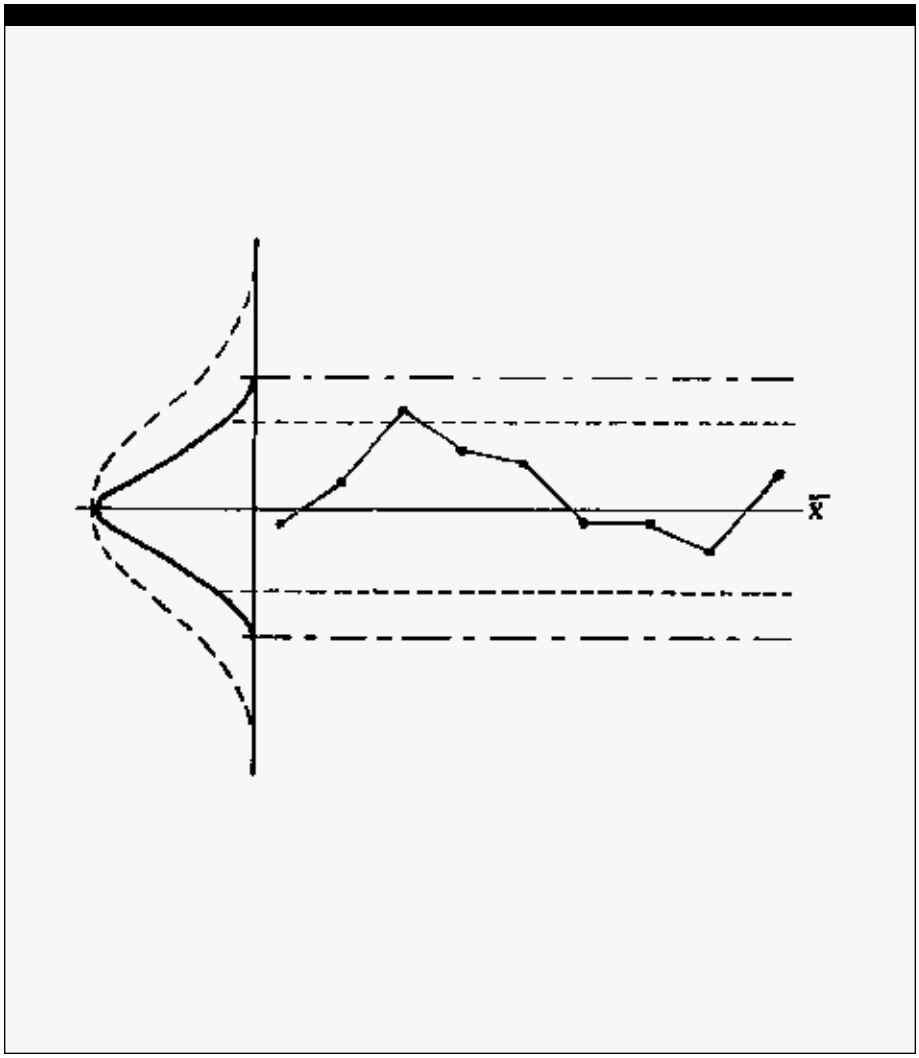
CRISP®/32

SPC

User's

Guide

SQUARE D COMPANY
CRISP AUTOMATION SYSTEMS



SPC User's Guide

CRISP® is a registered trademark of Square D Company

I/ONYX® is a registered trademark of Square D Company

Dedicated to Growth
Committed to Quality

SPC User's Guide

Square D Company
5160 Paul G. Blazer Memorial Parkway
Dublin, Ohio 43017
USA

(614) 764-4200

Dedicated to Growth
Committed to Quality

SPC

User's

Guide

SQUARE D COMPANY

CRISP AUTOMATION SYSTEMS

SPC
User'sGuide
Document number: 500 057 - 001, Rev. 1

Document History

Revision	Date	Pages affected/Description of change
1	6/93	Initial Release. ECN 4316

Software Version SPC Rev. 3.2 and Later

Table of Contents

Opcode Definitions (cont)

Opcode Definitions (cont)

Opcode Definitions (cont)

Introduction

General

This document describes the functionality and user interface for the CRISP® Statistical Process Control (SPC) product.

This manual is divided into the following sections.

Section	Description
Introduction *(page 1)*	Contains general information concerning the basic operation of the SPC software.
Configuration *(page 5)*	Describes the procedures for integrating the SPC product into a complete application.
Operation *(page 11)*	Describes how the software product performs the various SPC functions.
Opcode Definitions *(page 19)*	Defines the available opcodes in alphabetical order.
Token Definitions *(page 81)*	Describes the keywords that may be used in configuring the product.
Status/Error Messages *(page 83)*	Describes the potential messages returned by the product.
Glossary *(page 85)*	Describes technical terms, acronyms and mnemonics used in this document.
Appendix A *(page A-1)*	Archive File Definitions
Appendix B *(page B-1)*	Control Limit Calculation Constants
Appendix C *(page C-1)*	Sample UCF

Design Philosophy

CRISP/SPC is a software product designed to perform Statistical Process Control (SPC) calculations on data contained in one or more CRISP® databases and return the results of these calculations to a database or display on one or more workstations. The intent of this product is to provide a very flexible add-on application to a CRISP® system that is both real-time and "industrial strength" in terms of the amount of data analyses performed.

CRISP/SPC also provides many functions associated with Statistical Quality Control (SQC). These include automatic calculation of control limits and historical data analysis.

CRISP/SPC is configured by the Application Developer to interface with application CRISP database(s) through a User Configuration File (UCF). This text file can be created and edited using any ASCII-based text editor. On product startup, the contents of the UCF is read into memory and provides the program with detailed instructions for product execution.

Product Features

Some of the more important features of this product include the following.

- Supports major SPC chart types - CRISP/SPC provides support for building each of the popular chart types.
  - Control Charts
  - Histograms
  - CUSUM Charts
  - Pareto Charts
- Supports major SPC Analysis types - the analysis performed for each of these charts may be done using either of three data types.
  - Variable Data
  - Attribute Data (Defectives)
  - Attribute Data (Defects)
- Alarm detection functionality - both the raw data and the resulting calculations are checked against several user-defined limits in order to provide an alert to the user that a problem may exist on a particular SPC chart. This feature eliminates the need for manually reviewing each chart and allows the user to focus only on the exception conditions.
- Historical data archiving and retrieval - all data sampled by CRISP/SPC is written to disk storage for later recall. A user may view data for any past chart by simply entering a start date and requesting the data.

Control Chart Types

Depending on the type of data being collected and the size of the subgroups being used, CRISP/SPC will perform different calculations to support specific Control charts namely the following.

Variable Data

- Mean and Moving Range Control Charts (Subgroup size equal to one).
- Mean and Range Control Charts (Subgroup size greater than one).
- Mean and Standard Deviation Control Charts (Subgroup size greater than one).

Attribute Data (Defectives)

- np Control Charts (Number Defective, fixed lot size).
- p Control Charts (Proportion Defective, varying lot size).

Attribute Data (Defects)

- c Control Charts (Number of Defects, fixed lot size)
- u Control Charts (Number of Defects per Unit, varying lot size)

Product Model

Figure 1 shows the five software components that make up the CRISP SPC product. They include the following components.

- SPC Control Data (SCD) Area - This is a region of global memory assessable to each of the SPC processes that contains all of the essential data for the last 30 subgroups. This includes the results of statistical calculations, alarm limits, etc.
- SPC Sampler - This process is responsible for retrieving data samples from one or more CRISP databases and updating the SCD area.
- SPC Archiver - This process takes data periodically from the SCD area and writes it to disk for later recall.
- SPC Presenter - These process(es) recall data from the SCD area and disk storage, perform SPC calculations and write the results to one or more CRISP databases for display.
- SPC Controller - This supervisory process is responsible for coordinating all communications between the other processes and detection/reporting of SPC alarms.

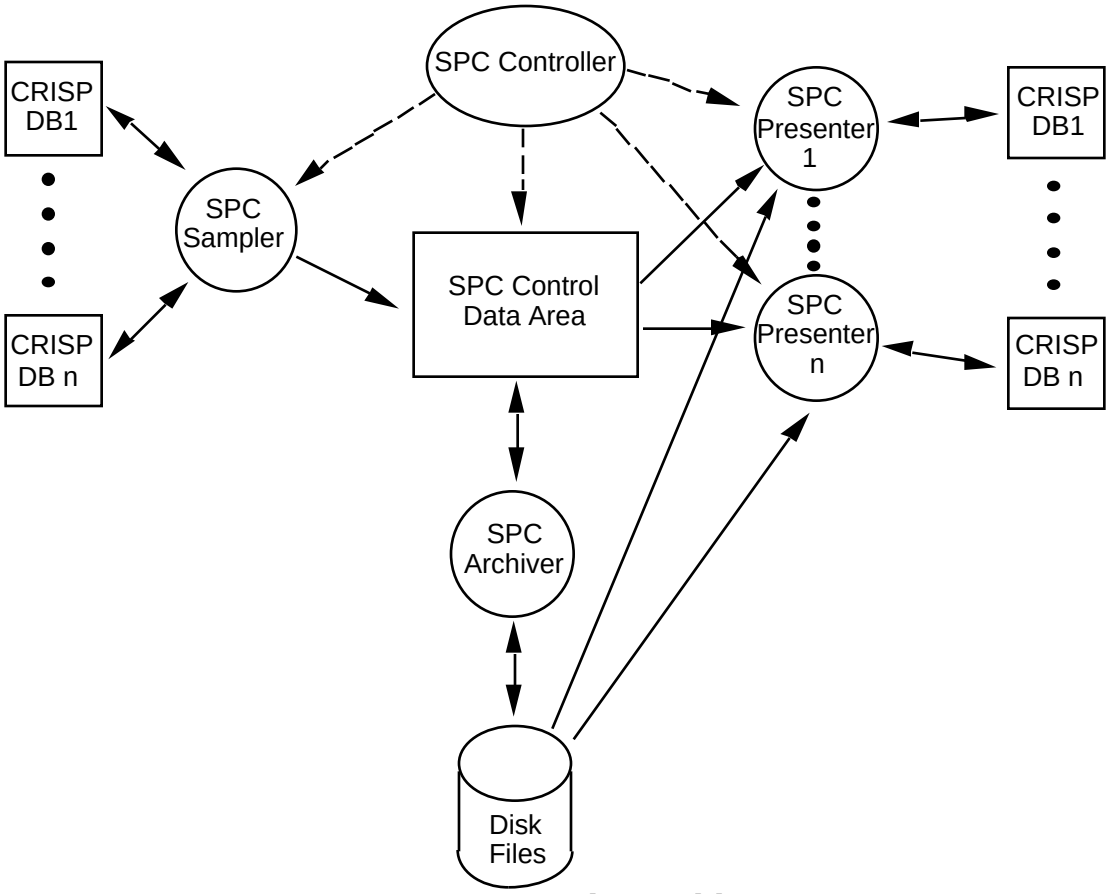


Figure 1. SPC Product Model

General

CRISP/SPC software is installed on a VAX/VMS system using Digital Equipment Corporation's standard VMSINSTAL utility. This process will load the product software and its associated data in the directory [CRISP.SPC]. Refer to the SPC Installation Guide for more detailed installation information. Please read this entire section before proceeding with the installation.

Product Configuration

Once CRISP/SPC has been installed, it must be configured for the particular environment. This is accomplished by running the following command procedure.

```
$ @CRISP$:USER_CONFIG_SPC  <ret>
```

This command procedure initially looks for the installed product files and if unable to locate them, it will prompt you for the disk drive ID where they are located.

The procedure will now ask a series of configuration questions. A suggested default answer is provided in the **[ ]** brackets. If the product has been configured before, then the defaults will be the answers entered previously. Enter a **"?"** to receive additional explanation about a particular item. If you wish to exit early from the config process, press **<CTRL> C** to abort. The questions are as follows.

1. `Name of Configuration file (UCF)? -` This file does not need to exist currently in order to complete this procedure. Remember to enter the full file specification, for example.

   ```
   DISK$USER:[CRISP.SPC]YOURPROJECT.UCF
   ```

2. `Type of data archive? -` The choices are either SPC (single daily file) or HPF (CRISP/Historian Point file format). The SPC format is generally more useful for smaller applications (less than 50 characteristics) and is the default. The HPF format uses the same high-capacity storage techniques contained in the CRISP/Historian product and allows ad-hoch queries to be performed using the CRISP application logic.

3. `Directory name for SPC data archive? -` The default directory is [CRISP.SPC]. You may redirect these data files to another disk or directory by entering a full path specification, for example.

   ```
   DISK$USER:[CRISP.SPC.DATA]
   ```

NOTE

HPF formatted data is always stored in the same general directory as the CRISP/Historian data files using different subdirectories and is not affected by this definition. Use the default when HPF format is selected.

Product Configuration (cont)

4. If you answered "SPC" for data archive then you will be asked to decide if Data file compression should be used. Compression is useful when large amounts of data will be collected using the COUNT mode of sampling. This feature will initiate a batch process that purges the archive files after a user-defined number of days. If you answer **YES** to this question, then you will also be prompted to supply a valid compression interval (1 to 30 days). The default interval is 7 days.

5. The next five questions relate to the tuning of the various SPC processes. The default values for scan interval (the frequency at that the processes wake up) are satisfactory for most applications. The minimum interval is 0.2 seconds.

   SPC Controller scan interval - 1 second
   SPC Initializer scan interval - 0.5 second
   SPC Archiver scan interval - 1 second
   SPC Sampler scan interval - 1 second
   SPC Presenter scan interval- 1 second

6. `Do you want SPC Presenter installed as sharable?` - The answer to this question depends on how many users are expected to access the SPC charts simultaneously. One VMS process is started for each presenter defined. If the answer is greater than one, then respond **YES** in order to conserve system resources.

7. `Number of presenters that will be started?` - This information is used for product memory tuning only and does not limit the number of presenter definitions in the UCF. The most common practice is to start with one Presenter during application development and expand the number after the logic has been debugged. A suggested guideline for the final configuration is one Presenter for each workstation.

NOTE

Remember to keep the product configuration and the UCF presenter definitions in agreement during application development by re-running the configuration process as more presenters are added.

Product Configuration (cont)

The configuration process completes by creating several startup command files including the following.

CRISP$CFG:USER_SETUP_SPC.COM - This file sets up VMS system logicals that configure the CRISP/SPC product. It also defines where error messages are directed and install/de-installs the Presenter process if requested.

CRISP$CFG:USER_START_SPC.COM - This file initiates the startup of the CRISP/SPC product. The CRSTART process looks for file names with the USER_START_* prefix and executes them after the databases and logics are in place. This process interprets the UCF and builds a section of global memory for Product execution.

NOTE

If CRISP/SPC is already running, the existing processes will be stopped and restarted by this command file.

CRISP$CFG:SPC_CONFIG.DAT - This file contains the answers supplied during product configuration. These answers will be recalled as defaults in the event that reconfiguration is required.

System Licensing

The CRISP/SPC product is licensed for single VAX node operation. The license information is read by the product at startup from the UCF defined during the configuration process. The information needed to complete the UCF comes in the form of a License Document that is shipped along with the product media and documentation.

To license CRISP/SPC type the license information into the UCF defined during configuration. It must be entered EXACTLY as shown on the license document, including all spaces, quotes and capitalization. Refer to the following example.

```
!
! YOURPROJECT.UCF
!
DEFINE>
CUSTOMERNAME>>              ;"CUST NAME"
CUSTOMERLOCATION>>          ;"CUST LOCATION"
CUSTOMERSWLICENSE>>         ;123456789
CUSTOMERUSELIMIT>>          ;1
CUSTOMERCONFIGCODE>>        ;1234567890
CUSTOMERCONFIGID>>          ;1234567890
```

User Configuration File Layout

The user-generated User Configuration File (UCF) will contain control parameters referred to as opcodes. These opcodes are converted into an optimized internal format at run time. The opcodes are used by the various processes that make up SPC to control the data sampling, maintenance, and presentation of SPC information.

In order to build an SPC UCF file, two concepts need to be introduced. They are as follows.

- SPC Characteristics
- SPC Presenters

SPC Characteristic - Each quantifiable measurement made of a finished part that the user wishes to perform SPC analysis is considered an SPC characteristic. In order to define a characteristic, several quantities need to be specified including the following.

- Raw data point location
- Sampling procedure
- Chart type
- Alarm Limits

By defining this information (plus several additional options) for each characteristic with opcodes in the UCF, the user permits SPC to automatically track characteristic performance. In short, SPC characteristic information defines how data flows into the SPC package and is analyzed.

SPC Presenter - Each user of the SPC charts has access to the data through a virtual display station. Each Presenter (or display station) is defined by opcodes that specify the location in the CRISP application database where the needed data is to be placed. This means that CRISP/SPC places the SPC data in the database and the user can incorporate the particular display capabilities of the various CRISP workstations to view the data. As a rule, one Presenter should be defined for each workstation configured with the system. This allows concurrent access to data by multiple users. Note that the opcodes included in each display station definition may be customized to access just the data needed by the user. For example, operators needing only control chart data and alarming would result in one display station definition. A process engineer needing all chart types, including histogram and pareto charts would require a second display station definition.

UCF Opcode Syntax

An opcode may begin at any location on the line as long as it is the first text item and is immediately followed by one or two right angle brackets. Most opcodes, with a few exceptions, allow up to two (usually optional) parameters. The first parameter will generally be an Application Data Base (ADB) variable name and the second parameter will usually be a default entry. Constant (keyword) definitions are supplied for default entries, where appropriate, in an attempt to reduce confusion. The general syntax of all SPC opcode statement is as follows.

```
OPCODE>> DB_NAME:NAME ; DEFAULT_VALUE !  COMMENT
```

UCF Opcode Syntax (cont)

Where OPCODE is one of the opcodes described in this manual; DB_NAME is the optional database identifier, always followed by a colon (":"); NAME is the name of the usually optional database variable used by the command, and DEFAULT_VALUE is the value used if the database variable is missing. DEFAULT_VALUE may be a literal numeric (integer or float), a pre-defined constant (keyword), or a quoted text-string, depending on the specific opcode used.

The single ">" indicates a continuously scanned statement, while ">>" indicates a statement that is executed only once at startup or restart. Unless otherwise noted, all SPC opcodes are executed only at startup, making the second right angle bracket optional. The exclamation point denotes the start of a comment. Anything to the right of the exclamation point is ignored. Imbedded comments are illegal and you may therefore start comments only to the right of any opcodes you wish to use. It is legal to have a line that contains only a comment. Further, all spaces and tabs, other than those enclosed in double quotes, as well as blank lines, will be ignored.

The ADB variable name will supply the runtime value for the associated opcode. Variable names should only be used where logically productive, that is, whenever the value supplied will need to change at runtime. There is a small amount of overhead involved with their use and therefore excessive or unproductive use of variables should be avoided.

The actual location of the variable in the ADB will be resolved once at startup or after a logical restart. If for some reason the variable cannot be resolved at this time, the process will use the default value, if supplied, or the system default value if no default value was supplied. The majority of opcodes have their own system defined default value. Usually the value need not be specified if the default value matches your system requirements. Any default entry supplied to the right of the semicolon is substituted for the system default value for the current opcode only. This means that subsequent uses of the same opcode will still have the original system default value.

The first opcode in every UCF must be the DEFINE> opcode. This opcode flags the start of the definition section. Any opcodes prior to the DEFINE> opcode are ignored. Until either the INIT> or the PROCESS> opcode is encountered, all statements in this section should relate to customer licensing information for the SPC product.

The INIT> opcode, that may optionally be declared immediately preceding the process section, indicates the beginning of the initialization section. Opcodes found in this section are considered to be initialization opcodes, and are processed only once at startup, or at restart. This section permits the user to define some opcodes that may otherwise be repeated throughout the UCF. The opcodes defined in the initialization section will become, in effect, the new defaults for that opcode. This new default is overridden for a particular SPC characteristic by defining that opcode in the corresponding SPC_CHARACTERISTIC> section.

UCF Opcode Syntax (cont)

The PROCESS> opcode, that flags the end of the initialization section and the beginning of the process section is also required. All following opcodes, until the END> opcode, are used to describe the statistical process control functions that are to take place.

The Process section contains the definitions of the SPC characteristics and the SPC display stations. An SPC characteristic begins with the SPC_CHARACTERISTIC> opcode and ends with the next occurrence of the SPC_CHARACTERISTIC> opcode or with the occurrence of the END> opcode. A SPC display station begins with the SPC_DISPLAY_STATION> opcode and ends with the next occurrence of the SPC_DISPLAY_STATION> opcode or with the occurrence of the END> opcode.

The END> opcode, that is required, signals the end of the UCF. All opcodes following the END> opcode will be ignored. Refer to the following summary.

- All opcodes end in ">"
- Not all opcodes support both parts of NAME;DEFAULT_VALUE and the ";" is required whenever either, or both are given
- All licensing information must follow DEFINE> and precede PROCESS>
- All initialization opcodes follow INIT> and precede PROCESS>
- Any text following an "!" on a line will be ignored
- All operations for a single characteristic must be grouped by characteristic, each following their respective SPC_CHARACTERISTIC> opcode
- Some opcodes propagate from SPC characteristic section to SPC characteristic section
- All operations for a single display station must be grouped by display station, each following their respective SPC_DISPLAY_STATION> opcode
- Some opcodes propagate from SPC display station section to SPC display station section
- Ordering of the opcodes within an SPC characteristic or display station section is not important

Alarms

CRISP SPC provides built-in support for numerous alarm conditions. These alarm conditions may be grouped in following two general types.

- Alarms based on user-defined limits (Threshold alarms)
- Alarms based on specific data patterns (Directional alarms)

Threshold Alarms

- X-Bar Control Limit Alarm - This alarm occurs whenever a single control point exceeds either the upper control limit (UCL) or lower control limit (LCL) established for a characteristic. The control limits are defined using 3.0 standard errors (sigma) either side of the process mean. The probability of a subgroup sample occurring randomly outside the control limits would be .13 percent (or 1 in 769) for a process under statistical control.

- RS Control Limit Alarm - This alarm occurs when a calculated Range or Standard Deviation value exceeds either the upper control limit (RSUCL) or lower control limit (RSLCL). These limits are statistically different from X-bar control limits since range and standard deviation values possess distributions that are positively skewed.

- X-Bar Warning Limit Alarm - The concept of a warning alarm is useful when process samples are collected infrequently. The usual action taken by the operator when a warning alarm occurs is to take another subgroup sample as soon as feasible. This is different from a Control limit alarm where some form of corrective action is usually warranted. The warning limits are calculated at 95 percent of UCL and LCL values using the following formula:

  UCL > Upper Warning Band > UCL - .05 * (UCL - LCL)
  LCL < Lower Warning Band < LCL + .05 * (UCL - LCL)

- The probability of a subgroup sample occurring randomly outside the control limits would be .21 percent (or 1 in 476) for a process under statistical control.

- RS Warning Limit Alarm - This alarm is similar to the X_Bar warning alarm in that the limits are automatically set at 95 percent of RSUCL and RSLCL.

RSUCL > Upper RS Warning Band > RSUCL - .05 * (RSUCL - RSLCL)
RSLCL < Lower RS Warning Band < RSLCL + .05 * (RSUCL - RSLCL)

- Cpk Limit Alarm - The Cpk index indicates both spread and setting of the process relative to the Upper Specification limits (USL) and Lower Specification limits (LSL). A Cpk alarm using a Cpk limit below 1.0 would indicate that the process is set off center or target.

Threshold Alarms (cont)

- CUSUM High Alarm - The CUSUM chart has an advantage over the control chart in that it is able to detect small but persistent changes more readily. CRISP/SPC uses the Two-sided Decision Interval procedure to compare accumulated sample changes to a limit interval (h). The CUSUM High alarm indicates that a series of samples greater than a user-defined target value (t) have exceeded a user-defined Limit value (h). Support for a user-defined Slack value (k) is available.

- CUSUM Low Alarm - The CUSUM Low alarm indicates that a series of samples less than a user-defined target value (t) have exceeded a user-defined Limit value (h). Support for a user-defined Slack value (k) is available.

- Validity Alarm - This alarm occurs when an operator is manually entering sample data. The purpose of the alarm is to detect data entry errors and prevent them from becoming permanently part of the control chart data. The SPC program compares the data sample against an Upper Validity limit (UVL) and a Lower Validity limit (LVL) and triggers an alarm for data outside these user-defined limits. An override operation is available if desired.

Directional Alarms

- Seven X-Bar points above or below the Grand Mean - This alarm is based on a statistically-significant pattern detected in the X-Bar chart data. A pattern of seven consecutive points above/below the mean is often indicative of the following.

  - New Material
  - New operator or new equipment
  - New setup or method

- Seven RS points above or below the RS Mean - This alarm is based on a statistically-significant pattern detected in the RS chart data. A pattern of seven consecutive points above the RS mean is similar in cause to the seven X-Bar points.

- Seven X-Bar points increasing or decreasing - This alarm is based on another statistically-significant pattern found in the X-Bar chart data. A pattern of 7 consecutive points sloping up or down is often indicative of the following.

  - Operator fatigue
  - Equipment/Tool wear
  - Environmental effects

- Seven RS points increasing or decreasing - This alarm is based on another statistically-significant pattern found in the RS chart data. A pattern of 7 consecutive points increasing/decreasing is similar in cause to the seven X-Bar points.

Control Limit Calculations

Control limits are usually derived from a statistical analysis of the historical sampled data. To automate this operation, CRISP/SPC provides opcodes that allow a user to request a recalculation of the X-Bar upper and lower control limits and the RS upper and lower control limits. The following is a summary of how these calculations are performed for each of the different chart types.

Variable X-Bar and Range Chart

$$XUCL = \bar{\bar{X}} + A2 * \bar{R}$$

$$XLCL = \bar{\bar{X}} - A2 * \bar{R}$$

$$RSUCL = D4 * \bar{R}$$

$$RSLCL = D3 * \bar{R}$$

Where: $\bar{\bar{X}}$ equals the calculated Grand mean (mean of means)

$\bar{R}$ equals the calculated range mean

A2 is a statistical constant for Normal (Gaussian) distributions
(a table of constants is contained in Appendix A)

D4 and D3 are statistical constants for positively skewed distributions
(see Appendix A).

Variable X-Bar and Standard Deviation Chart

$$XUCL = \bar{\bar{X}} + A3 * \bar{S}$$

$$XLCL = \bar{\bar{X}} - A3 * \bar{S}$$

$$RSUCL = B4 * \bar{S}$$

$$RSLCL = B3 * \bar{S}$$

Variable X-Bar and Standard Deviation Chart (cont)

Where: $\bar{\bar{X}}$ equals the calculated Grand mean (mean of means)

$\bar{S}$ equals the calculated standard deviation mean

A3 is a statistical constant for Normal (Gaussian) distributions
(a table of constants is contained in Appendix A)

B4 and B3 are statistical constants for positively skewed distributions
(see Appendix A).

Attribute p Control Chart

$$XUCL = \bar{p} + 3 * \frac{\sqrt{\bar{p} * (1 - \bar{p})}}{\sqrt{\bar{n}}}$$

$$XLCL = \bar{p} + 3 * \frac{\sqrt{\bar{p} * (1 - \bar{p})}}{\sqrt{\bar{n}}} \text{ (or zero if negative)}$$

Where: $\bar{p}$ equals the calculated average proportion defective

$\bar{n}$ equals the average sample size (variable).

Attribute np Control Chart

$$XUCL = n * p + 3 * \sqrt{n * p * (1 - p)}$$

$$XLCL = n * p - 3 * \sqrt{n * p * (1 - p)} \text{ (or zero if negative)}$$

Where: p equals the proportion defective

(1 - p) equals the proportion not defective

n equals the sample size (fixed).

Attribute u Control Chart

$$XUCL = \frac{\bar{u} + 3 * \sqrt{\bar{u}}}{\sqrt{\bar{n}}}$$

$$XLCL = \frac{\bar{u} - 3 * \sqrt{\bar{u}}}{\sqrt{\bar{n}}} \text{ (or zero if negative)}$$

Where: $\bar{u}$ equals the calculated average number of defects per unit

$\bar{n}$ equals the mean sample size (variable)

Attribute c Control Chart

$$XUCL = \bar{c} + 3 * \sqrt{\bar{c}}$$

$$XLCL = \bar{c} - 3 * \sqrt{\bar{c}} \text{ (or zero if negative)}$$

Where: $\bar{c}$ equals the calculated average number of defects

Data Archive

CRISP/SPC stores all information about each chart type in disk files for later recall. Any historical control chart will contain the full context (spec limits, control limits, etc.) that the raw sample data was originally taken. The date ranges on histograms and Pareto charts can be expanded or contracted to obtain different perspectives on the data. Because all charts are "on-line", the need to make printed copies of chart data is greatly reduced.

There is an opcode that can be used to disable the archiving function for particular characteristics if desired (see the DISABLE_ARCHIVING_IF> opcode.)

Another opcode can be used to clear the data history for a particular characteristic so that it may be reused for another analysis (see the CLEAR_CHARACTERISTIC_IF> opcode.)

Data Archive (cont)

The CRISP/SPC product provides two formats for archiving data to disk files. They are as follows.

- SPC - This format takes the internal SCD memory structure and extends it to a disk file format. One single file is created each day containing data from all characteristics for the day.

- HPF - This format takes the internal SCD memory structure and breaks it up into individual files (called Historian Point Files) that are compatible with the CRISP Historian product.

Appendix B provides a definition of the HPF files used by the CRISP/SPC product.

The choice of which format to use should be based on the following considerations.

1. Disk Capacity - The amount of disk memory consumed daily is a function of the total number of data samples taken and the period of historical data required. An application with 100 characteristics, sampled three times an hour, 16 hours a day, 7 days a week could consume about 45 MBytes of disk per month using the SPC format. The same data stored in HPF format would require approximately 20 MBytes.

2. Data Analysis - Data stored in HPF format is readily accessible to a number of other Third-party products, CRISP logic or custom C/Fortran programs. SPC formatted data is only recallable through the SPC Presenter processes. If data import/export is important then the HPF format should be selected.

Improving CPU Use

The following steps may be helpful in reducing CPU usage without loss of throughput or performance.

1. Avoid using the DISPLAY_WHILE> opcode - This opcode provides continuous updates to a fixed control chart. The realtime alarm function of CRISP/SPC permits operators to concentrate on their primary tasks and view control charts for individual characteristics only when there is cause to take action.

2. Judiciously use the ADB variable feature - Whenever a fixed value can be used, processing to resolve a variable is eliminated. The step of converting ADB variable references to fixed values is best taken following the initial phase of UCF creation, when variables may have been useful in development.

3. Define only the number of Characteristics really needed - Consider whether a realtime or historical trend analysis would substitute for certain types of measurements. One UCF can be defined for certification of new products and another for routine production.

4. Define only the number of Display Stations really needed - Consider whether screen exclusive access to SPC data is necessary and for how many users.

While there is no theoretical limit to the number of SPC characteristics or the number of virtual display stations, you should contact Square D / CRISP Automation for individual configurations with more than 100 characteristics or more than 10 display stations.

Notes:

ADB_IDENT>

Opcode which specifies the name or identity of the desired CRISP database. This opcode **must** be used prior to any CRISP variable reference, unless that opcode includes a CRISP database name followed by a colon. If the default entry is a string, then it must be enclosed in double quotes in order to avoid confusion with constants or absolute values.

Example: `ADB_IDENT> ;"TRAIN1"  !  Data Base name`

UCF Location: All sections
Variable Reference: No
Variable Type: n/a
System Default: None
Access: None
Propagation: Yes

ADB_LOCATION>

Opcode which supplies the first location in the CRISP Data Base to access for sampling. For characteristics with a subgroup size greater than one and a SAMPLE_TYPE> opcode defined as MANUAL, and a MANUAL_SUBTYPE> opcode defined as SUBGROUP, then the CRISP data element should be an array equal in size to the subgroup. The default entry should be a valid ADB variable name, enclosed in double quotes in order to avoid confusion with constants or absolute values.

Example: `ADB_LOCATION>  ;"SAMPLE(0)"  !  Starting variable`

UCF Location: Characteristic section
Variable Reference: No
Variable Type: Float, Numeric, Long
System Default: None
Access: None
Propagation: No

ADB_TYPE>

Opcode which specifies the Application Data Base (ADB) type. Currently only the CRISP ADB is supported. The only Legal database type is as follows.

Parameter	Constant
0	CRISP32

Example: `ADB_TYPE>  ;CRISP32  !  Database type`

UCF Location: All sections
Variable Reference: No
Variable Type: n/a
System Default: CRISP32
Access: None
Propagation: Yes

ALARM_CODE>

Opcode which defines the alarm code for a subgroup of sample values during manual data entry. This value would normally be used as a reference to a comment library in a CRISP application, which might indicate some irregularity about the subgroup of sample values. The primary purpose of collecting this data would be to prepare a Pareto chart of the Alarm types over time.

Example:
```
ALARM_CODE> ACODE;   !  Alarm code location
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: 0
Access: Read only
Propagation: No

CALC_CTRL_LIMITS_IF>

Opcode which defines the CRISP trigger variable that causes control limits to be recalculated. When the variable is nonzero (true), the upper and lower control limits for both the mean and range/standard deviation will be recalculated. New control limits will be written to the variables identified by the opcodes XUCL>, XLCL>, RSUCL>, and RSLCL> (if supplied with these opcode definitions). The opcode CTRL_CHART_TYPE> defines whether the calculated control limits written represent ranges or standard deviations of the subgroups. The opcode CALC_CTRL_LIMITS_TYPE> modifies the way control limits are calculated.

Example:
```
!  Recalc.  limits
CALC_CTRL_LIMITS_IF> RECALC_TRIG;
```

UCF Location: Characteristic or Initialization section
Variable Reference: Required
Variable Type: Logical
System Default: 0
Access: Read/Write
Propagation: Yes

CALC_CTRL_LIMITS_TYPE>

Opcode which defines the algorithm used to calculate the control limits. The value CURRENT indicates that the control limits are to be calculated immediately after the trigger is fired, using the memory resident subgroups (i.e. those subgroups currently residing in the control chart) as input to the control limit calculation. The value FUTURE indicates that the control limits are to be calculated after enough subgroups are acquired for the specified characteristic. The number of subgroups required for a FUTURE calculation is equal to the control chart size. When the value of FUTURE is selected, firing the trigger while subgroups are being collected for calculation has no effect.

The following is a summery of the choices for this opcode.

Parameter	Constant	Description
0	FUTURE	Use subgroups to be acquired
1	CURRENT	Use subgroups already acquired

Example:
```
!  Use current pts.
CALC_CTRL_LIMITS_TYPE> ;CURRENT
```

UCF Location: Characteristic or Initialization section
Variable Reference: Optional
Variable Type: Numeric, Long
System Default: CURRENT
Access: Read only
Propagation: Yes

CHARACTERISTIC_TYPE>

Opcode which determines the type of the characteristic that is being defined, VARIABLE or ATTRIBUTE. Refer to the Operations section of this manual for an explanation of these two characteristic types.

The following is a summery of the choices for this opcode.

Parameter	Constant	Description
0	VARIABLE	Variable characteristic
1	ATTRIBUTE	Attribute characteristic

Example:
```
!  Attribute char.
CHARACTERISTIC_TYPE> ;ATTRIBUTE
```

UCF Location: Characteristic section
Variable Reference: No
Variable Type: n/a
System Default: VARIABLE
Access: None
Propagation: Yes

CLEAR_CHARACTERISTIC_IF>

Opcode which allows the user to remove the memory resident subgroups (i.e. those subgroups currently residing in the control chart) for the specified characteristic. The subgroups to be cleared must have already been archived or had the archiving disabled while they were acquired.

Example:
```
!  Clear subgroups
CLEAR_CHARACTERISTIC_IF> RESET;
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: 0
Access: Read/Write
Propagation: No

COMMENT_CODE>

Opcode which defines the comment code for a sample value as part of manual data entry. This would normally be used as a reference to a comment library in a CRISP logic application, which might describe some irregularity about the sample. The primary purpose of collecting this data would be to report the results in the form of a Pareto chart over time.

Example:
```
COMMENT_CODE> CCODE;   !  Comment code location
```

UCF Location: Characteristic or Initialization section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: 0
Access: Read only
Propagation: No

CPK_ALARM_AT>

Opcode which defines the location in the ADB where the Cpk alarm status will be maintained. If the current Cpk value is less than the value specified by the CPK_LIMIT> opcode, a logical true will be written to the variable specified. If the Cpk value is greater than or equal to the value specified by the CPK_LIMIT> opcode, a logical false will be written to the variable specified. This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

Example:
```
!  Show Cpk alarm event here
CPK_ALARM_AT> CPK_ALM;
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

CPK_AT>

Opcode which defines the location in the ADB where the current Cpk value will be maintained. This data is continuously maintained by SPC, meaning that the Cpk value is available without referencing the characteristic at a display station.

Example: `CPK_AT> CPK_VALUE;  ! Show Cpk here`

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

CPK_LIMIT>

Opcode which defines the minimum Cpk value which will not cause an alarm. Any values lower than the specified limit will cause an alarm, which will be reported if the CPK_ALARM_AT> opcode has been specified for the characteristic.

Example: `CPK_LIMIT> ;0.97  ! Don't let Cpk fall below .97`

UCF Location: Characteristic or Initialization section
Variable Reference: Optional
Variable Type: Float
System Default: 1.0
Access: Read only
Propagation: Yes

CP_AT>

Opcode which defines the location in the ADB where the current Cpk value will be maintained. This data is continuously maintained by SPC, meaning that the Cp value is available without referencing the characteristic at a display station.

Example: `CP_AT> CP_VALUE;  ! Show Cp here`

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

CTRL_CHART_SIZE>

Opcode which defines the number of subgroups required for a complete control chart. This value is also the number of subgroups maintained in memory. Remember when specifying a value for this opcode, that large control chart sizes use up large amounts of memory. In general, the default of 30 subgroups should be sufficient for most applications.

If a variable is specified, this variable is read from the database at startup time. The value is then used to fix the control chart size. It is not effective to vary CTRL_CHART_SIZE> after startup.

Example:
```
CTRL_CHART_SIZE> ;20   !  Only 20 points
```

UCF Location: Characteristic section
Variable Reference: Optional
Variable Type: Numeric, Long
System Default: 30
Access: Read only
Propagation: Yes

CTRL_CHART_TYPE>

Opcode which defines the type of control chart that is used. When a VARIABLE characteristic is being defined, the user must select either X_BAR_AND_R (mean and range) charts or X_BAR_AND_S (mean and standard deviation) charts. In the case of a subgroup size of one, selecting X_BAR_AND_R for a variable characteristic provides mean and moving range charts. X-bar and S charts for a subgroup size of one would have no meaning. When an ATTRIBUTE characteristic is being defined, the user must select P (proportion defective) charts, NP (number defective) charts, C (number defects per unit) charts, or U (proportion defects per unit) charts.

The following is a summery of the choices for this opcode.

Parameter	Constant	Description
0	X_BAR_AND_R	Mean & range
1	X_BAR_AND_S	Mean & standard deviation
2	P	Proportion defective
3	NP	Number defective
4	C	Number defects per unit
5	U	Proportion defects per unit

Example:
```
!  Variable Mean & range data
CTRL_CHART_TYPE> ;X_BAR_AND_R
```

UCF Location: Characteristic section
Variable Reference: No
Variable Type: n/a
System Default: X_BAR_AND_R (VARIABLE char.) P (ATTRIBUTE char.)
Access: None
Propagation: Yes

CUSTOMER_CONFIG_CODE>

Opcode which specifies the customer's configuration code to SPC. Must be entered exactly as specified on your licensing agreement. Must be specified before the process section, immediately following the CUSTOMER_USE_LIMIT> opcode and immediately preceding the CUSTOMER_CONFIG_ID> opcode.

CUSTOMER_CONFIG_ID>

Opcode which specifies the customer's configuration identifier to SPC. Must be entered exactly as specified on your licensing agreement. Must be specified before the process section, immediately following the CUSTOMER_CONFIG_CODE> opcode.

CUSTOMER_LOCATION>

Opcode which specifies the customer's configuration location to SPC. Must be entered exactly as specified on your licensing agreement, enclosed in double quotes ("). Must be specified before the process section, immediately following the CUSTOMER_NAME> opcode and immediately preceding the CUSTOMER_SW_LICENSE> opcode.

CUSTOMER_NAME>

Opcode which specifies the customer's configuration name to SPC. Must be entered exactly as specified on your licensing agreement, enclosed in double quotes ("). Must be specified before the process section, immediately preceding the CUSTOMER_LOCATION> opcode.

CUSTOMER_SW_LICENSE>

Opcode which specifies the customer's configuration software license code to SPC. Must be entered exactly as specified on your licensing agreement. Must be specified before the process section, immediately following the CUSTOMER_LOCATION> opcode and immediately preceding the CUSTOMER_USE_LIMIT> opcode.

CUSTOMER_USE_LIMIT>

Opcode which specifies the customer's configuration use limit code to SPC. The use limit is the number of characteristics that you may define. Must be entered exactly as specified on your licensing agreement. Must be specified before the process section, immediately following the CUSTOMER_SW_LICENSE> opcode and immediately preceding the CUSTOMER_CONFIG_CODE> opcode.

CUSUM_HIGH_ALARM_AT>

Opcode which defines the location in the ADB where the current Cumulative Summation high alarm states will be maintained. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). This data is continuously maintained by SPC, meaning that the Cumulative Summation high alarm states are available without referencing the characteristic at a display station.

Example:
```
!  CuSum high alarms (Arrayof 30)
CUSUM_HIGH_ALARM_AT> CUSUM_HIGH_ALARM(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write Only
Propagation: No

CUSUM_LIMIT>

Opcode which defines the maximum Cumulative Summation value which will not cause an alarm. Any value higher than the specified limit will cause the appropriate Cumulative Summation high or low alarm, which will be reported if the appropriate CUSUM_HIGH_ALARM_AT or CUSUM_LOW_ALARM_AT opcode has been specified for the characteristic.

Example:
```
CUSUM_LIMIT> LIMIT;   !  CuSum limit
```

UCF Location: Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: 10.0
Access: Read only
Propagation: Yes

CUSUM_LOW_ALARM_AT>

Opcode which defines the location in the ADB where the current Cumulative Summation low alarm states will be maintained. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). This data is continuously maintained by SPC, meaning that the Cumulative Summation low alarm states are available without referencing the characteristic at a display station.

Example:
```
!  CuSum low alarms
CUSUM_LOW_ALARM_AT> CUSUM_LOW_ALARM(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write Only
Propagation: No

CUSUM_SLACK>

Opcode which defines the Cumulative Summation Slack limit value for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode.

Example: `CUSUM_SLACK> SLACK;  ! CuSum slack`

UCF Location: Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: 2.0
Access: Read only
Propagation: No

CUSUM_TARGET>

Opcode which defines the Cumulative Summation Target value for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode.

Example: `CUSUM_TARGET> TARGET;  ! CuSum target`

UCF Location: Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: 90.0
Access: Read only
Propagation: No

DEFINE>

Opcode which designates the start of the definition section used for product licensing. All opcodes in this section are automatically "one shot" regardless of the number of right angle brackets supplied.

Example: `DEFINE>  ! Begin definition section`

System Default: (No parameters)

DISABLE_ARCHIVING_IF>

Opcode which defines the trigger that controls whether of not the subgroups are written into the archive files as they are acquired. When the variable, or any value specified, is nonzero (true), the subgroups are NOT written into the archive files.

Example:
```
! Subgroups not archived
DISABLE_ARCHIVING_IF> ;1
```

UCF Location: Characteristic section
Variable Reference: Optional
Variable Type: Logical
System Default: 0
Access: Read Only
Propagation: No

DISPLAY_ALARM_CODE_AT>

Opcode which defines the first of an array of variables representing the alarm codes for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. These values would normally be used as a reference to a alarm library, which would describe some irregularity about the sample. A zero is equivalent to no alarm. The number of variables required in the array is the CTRL_CHART_SIZE> for the characteristic. The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Comment goes here (array 30)
DISPLAY_ALARM_CODE_AT> DS01_ACODE(0);
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Write only
Propagation: No

DISPLAY_CHARACTERISTIC>

Opcode which defines the characteristic to be displayed at the SPC display station. If a variable is defined, the string variable **must** contain the EXACT name of the SPC characteristic to be displayed when the trigger (defined by the DISPLAY_IF> or the DISPLAY_HISTOGRAM_IF> or the DISPLAY_PARETO_IF> opcode) is fired. The name of the characteristic is defined by the SPC_CHARACTERISTIC> opcode. Leading and trailing white space will be ignored. If a value is specified, it should be surrounded by double quote marks ("). This would permit only the specified characteristic to be displayed at the display station.

Example:
```
!  Disp. CNAME char
DISPLAY_CHARACTERISTIC> DS01_CNAME;
```

UCF Location: Display Station section
Variable Reference: Optional
Variable Type: String
System Default: None
Access: Read only
Propagation: No

DISPLAY_CHAR_TYPE_AT>

Opcode which defines the variable representing the characteristic type for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

The following is a summery of the output from this opcode.

Value	Description
0	Variable
1	Attribute

Example:
```
!  Characteristic type goes here
DISPLAY_CHAR_TYPE_AT> DS01_CTYPE;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Write only
Propagation: No

DISPLAY_COMMENT_CODE_AT>

Opcode which defines the first of an array of variables representing the comment codes for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. These values would normally be used as a reference to a comment library, which would describe some irregularity about the sample. A zero is equivalent to no comment. The number of variables required in the array is the product of the CTRL_CHART_SIZE> and the SUBGROUP_SIZE> for the characteristic. The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Comment Codes (Array 30 x 3)
DISPLAY_COMMENT_CODE_AT> DS01_CCODE(0);
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Write only
Propagation: No

DISPLAY_CPK_AT>

Opcode which defines the location where the Cpk value is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired. See the DISPLAY_USING> opcode for details concerning the algorithm used to calculate this value.

Example:
```
!  Cpk value placed here
DISPLAY_CPK_AT> DS01_CPK_VALUE;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_CP_AT>

Opcode which defines the location where the Cp value is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired. See the DISPLAY_USING> opcode for details concerning the algorithm used to calculate this value.

Example:
```
!  Cp value placed here
DISPLAY_CP_AT> DS01_CP_VALUE;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_CTRL_CHART_SIZE_AT>

Opcode which defines the location where the Control chart size value is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Chart Size placed here
DISPLAY_CTRL_CHART_SIZE_AT> DS01_CHT_SIZE;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Write only
Propagation: No

DISPLAY_CTRL_CHART_TYPE_AT>

Opcode which defines the location where the type of control chart is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The Control Chart type value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

The following is a summery of the data returned by this opcode.

Value	Description
0	Mean & range (X_BAR_AND_R)
1	Mean & standard deviation (X_BAR_AND_S)
2	Proportion defective (P)
3	Number defective (NP)
4	Number defects per unit (C)
5	Proportion defects per unit (U)

Example:
```
!  Chart Type for Char
DISPLAY_CTRL_CHART_TYPE_AT> ;DS01_CHT_TYPE
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Write only
Propagation: No

DISPLAY_CUSUM_HIGH_ALARM_AT>

Opcode which defines the location in the ADB where the current Cumulative Summation high alarm states are written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default is 30). The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  CuSum high alarms (Array of 30)
DISPLAY_CUSUM_HIGH_ALARM_AT> DS01_CUSUM_HIGH_ALARM(0);
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Logical
System Default: N/A
Access: Write Only
Propagation: No

DISPLAY_CUSUM_HIGH_AT>

Opcode which defines the location in the ADB where the current Cumulative Summation high chart points are written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default is 30). The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  CuSum high data  (Array of 30)
DISPLAY_CUSUM_HIGH_AT> DS01_CUSUM_HIGH(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_CUSUM_LIMIT_AT>

Opcode which defines the location where the Cusum Limit value is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Cusum limit placed here
DISPLAY_CUSUM_LIMIT_AT> DS01_CUSUM_LIMIT;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_CUSUM_LOW_ALARM_AT>

Opcode which defines the location in the ADB where the current Cumulative Summation low alarm states are written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default is 30). The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:

```
!  CuSum low alarms (Array of 30)
DISPLAY_CUSUM_LOW_ALARM_AT> DS01_CUSUM_LOW_ALARM(0);
```

UCF Location:	Display Station section
Variable Reference:	Required
Variable Type:	Logical
System Default:	N/A
Access:	Write Only
Propagation:	No

DISPLAY_CUSUM_LOW_AT>

Opcode which defines the location in the ADB where the current Cumulative Summation low alarm states are written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default is 30). The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:

```
!  CuSum low data  (Array of 30)
DISPLAY_CUSUM_LOW_AT> DS01_CUSUM_LOW(0);
```

UCF Location:	Characteristic section
Variable Reference:	Required
Variable Type:	Float
System Default:	None
Access:	Write only
Propagation:	No

DISPLAY_CUSUM_SLACK_AT>

Opcode which defines the location where the Cusum Slack value is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Cusum slack placed here
DISPLAY_CUSUM_SLACK_AT> DS01_CUSUM_SLACK;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_CUSUM_TARGET_AT>

Opcode which defines the location where the Cusum Target value is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Cusum target placed here
DISPLAY_CUSUM_TARGET_AT> DS01_CUSUM_TARGET;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_DEFECTS_AT>

Opcode which defines the location in the ADB where the Defect control chart points are written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. This opcode is only valid for the P and NP Attribute control chart types. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode. The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  P control chart data (Array of 30)
DISPLAY_DEFECTS_AT> DS01_DEFECTS(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_END_RETRIEVAL_TIME>

Opcode which defines the location in the ADB where the end retrieval time string is located. This opcode, in conjunction with the DISPLAY_RETRIEVAL_TIME> opcode, permits the user to define a time period from which a histogram or a pareto diagram will be built for the characteristic selected by the DISPLAY_CHARACTERISTIC> opcode.

This opcode is only used for retrieving histogram or pareto information, therefore it is only used when the trigger defined by DISPLAY_HISTOGRAM_IF> or DISPLAY_PARETO_IF> goes from false to true. The correct date/time format (standard VMS format) is DD-MMM-YYYY HH:MM:SS, or some subset of this string.

If the string variable defined for this opcode is blank or of an improper format, the time used in the case of the histogram will be one of the following.

1) Last histogram reset time
2) Time of last specification limit (upper or lower) change
3) Time of the 400th sample point

Example:
```
!  End bound
DISPLAY_END_RETRIEVAL_TIME> DS01_END_TIME;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: String[20]
System Default: See Above
Access: Read only
Propagation: No

DISPLAY_GRAND_MEAN_AT>

Opcode which defines the location where the grand mean (X-bar-bar, the mean of the means) is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The grand mean is the average of all the X-bar points on the control chart. The total number of points involved equals the CTRL_CHART_SIZE>. The grand mean is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  X-barbar goes here
DISPLAY_GRAND_MEAN_AT> DS01_X_BAR_BAR;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_GROUP_SIZE_AT>

Opcode which defines the location where the Group size value is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Group size placed here
DISPLAY_GROUP_SIZE_AT> DS01_GROUP_SIZE;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Write only
Propagation: No

DISPLAY_HISTOGRAM_AT>

Opcode which defines the first of 42 locations where data necessary to plot a histogram is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The histogram data is written when the histogram trigger (defined by the DISPLAY_HISTOGRAM_IF> opcode) is fired. The following data is provided.

Array Index	Element Description	Bucket #
0	Actual average of values	
1	Standard deviation of values	
2	Upper 3 sigma limit	
3	Lower 3 sigma limit	
4	6 sigma spread	
5	Specification average	
6	Specification spread	
7	Percent below specification limit	
8	Percent above specification limit	
9	Percent of range used	
10	Bucket size	
11	Total number of bucket hits	
12	Number of items too small for histogram	1
13	Bucket holding smallest values	2
14	.	3
15	.	4
16	.	5
17		6
18		7
19		8
20		9
21		10
22	.	11
23	.	12

DISPLAY_HISTOGRAM_AT> (cont)

24	.	13
25	Bucket holding largest values	14
26	Number of items too large for histogram	15
27	Lower limit of bucket 1	
28	Lower limit of bucket 2 (upper limit of 1)	
29	Lower limit of bucket 3 (upper limit of 2)	
30	Lower limit of bucket 4 (upper limit of 3)	
31	Lower limit of bucket 5 (upper limit of 4)	
32	Lower limit of bucket 6 (upper limit of 5)	
33	Lower limit of bucket 7 (upper limit of 6)	
34	Lower limit of bucket 8 (upper limit of 7)	
35	Lower limit of bucket 9 (upper limit of 8)	
36	Lower limit of bucket 10 (upper limit of 9)	
37	Lower limit of bucket 11 (upper limit of 10)	
38	Lower limit of bucket 12 (upper limit of 11)	
39	Lower limit of bucket 13 (upper limit of 12)	
40	Lower limit of bucket 14 (upper limit of 13)	
41	Lower limit of bucket 15 (upper limit of 14)	

Example:
```
!  Histogram data
DISPLAY_HISTOGRAM_AT> HIST_ARRAY;
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_HISTOGRAM_BUCKETS>

Opcode which defines the location where the number of Histogram buckets is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_HISTOGRAM_IF> opcode) is fired.

Example:
```
!  Bucket qty placed here
DISPLAY_HISTOGRAM_BUCKETS> DS01_HISTO_BUCKETS;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Write only
Propagation: No

DISPLAY_HISTOGRAM_IF>

Opcode which defines the trigger that causes the DISPLAY_HISTOGRAM_AT> opcode to be executed. When the variable changes from zero (false) to nonzero (true), histogram data is provided to the application database as determined by the DISPLAY_HISTOGRAM_AT> opcode, as well as by the DISPLAY_RETRIEVAL_TIME> and DISPLAY_END_RETRIEVAL_TIME> opcodes. SPC will reset (make false) the trigger specified after the display actions are complete.

Example:
```
!  Show histogram
DISPLAY_HISTOGRAM_IF> HISTO_TRIG;
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: 0
Access: Read/Write
Propagation: Yes

DISPLAY_HISTO_RESET_TIME_AT>

Opcode which defines the location in the ADB where the last Histogram reset time string is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The format for this string is DD-MMM-YYYY HH:MM:SS (standard VMS format). A histogram reset is triggered by the HISTOGRAM_RESET_IF> opcode. The value is only written when the trigger (defined by the DISPLAY_HISTOGRAM_IF> opcode) is fired.

Example:
```
! Show reset date/time
DISPLAY_HISTOGRAM_RESET_TIME_AT> HISTO_RESET_TIME;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: String[20]
System Default: None
Access: Write only
Propagation: No

DISPLAY_IF>

Opcode which defines the trigger that causes the various display station opcodes to be executed (ex. DISPLAY_MEAN_AT>, DISPLAY_RS_AT>, etc.) When the variable changes from zero (false) to nonzero (true), display station-related data is written once to the application database. SPC will reset (make false) the trigger specified after the display actions are complete. Refer to the DISPLAY_WHILE> opcode for level triggered (continuous) functionality.

Example:
```
!  Show Control chart
DISPLAY_IF> CONTROL_CHART_TRIG;
```

DISPLAY_IF> (cont)

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: 0
Access: Read/Write
Propagation: Yes

DISPLAY_LOT_SIZE_AT>

Opcode which defines the location where the number of samples per lot is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. This opcode is only valid for the P, NP, U and C attribute control chart types. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Lot size placed here
DISPLAY_LOT_SIZE_AT> DS01_LOST_SIZE;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Write only
Propagation: No

DISPLAY_LSL_AT>

Opcode which defines the location where the lower Specification limit is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Lower spec limit placed here
DISPLAY_LSL_AT> DS01_LOWER_SPEC_LIMIT;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_LVL_AT>

Opcode which defines the location where the lower validity limit is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Lower validity limit placed here
DISPLAY_LVL_AT> DS01_LOWER_VALIDITY_LIMIT;
```

DISPLAY_LVL_AT> (cont)

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_MEAN_AT>

Opcode which defines the first of a control chart set of consecutive variables representing the means (X-bar) of each of the subgroups for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). These values can be used to plot the control chart for either Variable or Attribute SPC.

Example:
```
! Mean array placed here
DISPLAY_MEAN_AT> X_BAR_ARRAY(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_PARETO_AT>

Opcode which defines the first of 30 locations where data necessary to plot a pareto chart is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The pareto data is written when the pareto trigger (defined by the DISPLAY_PARETO_IF> opcode) is fired. The following data is provided.

Array Index	Description	Bucket #
0	Largest number of hits	1
1	Next largest number of hits	2
2	.	3
3	.	4
4	.	5
5		6
6		7
7		8
8		9
9		10
10	.	11
11	.	12
12	.	13

DISPLAY_PARETO_AT> (cont)

13	Next smallest number of hits	14
14	Smallest number of hits	15
15	Code with largest number of hits	1
16	Code with next largest number of hits	2
17	.	3
18	.	4
19	.	5
20		6
21		7
22		8
23		9
24		10
25	.	11
26	.	12
27	.	13
28	Code with next smallest number of hits	14
29	Code with smallest number of hits	15

UCF Location: Display station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: 0
Access: Write only
Propagation: No

DISPLAY_PARETO_BUCKETS>

Opcode which defines the location where the number of Pareto buckets is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_PARETO_IF> opcode) is fired.

Example:
```
!  Bucket qty placed here
DISPLAY_PARETO_BUCKETS> DS01_PARETO_BUCKETS;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Write only
Propagation: No

DISPLAY_PARETO_IF>

Opcode which defines the trigger that causes the display pareto chart opcodes to be executed (ex. DISPLAY_PARETO_AT>, DISPLAY_PARETO_BUCKETS>, etc.) When the variable changes from zero (false) to nonzero (true), pareto chart related data is provided to the application database as determined by the DISPLAY_PARETO_AT> opcode, as well as by the DISPLAY_RETRIEVAL_TIME> and the DISPLAY_END_RETRIEVAL_TIME> opcode. SPC will reset (make false) the trigger specified after the display actions are complete.

Example:
```
!  Show Pareto chart
DISPLAY_PARETO_IF> PARETO_CHART_TRIG;
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: 0
Access: Read/Write
Propagation: Yes

DISPLAY_PARETO_RESET_TIME_AT>

Opcode which defines the location in the ADB where the last Pareto reset time string is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The format for this string is DD-MMM-YYYY HH:MM:SS (standard VMS format). A pareto reset is triggered by the PARETO_RESET_IF> opcode. The value is only written when the trigger (defined by the DISPLAY_PARETO_IF> opcode) is fired.

Example:
```
!  Show reset date/time
DISPLAY_PARETO_RESET_TIME_AT> PARETO_RESET_TIME;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: String[20]
System Default: None
Access: Write only
Propagation: No

DISPLAY_PARETO_TYPE>

Opcode which determines the source of input data for the pareto chart being defined. The value is only written when the trigger (defined by the DISPLAY_PARETO_IF> opcode) is fired.

The following is a summery of the output from this opcode.

Parameter	Constant	Description
0	ALARM	Use alarm code values
1	COMMENT	Use comment code values

Example:
```
!  Display pareto type
DISPLAY_PARETO_TYPE> PARETO_TYPE;
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: 0 (ALARM)
Access: Write only
Propagation: No

DISPLAY_PCT_DEFECTS_AT>

Opcode which defines the location in the ADB where the Percent Defective control chart points are written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. This opcode is only valid for the U and C attribute control chart types. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode. The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  U control chart data (Array of 30)
DISPLAY_PCT_DEFECTS_AT> DS01_PCT_DEFECTS(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_RETRIEVAL_TIME>

Control Charts: Opcode which defines the location in the ADB where the retrieval time string is located. The retrieval time string permits the user to re-define the starting time period for the control chart data displayed by the DISPLAY_MEAN_AT> DISPLAY_RS_AT, DISPLAY_DEFECTS_AT> and DISPLAY_PCT_DEFECTS_AT> opcodes. The control chart data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

DISPLAY_RETRIEVAL_TIME> (cont)

Histogram/Pareto Charts: This opcode, in optional conjunction with the DISPLAY_END_RETRIEVAL_TIME> opcode, permits the user to define a time period from which a histogram or a pareto diagram will be built for the characteristic selected by the DISPLAY_CHARACTERISTIC> opcode. The Histogram/Pareto data is written to the database when the trigger (defined by the DISPLAY_HISTOGRAM_IF> or DISPLAY_PARETO_IF> opcode) is fired.

The correct date/time format (standard VMS format) is DD-MMM-YYYY HH:MM:SS, or some subset of this string.

If the string variable defined for this opcode is blank or of an improper format, the time used will be the most recent data.

Example:
```
!  Start time boundary
DISPLAY_RETRIEVAL_TIME> DS01_START_TIME;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: String[20]
System Default: None
Access: Read only
Propagation: No

DISPLAY_RSLCL_AT>

Opcode which defines the location where the Range (R) or Standard Deviation (S) lower control limit is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The opcode CTRL_CHART_TYPE> defines whether the values written apply to ranges or standard deviations of the subgroups. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Lower control limit placed here
DISPLAY_RSLCL_AT> DS01_RANGE_LCL_LIMIT;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_RSUCL_AT>

Opcode which defines the location where the Range (R) or Standard Deviation (S) upper control limit is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The opcode CTRL_CHART_TYPE> defines whether the values written apply to ranges or standard deviations of the subgroups. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Lower control limit placed here
DISPLAY_RSUCL_AT> DS01_STDDEV_UCL_LIMIT;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_RS_AT>

Opcode which defines the first of a control chart set of consecutive variables representing either the Range, Moving Range or Standard Deviation of each of the subgroups for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent ranges or standard deviations of the subgroups. If the subgroup size is one, (defined by the SUBGROUP_SIZE> opcode) the values written represent a Moving Range chart.

Example:
```
!  Ranges placed here
DISPLAY_RS_AT> RANGE_ARRAY(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_RS_BAR_7_ABOVE_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive range or standard deviation control points being above the range/standard deviation mean. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent alarms relating to ranges or standard deviations of the subgroups. This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup ranges/standard deviations are found to be above the mean, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition. The data is written when the trigger (defined by the DISPLAY_IF> opcode) goes from false to true.

Example:
```
!  7 pts above alarms
DISPLAY_RS_BAR_7_ABOVE_AT> DS01_ALM_7ABOVE(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_RS_BAR_7_BELOW_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive range or standard deviation control points being below the range/standard deviation mean. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent alarms relating to ranges or standard deviations of the subgroups. This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup ranges/standard deviations are found to be above the mean, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition. The data is written when the trigger (defined by the DISPLAY_IF> opcode) goes from false to true.

Example:
```
!  7 pts below alarms
DISPLAY_RS_BAR_7_BELOW_AT> DS01_ALM_7BELOW(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_RS_BAR_7_DECR_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive range or standard deviation control points in a steadily decreasing pattern. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent alarms relating to ranges or standard deviations of the subgroups. This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup ranges/standard deviations are found to be smaller than their predecessors, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition. The data is written when the trigger (defined by the DISPLAY_IF> opcode) goes from false to true.

Example:
```
!  7 pts decreasing alarms
DISPLAY_RS_BAR_7_DECR_AT> DS01_ALM_7DECR(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_RS_BAR_7_INCR_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive range or standard deviation control points in a steadily increasing pattern. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent alarms relating to ranges or standard deviations of the subgroups. This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup ranges/standard deviations are found to be larger than their predecessors, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition. The data is written when the trigger (defined by the DISPLAY_IF> opcode) goes from false to true.

Example:
```
!  7 pts increasing alarms
DISPLAY_RS_BAR_7_INCR_AT> DS01_ALM_7INCR(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_RS_BAR_CTRL_ALARM_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by a single range or standard deviation control point exceeding the limits defined by the RSUCL> or RSLCL> opcodes. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent alarms relating to ranges or standard deviations of the subgroups. When a subgroup is either larger than the upper control limit, the alarm variable corresponding to the subgroup in alarm will be set true. The data is written when the trigger (defined by the DISPLAY_IF> opcode) goes from false to true.

Example:
```
!  RS Ctrl alarms
DISPLAY_RS_BAR_CTRL_ALARM_AT> DS01_ALM_RS_MEAN(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_RS_BAR_CTRL_WARN_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by a single range or standard deviation control point exceeding the SPC-calculated warning limits. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent alarms relating to ranges or standard deviations of the subgroups. A control point falls in the warning band if its value is greater than the upper control limit minus five percent of the specification spread, or is less than the lower control limit plus 5 percent of the specification spread, yet not outside the control limits.

RSUCL > Upper warning band > RSUCL - (.05 * (RSUSL - RSLSL))
RSLCL < Lower warning band < RSLCL + (.05 * (RSUSL - RSLSL))

When the subgroup falls in this band, the warning bit will be set true. The data is written when the trigger (defined by the DISPLAY_IF> opcode) goes from false to true.

Example:
```
!  RS Warning alarms
DISPLAY_RS_BAR_CTRL_WARN_AT> DS01_ALM_RS_WARNING(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_RS_MEAN_AT>

Opcode which defines the location where the mean of range or standard deviation arrays is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The RS mean is the average of all the range or standard deviation points on the control chart. The opcode CTRL_CHART_TYPE> defines whether the values written represent ranges or standard deviations of the subgroups. The total number of points involved equals the CTRL_CHART_SIZE>. The RS mean is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Range mean goes here
DISPLAY_RS_MEAN_AT> DS01_RS_BAR;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_RS_TARGET_AT>

Opcode which defines the location where the target for range or standard deviation data is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The RS target is not currently used as part of any SPC calculations, but is for display purposes only. The opcode CTRL_CHART_TYPE> defines whether the values written represent ranges or standard deviations of the subgroups. The RS target is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Range target goes here
DISPLAY_RS_TARGET_AT> DS01_RS_TARGET;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_SAMPLES_AT>

Opcode which defines the first of an array of variables representing the raw sample data for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The data is arranged such that sample one of subgroup one is the first item in the array, followed by the second sample of the first subgroup, etc. The values placed in the database represent the value of each individual sample that went into making the subgroups for the current control chart. The number of variables required in the array is the product of the CTRL_CHART_SIZE> and the SUBGROUP_SIZE> for the characteristic. The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Sample data  (Array 30 x 3)
DISPLAY_SAMPLES_AT> SAMPLE_DATA(0);
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_SAMPLE_TIME_AT>

Opcode which defines the first of an array of variables representing the times associated with raw sample data for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The data is arranged such that sample time one of subgroup one is the first item in the array, followed by the second sample time of the first subgroup, etc. The values placed in the database represent the time at which each value of each individual sample was entered into the subgroups for the current control chart. The number of variables required in the array is the product of the CTRL_CHART_SIZE> and the SUBGROUP_SIZE> for the characteristic. The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Sample times  (Array 30 x 3)
DISPLAY_SAMPLES_TIME_AT> SAMPLE_TIME(0);
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: String[20]
System Default: None
Access: Write only
Propagation: No

DISPLAY_SAMPLE_TYPE_AT>

Opcode which defines the location where the sampling type is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The sample type is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

The following is a summery of the output from this opcode.

Value	Description
0	Event based automatic sampling
1	Time based automatic sampling
2	Count based automatic sampling
3	Manual sampling

UCF Location: Display station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Write only
Propagation: No

DISPLAY_SHIFT_NUMBER_AT>

Opcode which defines the location in the ADB where the shift number array data is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode. The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Shift data goes here
DISPLAY_SHIFT_NUMBER_AT> DS01_SHIFT(0);
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Write only
Propagation: No

DISPLAY_SPEC_ALARM_AT>

Opcode which defines the first of an array of variables representing the alarm status for the raw sample data for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The data is arranged such that sample one of subgroup one is the first bit in the array, followed by the second sample of the first subgroup, etc. When the bit is logically true, it indicates that the represented sample is outside the specification tolerances. The number of variables required in the array is the product of the CTRL_CHART_SIZE> and the SUBGROUP_SIZE> for the characteristic. The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Show spec. status here
DISPLAY_SPEC_ALARM_AT> DS01_SAMPLE_ALMS(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_SUBGROUP_SIZE_AT>

Opcode which defines the location in the ADB where the subgroup size array data is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode. The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired. This functionality is useful when the END_SUBGROUP_IF> opcode or the TMP_SUBGROUP_SIZE> opcode is used, otherwise the subgroup size is fixed by the SUBGROUP_SIZE> opcode.

Example:
```
!  Show size here
DISPLAY_SUBGROUP_SIZE_AT> DS01_SG_SIZE(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Write only
Propagation: No

DISPLAY_SUBGROUP_TIME_AT>

Opcode which defines the location in the ADB where the subgroup date/time array data is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The format for this string is DD-MMM-YYYY HH:MM:SS (standard VMS format). The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode. The data is written to the database when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Show date/time here
DISPLAY_SUBGROUP_TIME_AT> DS01_SG_TIME(0);
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: String[20]
System Default: None
Access: Read only
Propagation: No

DISPLAY_USING>

Opcode which defines whether Cp, Cpk, grand mean, and range/standard deviation mean are recalculated when retrieving archived SPC data. The constant ARCHIVED causes these values to be the same as they were when the subgroup was created. The constant CALCULATED causes these values to be calculated from the current 30 subgroups on the chart. The following is a summery of the choices for this opcode.

Parameter	Constant	Description
0	ARCHIVED	Use archived values
1	CALCULATED	Calculate values from 30 chart points

Example:
```
DISPLAY_USING>   ;CALCULATED   !  Use calc. type
```

UCF Location: Display station section
Variable Reference: Optional
Variable Type: Numeric, Long
System Default: ARCHIVED
Access: Read only
Propagation: Yes

DISPLAY_USL_AT>

Opcode which defines the location where the current Upper Spec limit is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
! USL value placed here
DISPLAY_USL_AT> DS01_USL;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_UVL_AT>

Opcode which defines the location where the current Upper Validity limit is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
! UVL value placed here
DISPLAY_UVL_AT> DS01_UVL;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_WHILE>

Opcode which defines the trigger that continuously causes the various display station opcodes to be executed (ex. DISPLAY_MEAN_AT>, DISPLAY_RS_AT>, etc.) When the variable changes from zero (false) to nonzero (true), display station related data is provided to the application database. SPC will NOT reset the trigger specified after the display actions are complete. Refer to the DISPLAY_IF> opcode for edge triggered (one shot) functionality.

Example:
```
! Show Control chart
DISPLAY_WHILE> CONTROL_CHART_TRIG;
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: 0
Access: Read Only
Propagation: Yes

DISPLAY_XLCL_AT>

Opcode which defines the location where the current Lower Control limit is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
! X_bar LCL value placed here
DISPLAY_XLCL_AT> DS01_XLCL;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_XUCL_AT>

Opcode which defines the location where the current Upper Control limit is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The value is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
! X_bar UCL value placed here
DISPLAY_XUCL_AT> DS01_XUCL;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

DISPLAY_X_BAR_7_ABOVE_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive mean control points being above the grand mean. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup means are found to be above the grand mean, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition. The data is written when the trigger (defined by the DISPLAY_IF> opcode) goes from false to true.

Example:
```
! 7 pts above alarms
DISPLAY_X_BAR_7_ABOVE_AT> DS01_X_BAR_ALM_7ABOVE(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_X_BAR_7_BELOW_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive mean control points being below the grand mean. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup means are found to be below the grand mean, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition. The data is written when the trigger (defined by the DISPLAY_IF> opcode) goes from false to true.

Example:
```
!  7 pts below alarms
DISPLAY_X_BAR_7_BELOW_AT> DS01_X_BAR_ALM_7BELOW(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_X_BAR_7_DECR_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive mean control points in a steadily decreasing pattern. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup mean values are found to be smaller than their predecessors, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition. The data is written when the trigger (defined by the DISPLAY_IF> opcode) goes from false to true.

Example:
```
!  7 pts decreasing alarms
DISPLAY_X_BAR_7_DECR_AT> DS01_X_BAR_ALM_7DECR(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_X_BAR_7_INCR_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive mean control points in a steadily increasing pattern. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup mean values are found to be larger than their predecessors, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition. The data is written when the trigger (defined by the DISPLAY_IF> opcode) goes from false to true.

Example:
```
!  7 pts increasing alarms
DISPLAY_X_BAR_7_INCR_AT> DS01_X_BAR_ALM_7INCR(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_X_BAR_CTRL_ALARM_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by a single mean control point exceeding the limits defined by the XUCL> or XLCL> opcodes. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). When a subgroup is either larger than the upper control limit, the alarm variable corresponding to the subgroup in alarm will be set true. The data is written when the trigger (defined by the DISPLAY_IF> opcode) goes from false to true.

Example:
```
!  X_bar Ctrl alarms
DISPLAY_X_BAR_CTRL_ALARM_AT> DS01_ALM_X_MEAN(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_X_BAR_CTRL_WARN_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by a single mean control point exceeding the SPC-calculated warning limits. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). A control point falls in the warning band if its value is greater than the upper control limit minus five percent of the specification spread, or is less than the lower control limit plus 5 percent of the specification spread, yet not outside the control limits.

XUCL > Upper warning band > XUCL - (.05 * (XUSL - XLSL))
XLCL < Lower warning band < XLCL + (.05 * (XUSL - XLSL))

When the subgroup falls in this band, the warning bit will be set true. The data is written when the trigger (defined by the DISPLAY_IF> opcode) goes from false to true.

Example:
```
!  X Warning alarms
DISPLAY_X_BAR_CTRL_WARN_AT> DS01_ALM_X_WARNING(0);
```

UCF Location: Display station section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

DISPLAY_X_TARGET_AT>

Opcode which defines the location where the target for mean control chart data is written for the characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The mean target is not currently used as part of any SPC calculations, but is for display purposes only. The opcode CTRL_CHART_TYPE> defines whether the values written represent ranges or standard deviations of the subgroups. The X target is written when the trigger (defined by the DISPLAY_IF> opcode) is fired.

Example:
```
!  Mean target goes here
DISPLAY_X_TARGET_AT> DS01_X_TARGET;
```

UCF Location: Display Station section
Variable Reference: Required
Variable Type: Float
System Default: None
Access: Write only
Propagation: No

END>

Opcode which defines the end of the UCF file. Anything following this opcode is ignored.

Example: `END>  ! End of processing`

UCF Location: Display Station section
Variable Reference: No
Variable Type: n/a
System Default: None
Access: None
Propagation: No

END_SUBGROUP_IF>

Opcode which defines the trigger that terminates acquisition of sample values for the current subgroup. When the variable is nonzero (true) and at least one sample value has been acquired, the subgroup is considered complete. If no sample values have been acquired, this function is ignored. After this function is completed, trigger is reset to zero (false). This function is used when acquiring subgroup information one sample at a time and enough samples have not been acquired to satisfy the samples per subgroup requirement, defined by the SUBGROUP_SIZE> opcode. This opcode is recognized only if the characteristic type is VARIABLE, as defined by the CHARACTERISTIC_TYPE> opcode, the type of sampling is MANUAL, as defined by the SAMPLE_TYPE> opcode, and the manual sampling subtype is SAMPLE, as defined by the MANUAL_SUBTYPE> opcode. See documentation on TMP_SUBGROUP_SIZE> if the manual sampling subtype is SUBGROUP.

Example: `END_SUBGROUP_IF> STOP_SAMPLE;  ! End subgroup`

UCF Location: Characteristic or Initialization section
Variable Reference: Required
Variable Type: Logical
System Default: 0
Access: Read/Write
Propagation: Yes

EXIT_SPC_IF>

Opcode which defines the trigger that causes SPC to exit. When the variable, or any value specified, is non-zero (true), SPC will exit. Only the last EXIT_SPC_IF> opcode in the UCF will be recognized. A manual restart of SPC will be required after exiting in this manner.

Example: `EXIT_SPC_IF> EXIT;  ! Exit trigger`

UCF Location: Initialization or Display station sections
Variable Reference: Required
Variable Type: Logical
System Default: 0
Access: Read/Write
Propagation: No

GROUP_COMPRESS_SIZE>

Opcode which defines the sampling group size to use when compressing 100 percent group sampling. This opcode is only meaningful when the characteristic type is VARIABLE, as defined by the CHARACTERISTIC_TYPE> opcode and the sampling type is COUNT, as defined by the SAMPLE_TYPE> opcode.

Example:
```
!  Comp.  population size
GROUP_COMPRESS_SIZE> ;50
```

UCF Location: Characteristic or Initialization section
Variable Reference: Optional
Variable Type: Numeric, Long
System Default: 100
Access: Read only
Propagation: Yes

GROUP_SIZE>

Opcode which defines the sampling group size from which the sample values will be acquired. Only the last subgroup size worth of sample values are acquired; the sample values previous to the last subgroup size worth are ignored. This opcode is only meaningful when the characteristic type is VARIABLE, as defined by the CHARACTERISTIC_TYPE> opcode and the sampling type is COUNT, as defined by the SAMPLE_TYPE> opcode.

Example:
```
GROUP_SIZE> ;50   !  Population size
```

UCF Location: Characteristic or Initialization section
Variable Reference: Optional
Variable Type: Numeric, Long
System Default: 100
Access: Read only
Propagation: Yes

HISTOGRAM_RESET_IF>

Opcode which defines the trigger that controls the re-zeroing of the histogram accumulators by recording a histogram reset time. When the variable is non-zero (true), the histogram reset time will be updated with the current time. After this function is completed, trigger is reset to zero (false). The reset time is NOT permanently saved and will become unset if either CRISP or SPC is restarted. Resetting each histogram periodically, either manually or through application logic, will improve histogram response times greatly. When the HISTOGRAM_RESET_TIME_AT> opcode has been defined, the variable associated with that opcode will be updated.

Example:
```
!  Reset histogram
HISTOGRAM_RESET_IF> HISTO_RESET_TRIG;
```

HISTOGRAM_RESET_IF> (cont)

UCF Location: Characteristic or Initialization section
Variable Reference: Optional
Variable Type: Logical
System Default: 0
Access: Read/Write
Propagation: Yes

HISTOGRAM_RESET_TIME_AT>

Opcode which defines the location in the ADB where the current histogram reset time will be maintained. The histogram reset time will be in standard VMS time string format, DD-MMM-YYYY HH:MM:SS.

Example:
```
!  Show time
HISTOGRAM_RESET_TIME_AT> RESET_TIME;
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: String[20]
System Default: None
Access: Write only
Propagation: No

INIT>

Optional opcode which designates the start of the initialization section. All opcodes in this section are automatically "one shot" regardless of the number of right angle brackets supplied.

Example:
```
INIT>   !  Begin initialization section
```

UCF Location: Initialization section
Variable Reference: No
Variable Type: n/a
System Default: None
Access: None
Propagation: No

LOT_SIZE>

Opcode which defines the total number of items for which the sample value is associated with. This opcode is only meaningful when the characteristic type is ATTRIBUTE, as defined by the CHARACTERISTIC_TYPE> opcode.

Example:
```
LOT_SIZE> LOT_SIZE;   !  Total number of items
```

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Numeric, Long
System Default: 100
Access: Read only
Propagation: No

LSL>

Opcode which defines the lower specification limit for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The lower spec limit is important for performing Cpk/Cp calculations and boundary definition for the Histogram. When a variable name is included with this opcode, the value is read each time the data sample trigger (defined by the SAMPLE_IF> opcode) is fired.

Example: `LSL> LOWER_SPEC_LMT;  ! Lower spec limit`

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: 10.0
Access: Read only
Propagation: Yes

LVL>

Opcode which defines the lower limit for manually entered sample values which will not cause a validity alarm. Validity alarm checking is performed only when the characteristic type is VARIABLE, as defined by the CHARACTERISTIC_TYPE> opcode, and the sample type is MANUAL, as defined by the SAMPLE_TYPE> opcode. Any value lower than the specified limit will cause a validity alarm, which will be reported if the VALIDITY_ALARM_AT> opcode has been specified for the characteristic. When a validity alarm occurs, the sample value will be ignored. The OVERRIDE_VALIDITY_IF> opcode may be used to disable the validity checking and allow the acquisition of sample values which are outside of the validity limits. When a variable name is included with this opcode, the value is read each time the data sample trigger (defined by the SAMPLE_IF> opcode) is fired.

Example: `LVL> LOW_VALIDITY_LIMIT;  ! Lower validity limit`

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: -100000000
Access: Read only
Propagation: Yes

MANUAL_SUBTYPE>

Opcode which determines the type of manual sampling that is being defined, SAMPLE or SUBGROUP. A manual subtype of SAMPLE acquires a sample value each time the trigger is non-zero (true). A manual subtype of SUBGROUP acquires a subgroup size worth of sample values each time the trigger is non-zero (true). The trigger is defined by the SAMPLE_IF> opcode. This opcode is only meaningful when the characteristic type is VARIABLE, as defined by the CHARACTERISTIC_TYPE> opcode, and the sample type is MANUAL, as defined by the SAMPLE_TYPE> opcode.

The following is a summery of the choices for this opcode.

Parameter	Constant	Description
0	SAMPLE	Trigger each sample
1	SUBGROUP	Trigger each subgroup

Example:
```
MANUAL_SUBTYPE> ;SAMPLE   !  Trigger each sample
```

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Numeric, Long
System Default: SAMPLE
Access: Read only
Propagation: Yes

MULTIPLIER>

Opcode which defines a scaling factor to be applied to each sample value. The scaling factor is applied prior to any validity or alarm checking.

Example:
```
MULTIPLIER> ;.001   !  Scale factor
```

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: 1.0
Access: Read only
Propagation: Yes

OVERRIDE_VALIDITY_IF>

Opcode that controls whether or not validity checking is performed on the sample values. This opcode is only meaningful when the sampling type is MANUAL, as defined by the SAMPLE_TYPE> opcode and validity limits are defined by the LVL> and UVL> opcodes.

Example:
```
!  Disable Validity Checking
OVERRIDE_VALIDITY_IF> OVERRIDE_ENABLE;
```

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Logical
System Default: 0
Access: Read only
Propagation: Yes

PARETO_RESET_IF>

Opcode which defines the trigger that controls the re-zeroing of the pareto chart accumulators by recording a pareto reset time. When the variable is non-zero (true), the pareto reset time will be updated with the current time. After this function is completed, trigger is reset to zero (false). The reset time is NOT permanently saved and will become unset if either CRISP or SPC is restarted. Resetting each pareto chart periodically, either manually or through application logic, will improve pareto response times greatly. When the PARETO_RESET_TIME_AT> opcode has been defined, the variable associated with that opcode will be updated.

Example:
```
!  Reset pareto
PARETO_RESET_IF>  PARETO_RESET_TRIG;
```

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Logical
System Default: 0
Access: Read/Write
Propagation: Yes

PARETO_RESET_TIME_AT>

Opcode which defines the location in the ADB where the current pareto reset time will be maintained. The pareto reset time will be in standard VMS time string format, DD-MMM-YYYY HH:MM:SS.

Example:
```
PARETO_RESET_TIME_AT> RESET_TIME;   !  Show time
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: String[20]
System Default: None
Access: Write only
Propagation: No

PROCESS>

Opcode which defines the start of the process section. This is included in the UCF for compatibility with IDI. Some opcodes prior to the PROCESS> opcode are common to all characteristics or display stations.

Example:
```
PROCESS>   !  Begin process section
```

UCF Location: Following the Initialization section
Variable Reference: No
Variable Type: n/a
System Default: None
Access: None
Propagation: No

RSLCL>

Opcode which defines the lower range or standard deviation control limit for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. Any value lower than the specified limit will cause a control alarm. The CTRL_CHART_TYPE> opcode determines whether the control limit is for the subgroup ranges or the standard deviations. When a variable name is included with this opcode, the value is read each time the data sample trigger (defined by the SAMPLE_IF> opcode) is fired. If the limit calculation function is used (as defined by the CALC_CONTROL_LIMITS_IF> opcode) then the current value associated with this opcode will be revised.

Example:
```
!  Lower range/stddev control limit
RSLCL> RS_LOWER_CONTROL_LMT;
```

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: 1.0
Access: Read/Write
Propagation: Yes

RSUCL>

Opcode which defines the upper range or standard deviation control limit for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. Any value higher than the specified limit will cause a control alarm. The CTRL_CHART_TYPE> opcode determines whether the control limit is for the subgroup ranges or the standard deviations. When a variable name is included with this opcode, the value is read each time the data sample trigger (defined by the SAMPLE_IF> opcode) is fired. If the limit calculation function is used (as defined by the CALC_CONTROL_LIMITS_IF> opcode) then the current value associated with this opcode will be revised.

Example:
```
!  Upper range/stddev control limit
RSLCL> RS_UPPPER_CONTROL_LMT;
```

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: 10.0
Access: Read/Write
Propagation: Yes

RS_BAR_7_ABOVE_ALARM_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive range or standard deviation control points being above the range/standard deviation mean. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent alarms relating to ranges or standard deviations of the subgroups. This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

RS_BAR_7_ABOVE_ALARM_AT> (cont)

This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup ranges/standard deviations are found to be above the mean, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition.

Example:
```
!  7 pts above alarms
RS_BAR_7_ABOVE_AT> ALM_7ABOVE(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

RS_BAR_7_BELOW_ALARM_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive range or standard deviation control points being below the range/standard deviation mean. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent alarms relating to ranges or standard deviations of the subgroups. This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup ranges/standard deviations are found to be above the mean, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition.

Example:
```
!  7 pts below alarms
RS_BAR_7_BELOW_AT> ALM_7BELOW(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

RS_BAR_7_DECR_ALARM_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive range or standard deviation control points in a steadily decreasing pattern. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent alarms relating to ranges or standard deviations of the subgroups. This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup ranges/standard deviations are found to be smaller than their predecessors, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition.

Example:
```
!  7 pts decreasing alarms
RS_BAR_7_DECR_AT> ALM_7DECR(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

RS_BAR_7_INCR_ALARM_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive range or standard deviation control points in a steadily increasing pattern. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent alarms relating to ranges or standard deviations of the subgroups. This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup ranges/standard deviations are found to be larger than their predecessors, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition.

Example:
```
!  7 pts increasing alarms
RS_BAR_7_INCR_AT> ALM_7INCR(0);
```

RS_BAR_7_INCR_ALARM_AT> (cont)

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

RS_BAR_CTRL_ALARM_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by a single range or standard deviation control point exceeding the limits defined by the RSUCL> or RSLCL> opcodes. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent alarms relating to ranges or standard deviations of the subgroups. When a subgroup is either larger than the upper control limit, the alarm variable corresponding to the subgroup in alarm will be set true. This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

Example:
```
!  RS Ctrl alarms
RS_BAR_CTRL_ALARM_AT> ALM_RS_MEAN(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

RS_BAR_CTRL_WARN_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by a single range or standard deviation control point exceeding the SPC-calculated warning limits. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). The opcode CTRL_CHART_TYPE> defines whether the values written represent alarms relating to ranges or standard deviations of the subgroups. This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

A control point falls in the warning band if its value is greater than the upper control limit minus five percent of the specification spread, or is less than the lower control limit plus 5 percent of the specification spread, yet not outside the control limits.

RSUCL > Upper warning band > RSUCL - (.05 * (RSUCL - RSLCL))
RSLCL < Lower warning band < RSLCL + (.05 * (RSUCL - RSLCL))

RS_BAR_CTRL_WARN_AT> (cont)

When the subgroup falls in this band, the warning bit will be set true.

Example:
```
!  RS Warning alarms
RS_BAR_CTRL_WARN_AT> ALM_RS_WARNING(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

RS_TARGET>

Opcode which defines the ideal average range or standard deviation of the sample values in the subgroup. The data referenced by this opcode is used for building chart graphics and is not part of any supporting calculations.

Example:
```
RS_TARGET> ;4.28   !  Range target
```

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: (RSUCL + RSLCL) / 2
Access: Read only
Propagation: Yes

SAMPLE_IF>

Opcode which defines the trigger that causes a sample value to be acquired from the ADB location. When the variable specified is non-zero (true), the sample value is acquired and the variable is set to zero (false).

Example:
```
SAMPLE_IF> READ_TRIG;   !  Read sample trigger
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: 0
Access: Read/Write
Propagation: Yes

SAMPLE_INTERVAL>

Opcode which defines the rate at which a sample value is acquired when the characteristic type is VARIABLE, as defined by the CHARACTERISTIC_TYPE> opcode, and the sample type is TIME, as defined by the SAMPLE_TYPE> opcode. The value provided should conform to the standard VMS delta time format, "DDD HH:MM:SS.CC". The delta time interval can be no smaller than one second.

Example:
```
SAMPLE_INTERVAL> INTERVAL;   !  Interval
```

SAMPLE_INTERVAL> (cont)

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: String
System Default: 0 00:01:00.00 (once a minute)
Access: Read only
Propagation: Yes

SAMPLE_TIME>

Opcode which defines the timestamp for a sample value when the characteristic type is VARIABLE, as defined by the CHARACTERISTIC_TYPE> opcode, and the sample type is MANUAL, as defined by the SAMPLE_TYPE> opcode. The values supplied should conform with the standard VMS time format, "DD-MMM-YYYY HH:MM:SS.CC". This opcode should be used when a sample timestamp other than the current system time is required for a sample value.

Example:
```
! Time of sample
SAMPLE_TIME>    ACT_SAMPLE_TIME(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: String[23]
System Default: Current time
Access: Read only
Propagation: No

SAMPLE_TYPE>

Opcode which determines the type of sampling being defined for a VARIABLE characteristic; **EVENT**, **TIME**, **COUNT** or **MANUAL**. The sample types **EVENT**, **TIME** and **COUNT** collect sample values based on a given collection criteria.

When **EVENT** sampling is defined, a sample value is acquired when the CRISP application sets the associated trigger, as defined by the SAMPLE_IF> opcode, to non-zero (true). After the sample value has been acquired from the location defined by the ADB_LOCATION> opcode, the trigger is set to zero (false).

When **TIME** sampling is defined, a sample value is acquired automatically from the location defined by the ADB_LOCATION> opcode at the end of a specified time interval, as defined by the SAMPLE_INTERVAL> opcode.

When **COUNT** sampling is defined, a sample value is acquired when the CRISP application sets the associated trigger, as defined by the SAMPLE_IF> opcode, to non-zero (true). After the sample value has been acquired from the location defined by the ADB_LOCATION> opcode, the trigger is set to zero (false). Whether the sample value is stored or not is determined by the values associated with the SUBGROUP_SIZE> and GROUP_SIZE> opcodes. Only the last subgroup size worth of sample values are actually stored; the rest of the group size worth of sample values are ignored.

SAMPLE_TYPE> (cont)

When **MANUAL** sampling is defined, a sample value or subgroup size worth of sample values is acquired when the trigger (defined by the SAMPLE_IF> opcode) goes non-zero (true). The MANUAL_SUBTYPE> opcode determines whether a single sample value or a subgroup size worth of sample values is acquired. After the sample value(s) has been acquired from the location defined by the ADB_LOCATION> opcode, the trigger is set to zero (false). Validity checking is performed on the sample value(s) unless the OVERIDE_VALIDITY_IF> trigger is set to non-zero (true). If the sample value(s) is outside the validity limits, the sample value(s) is ignored.

The following is a summery of the choices for this opcode.

Parameter	Constant	Description
0	EVENT	Event based automatic sampling
1	TIME	Time based automatic sampling
2	COUNT	Count based automatic sampling
3	MANUAL	Manual sampling

Example:
```
SAMPLE_TYPE> ;COUNT   !  Use count sampling
```

UCF Location: Characteristic section
Variable Reference: No
Variable Type: n/a
System Default: EVENT
Access: Read only
Propagation: Yes

SHIFT_NUMBER>

Opcode which defines the shift number for a sample value.

Example:
```
SHIFT_NUMBER> CURRENT_SHIFT;   !  Shift number
```

UCF Location: Initialization or Characteristic section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: 1
Access: Write only
Propagation: Yes

SPC_CHARACTERISTIC>

Opcode which determines the start of a SPC characteristic definition. A SPC characteristic definition section identifies the SPC characteristic name, sample rules, alarm rules, ADB location, specification limits, control limits and validity limits associated with it. All SPC characteristic definition opcodes following the SPC_CHARACTERISTIC> opcode pertain to the same characteristic until the next SPC_CHARACTERISTIC> opcode or END> opcode is encountered. A SPC characteristic definition section and a SPC display station definition are allowed to overlap.

SPC_CHARACTERISTIC> (cont)

The value portion of the statement **must** be a string value, enclosed in quotes, containing the descriptive name of the characteristic being defined. The name may be up to 35 characters in length with A-Z, 0-9, $, and _. A space is NOT a valid character. The name is not case sensitive; all lower case letters will be forced to upper case.

Example: `SPC_CHARACTERISTIC> ;"FURNACE_1_TEMP"  ! Name`

UCF Location: Characteristic section
Variable Reference: No
Variable Type: n/a
System Default: None
Access: Read only
Propagation: No

SPC_DISPLAY_STATION>

Opcode which determines the start of a SPC display station definition. A SPC display station definition section identifies the SPC display station control opcodes and display opcodes associated with it. All SPC display station definition opcodes following the SPC_DISPLAY_STATION> opcode pertain to the same display station until the next SPC_DISPLAY_STATION> opcode or END> opcode is encountered. A SPC display station definition section and a SPC characteristic definition are allowed to overlap.

Example: `SPC_DISPLAY_STATION> ;  ! Display station`

UCF Location: Display station section
Variable Reference: No
Variable Type: n/a
System Default: None
Access: None
Propagation: No

SPEC_ALARM_AT>

Opcode which defines the first of an array of variables representing the alarm status for the raw sample data for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The data is arranged such that sample one of subgroup one is the first bit in the array, followed by the second sample of the first subgroup, etc. When the bit is logically true, it indicates that the represented sample is outside the specification tolerances. The number of variables required in the array is the product of the CTRL_CHART_SIZE> and the SUBGROUP_SIZE> for the characteristic.

This data is continuously maintained by SPC, meaning that the specification alarm statuses are available without referencing the characteristic at a display station.

Example:
```
! Show spec. status here
SPEC_ALARM_AT> SAMPLE_ALMS(0);
```

SPEC_ALARM_AT> (cont)

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

SUBGROUP_SIZE>

Opcode which defines the number of sample values that will be acquired to create a subgroup, a single point on a control chart. This opcode is only meaningful when the characteristic type is VARIABLE, as defined by the CHARACTERISTIC_TYPE> opcode.

Example:
```
SUBGROUP_SIZE> ;6   !  Subgroup size
```

UCF Location: Initialization or Characteristic section
Variable Reference: No
Variable Type: n/a
System Default: 5
Access: Read only
Propagation: Yes

TMP_SUBGROUP_SIZE>

Opcode which defines the number of valid sample values in a subgroup. When the value is a positive number less than the subgroup size, that many sample values will be acquired for the current subgroup. This opcode is recognized only if the characteristic type is VARIABLE, as defined by the CHARACTERISTIC_TYPE> opcode, the type of sampling is MANUAL, as defined by the SAMPLE_TYPE> opcode, and the manual sampling subtype is SUBGROUP, as defined by the MANUAL_SUBTYPE> opcode. The CRISP application is responsible for setting the temporary subgroup size to zero after the subgroup has been acquired. This function is used when acquiring a subgroup's worth of sample values at a time and not enough sample values are available.

Example:
```
!  Actual subgroup size
TMP_SUBGROUP_SIZE>   ACT_SUBGROUP_SIZE;
```

UCF Location: Initialization or Characteristic section
Variable Reference: Required
Variable Type: Numeric, Long
System Default: None
Access: Read only
Propagation: Yes

USL>

Opcode which defines the upper specification limit for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. The upper spec limit is important for performing Cpk/Cp calculations and for boundary definition for the Histogram. When a variable name is included with this opcode, the value is read each time the data sample trigger (defined by the SAMPLE_IF> opcode) is fired.

Example:
```
USL> UPPER_SPEC_LMT;   !  Upper spec limit
```

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: 90.0
Access: Read only
Propagation: Yes

UVL>

Opcode which defines the upper limit for manually entered sample values which will not cause a validity alarm. Validity alarm checking is performed only when the characteristic type is VARIABLE, as defined by the CHARACTERISTIC_TYPE> opcode, and the sample type is MANUAL, as defined by the SAMPLE_TYPE> opcode. Any value lower than the specified limit will cause a validity alarm, which will be reported if the VALIDITY_ALARM_AT> opcode has been specified for the characteristic. When a validity alarm occurs, the sample value will be ignored. The OVERRIDE_VALIDITY_IF> opcode may be used to disable the validity checking and allow the acquisition of sample values which are outside of the validity limits. When a variable name is included with this opcode, the value is read each time the data sample trigger (defined by the SAMPLE_IF> opcode) is fired.

Example:
```
!  Upper validity limit
UVL> UPPER_VALIDITY_LIMIT;
```

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: 100000000
Access: Read only
Propagation: Yes

VALIDITY_ALARM_AT>

Opcode which defines the location in the ADB where the current validity alarm states will be maintained by SPC. This opcode only applies for a sample type of MANUAL, as defined by SAMPLE_TYPE> opcode. When a sample value falls outside the validity limits, all sample values acquired at that time will be ignored by SPC unless the OVERRIDE_VALIDITY_IF> opcode value is non-zero (true).

Example:
```
VALIDITY_ALARM_AT>  BAD_DATA_ALM(0);   !  Validity
alarms
```

VALIDITY_ALARM_AT> (cont)

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: 0
Access: Write only
Propagation: No

XLCL>

Opcode which defines the lower mean control limit for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. Any value lower than the specified limit will cause a control alarm. When a variable name is included with this opcode, the value is read each time the data sample trigger (defined by the SAMPLE_IF> opcode) is fired. If the limit calculation function is used (as defined by the CALC_CONTROL_LIMITS_IF> opcode) then the current value associated with this opcode will be revised.

Example:
```
!  lower mean control limit
XLCL> RS_LOWER_CONTROL_LMT;
```

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: 30.0
Access: Read/Write
Propagation: Yes

XUCL>

Opcode which defines the upper mean control limit for the SPC characteristic defined by the DISPLAY_CHARACTERISTIC> opcode. Any value higher than the specified limit will cause a control alarm. When a variable name is included with this opcode, the value is read each time the data sample trigger (defined by the SAMPLE_IF> opcode) is fired. If the limit calculation function is used (as defined by the CALC_CONTROL_LIMITS_IF> opcode) then the current value associated with this opcode will be revised.

Example:
```
!  Upper mean control limit
XUCL> RS_UPPER_CONTROL_LMT;
```

UCF Location: Initialization or Characteristic section
Variable Reference: Optional
Variable Type: Float
System Default: 70.0
Access: Read/Write
Propagation: Yes

X_BAR_7_ABOVE_ALARM_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive mean control points being above the grand mean. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup means are found to be above the grand mean, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition.

Example:
```
!  7 pts above alarms
X_BAR_7_ABOVE_AT>  X_BAR_ALM_7ABOVE(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

X_BAR_7_BELOW_ALARM_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive mean control points being below the grand mean. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup means are found to be below the grand mean, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition.

Example:
```
!  7 pts below alarms
X_BAR_7_BELOW_AT>  X_BAR_ALM_7BELOW(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

X_BAR_7_DECR_ALARM_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive mean control points in a steadily decreasing pattern. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup mean values are found to be smaller than their predecessors, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition.

Example:
```
!  7 pts decreasing alarms
X_BAR_7_DECR_AT>  X_BAR_ALM_7DECR(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

X_BAR_7_INCR_ALARM_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by seven consecutive mean control points in a steadily increasing pattern. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

This opcode is used to find statistically significant patterns in the data. When seven consecutive subgroup mean values are found to be larger than their predecessors, the alarm variable corresponding to the seventh subgroup will be set true, indicating that the process may be tending toward an out of control condition.

Example:
```
!  7 pts increasing alarms
X_BAR_7_INCR_AT> X_BAR_ALM_7INCR(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

X_BAR_CTRL_ALARM_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by a single mean control point exceeding the limits defined by the XUCL> or XLCL> opcodes. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). When a subgroup is either larger than the upper control limit, the alarm variable corresponding to the subgroup in alarm will be set true. This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

Example:
```
!  X_bar Ctrl alarms
X_BAR_CTRL_ALARM_AT> ALM_X_MEAN(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

X_BAR_CTRL_WARN_AT>

Opcode which defines the first of a control chart set of consecutive variables representing alarms caused by a single mean control point exceeding the SPC-calculated warning limits. The number of variables required in the array is defined by the CTRL_CHART_SIZE> opcode for the characteristic (default 30). A control point falls in the warning band if its value is greater than the upper control limit minus five percent of the specification spread, or is less than the lower control limit plus 5 percent of the specification spread, yet not outside the control limits.

XUCL > Upper warning band > XUCL - (.05 * (XUSL - XLSL))
XLCL < Lower warning band < XLCL + (.05 * (XUSL - XLSL))

When the subgroup falls in this band, the warning bit will be set true. This alarm status is continuously maintained by SPC, meaning that the alarm status is available without referencing the characteristic at a display station.

Example:
```
!  X Warning alarms
X_BAR_CTRL_WARN_AT> ALM_X_WARNING(0);
```

UCF Location: Characteristic section
Variable Reference: Required
Variable Type: Logical
System Default: None
Access: Write only
Propagation: No

X_TARGET>

Opcode which defines the ideal average mean of the sample values in the subgroup. The data referenced by this opcode is used for building chart graphics and is not part of any supporting calculations.

Example: `X_TARGET> ;50.  ! mean target`

UCF Location:	Initialization or Characteristic section
Variable Reference:	Optional
Variable Type:	Float
System Default:	(XUCL + XLCL) / 2
Access:	Read only
Propagation:	Yes

Notes:

General

The keywords (called Token), which may be used in place of literal numeric values, are listed here in alphabetical order. Refer to the specific opcode for more details.

Token	Value	Opcode(s)
ALARM	0	DISPLAY_PARETO_TYPE>
ARCHIVED	0	DISPLAY_USING>
ATTRIBUTE	1	CHARACTERISTIC_TYPE>
C	4	CTRL_CHART_TYPE>
CALCULATED	1	DISPLAY_USING>
COMMENT	1	DISPLAY_PARETO_TYPE>
COUNT	2	SAMPLE_TYPE>
CRISP/32	0	ADB_TYPE>
CURRENT	1	CALC_CTRL_LIMITS_TYPE>
EVENT	0	SAMPLE_TYPE>
FUTURE	0	CALC_CTRL_LIMITS_TYPE>
MANUAL	3	SAMPLE_TYPE>
NP	3	CTRL_CHART_TYPE>
P	2	CTRL_CHART_TYPE>
SAMPLE	0	MANUAL_SUBTYPE>

General (cont)

Token	Value	Opcode(s)
SUBGROUP	1	MANUAL_SUBTYPE>
TIME	1	SAMPLE_TYPE>
U	5	CTRL_CHART_TYPE>
VARIABLE	0	CHARACTERISTIC_TYPE>
X_BAR_AND_R	0	CTRL_CHART_TYPE>
X_BAR_AND_S	1	CTRL_CHART_TYPE>

```
%SERVICES-I-INFO_MSG, The following is a Product Specific Message:
%C32_SPC -E-RSVLFAIL Resolve of CRISP32/DATABASE:VARIABLE FAILED, %ERR: 134381914

             This Variable will NOT be accessed !
```

Message Type: Error.
Explanation: The DATABASE and/or VARIABLE defined in the UCF does not exist.
User Action: Check the spelling of the referenced variable against the C32 file declarations.

```
%SERVICES-I-INFO_MSG, The following is a Product Specific Message:
C32_SPC - No such ADB symbol "VARIABLE", for opcode "SPC_OPCODE>"
```

Message Type: Error.
Explanation: The DATABASE and/or VARIABLE defined in the UCF does not exist.
User Action: Check the spelling of the referenced variable against the C32 file declarations.

```
%SERVICES-I-INFO_MSG, The following is a Product Specific Message:
INI_0 - Characteristic CHAR_NAME has not been initialized, code 00018292
```

Message Type: Information.
Explanation: No data has been stored for characteristic CHAR_NAME at the present time.
User Action: None required.

```
%SERVICES-I-INFO_MSG, The following is a Product Specific Message:
C32_SPC - SPC Initializer has been activated
%SPCARC-I-HPF, Storing data in Historian point file format
```

Message Type: Information.
Explanation: A temporary process called, "the Initializer" has begun reading historical data (found in point files) into memory to support real-time SPC processing.

```
%SPCINI-I-EXITMSG, Exit message received at 10:52:08.60
```

Message Type: Information.
Explanation: A temporary process called, "the Initializer" has completed reading historical data (found in point files) into memory. The SPC product is now ready to accept new data.

Notes:

General

The following definitions are explanations of technical terms, acronyms, and mnemonics.

ADB — Application database (ex. a user-created database created from a C32 file definition).

Attribute SPC — The analysis of data that may be judged as either present or absent, acceptable or defective (ex. bubbles of air in a windscreen, general appearance of a painted surface, etc).

Binomial Distribution — A table of probabilities which govern the behavior of Attribute SPC for P and NP charts.

C Chart — An Attribute control chart showing the number of defects being produced by a process for a constant sample (lot) size.

Capability Index (Cp and Cpk) — Cp: indicates the spread of the process.

Cpk: indicates the spread and the setting of the process.

Values for Cp and Cpk greater than 1 indicate that the process is capable of achieving the specified tolerances.

Characteristic — A collection of opcodes in the User Configuration file which define the sampling and calculation rules for the SPC product.

Control Chart — The general name given to a set of SPC charts (ex. XBAR_AND_R, P, U, etc.)

Control Point — A point plotted on an XBAR control chart. The average of a subgroup of data samples.

Cusum Chart (CUSUM) — A chart of the accumulation of the positive and negative deviations between consecutive samples which is compared against a user-defined decision interval (limit). An alternate charting technique to the Shewhart control chart.

Defective(s) — The measured attribute used to create NP and P charts.

Defect(s) — The measured attribute used to create C and U charts.

Display station — A collection of opcodes in the User Configuration file which define the data presentation rules for the SPC product.

Gaussian Distribution — A table of probabilities which govern the behavior of Variable SPC for XBAR_AND_R and XBAR_AND_S charts.

Grand Mean — The average of mean data contained in a control chart.

Group — The population of data samples from which a single subgroup of data is taken.

Histogram — A chart which shows sample data as a frequency distribution.

LCL Lower Control Limit: A user-defined limit used for control chart alarming on a XBAR, NP, P, C or U chart.

LSL Lower Specification Limit: A user-defined limit which defines the engineering limits for acceptable product. Used for Cp, Cpk and Histogram calculations.

LVL Lower Validity Limit: A user-defined limit used for edit-checking of manually-entered sample data.

Lot Size The sample size removed from the process for the purpose of measuring attributes.

Moving Range Chart A chart used for Variable SPC analysis where the subgroup size is equal to 1. The moving range value is calculated by computing the absolute value of the difference between the current sample point and the previous sample point.

NP Chart An Attribute control chart showing the number of defective being produced by a process for a constant sample (lot) size.

Opcode Operation code: statements in a UCF file which configure a layered product to perform a function.

P Chart An Attribute control chart showing the proportion defective being produced by a process for a varying sample (lot) size.

Pareto Chart A chart which sorts and arranges SPC alarm code or comment code data according to frequency of occurrence.

Poisson Distribution A table of probabilities which govern the behavior of Attribute SPC for C and U charts.

Range Chart A chart used for Variable SPC analysis where the subgroup size is greater than 1. The range value is calculated by finding the min. and max. sample within a subgroup and computing the absolute value of the difference.

RSLCL Range/Standard Deviation Lower Control Limit: A user-defined limit used for control chart alarming on a XBAR_AND_R or XBAR_AND_S chart.

Sample The value taken for purposes of performing SPC analysis. Multiple samples go into making a subgroup.

Shewhart Chart Another name for the XBAR_AND_R and XBAR_AND_S control charts.

Standard Deviation A measure of the absolute deviation of samples from the mean.

Subgroup A contiguous collection of samples which are used to perform XBAR, Range and Standard Deviation calculations.

U Chart An Attribute control chart showing the number of defects being produced by a process for a varying sample (lot) size.

UCF	User Configuration File: a collection of Opcodes and data values which configure a layered product for a custom application.
UCL	Upper Control Limit: A user-defined limit used for control chart alarming on a XBAR, NP, P, C or U chart.
USL	Upper Specification Limit: A user-defined limit which defines the engineering limits for acceptable product. Used for Cp, Cpk and Histogram calculations.
UVL	Upper Validity Limit: A user-defined limit used for edit-checking of manually-entered sample data.
Variable SPC	The analysis of data which has been measured (ex. rod ;etc.)
XBAR Chart	A chart used for Variable SPC analysis where the mean of subgroup size of sample data is plotted against upper and lower control limits.

Notes:

Appendix A

Control Limit Calculation Constants						
Subgroup Size	A2	A3	B3	B4	D3	D4
2	1.880	2.659	0.000	3.267	0.000	3.267
3	1.023	1.954	0.000	2.568	0.000	2.574
4	0.729	1.628	0.000	2.266	0.000	2.282
5	0.577	1.427	0.000	2.089	0.000	2.114
6	0.483	1.287	0.030	1.970	0.000	2.004
7	0.419	1.182	0.118	1.882	0.076	1.924
8	0.373	1.099	0.185	1.815	0.136	1.864
9	0.337	1.032	0.239	1.761	0.184	1.816
10	0.308	0.975	0.284	1.716	0.223	1.777

Notes:

Appendix B

Historian point file directory definitions

Index	Subdirectory	Data Description
1	SPC010	Control
2	SPC020	Subgroup size
3	SPC030	Control chart type
4	SPC040	Sample type
5	SPC050	Upper specification limit
6	SPC060	Lower specification limit
7	SPC070	Upper validity limit
8	SPC080	Lower validity limit
9	SPC090	Mean upper control limit
10	SPC100	Mean lower control limit
11	SPC110	Range/Stddev upper control limit
12	SPC120	Range/Stddev lower control limit
13	SPC130	Mean target
14	SPC140	Range/Stddev target
15	SPC150	CUSUM limit
16	SPC160	CUSUM slack
17	SPC170	CUSUM target
18	SPC180	Subgroup control
19	SPC190	Subgroup alarm assistance
20	SPC200	Subgroup alarm
21	SPC210	Subgroup alarm code
22	SPC220	Subgroup CUSUM high
23	SPC230	Subgroup CUSUM low
24	SPC240	Subgroup shift number
25	SPC250 - SPC259	Subgroup comment code
26	SPC260 - SPC269	Subgroup timestamp
27	SPC270	Variable subgroup sample size
28	SPC280	Variable subgroup group size
29	SPC290 - SPC299	Variable subgroup value
30	SPC300	Attribute subgroup defect number
31	SPC310	Attribute subgroup lot size

Notes:

Appendix C - Example

```
!=============================================================================
!
!
!      CRISP STATISICAL PROCESS CONTROL SOFTWARE
!
!               USER CONFIGURATION FILE
!
!
!=============================================================================
!
!      Char      Data       Chart          Sample    Subgroup
!      Name      Type       Type           Type      Size
!      -------   --------   -----------    -------   --------
!
! 1.   CHAR_01   VARIABLE   X_BAR_AND_R    EVENT     1
!
! 2.   CHAR_02   VARIABLE   X_BAR_AND_R    TIME      1
!
! 3.   CHAR_03   VARIABLE   X_BAR_AND_R    MANUAL    5
!                                         (SAMPLE)
!
! 4.   CHAR_04   VARIABLE   X_BAR_AND_S    MANUAL    5
!                                         (SUBGRP)
!
! 5.   CHAR_05   VARIABLE   X_BAR_AND_S    COUNT     5
!
! 6.   CHAR_06   ATTRIBUTE  NP             MANUAL
!                                         (SAMPLE)
!
! 7.   CHAR_07   ATTRIBUTE  P              EVENT
!
! 8.   CHAR_08   ATTRIBUTE  C              MANUAL
!                                         (SAMPLE)
!
! 9.   CHAR_09   ATTRIBUTE  U              EVENT
!
!=============================================================================

!      Enter Product license information here
!      --------------------------------------
!
DEFINE>>

       CUSTOMER_NAME>>                        ;"CUSTOMER NAME"
       CUSTOMER_LOCATION>>                    ;"CUSTOMER LOCATION"
       CUSTOMER_SW_LICENSE>>        ;1234567890
       CUSTOMER_USE_LIMIT>>                   ;500
       CUSTOMER_CONFIG_CODE>>                 ;0987654321
       CUSTOMER_CONFIG_ID>>                   ;1234567890

!
!      Enter Global Opcodes here
!      -------------------------
!
INIT>>

       ADB_TYPE>                              ;CRISP32             ! CRISP type database
       ADB_IDENT>                             ;"SPCTST"            ! Database name
       SHIFT_NUMBER>                          SHIFT_NO;            ! Current shift ID
       CALC_CTRL_LIMITS_TYPE>                 ;CURRENT             ! Use existing data
       CTRL_CHART_SIZE>                       ;30        ! 30 control points in memory
```

```
!
!      Define Characteristics here
!      ---------------------------
!

PROCESS>>

!
!      ------------------------------------------------
!      Variable Characteristic for EVENT-Based sampling
!      ------------------------------------------------
!
       SPC_CHARACTERISTIC>                    ;"CHAR_01"

              CHARACTERISTIC_TYPE>            ;VARIABLE
              CTRL_CHART_TYPE>                ;X_BAR_AND_R
              SUBGROUP_SIZE>              ;1                    ! 1 sample per control pt.

              SAMPLE_TYPE>                    ;EVENT

              ADB_LOCATION>                   ;"DATA_VALUE_01"      ! Sample variable
              SAMPLE_IF>                      READ_DATA_TRIG(1);

              USL>                            CFG_XUSL(1);
              XUCL>                           CFG_XUCL(1);
              RSUCL>                          CFG_RSUCL(1);
              LSL>                            CFG_XLSL(1);
              XLCL>                           CFG_XLCL(1);
              RSLCL>                          CFG_RSLCL(1);
              CPK_LIMIT>                      CFG_CPK_LCL(1);
              CUSUM_LIMIT>                    CFG_CUSUM_H(1);
              CUSUM_SLACK>                    CFG_CUSUM_K(1);
              CUSUM_TARGET>                   CFG_CUSUM_TGT(1);

              CALC_CTRL_LIMITS_IF>            CALC_LIMITS_TRIG(1);
              HISTOGRAM_RESET_IF>             HISTO_RESET_TRIG(1);
              HISTOGRAM_RESET_TIME_AT>        HISTO_RESET_TIME(1);
              PARETO_RESET_IF>                PARETO_RESET_TRIG(1);
              PARETO_RESET_TIME_AT>       PARETO_RESET_TIME(1);
              CLEAR_CHARACTERISTIC_IF>        CLEAR_CHAR_TRIG(1);   ! Start over

              CPK_ALARM_AT>                   CPK_ALRM_01;
              X_BAR_CTRL_ALARM_AT>            X_BAR_CTRL_ALRM_01(0);! 30 control pts.
              X_BAR_7_ABOVE_ALARM_AT>         X_BAR_7ABV_ALRM_01(0);
              X_BAR_7_BELOW_ALARM_AT>         X_BAR_7BLW_ALRM_01(0);
              X_BAR_7_DECR_ALARM_AT>          X_BAR_7DEC_ALRM_01(0);
              X_BAR_7_INCR_ALARM_AT>          X_BAR_7INC_ALRM_01(0);
              X_BAR_CTRL_WARN_AT>             X_BAR_WARN_ALRM_01(0);
              RS_BAR_CTRL_ALARM_AT>       RANGE_CTRL_ALRM_01(0);
              RS_BAR_7_ABOVE_ALARM_AT>        RANGE_7ABV_ALRM_01(0);
              RS_BAR_7_BELOW_ALARM_AT>        RANGE_7BLW_ALRM_01(0);
              RS_BAR_7_DECR_ALARM_AT>         RANGE_7DEC_ALRM_01(0);
              RS_BAR_7_INCR_ALARM_AT>         RANGE_7INC_ALRM_01(0);
              RS_BAR_CTRL_WARN_AT>            RANGE_WARN_ALRM_01(0);
              CUSUM_HIGH_ALARM_AT>            CUSUM_HI_ALRM_01(0);
              CUSUM_LOW_ALARM_AT>             CUSUM_LO_ALRM_01(0);
```

```
!
!       ------------------------------------------------
!       Variable Characteristic for TIME-Based sampling
!       ------------------------------------------------
!
        SPC_CHARACTERISTIC>                     ;"CHAR_02"

                CHARACTERISTIC_TYPE>            ;VARIABLE
                CTRL_CHART_TYPE>                ;X_BAR_AND_R
                SUBGROUP_SIZE>          ;1                          ! 1 sample per subgroup

                SAMPLE_TYPE>                    ;TIME
                SAMPLE_INTERVAL>                ;"0 01:00:00.00"      ! 1 sample each hour

                ADB_LOCATION>                   ;"DATA_VALUE_02"

                USL>                            CFG_XUSL(2);
                XUCL>                           CFG_XUCL(2);
                RSUCL>                          CFG_RSUCL(2);
                LSL>                            CFG_XLSL(2);
                XLCL>                           CFG_XLCL(2);
                RSLCL>                          CFG_RSLCL(2);
                CPK_LIMIT>                      CFG_CPK_LCL(2);
                CUSUM_LIMIT>                    CFG_CUSUM_H(2);
                CUSUM_SLACK>                    CFG_CUSUM_K(2);
                CUSUM_TARGET>                   CFG_CUSUM_TGT(2);

                CALC_CTRL_LIMITS_IF>            CALC_LIMITS_TRIG(2);
                HISTOGRAM_RESET_IF>             HISTO_RESET_TRIG(2);
                HISTOGRAM_RESET_TIME_AT>        HISTO_RESET_TIME(2);
                PARETO_RESET_IF>                PARETO_RESET_TRIG(2);
                PARETO_RESET_TIME_AT> PARETO_RESET_TIME(2);
                CLEAR_CHARACTERISTIC_IF>        CLEAR_CHAR_TRIG(2);

                CPK_ALARM_AT>                   CPK_ALRM_02;
                X_BAR_CTRL_ALARM_AT>            X_BAR_CTRL_ALRM_02(0);! 30 control pts.
                X_BAR_7_ABOVE_ALARM_AT>         X_BAR_7ABV_ALRM_02(0);
                X_BAR_7_BELOW_ALARM_AT>         X_BAR_7BLW_ALRM_02(0);
                X_BAR_7_DECR_ALARM_AT>          X_BAR_7DEC_ALRM_02(0);
                X_BAR_7_INCR_ALARM_AT>          X_BAR_7INC_ALRM_02(0);
                X_BAR_CTRL_WARN_AT>             X_BAR_WARN_ALRM_02(0);
                RS_BAR_CTRL_ALARM_AT> RANGE_CTRL_ALRM_02(0);
                RS_BAR_7_ABOVE_ALARM_AT>        RANGE_7ABV_ALRM_02(0);
                RS_BAR_7_BELOW_ALARM_AT>        RANGE_7BLW_ALRM_02(0);
                RS_BAR_7_DECR_ALARM_AT>         RANGE_7DEC_ALRM_02(0);
                RS_BAR_7_INCR_ALARM_AT>         RANGE_7INC_ALRM_02(0);
                RS_BAR_CTRL_WARN_AT>            RANGE_WARN_ALRM_02(0);
                CUSUM_HIGH_ALARM_AT>            CUSUM_HI_ALRM_02(0);
                CUSUM_LOW_ALARM_AT>             CUSUM_LO_ALRM_02(0);
```

```
!
!      ------------------------------------------------------------
!      Variable Characteristic for MANUAL-Based sampling (Sample)
!      ------------------------------------------------------------
!
       SPC_CHARACTERISTIC>                    ;"CHAR_03"

              CHARACTERISTIC_TYPE>            ;VARIABLE
              CTRL_CHART_TYPE>                ;X_BAR_AND_R
              SUBGROUP_SIZE>       ;5                          ! 5 samples per control pt.

              SAMPLE_TYPE>                    ;MANUAL
              MANUAL_SUBTYPE>                 ;SAMPLE              ! Read single sample

              ADB_LOCATION>                   ;"DATA_VALUE_03"
              COMMENT_CODE>                   COMMENT_03(0);       ! 1 comment per sample
              ALARM_CODE>                     ALRM_CODE_03;        ! 1 comment per subgroup
              SAMPLE_IF>                      READ_DATA_TRIG(3);
              END_SUBGROUP_IF>                END_SUBGROUP_TRIG(3); ! Trigger if subgroup < 5

              VALIDITY_ALARM_AT>              VALIDITY_ALRM_03;    !
              OVERRIDE_VALIDITY_IF> VALIDITY_OVERRIDE_ENABLE(3);

              UVL>                            CFG_XUVL(3);
              USL>                            CFG_XUSL(3);
              XUCL>                           CFG_XUCL(3);
              RSUCL>                          CFG_RSUCL(3);
              LVL>                            CFG_XLVL(3);
              LSL>                            CFG_XLSL(3);
              XLCL>                           CFG_XLCL(3);
              RSLCL>                          CFG_RSLCL(3);
              CPK_LIMIT>                      CFG_CPK_LCL(3);
              CUSUM_LIMIT>                    CFG_CUSUM_H(3);
              CUSUM_SLACK>                    CFG_CUSUM_K(3);
              CUSUM_TARGET>                   CFG_CUSUM_TGT(3);

              CALC_CTRL_LIMITS_IF>            CALC_LIMITS_TRIG(3);
              HISTOGRAM_RESET_IF>             HISTO_RESET_TRIG(3);
              HISTOGRAM_RESET_TIME_AT>        HISTO_RESET_TIME(3);
              PARETO_RESET_IF>                PARETO_RESET_TRIG(3);
              PARETO_RESET_TIME_AT> PARETO_RESET_TIME(3);
              CLEAR_CHARACTERISTIC_IF>        CLEAR_CHAR_TRIG(3);

              CPK_ALARM_AT>                   CPK_ALRM_03;
              X_BAR_CTRL_ALARM_AT>            X_BAR_CTRL_ALRM_03(0); ! 30 control pts.
              X_BAR_7_ABOVE_ALARM_AT>         X_BAR_7ABV_ALRM_03(0);
              X_BAR_7_BELOW_ALARM_AT>         X_BAR_7BLW_ALRM_03(0);
              X_BAR_7_DECR_ALARM_AT>          X_BAR_7DEC_ALRM_03(0);
              X_BAR_7_INCR_ALARM_AT>          X_BAR_7INC_ALRM_03(0);
              X_BAR_CTRL_WARN_AT>             X_BAR_WARN_ALRM_03(0);
              RS_BAR_CTRL_ALARM_AT> RANGE_CTRL_ALRM_03(0);
              RS_BAR_7_ABOVE_ALARM_AT>        RANGE_7ABV_ALRM_03(0);
              RS_BAR_7_BELOW_ALARM_AT>        RANGE_7BLW_ALRM_03(0);
              RS_BAR_7_DECR_ALARM_AT>         RANGE_7DEC_ALRM_03(0);
              RS_BAR_7_INCR_ALARM_AT>         RANGE_7INC_ALRM_03(0);
              RS_BAR_CTRL_WARN_AT>            RANGE_WARN_ALRM_03(0);
              CUSUM_HIGH_ALARM_AT>            CUSUM_HI_ALRM_03(0);
              CUSUM_LOW_ALARM_AT>             CUSUM_LO_ALRM_03(0);
```

```
!
!       ------------------------------------------------
!       Characteristic for MANUAL-Based sampling (Subgroup)
!       ------------------------------------------------
!
        SPC_CHARACTERISTIC>                     ;"CHAR_04"

                CHARACTERISTIC_TYPE>            ;VARIABLE
                CTRL_CHART_TYPE>                ;X_BAR_AND_S
                SUBGROUP_SIZE>        ;5                        ! 5 samples per control pt.
                TMP_SUBGROUP_SIZE>              TMP_SUBGROUP_SIZE_02; ! Enter if subgroup < 5

                SAMPLE_TYPE>                    ;MANUAL
                MANUAL_SUBTYPE>                 ;SUBGROUP             ! Read total subgroup

                ADB_LOCATION>                   ;"DATA_VALUE_04(0)"   ! data samples (5)
                SAMPLE_TIME>                    SAMPLE_TIME_04(0);    ! Actual sample times (optional)
                COMMENT_CODE>                   COMMENT_04(0);        ! data comment codes (5)
                ALARM_CODE>                     ALRM_CODE_04;         ! 1 comment per subgroup
                SAMPLE_IF>                      READ_DATA_TRIG(4);

                VALIDITY_ALARM_AT>              VALIDITY_ALRM_04(0);  ! 1 alarm per sample (5)
                OVERRIDE_VALIDITY_IF> VALIDITY_OVERRIDE_ENABLE(4);

                UVL>                            CFG_XUVL(4);
                USL>                            CFG_XUSL(4);
                XUCL>                           CFG_XUCL(4);
                RSUCL>                          CFG_RSUCL(4);
                LVL>                            CFG_XLVL(4);
                LSL>                            CFG_XLSL(4);
                XLCL>                           CFG_XLCL(4);
                RSLCL>                          CFG_RSLCL(4);
                CUSUM_LIMIT>                    CFG_CUSUM_H(4);
                CUSUM_SLACK>                    CFG_CUSUM_K(4);
                CUSUM_TARGET>                   CFG_CUSUM_TGT(4);

                CALC_CTRL_LIMITS_IF>            CALC_LIMITS_TRIG(4);
                HISTOGRAM_RESET_IF>             HISTO_RESET_TRIG(4);
                HISTOGRAM_RESET_TIME_AT>        HISTO_RESET_TIME(4);
                PARETO_RESET_IF>                PARETO_RESET_TRIG(4);
                PARETO_RESET_TIME_AT> PARETO_RESET_TIME(3);

                CPK_ALARM_AT>                   CPK_ALRM_04;
                X_BAR_CTRL_ALARM_AT>            X_BAR_CTRL_ALRM_04(0); ! 30 control pts.
                X_BAR_7_ABOVE_ALARM_AT>         X_BAR_7ABV_ALRM_04(0);
                X_BAR_7_BELOW_ALARM_AT>         X_BAR_7BLW_ALRM_04(0);
                X_BAR_7_DECR_ALARM_AT>          X_BAR_7DEC_ALRM_04(0);
                X_BAR_7_INCR_ALARM_AT>          X_BAR_7INC_ALRM_04(0);
                X_BAR_CTRL_WARN_AT>             X_BAR_WARN_ALRM_04(0);
                RS_BAR_CTRL_ALARM_AT> RANGE_CTRL_ALRM_04(0);
                RS_BAR_7_ABOVE_ALARM_AT>        RANGE_7ABV_ALRM_04(0);
                RS_BAR_7_BELOW_ALARM_AT>        RANGE_7BLW_ALRM_04(0);
                RS_BAR_7_DECR_ALARM_AT>         RANGE_7DEC_ALRM_04(0);
                RS_BAR_7_INCR_ALARM_AT>         RANGE_7INC_ALRM_04(0);
                RS_BAR_CTRL_WARN_AT>            RANGE_WARN_ALRM_04(0);
                CUSUM_HIGH_ALARM_AT>            CUSUM_HI_ALRM_04(0);
                CUSUM_LOW_ALARM_AT>             CUSUM_LO_ALRM_04(0);
```

```
!
!      ------------------------------------------------
!      Variable Characteristic for COUNT-Based sampling
!      ------------------------------------------------
!
       SPC_CHARACTERISTIC>                    ;"CHAR_05"

              CHARACTERISTIC_TYPE>            ;VARIABLE
              CTRL_CHART_TYPE>                ;X_BAR_AND_S
              GROUP_SIZE>                     ;100                ! 100 samples per group
              SUBGROUP_SIZE>       ;5                          ! 5 samples per control pt.

              SAMPLE_TYPE>                    ;COUNT

              ADB_LOCATION>                   ;"DATA_VALUE_05"     ! Sample variable
              SAMPLE_IF>                      READ_DATA_TRIG(5);

              USL>                            CFG_XUSL(5);
              XUCL>                           CFG_XUCL(5);
              RSUCL>                          CFG_RSUCL(5);
              LSL>                            CFG_XLSL(5);
              XLCL>                           CFG_XLCL(5);
              RSLCL>                          CFG_RSLCL(5);
              CPK_LIMIT>                      CFG_CPK_LCL(5);
              CUSUM_LIMIT>                    CFG_CUSUM_H(5);
              CUSUM_SLACK>                    CFG_CUSUM_K(5);
              CUSUM_TARGET>                   CFG_CUSUM_TGT(5);

              CALC_CTRL_LIMITS_IF>            CALC_LIMITS_TRIG(5);
              HISTOGRAM_RESET_IF>             HISTO_RESET_TRIG(5);
              HISTOGRAM_RESET_TIME_AT>        HISTO_RESET_TIME(5);
              PARETO_RESET_IF>                PARETO_RESET_TRIG(5);
              PARETO_RESET_TIME_AT> PARETO_RESET_TIME(5);
              CLEAR_CHARACTERISTIC_IF>        CLEAR_CHAR_TRIG(5);   ! Start over

              CPK_ALARM_AT>                   CPK_ALRM_05;
              X_BAR_CTRL_ALARM_AT>            X_BAR_CTRL_ALRM_05(0);! 30 control pts.
              X_BAR_7_ABOVE_ALARM_AT>         X_BAR_7ABV_ALRM_05(0);
              X_BAR_7_BELOW_ALARM_AT>         X_BAR_7BLW_ALRM_05(0);
              X_BAR_7_DECR_ALARM_AT>          X_BAR_7DEC_ALRM_05(0);
              X_BAR_7_INCR_ALARM_AT>          X_BAR_7INC_ALRM_05(0);
              X_BAR_CTRL_WARN_AT>             X_BAR_WARN_ALRM_05(0);
              RS_BAR_CTRL_ALARM_AT> RANGE_CTRL_ALRM_05(0);
              RS_BAR_7_ABOVE_ALARM_AT>        RANGE_7ABV_ALRM_05(0);
              RS_BAR_7_BELOW_ALARM_AT>        RANGE_7BLW_ALRM_05(0);
              RS_BAR_7_DECR_ALARM_AT>         RANGE_7DEC_ALRM_05(0);
              RS_BAR_7_INCR_ALARM_AT>         RANGE_7INC_ALRM_05(0);
              RS_BAR_CTRL_WARN_AT>            RANGE_WARN_ALRM_05(0);
              CUSUM_HIGH_ALARM_AT>            CUSUM_HI_ALRM_05(0);
              CUSUM_LOW_ALARM_AT>             CUSUM_LO_ALRM_05(0);
```

```
!
!       ------------------------------------------------
!       Attribute Cracteristic for Manual-Based sampling
!       ------------------------------------------------
!
        SPC_CHARACTERISTIC>                     ;"CHAR_06"

                CHARACTERISTIC_TYPE>            ;ATTRIBUTE
                CTRL_CHART_TYPE>                ;NP              ! np chart
                LOT_SIZE>                       ;100                ! Constant 100 samples per lot

                SAMPLE_TYPE>                    ;MANUAL
                MANUAL_SUBTYPE>                 ;SAMPLE             ! Read single sample

                ADB_LOCATION>                   ;"DATA_VALUE_06"    ! Number defective
                COMMENT_CODE>                   COMMENT_06;         ! 1 comment per sample
                ALARM_CODE>                     ALRM_CODE_06;       ! 1 comment per sample
                SAMPLE_IF>                      READ_DATA_TRIG(6);

                UVL>                            CFG_XUVL(6);
                USL>                            CFG_XUSL(6);
                XUCL>                           CFG_XUCL(6);
                LVL>                            CFG_XLVL(6);
                LSL>                            CFG_XLSL(6);
                XLCL>                           CFG_XLCL(6);
                CUSUM_LIMIT>                    CFG_CUSUM_H(6);
                CUSUM_SLACK>                    CFG_CUSUM_K(6);
                CUSUM_TARGET>                   CFG_CUSUM_TGT(6);

                CALC_CTRL_LIMITS_IF>            CALC_LIMITS_TRIG(6);
                HISTOGRAM_RESET_IF>             HISTO_RESET_TRIG(6);
                HISTOGRAM_RESET_TIME_AT>        HISTO_RESET_TIME(6);
                PARETO_RESET_IF>                PARETO_RESET_TRIG(6);
                PARETO_RESET_TIME_AT> PARETO_RESET_TIME(6);
                CLEAR_CHARACTERISTIC_IF>        CLEAR_CHAR_TRIG(6);   ! Start over

                X_BAR_CTRL_ALARM_AT>            X_BAR_CTRL_ALRM_06(0);! 30 control pts.
                X_BAR_7_ABOVE_ALARM_AT>         X_BAR_7ABV_ALRM_06(0);
                X_BAR_7_BELOW_ALARM_AT>         X_BAR_7BLW_ALRM_06(0);
                X_BAR_7_DECR_ALARM_AT>          X_BAR_7DEC_ALRM_06(0);
                X_BAR_7_INCR_ALARM_AT>          X_BAR_7INC_ALRM_06(0);
                X_BAR_CTRL_WARN_AT>             X_BAR_WARN_ALRM_06(0);
                CUSUM_HIGH_ALARM_AT>            CUSUM_HI_ALRM_06(0);
                CUSUM_LOW_ALARM_AT>             CUSUM_LO_ALRM_06(0);
```

```
!
!      ------------------------------------------------
!      Attribute Characteristic for EVENT-Based sampling
!      ------------------------------------------------
!
       SPC_CHARACTERISTIC>                    ;"CHAR_07"

              CHARACTERISTIC_TYPE>            ;ATTRIBUTE
              CTRL_CHART_TYPE>                ;P                    ! P chart
              LOT_SIZE>                       ;LOT_SIZE_07          ! Varying Lot size

              SAMPLE_TYPE>                    ;EVENT

              ADB_LOCATION>                   ;"DATA_VALUE_07"      ! Sample variable
              SAMPLE_IF>                      READ_DATA_TRIG(7);    ! Proportion Defective

              USL>                            CFG_XUSL(7);
              XUCL>                           CFG_XUCL(7);
              LSL>                            CFG_XLSL(7);
              XLCL>                           CFG_XLCL(7);
              CUSUM_LIMIT>                    CFG_CUSUM_H(7);
              CUSUM_SLACK>                    CFG_CUSUM_K(7);
              CUSUM_TARGET>                   CFG_CUSUM_TGT(7);

              CALC_CTRL_LIMITS_IF>            CALC_LIMITS_TRIG(7);
              HISTOGRAM_RESET_IF>             HISTO_RESET_TRIG(7);
              HISTOGRAM_RESET_TIME_AT>        HISTO_RESET_TIME(7);
              PARETO_RESET_IF>                PARETO_RESET_TRIG(7);
              PARETO_RESET_TIME_AT> PARETO_RESET_TIME(7);
              CLEAR_CHARACTERISTIC_IF>        CLEAR_CHAR_TRIG(7);   ! Start over

              X_BAR_CTRL_ALARM_AT>            X_BAR_CTRL_ALRM_07(0);! 30 control pts.
              X_BAR_7_ABOVE_ALARM_AT>         X_BAR_7ABV_ALRM_07(0);
              X_BAR_7_BELOW_ALARM_AT>         X_BAR_7BLW_ALRM_07(0);
              X_BAR_7_DECR_ALARM_AT>          X_BAR_7DEC_ALRM_07(0);
              X_BAR_7_INCR_ALARM_AT>          X_BAR_7INC_ALRM_07(0);
              X_BAR_CTRL_WARN_AT>             X_BAR_WARN_ALRM_07(0);
              CUSUM_HIGH_ALARM_AT>            CUSUM_HI_ALRM_07(0);
              CUSUM_LOW_ALARM_AT>             CUSUM_LO_ALRM_07(0);
```

```
!
!       ------------------------------------------------
!       Attribute Chacteristic for Manual-Based sampling
!       ------------------------------------------------
!
        SPC_CHARACTERISTIC>                     ;"CHAR_08"

                CHARACTERISTIC_TYPE>            ;ATTRIBUTE
                CTRL_CHART_TYPE>                ;C                  ! C chart
                LOT_SIZE>                       ;100                ! Constant samples per lot

                SAMPLE_TYPE>                    ;MANUAL
                MANUAL_SUBTYPE>                 ;SAMPLE             ! Read single sample

                ADB_LOCATION>                   ;"DATA_VALUE_08"    ! Number of defects
                COMMENT_CODE>                   COMMENT_08;         ! 1 comment per sample
                ALARM_CODE>                     ALRM_CODE_08;       ! 1 comment per sample
                SAMPLE_IF>                      READ_DATA_TRIG(8);

                UVL>                            CFG_XUVL(8);
                USL>                            CFG_XUSL(8);
                XUCL>                           CFG_XUCL(8);
                LVL>                            CFG_XLVL(8);
                LSL>                            CFG_XLSL(8);
                XLCL>                           CFG_XLCL(8);
                CUSUM_LIMIT>                    CFG_CUSUM_H(8);
                CUSUM_SLACK>                    CFG_CUSUM_K(8);
                CUSUM_TARGET>                   CFG_CUSUM_TGT(8);

                CALC_CTRL_LIMITS_IF>            CALC_LIMITS_TRIG(8);
                HISTOGRAM_RESET_IF>             HISTO_RESET_TRIG(8);
                HISTOGRAM_RESET_TIME_AT>        HISTO_RESET_TIME(8);
                PARETO_RESET_IF>                PARETO_RESET_TRIG(8);
                PARETO_RESET_TIME_AT> PARETO_RESET_TIME(8);
                CLEAR_CHARACTERISTIC_IF>        CLEAR_CHAR_TRIG(8);   ! Start over

                X_BAR_CTRL_ALARM_AT>            X_BAR_CTRL_ALRM_08(0);! 30 control pts.
                X_BAR_7_ABOVE_ALARM_AT>         X_BAR_7ABV_ALRM_08(0);
                X_BAR_7_BELOW_ALARM_AT>         X_BAR_7BLW_ALRM_08(0);
                X_BAR_7_DECR_ALARM_AT>          X_BAR_7DEC_ALRM_08(0);
                X_BAR_7_INCR_ALARM_AT>          X_BAR_7INC_ALRM_08(0);
                X_BAR_CTRL_WARN_AT>             X_BAR_WARN_ALRM_08(0);
                CUSUM_HIGH_ALARM_AT>            CUSUM_HI_ALRM_08(0);
                CUSUM_LOW_ALARM_AT>             CUSUM_LO_ALRM_08(0);
```

```
!
!      ------------------------------------------------
!      Attribute Chacteristic for Manual-Based sampling
!      ------------------------------------------------
!
       SPC_CHARACTERISTIC>                    ;"CHAR_09"

              CHARACTERISTIC_TYPE>            ;ATTRIBUTE
              CTRL_CHART_TYPE>                ;U                    ! U chart
              LOT_SIZE>                       ;LOT_SIZE_09          ! Varying samples per lot

              SAMPLE_TYPE>                    ;EVENT

              ADB_LOCATION>                   ;"DATA_VALUE_09"      ! Number of defects per unit
              SAMPLE_IF>                      READ_DATA_TRIG(9);

              UVL>                            CFG_XUVL(9);
              USL>                            CFG_XUSL(9);
              XUCL>                           CFG_XUCL(9);
              LVL>                            CFG_XLVL(9);
              LSL>                            CFG_XLSL(9);
              XLCL>                           CFG_XLCL(9);
              CUSUM_LIMIT>                    CFG_CUSUM_H(9);
              CUSUM_SLACK>                    CFG_CUSUM_K(9);
              CUSUM_TARGET>                   CFG_CUSUM_TGT(9);

              CALC_CTRL_LIMITS_IF>            CALC_LIMITS_TRIG(9);
              HISTOGRAM_RESET_IF>             HISTO_RESET_TRIG(9);
              HISTOGRAM_RESET_TIME_AT>        HISTO_RESET_TIME(9);
              PARETO_RESET_IF>                PARETO_RESET_TRIG(9);
              PARETO_RESET_TIME_AT> PARETO_RESET_TIME(9);
              CLEAR_CHARACTERISTIC_IF>        CLEAR_CHAR_TRIG(9);   ! Start over

              X_BAR_CTRL_ALARM_AT>            X_BAR_CTRL_ALRM_09(0);! 30 control pts.
              X_BAR_7_ABOVE_ALARM_AT>         X_BAR_7ABV_ALRM_09(0);
              X_BAR_7_BELOW_ALARM_AT>         X_BAR_7BLW_ALRM_09(0);
              X_BAR_7_DECR_ALARM_AT>          X_BAR_7DEC_ALRM_09(0);
              X_BAR_7_INCR_ALARM_AT>          X_BAR_7INC_ALRM_09(0);
              X_BAR_CTRL_WARN_AT>             X_BAR_WARN_ALRM_09(0);
              CUSUM_HIGH_ALARM_AT>            CUSUM_HI_ALRM_09(0);
              CUSUM_LOW_ALARM_AT>             CUSUM_LO_ALRM_09(0);
```

```
!
!       ====================================================================
!       Display station definitions
!       ====================================================================
!

!
!       ------------------------------
!       Variable Chart Display Station
!       ------------------------------
!

       SPC_DISPLAY_STATION>

              DISPLAY_CHARACTERISTIC>          DS01_CHAR_NAME;
              DISPLAY_CTRL_CHART_TYPE>         DS01_CHART_TYPE;
              DISPLAY_GRAND_MEAN_AT>           DS01_X_BARBAR;
              DISPLAY_RS_MEAN_AT>              DS01_RS_MEAN;
              DISPLAY_MEAN_AT>                 DS01_X_BAR(0);
              DISPLAY_RS_AT>         DS01_RANGE(0);
              DISPLAY_UVL_AT>                  DS01_XUVL;
              DISPLAY_LVL_AT>                  DS01_XLVL;
              DISPLAY_USL_AT>                  DS01_XUSL;
              DISPLAY_LSL_AT>                  DS01_XLSL;
              DISPLAY_XUCL_AT>                 DS01_XUCL;
              DISPLAY_XLCL_AT>                 DS01_XLCL;
              DISPLAY_RSUCL_AT>                DS01_RSUCL;
              DISPLAY_RSLCL_AT>                DS01_RSLCL;
              DISPLAY_CPK_AT>                  DS01_CPK;
              DISPLAY_CP_AT>         DS01_CP;
              DISPLAY_X_TARGET_AT>             DS01_X_TARGET;
              DISPLAY_RS_TARGET_AT> DS01_RS_TARGET;
              DISPLAY_SAMPLES_AT>              DS01_SAMPLE(0);
              DISPLAY_SPEC_ALARMS_AT>          DS01_SAMPLE_ALMS(0);
              DISPLAY_SAMPLE_TIME_AT>          DS01_SAMPLE_TIME(0);
              DISPLAY_SUBGROUP_SIZE_AT>        DS01_SUBGRP_SIZE(0);
              DISPLAY_SUBGROUP_TIME_AT>        DS01_SUBGRP_TIME(0);
              DISPLAY_HISTOGRAM_AT> DS01_HISTO_AVG;
              DISPLAY_HISTO_RESET_TIME_AT>     DS01_HISTO_RESET_TIME;
              DISPLAY_CUSUM_HIGH_AT>           DS01_CUSUM_HIGH(0);
              DISPLAY_CUSUM_LOW_AT> DS01_CUSUM_LOW(0);
              DISPLAY_CUSUM_LIMIT_AT>          DS01_CUSUM_H;
              DISPLAY_CUSUM_SLACK_AT>          DS01_CUSUM_K;
              DISPLAY_CUSUM_TARGET_AT>         DS01_CUSUM_TGT;
              DISPLAY_PARETO_AT>               DS01_PARETO_CNT(0);
              DISPLAY_PARETO_RESET_TIME_AT>    DS01_PARETO_RESET_TIME;
              DISPLAY_PARETO_BUCKETS>          DS01_PARETO_BUCKETS;
              DISPLAY_PARETO_TYPE>             DS01_PARETO_TYPE;
              DISPLAY_SHIFT_NO_AT>             DS01_SHIFT_NO(0);
              DISPLAY_ALARM_CODE_AT>           DS01_ALARM_CODE(0);
              DISPLAY_COMMENT_CODE_AT>         DS01_COMMENT_CODE(0);

       DISPLAY_RETRIEVAL_TIME>                 DS01_RETRIEVAL_TIME;
       DISPLAY_END_RETRIEVAL_TIME>             DS01_END_RETRIEVAL_TIME;

       DISPLAY_PARETO_IF>                      DS01_PARETO_DSP_TRIG;
       DISPLAY_HISTOGRAM_IF>         DS01_HISTO_DSP_TRIG;
       DISPLAY_IF>                             DS01_DSP_TRIG;
```

```
!
!      ------------------------------
!      Attribute Chart Display Station
!      ------------------------------
!

       SPC_DISPLAY_STATION>

              DISPLAY_CHARACTERISTIC>          DS02_CHAR_NAME;
              DISPLAY_CTRL_CHART_TYPE>         DS02_CHART_TYPE;
              DISPLAY_GRAND_MEAN_AT>           DS02_X_BARBAR;
              DISPLAY_PCT_DEFECTS_AT>          DS02_PCT_DEFECTS(0);
              DISPLAY_DEFECTS_AT>              DS02_DEFECTS(0);
              DISPLAY_LOT_SIZE_AT>             DS02_LOT_SIZE(0);
              DISPLAY_UVL_AT>                  DS02_XUVL;
              DISPLAY_LVL_AT>                  DS02_XLVL;
              DISPLAY_USL_AT>                  DS02_XUSL;
              DISPLAY_LSL_AT>                  DS02_XLSL;
              DISPLAY_XUCL_AT>                 DS02_XUCL;
              DISPLAY_XLCL_AT>                 DS02_XLCL;
              DISPLAY_X_TARGET_AT>             DS02_X_TARGET;
              DISPLAY_SAMPLES_AT>              DS02_SAMPLE(0);
              DISPLAY_SPEC_ALARMS_AT>          DS02_SAMPLE_ALMS(0);
              DISPLAY_SAMPLE_TIME_AT>          DS02_SAMPLE_TIME(0);
              DISPLAY_HISTOGRAM_AT> DS02_HISTO_AVG;
              DISPLAY_HISTO_RESET_TIME_AT>     DS02_HISTO_RESET_TIME;
              DISPLAY_CUSUM_HIGH_AT>           DS02_CUSUM_HIGH(0);
              DISPLAY_CUSUM_LOW_AT> DS02_CUSUM_LOW(0);
              DISPLAY_CUSUM_LIMIT_AT>          DS02_CUSUM_H;
              DISPLAY_CUSUM_SLACK_AT>          DS02_CUSUM_K;
              DISPLAY_CUSUM_TARGET_AT>         DS02_CUSUM_TGT;
              DISPLAY_PARETO_AT>               DS02_PARETO_CNT(0);
              DISPLAY_PARETO_RESET_TIME_AT>    DS02_PARETO_RESET_TIME;
              DISPLAY_PARETO_BUCKETS>          DS02_PARETO_BUCKETS;
              DISPLAY_PARETO_TYPE>             DS02_PARETO_TYPE;
              DISPLAY_SHIFT_NO_AT>             DS02_SHIFT_NO(0);
              DISPLAY_ALARM_CODE_AT>           DS02_ALARM_CODE(0);
              DISPLAY_COMMENT_CODE_AT>         DS02_COMMENT_CODE(0);

       DISPLAY_RETRIEVAL_TIME>                 DS02_RETRIEVAL_TIME;
       DISPLAY_END_RETRIEVAL_TIME>             DS02_END_RETRIEVAL_TIME;

       DISPLAY_PARETO_IF>                      DS02_PARETO_DSP_TRIG;
       DISPLAY_HISTOGRAM_IF>        DS02_HISTO_DSP_TRIG;
       DISPLAY_IF>                             DS02_DSP_TRIG;

END>
```