

1. Introduction

This topic sheet discusses the operations of CPC-NT's "SPC Analysis & Reporting" capabilities.

What is SPC?

SPC, or Statistical Process Control, is a term used to define an engineering practice of analyzing sub-sets of long-term historical data to 1.) determine the "fitness" of a process and 2.) identify degradation in equipment and process repeatability.

Why use SPC analysis in CPC-controlled processes?

SPC analysis in CPC-NT can identify the following anomalies quickly and easily:

- Degradation in equipment efficiency, by visualizing long-term trends in run durations for specific recipes or groups of recipes.
- Changes in process parameters, by visualizing long-term trends in segment durations and run durations for specific parts.
- Equipment utilization for a specific time period, including "door open" times.
- Custom data tracking, including motor utilization, alarm counts, and other user-specified data.

2. Run Database

The Run Database is the source of all data used by the SPC Analysis module. In order to utilize SPC you will first need to be familiar with the structure and data contained in the Run Database.

Run Database defined

The run database is maintained on each CPC system and contains generalized information and data from every run that the system has controlled.

Where is the database?

The database file is located on the hard-drive of the system, usually in the Data folder. The file is normally called RunDB.MDB or RunDatabase.mdb. *Note: The location can be changed via the Object Editor in the object labeled "RecipeProcessor" and "DataAnalyze".*

Database type

The default run database that CPC-NT creates is a Microsoft Access database. The version of the database is Access-97 (for CPC-NT versions before March-2002) and Access 2000-2002 for recent CPC-NT versions dated.

Changing the database version in Access

Because the older Run Database (created before March-2002) is an Access-97 version, if you load the Run Database file into Access 2000/2002/XP, the system will ask whether you want to convert the database to the newest version. We recommend that you convert the database to the 2000/2002/XP version **only if** you also have the newest CPC-NT upgrade installation (Dated March-2002 or later).

*** Please do not convert the database if you haven't installed the newest CPC-NT software.*

Database contents

Tables

The database includes a number of tables that are used to organize the data. Those tables are:

- Attachments – Contains information on part attachments
- Equipment – Contains records that identify each piece of controlled equipment
- Messages – Contains records for each message that is generated during a run
- Notes – Contains information on engineering and QA notes
- Parts – Contains information on parts, part fields, and attachment quantities
- QCStatus – Contains information on pass/fail for systems that utilize quality cards.
- Recipes – Contains records which identify every recipe (name, version, etc.) used during a run.
- Runs – Contains an individual record for every run initiated by the system, including fields for starttime, endtime, duration, recipe, etc.

- SPCVariables – Custom set of variables that can be configured to track any system element. *Note: This is only utilized if customer requests a specific tracking.*
- Tools – Contains records which identify each and every tool used during runs. *Note: This is only utilized on systems that use the Tool Database feature.*

No raw data

Unlike each individual CPC-NT datafile, the run database does not include interval data. Though this exclusion presents some drawbacks with regard to long-term analysis, it does however allow the database to be small and easily and quickly queried.

Using database tables

Relational structure

The Run Database is considered a relational database. This term indicates that records in different database tables are connected by linked fields. For example, the Runs table has a field called RecipeID which corresponds to a similar field in the Recipes table. The individual records in each table that have the same RecipeID value are considered “linked”.

Linking tables with key fields

In a relational database, the different tables in the database can be linked together by certain “Key” fields. By using these key fields we can link multiple tables together so that the collective “joined tables” can be queried and organized as if they are one table.

The linking, or “joining”, of these tables is a required practice when you generate database queries.

Database fields

The following section defines the fields contained in the individual tables:

Attachments

This table includes a record for each part attachment used for each run on each piece of equipment controlled by the system. Attachments typically include part thermocouples (ie. PTC), vacuum probes (ie. MON or VPRB), and vacuum sources (ie. SUP or VSRC).

The following fields are provided:

AttachmentID This numeric field is the “key field” used to identify each separate record.

EquipmentID This numeric field is the link field that indicates the matching record in Equipment table.

RunID This numeric field is the link field that indicates the matching record in Runs table.

PartID This numeric field is the link field that indicates the matching record in Parts table.

Name This is the name of the attachment (ie. PTC5)

Type This field indicates the type of attachment (ie. PartTC, Probe, Source, etc.)

Action This field indicates the action for the attachment at the beginning of the run (ie. VAC, VENT). *This is used only for vacuum source attachments.*

Equipment

This table includes a record for each piece of equipment being controlled.

The following fields are provided:

<i>EquipmentID</i>	This numeric field is the “key field” used to identify each separate record.
<i>Name</i>	This is the name of the general name of the equipment (ie. Autoclave). <i>On some systems the Description is the field that actually describes the equipment.</i>
<i>Description</i>	This is the description of the equipment (ie. Autoclave AC2). This field is the better field to use for searching for a specific piece of equipment.
<i>Type</i>	This field is sometimes used to indicate the type of equipment (ie. Autoclave, Oven, Press, etc.)
<i>Manufacturer</i>	This field is sometimes used to indicate the manufacturer of the equipment.

Messages

This table includes a record for each segment change message that occurs during a run.

The following fields are provided:

<i>MessageID</i>	This numeric field is the “key field” used to identify each separate record.
<i>PartID</i>	This numeric field is the link field that indicates the matching record in Parts table.
<i>ToolID</i>	This numeric field is the link field that indicates the matching record in Tools table.
<i>QCStatusID</i>	This numeric field is the link field that indicates the matching record in QCStatus table.
<i>UserName</i>	This indicates the user name of the message. This is usually “Auto”.
<i>Date</i>	This is the date & time that the message was logged.
<i>MessageType</i>	This is the type of message issued (ie. System, Alarm, Manual, etc.)
<i>MessageText</i>	This is the actual message text displayed during the run.

Notes

This table includes a record for each note added manually or automatically during data-analysis operations in the reporting screen.

The following fields are provided:

<i>NoteID</i>	This numeric field is the “key field” used to identify each separate record.
<i>Date</i>	This is the date & time that the message was logged.
<i>Type</i>	This indicates the type of message (ie. NOTE, SYSTEMQA, etc.)
<i>SubType</i>	This indicates the status of QA notes (ie. PASS, FAIL)
<i>Author</i>	This indicates the author of the note (ie. SYSTEM, or username)
<i>Text</i>	This is the note text.
<i>RunID</i>	This numeric field is the link field that indicates the matching record in Runs table.

<i>PartID</i>	This numeric field is the link field that indicates the matching record in Parts table.
<i>ToolID</i>	This numeric field is the link field that indicates the matching record in Tools table.
<i>AttachmentID</i>	This numeric field is the link field that indicates the matching record in Attachments table.

Parts

This table includes a record for each part processed in a run. The parts table includes all part fields as well as leak test information and index information.

The following fields are provided:

<i>PartID</i>	This numeric field is the “key field” used to identify each separate record.
<i>RunID</i>	This numeric field is the link field that indicates the matching record in Runs table.
<i>EquipmentID</i>	This numeric field is the link field that indicates the matching record in Equipment table.
<i>PartIndex</i>	This is the index number of the part in the run’s batch. For example, the first part would have an index = 1.
<i>ParentPartIndex</i>	This is the index number of the part’s parent part or tool. This is used only for Part-on-tool or Multi-part applications.
<i>LeakValue</i>	This indicates the amount (ie. “Hg) that the part leaked during pre-run leak test.
<i>LeakRequirement</i>	This indicates the leak test requirement that should have been achieved.
<i>RecordID</i>	This is the numeric field that links this part to the record in the PartDatabase Parts table (used during part entry).
<i>TemplateName</i>	This is a part field that indicates the entry template used during part entry.
<i>RecordName</i>	This is a part field used to identify the part record.
<i>Other fields...</i>	Depending on the system, there can be many different fields including PartNumber, SerialNumber, PartName, Description, etc.

QCStatus

This table includes a record for each part that has been electronically qualified by the quality card system.

The following fields are provided:

<i>QCStatusID</i>	This numeric field is the “key field” used to identify each separate record.
<i>RunID</i>	This numeric field is the link field that indicates the matching record in the Runs table.
<i>EquipmentID</i>	This numeric field is the link field that indicates the matching record in the Equipment table.
<i>PartID</i>	This numeric field is the link field that indicates the matching record in the Parts table.

<i>ToolID</i>	This numeric field is the link field that indicates the matching record in the Tools table.
<i>Status</i>	This indicates the PASS or FAIL status of the automatic QA buyoff for the specified part.
<i>UserName</i>	This identifies the user that added the QCStatus record.

Runs

This table includes a single record for each run that occurs on the equipment.

The following fields are provided:

<i>RunID</i>	This numeric field is the “key field” used to identify each separate record.
<i>EquipmentID</i>	This numeric field is the link field that indicates the matching record in the Equipment table.
<i>RecipeID</i>	This numeric field is the link field that indicates the matching record in the Recipe table. This linked record will identify the recipe used for the run.
<i>StartTime</i>	This date/time field indicates the start of the run.
<i>EndTime</i>	This date/time field indicates the end of the run.
<i>Duration</i>	This numeric field identifies the duration of the run in Minutes.
<i>Datafile</i>	This field identifies the datafile path/name that was generated during the run.
<i>PartQty</i>	This indicates the number of parts that were processed in the run.
<i>PartTCQty</i>	This indicates the number of part thermocouples that were utilized in the run.
<i>ProbeQty</i>	This indicates the number of vacuum probes that were utilized in the run.
<i>QCComplete</i>	This True/False field indicates whether the part data for the run was inspected by the automatic quality system.

Recipes

This table includes a single record for each recipe that was utilized on the system.

The following fields are provided:

<i>RecipeID</i>	This numeric field is the “key field” used to identify each separate record.
<i>EquipmentID</i>	This numeric field is the link field that indicates the matching record in the Equipment table.
<i>Name</i>	Recipe name, including folder (ie. Recipes\TestRecipe).
<i>Description</i>	Description of recipe.
<i>Comment</i>	Recipe comment
<i>Author</i>	Recipe author
<i>Specification</i>	Recipe specification
<i>Revision</i>	Recipe revision # or letter
<i>RevisionDate</i>	Recipe revision date.

ModificationDate Latest modification date for the recipe

TotalRuns Total runs using that recipe

LastRunDate Last date that the recipe was used.

SPCVariables

This table includes a single record for each run that occurs. This table is only used if an accompanying SPCVariables object is configured in CPC-NT.

The following fields are provided:

VariableID This numeric field is the “key field” used to identify each separate record.

RunID This numeric field is the link field that indicates the matching record in the Runs table.

EquipmentID This numeric field is the link field that indicates the matching record in the Equipment table.

Other fields... A user-configured list of variables that can track motor hours, events, and counting of specific conditions.

Tools

This table includes a single record for each tool that is used in the run. This table is only updated if the system is configured to utilize the ToolsTable in the PartDatabase.

The following fields are provided:

ToolID This numeric field is the “key field” used to identify each separate record.

RunID This numeric field is the link field that indicates the matching record in the Runs table.

EquipmentID This numeric field is the link field that indicates the matching record in the Equipment table.

PartID This numeric field is the link field that indicates the matching record in the Parts table.

Other fields.. Each tool related field defined in the part database will be displayed in this record. Those fields can include ToolName, ToolNumber, CycleCount, MaxCycles, etc.

Variables

This table includes a single record for each run that occurs. This table is only used to track and record specific variables, including LoadNumber, OperatorName, and any other system-specific information. The table mirrors the Variables object in the Object Editor.

The following fields are provided:

- | | |
|------------------------|---|
| <i>VariableID</i> | This numeric field is the “key field” used to identify each separate record. |
| <i>RunID</i> | This numeric field is the link field that indicates the matching record in the Runs table. |
| <i>EquipmentID</i> | This numeric field is the link field that indicates the matching record in the Equipment table. |
| <i>Other fields...</i> | A list of variables maintained by the system in the Variables object. The list usually includes <u>LoadNumber</u> and other status information. |

3. Programming queries

A query is a user-defined search string that defines what information to extract from a database. Queries are the main method used to organize database information into a table of useful data.

SQL language

SQL, or Structured Query Language, is the language used to define a database query. The language includes a number of reserved keywords which may be linked together to create a query.

Field identification

In queries the database table fields are defined by using the following syntax:

Table.Field

For example, the name of a recipe would be identified as :

Recipes.Name

SELECT and FROM statement

In a query the SELECT and FROM statement defines the list of fields that will be “selected” and the table to select them from.

For example, the following query select the start time, end time, and duration from the Runs table.

SELECT Runs.StartTime, Runs.EndTime, Runs.Duration FROM Runs;

You will notice that the reserved words are capitalized – this is not required, but is good practice. You will also notice that the query ends with a semicolon – this is required.

WHERE statement

The WHERE statement defines a search criteria that will be used to extract a subset of the table records. The WHERE clause can use operators such as >, <, =, or can utilize the keyword LIKE to define a wild card statement.

Example #1: Get all datafiles where the run run duration was greater than 2 hrs.

SELECT Runs.Datafile FROM Runs WHERE Runs.Duration > 120;

Example #2: Get all datafiles where between 1/1/2002 and 2/1/2002. Notice that when using date literals, you must encase the date in the # symbol.

**SELECT Runs.Datafile FROM Runs WHERE Runs.StartTime > #1/1/2001# AND
Runs.EndTime < #2/1/2001#;**

Example #3: Get all datafiles where the datafile name has “CLAVE” is in the file name. Notice that this example uses apostrophe marks ‘*clave*’ to enclose a string. It also uses an asterisk * to define a wild card.

SELECT Runs.Datafile FROM Runs WHERE Runs.Datafile LIKE ‘*CLAVE*’;

INNER JOIN and ON statements

The JOIN statement is used to join multiple tables into one table by matching link fields in the tables. INNER JOIN is the most common join, and will return a mix of both tables where the ON field is equivalent.

Example #1: Display all part numbers for parts processed during the day 2/3/02:

```
SELECT Parts.PartNumber FROM Parts INNER JOIN Runs ON Parts.RunID =  
Runs.RunID WHERE Runs.StartTime > #2/3/02# AND Runs.StartTime < #2/4/02#;
```

In this example the Runs and Parts tables are joined on a matching link field RunID. Once linked, the query only returns the records where the run started on 2/3/02.

Example #2: Display part number, recipe name, and run duration for all runs in January, 2002:

```
SELECT Runs.StartTime, Recipes.Name, Parts.PartNumber FROM (Parts INNER JOIN  
Runs ON Parts.RunID = Runs.RunID) INNER JOIN Recipes ON Runs.RecipeID =  
Recipes.RecipeID WHERE Runs.StartTime > #12/31/01# AND Runs.StartTime <  
#1/31/02#;
```

In this example uses a nested INNER JOIN to link three tables. The method is to link two tables first, and then link that linked table to the third. In the example the Parts table is linked to Runs, and then that linked table (Parts-Runs) is linked to Recipes. The link for the Recipe is using RecipeID, not RunID, because there are multiple runs possible using the same recipe.

ORDER BY statement

The ORDER BY statement is used to sort the returned records in the query. The ordering can be ascending (default) or descending.

Example #1: Display all part numbers for parts processed during the day 2/3/02:

```
SELECT Parts.PartNumber FROM Parts INNER JOIN Runs ON Parts.RunID =  
Runs.RunID WHERE Runs.StartTime > #2/3/02# AND Runs.StartTime < #2/4/02#;
```

Using variables in queries

By using variables, you can program a custom query which will automatically request an entry from the operator on activation. This provides a much more flexible query.

Example #1: List all parts between two operator entered dates:

```
SELECT Parts.PartNumber FROM Parts INNER JOIN Runs ON Parts.RunID =  
Runs.RunID WHERE Runs.StartTime > CDATE(EnterStartTime) AND Runs.StartTime <  
CDATE(EnterEndTime);
```

Because the name "EnterStartTime" and "EnterEndTime" do not exist in the current query set, the system will assume that those are variables that the operator must enter. The CDATE command is used to convert the variable string into a date variable.

Using FORMAT and AS keyword

You can utilize the FORMAT keyword to change the way data is presented. When manipulating the data, the keyword AS must also be used to assign a name to the new modified field.

Example #1: List the days for all runs:

```
SELECT FORMAT(Runs.StartTime, "MM:DD:YY") AS StartDate FROM Runs;
```

This will display only the month, day, and year (not the time) of the runs. Format can also be used to format data.

Example #2: List run durations for all runs (no decimal points):

```
SELECT FORMAT(Runs.Duration, "###") AS RunDur FROM Runs;
```

Using DISTINCT keyword

You can list distinct records by using the DISTINCT keyword. This is useful if you don't want duplicate records displayed.

Example #1: List the days for all runs, but only list one record for each day, even if multiple runs occurred during that day:

```
SELECT DISTINCT FORMAT(Runs.StartTime, "MM:DD:YY") AS StartDate FROM Runs;
```

Using conversion commands

The following commands are useful in order to convert variables and fields into the proper data type. Each utilizes a parantheses (ie. VAL(Variables.LoadNumber))

VAL Get value of field (numeric or string).

CINT Convert to integer

CSNG Convert to single-precision number

CDBL Convert to double-precision number

CLNG Convert to long integer

CSTR Convert to string

CDATE Convert to date

Where are queries stored?

Queries are maintained in the database file. This storage location allows queries to be created via either CPC-NT or by using Microsoft Access.

Creating queries in Access

Any query created in Microsoft Access will be automatically available to you in CPC-NT. ASC recommends the use of Access for query programming for the novice query programmer. Access has a number of nice features, including a query wizard, which provide a simple method to develop complex queries.

4. CPC Query Builder

The following chapter describes the method used to build custom database queries in CPC.

How to access the SPC reporting screen

- 1.) From the CPC-NT main screen click the “Reporting” button to access the reporting screen.
- 2.) From the reporting screen click the “SPC Analysis” button to access the SPC Reporting screen.

The screenshot shows the 'SPC Reporting Screen' interface. It features a 'Query Selections' dropdown menu with 'All Runs' selected. Below it is a 'Selected Query' text area containing the following SQL query:

```
SELECT Runs.Duration  
FROM Runs  
WHERE Runs.StartTime > #6/1/01# and Runs.StartTime <  
#6/30/01#
```

 To the right of the query area are buttons for 'Add', 'Update', and 'Delete'. Further right are 'SQL Operators' (SELECT, DISTINCT, FROM, WHERE, ON, INNER JO, OUTER JO, ORDER B) and 'Tables in Database' (Attachments, Equipment, Messages, Parts, QCStatus, Recipes, Runs). A 'Fields in Table' list shows AttachmentID, EquipmentID, RunID, PartID, Name, Type, and Action. The 'Do Query' button is at the top left, and 'Export Data' is below it. A 'Record Count' field is on the left. The 'Query Results' area is a large empty grey box. At the bottom, there is a status bar with 'ClientPC: MASONLAPTOP2 | ServerPC: MASONLAPTOP2 | User: [blank] Back'.

How to select a query

All pre-programmed queries are available by clicking the combo button (arrow) below the label “Query Selections”.

When you click the combo selection, a drop-down listing will show all programmed queries.

How to perform the query

Having selected the query from the combo box, click the “Do Query” button to activate the query. If the query is valid, then the desired data will be shown in the “Query Results” spreadsheet.

For example, the following query resulted in the following eight (8) records.

SPC Reporting Screen

Query Selections: TimeSurvey | Report Type: Spreadsheet | Report | Add | Update | Delete

SQL Operators: SELECT, DISTINCT, FROM, WHERE, ON, INNER JO, OUTER JO, ORDER B

Tables in Database: Attachments, Equipment, Messages, Notes, Parts, OCStatus, Recipes, Runs

Fields in Table: AttachmentID, EquipmentID, RunID, PartID, Name, Type, Action

Do Query | Export Data | Record Count: 8

Selected Query:
SELECT Runs.StartTime, Runs.EndTime, Runs.Duration
FROM Runs
WHERE Runs.StartTime > #8/1/01# and Runs.StartTime <
#8/30/01#

Query Results

#	StartTime	EndTime	Duration
1	8/16/2001 2:59:14 PM	8/16/2001 4:23:28 PM	84.23333
2	8/17/2001 11:57:46 AM	8/17/2001 11:58:56 AM	1.166667
3	8/17/2001 12:00:53 PM	8/17/2001 12:15:58 PM	15.08333
4	8/17/2001 2:36:41 PM	8/17/2001 8:25:59 PM	349.3
5	8/19/2001 2:03:02 PM	8/19/2001 2:22:19 PM	19.28333
6	8/19/2001 2:32:02 PM	8/19/2001 4:41:39 PM	129.6167
7	8/19/2001 4:53:08 PM	8/19/2001 4:53:15 PM	0.1166667
8	8/19/2001 5:13:20 PM	8/19/2001 5:13:52 PM	0.5333334

ClientPC: MASONLAPTOP2 | ServerPC: MASONLAPTOP2 | User: Back

How to modify a query

- 1.) Select the query from the combo box
- 2.) Edit the query in the “Selected Query” text area. Once you change the text the button titled “Update” will become enabled.
- 3.) To accept the changes, press the “Update” button.

How to delete a query

- 1.) Select the query from the combo box
- 2.) Click the “Delete” button. Confirm the acknowledgement.

How to add a new query

- 1.) Formulate a query name. We suggest that you name the query with a text string identifying the purpose of the query (ie. List All Parts in January).

- 2.) Enter the name in the “Query Selections” combo box.
- 3.) Enter the query text in the “Selected Query” text area.
- 4.) Click the “Add” button to accept the new query.

Quick-entry of statements, table names, and fields

When programming the query you can utilize the list boxes on the right-side of the screen to enter fields, tables, and statements.

By click a specific table, the system will update the fields listing for that specified table. By double-clicking a desired field, the system will automatically enter the field (ie. Parts.PartNumber) in the current query at the current cursor position.

ASC recommends the use of the quick-entry lists for all novice programmers.

Viewing a report of the resulting data

You can view and print the resulting data in spread-sheet form by the following procedure:

- 1.) Select a query
- 2.) Run the query via “Do Query”
- 3.) Click the “Report Type” combo box and select “Spreadsheet” from the listing.
- 4.) Click the “Report” button to view/print the report .

5. SPC Plotting and Reporting

The following chapter describes some SPC related plotting, bar graphs, and time surveys that are available. Please contact ASC for details on charts or graphs that may not be immediately supported.

Time survey report

A useful report that is provided in CPC-NT is an equipment utilization report. This report provides the equipment utilization between two dates, including the door-open time and the total running time.

- 1.) Create a time survey query called "Equipment Utilization":

```
SELECT Runs.StartTime, Runs.EndTime, Runs.Duration FROM Runs WHERE Runs.StartTime > CDATE(StartDate) and Runs.StartTime < CDATE(EndDate);
```
- 3.) Click the "Report Type" combo box and select "Time Survey" from the listing.
- 4.) Click "Report" button to show the report.
- 5.) The following report shows a Time Survey from 8/1/01 to 9/1/01 for a piece of equipment.

The screenshot displays the "SPC Reporting Screen" interface. At the top, there are controls for "View Page #" (set to 1) and "Copies" (set to 1), along with "Print", "Save", and "Load" buttons. The main content area is titled "Database Report - Time Survey" and contains the following information:

Date:	Thursday, April 18, 2002	Time:	4:55:05 PM
Query:	TimeSurvey		
Comment:			
StartDate:	8/1/01		
EndDate:	9/1/01		

Summary statistics:

- Total records = 8
- Total runs = 8
- First run = 8/16/2001 2:59:14 PM
- Last run = 8/19/2001 5:13:52 PM
- Total time = 74H 14M 38S
- Total running time = 9H 59M 20S
- Total dormant time = 64H 15M 18S
- Utilization = 13.45%

The bottom status bar shows "ClientPC: MASONLAPTOP2 | ServerPC: MASONLAPTOP2 | User: Back".

Duration reports

The duration report can be used for specific runs that have the same recipe. As long as the first field is a date, the system will plot by date. If the first field is Duration, then the system will plot by record index #.

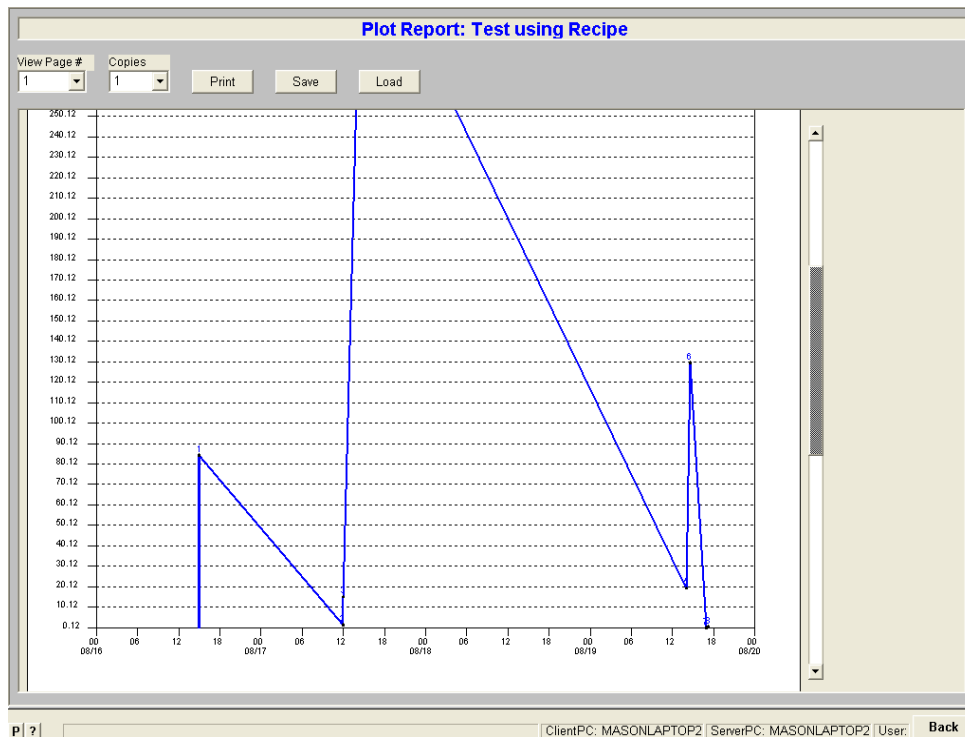
Duration of Run by Recipe

This query and report plots the run durations for a set of runs. This report can quickly identify degradation in heating capabilities, or changes in process parameters.

- 1.) Create a Duration query called "Duration of Run by Recipe":

```
SELECT Runs.StartTime, Runs.Duration FROM Runs INNER JOIN Recipes ON  
Runs.RecipeID = Recipes.RecipeID WHERE Runs.StartTime > CDATE(StartDate)  
and Runs.StartTime < CDATE(EndDate) and Recipes.RecipeName LIKE  
RecipeSearch;
```

- 2.) Click the "Do Query" button.
- 3.) Enter the StartDate, EndDate, and the RecipeSearch. RecipeSearch can be a recipe name (ie. "Recipes\Uniformity") or a partial recipe name with wild cards (ie. "*Unif*")
- 4.) Select "Plot" from the "ReportType" combo box.
- 5.) The plot shown below will plot the durations of all of the runs for that recipe on a time/date scale.



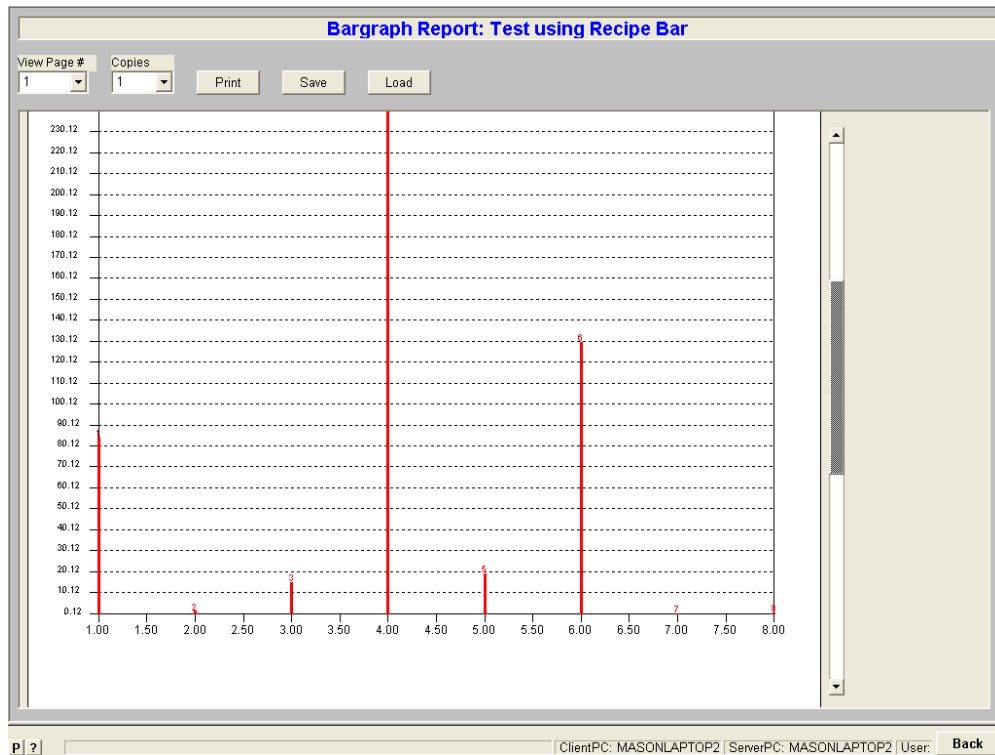
Duration of Run by Recipe bar chart

This query and report shows bar graphs for each duration, plotted on an index scale (not time).

- 6.) Create a Duration query called "Duration of Run by Recipe Bar":

```
SELECT Runs.Duration FROM Runs INNER JOIN Recipes ON Runs.RecipeID =  
Recipes.RecipeID WHERE Runs.StartTime > CDATE(StartDate) and  
Runs.StartTime < CDATE(EndDate) and Recipes.RecipeName LIKE  
RecipeSearch;
```

- 7.) Click the "Do Query" button.
- 8.) Enter the StartDate, EndDate, and the RecipeSearch. RecipeSearch can be a recipe name (ie. "Recipes\Uniformity") or a partial recipe name with wild cards (ie. "*Unif*")
- 9.) Select "Bar" from the "ReportType" combo box.



Other SPC reports

With the tools provided you should be able to create custom queries of specific data - by part, by recipe, or by run. Once the data is collected in a table, the spread-sheet, plot, or bar report can provide a useful graphical view of trends of that data.

ASC requests

Please contact ASC if the tools provided are not sufficient for the type of SPC analysis you would like to achieve.