OVERVIEW OF "THE SCANS" IN THE CENTRAL CONTROL SYSTEM OF TRIUMF'S 500 MeV CYCLOTRON

J. J. Pon, K. S. Lee, M. M. Mouat, P. J. Yogendran, TRIUMF, Vancouver, Canada
B. Davison, Simon Fraser University, Burnaby, Canada

Abstract

The Controls Group for TRIUMF's 500 MeV cyclotron developed, runs and maintains a software application known as The Scans whose purpose is to: a) log events, b) enunciate alarms and warnings, c) perform simple actions on the hardware, and d) provide software interlocks for machine protection. Since its inception more than 35 years ago, The Scans has increasingly become an essential part for the proper operation of the Cyclotron. This paper gives an overview of The Scans, its advantages and limitations, and desired improvements.

INTRODUCTION

The Scans is a primary software utility [1] in the Central Control System (CCS) of TRIUMF's 500 MeV cyclotron. More than 35 years ago the first version of The Scans was created. Since then this utility has undergone, and continues to undergo, significant evolution. In the present configuration of the CCS, The Scans is an essential component.

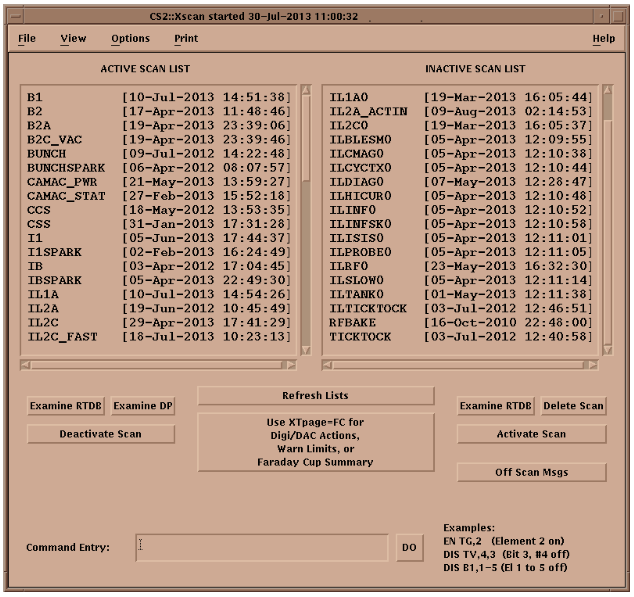
This utility has a master process, which on startup spawns one subprocess for each scan that is in a list of active scans. Each subprocess (a child scan) then reads its corresponding scan definition file (a script file) and runs autonomously for the most part. Each definition file has a similar structure, which will be described later but in brief has a user defined number of elements, where each element has three parts: a) a header, b) a test condition, c) actions.

From one perspective The Scans is structured like a programmable logic controller (PLC). Each scan follows a PLC-like sequence of: a) acquire data, b) test the data, c) take actions, and d) repeat the sequence. One of the prime differences is that The Scans executes as a regular software process in a common server while a real PLC needs dedicated, specialized hardware.

Operators interact with The Scans via an X Window graphical user interface known as Xscan (see Fig. 1) and multiple instances of Xscan can be running.

The scan definition files can be categorized into two basic types, regular and interlock. Both types have the same structure but by convention only interlock scans take interlock actions (e.g., trip the beam). Detailed information on the state of interlock scans is displayed in a different X Window application called XTpage (see Fig. 2).

Individual scan elements can perform a wide variety of actions such as provide machine protection (beam trips and "soft" trips [2][3]) and warnings, diagnostics, initializations, watchdog monitoring, event notification (messages to activity logs), automatic resetting of equipment, and dynamic adjustment of gain settings to avoid saturation.



CS2::Xscan started 30-Jul-2013 11:00:32

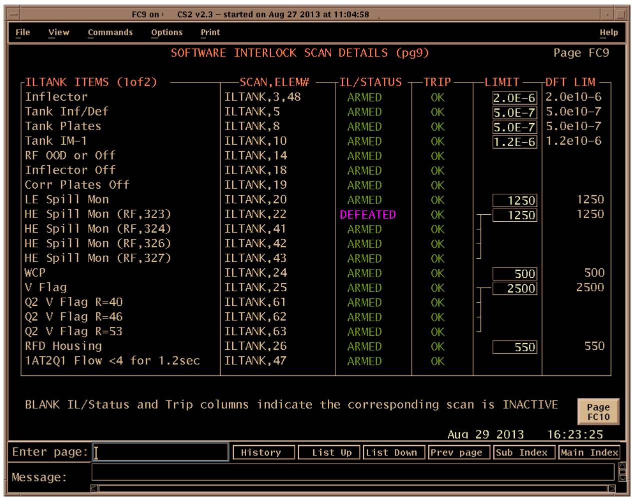
ACTIVE SCAN LIST		INACTIVE SCAN LIST	
B1	[10-Jul-2013 14:51:38]	IL1A0	[19-Mar-2013 16:05:44]
B2	[17-Apr-2013 11:48:46]	IL2A_ACTIN	[09-Aug-2013 02:14:53]
B2A	[19-Apr-2013 23:39:06]	IL2C0	[19-Mar-2013 16:05:37]
B2C_VAC	[19-Apr-2013 23:39:46]	ILBLESM0	[05-Apr-2013 12:09:55]
BUNCH	[09-Jul-2012 14:22:48]	ILCMAG0	[05-Apr-2013 12:10:38]
BUNCHSPARK	[06-Apr-2012 08:07:57]	ILCYCTX0	[05-Apr-2013 12:10:44]
CAMAC_PWR	[21-May-2013 13:59:27]	ILDIAG0	[07-May-2013 12:28:47]
CAMAC_STAT	[27-Feb-2013 15:52:18]	ILHICUR0	[05-Apr-2013 12:10:48]
CCS	[18-May-2012 13:53:35]	ILINF0	[05-Apr-2013 12:10:52]
CSS	[31-Jan-2013 17:31:28]	ILINFSK0	[05-Apr-2013 12:10:58]
I1	[05-Jun-2013 17:44:37]	ILISIS0	[05-Apr-2013 12:11:01]
I1SPARK	[02-Feb-2013 16:24:49]	ILPROBE0	[05-Apr-2013 12:11:05]
IB	[03-Apr-2012 17:04:45]	ILRF0	[23-May-2013 16:32:30]
IBSPARK	[05-Apr-2013 22:49:30]	ILSLOW0	[05-Apr-2013 12:11:14]
IL1A	[10-Jul-2013 14:54:26]	ILTANK0	[01-May-2013 12:11:38]
IL2A	[19-Jun-2012 10:45:49]	ILTICKTOCK	[03-Jul-2012 12:46:51]
IL2C	[29-Apr-2013 17:41:29]	RFBAKE	[16-Oct-2010 22:48:00]
IL2C_FAST	[18-Jul-2013 10:23:13]	TICKTOCK	[03-Jul-2012 12:40:58]

Use XTpage=FC for Digi/DAC Actions, Warn Limits, or Faraday Cup Summary

Command Entry:

Examples:
EN TG,2 (Element 2 on)
DIS TV,4,3 (Bit 3, #4 off)
DIS B1,1–5 (El 1 to 5 off)

Figure 1: Xscan – User interface to The Scans.



FC9 on CS2 v2.3 – started on Aug 27 2013 at 11:04:58

SOFTWARE INTERLOCK SCAN DETAILS (pg9) Page FC9

ILTANK ITEMS (1of2)	SCAN,ELEM#	IL/STATUS	TRIP	LIMIT	DFT LIM
Inflector	ILTANK,3,48	ARMED	OK	2.0E-6	2.0e10-6
Tank Inf/Def	ILTANK,5	ARMED	OK	5.0E-7	5.0e10-7
Tank Plates	ILTANK,8	ARMED	OK	5.0E-7	5.0e10-7
Tank IM-1	ILTANK,10	ARMED	OK	1.2E-6	1.2e10-6
RF OOD or Off	ILTANK,14	ARMED	OK		
Inflector Off	ILTANK,18	ARMED	OK		
Corr Plates Off	ILTANK,19	ARMED	OK		
LE Spill Mon	ILTANK,20	ARMED	OK	1250	1250
HE Spill Mon (RF,323)	ILTANK,22	DEFEATED	OK	1250	1250
HE Spill Mon (RF,324)	ILTANK,41	ARMED	OK		
HE Spill Mon (RF,326)	ILTANK,42	ARMED	OK		
HE Spill Mon (RF,327)	ILTANK,43	ARMED	OK		
WCP	ILTANK,24	ARMED	OK	500	500
V Flag	ILTANK,25	ARMED	OK	2500	2500
Q2 V Flag R=40	ILTANK,61	ARMED	OK		
Q2 V Flag R=46	ILTANK,62	ARMED	OK		
Q2 V Flag R=53	ILTANK,63	ARMED	OK		
RFD Housing	ILTANK,26	ARMED	OK	550	550
1AT2Q1 Flow <4 for 1.2sec	ILTANK,47	ARMED	OK		

BLANK IL/Status and Trip columns indicate the corresponding scan is INACTIVE

Aug 29 2013 16:23:25

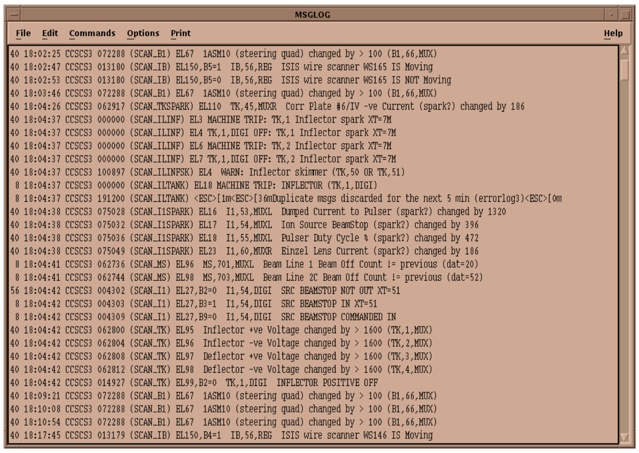
Enter page:

Message:

Figure 2: XTpage display of an interlock scan.

Typically there are more than 50 active scans and more than 4500 enabled (active) scan elements. The message logs regularly receive several thousand event messages a day from scan elements (see Fig. 3). The Scans runs 24 hours a day, 7 days a week, even during maintenance and shutdown periods when possible.

ISBN 978-3-95450-139-7



```
40 18:02:25 CCSCS3 072288 (SCAN_B1) EL67  1ASM10 (steering quad) changed by > 100 (B1,66,MUX)
40 18:02:47 CCSCS3 013180 (SCAN_IB) EL150,B5=1  IB,56,REG  ISIS wire scanner WS165 IS Moving
40 18:02:53 CCSCS3 013180 (SCAN_IB) EL150,B5=0  IB,56,REG  ISIS wire scanner WS165 IS NOT Moving
40 18:03:46 CCSCS3 072288 (SCAN_B1) EL67  1ASM10 (steering quad) changed by > 100 (B1,66,MUX)
40 18:04:26 CCSCS3 062917 (SCAN_TKSPARK) EL110  TK,45,MUXR  Corr Plate #6/IV -ve Current (spark?) changed by 186
40 18:04:37 CCSCS3 000000 (SCAN_ILINF) EL3 MACHINE TRIP: TK,1 Inflector spark XT=7M
40 18:04:37 CCSCS3 000000 (SCAN_ILINF) EL4 TK,1,DIGI OFF: TK,1 Inflector spark XT=7M
40 18:04:37 CCSCS3 000000 (SCAN_ILINF) EL6 MACHINE TRIP: TK,2 Inflector spark XT=7M
40 18:04:37 CCSCS3 000000 (SCAN_ILINF) EL7 TK,1,DIGI OFF: TK,2 Inflector spark XT=7M
40 18:04:37 CCSCS3 100897 (SCAN_ILINFSK) EL4  WARN: Inflector skimmer (TK,50 OR TK,51)
 8 18:04:37 CCSCS3 000000 (SCAN_ILTANK) EL18 MACHINE TRIP: INFLECTOR (TK,1,DIGI)
 8 18:04:37 CCSCS3 191200 (SCAN_ILTANK) <ESC>[1m<ESC>[36mDuplicate msgs discarded for the next 5 min (errorlog3)<ESC>[0m
40 18:04:38 CCSCS3 075028 (SCAN_I1SPARK) EL16  I1,53,MUXL  Dumped Current to Pulser (spark?) changed by 1320
40 18:04:38 CCSCS3 075032 (SCAN_I1SPARK) EL17  I1,54,MUXL  Ion Source BeamStop (spark?) changed by 396
40 18:04:38 CCSCS3 075036 (SCAN_I1SPARK) EL18  I1,55,MUXL  Pulser Duty Cycle % (spark?) changed by 472
40 18:04:38 CCSCS3 075049 (SCAN_I1SPARK) EL23  I1,60,MUXR  Einzel Lens Current (spark?) changed by 186
 8 18:04:41 CCSCS3 062736 (SCAN_MS) EL96  MS,701,MUXL  Beam Line 1 Beam Off Count != previous (dat=20)
 8 18:04:41 CCSCS3 062744 (SCAN_MS) EL98  MS,703,MUXL  Beam Line 2C Beam Off Count != previous (dat=52)
56 18:04:42 CCSCS3 004302 (SCAN_I1) EL27,B2=0  I1,54,DIGI  SRC BEAMSTOP NOT OUT XT=51
 8 18:04:42 CCSCS3 004303 (SCAN_I1) EL27,B3=1  I1,54,DIGI  SRC BEAMSTOP IN XT=51
 8 18:04:42 CCSCS3 004309 (SCAN_I1) EL27,B9=0  I1,54,DIGI  SRC BEAMSTOP COMMANDED IN
40 18:04:42 CCSCS3 062800 (SCAN_TK) EL95  Inflector +ve Voltage changed by > 1600 (TK,1,MUX)
40 18:04:42 CCSCS3 062804 (SCAN_TK) EL96  Inflector -ve Voltage changed by > 1600 (TK,2,MUX)
40 18:04:42 CCSCS3 062808 (SCAN_TK) EL97  Deflector +ve Voltage changed by > 1600 (TK,3,MUX)
40 18:04:42 CCSCS3 062812 (SCAN_TK) EL98  Deflector -ve Voltage changed by > 1600 (TK,4,MUX)
40 18:04:42 CCSCS3 014927 (SCAN_TK) EL99,B2=0  TK,1,DIGI  INFLECTOR POSITIVE OFF
40 18:09:21 CCSCS3 072288 (SCAN_B1) EL67  1ASM10 (steering quad) changed by > 100 (B1,66,MUX)
40 18:10:08 CCSCS3 072288 (SCAN_B1) EL67  1ASM10 (steering quad) changed by > 100 (B1,66,MUX)
40 18:10:54 CCSCS3 072288 (SCAN_B1) EL67  1ASM10 (steering quad) changed by > 100 (B1,66,MUX)
40 18:17:45 CCSCS3 013179 (SCAN_IB) EL150,B4=1  IB,56,REG  ISIS wire scanner WS146 IS Moving
```

Figure 3: Sample messages logged by The Scans.

FRAMEWORK AND SOFTWARE ENVIRONMENT (DEVELOPMENT & PRODUCTION)

The Scans is written in C/C++ and runs on an OpenVMS machine which has access to all of the CCS devices that need to be monitored or controlled.

A third-party messaging application called Oracle MessageQ is used for communication between the master process and its active child scans for requests such as enabling/disabling scan elements. A typical operator request follows this sequence: a) the operator uses Xscan to make a request, b) the request goes via the MessageQ to the master process, c) the master process identifies the relevant child scan and relays the message to it using the MessageQ, and d) the child scan receives the message and acts accordingly.

With Xscan, operators can request actions such as a) activate/deactivate entire scans, b) enable/disable individual scan elements, c) display a list of active and inactive scans, and d) display currently disabled scan elements. If for some reason The Scans is restarted, the state of active and inactive scans is preserved but the list of disabled scan elements is cleared.

The MessageQ is also used extensively by individual child scans to send warning and event messages to activity logs. This messaging system is flexible enough to allow passing messages not only between processes in the same computer but also across different machines and even clusters. Every message is logged in two separate and independent computer clusters for redundancy.

The functional part of Xscan (the X Window application) is written in C/C++ and the graphical part is written in UIL. Xscan can be displayed on any platform that has an X Window server application. On a Linux machine the X server is native and Xscan can be readily displayed. On a Windows machine you would need X Server software like X-Win32 or Xming. On the Apple iPad/iPod touch/iPhone there is an X server app called iSSH available.

Fig. 4 shows a diagram of how The Scans is functionally organized.

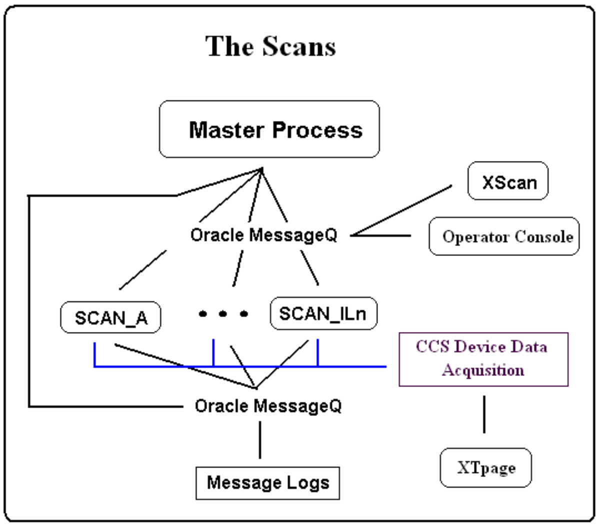


Figure 4: Primary application components of The Scans.

SCAN DEFINITION FILES

For each child scan there is a corresponding scan definition file. By convention each definition file contains scan elements that are primarily related by function. Thus, there are scan files for beamlines, safety, magnets, over current interlocks, RF devices, and watchdogs to name a few.

The scan definition files are written in ASCII text. This was a conscious decision to allow operators to quickly view, understand, and even make changes in urgent cases. The definition files are initially read and parsed by the scan software and subsequently run as executable code, not interpreted. If the scripts are modified, the corresponding scan needs to be reactivated to pick up the changes. Reactivation of individual scans is fast with minimal or no beam disruption.

Each scan file has a similar format and starts with a scan header. In the scan header we define the scan frequency. Other information in the scan header relates to the number of scan elements, a history of revisions, and helpful comments about the particular scan.

Next follow the scan elements. Each element adheres to the following format: a) an element header, b) a test condition, and c) actions to perform. This is the syntax for a single element:

```
ELEMENT i:<DIS|EN>ABLED,ACTIONS: n[,args,...]
<test condition>
<action 1>
   :
<action n>
```

ELEMENT i defines the element number.

<DIS|EN>ABLED specifies whether the element is enabled or disabled on startup (i.e., scan activation).

ACTIONS: n is the number of actions to perform when the test condition is true.

ISBN 978-3-95450-139-7

[,args,...] are optional arguments. Common options are:

TRUE_COUNT specifies how many consecutive scan cycles the test condition must be true before taking action (used for timing and de-glitching). If the keyword is omitted the element infers a TRUE_COUNT of 1.

NO_AUTODISABLE prevents the element from disabling itself when there is a device error. By default, if an element tests a device that returns a data acquisition error (i.e., hardware error) the element performs its actions and then automatically disables itself to prevent it from continuously enunciating it is in an error state. NO_AUTODISABLE is useful in cases where interlock elements need to stay enabled and keep providing protection even during hardware errors. The element can still be manually disabled by operators if they decide it is safe to run without a particular interlock protection.

<test condition> must return true for the element to perform actions. The syntax for the test conditions can include the relational operators =, !=, >, >=, <, <=; and conditional operators like AND, OR. There is also a special operator != PREVIOUS which compares the current value to the one acquired in the previous scan cycle. In cases where an element is needed to perform its actions on every scan cycle we have the keyword ALWAYS. Alternatively, the keyword FIRST TIME can be used in instances where we need the element to perform only on the first scan cycle such as when initializing temporary variables.

If <test condition> evaluates to true (and TRUE_COUNT is satisfied) then the element performs its assigned actions sequentially. Each action starts with the keyword DO followed by a command. The most frequently used commands are:

- INSERT FARADAY CUP (to trip the beam).
- LOG (to enunciate a message to the logs).
- CALC (to perform a mathematical calculation).
- WRITE (to write a value to a device).

Here is a typical example of a scan element:

```
ELEMENT 8: ENABLED,ACTIONS: 4, NO_AUTODISABLE
TV,56,MUX < 1.1E-9 OR TV,56,MUX > TK,531,MUX
DO INSERT FARADAY CUP
DO LOG opslog “CORRECTION PLATE TRIP: VACUUM”
DO CALC TEMP(90) = 15 * 65535 + 101
DO WRITE TEMP(90) TO ET,100,DAC
```

FEATURES

- Frequency: Each child scan runs at its own predefined time period to ensure the conditions are tested at the required rate, but not so often that CPU cycles are wasted. At present we have some scans running at 10 Hz while others run as slow as once per minute. Regular scans typically run every 6 seconds while interlock scans most commonly run at 5 Hz. In principle, scans could run at up to 100 Hz.
- Defeats: Elements can be defined as enabled or disabled on scan activation. Operators can manually activate or deactivate an entire scan. Also, they can enable or disable an element or a bit within an element without needing to reactivate the whole scan.
- Inter-process communications: The Scans conveys useful information like: a) a list of elements currently disabled, b) a list of active and inactive scans, c) trip status, d) last trip, e) interlock defeats. This feature allows other applications to display the data. The Scans also listens for external commands and takes actions.
- Multiple actions: A scan element can perform multiple actions. For example, an interlock element a) inserts the faraday cup, b) logs the event, c) performs some mathematical calculation, d) exports its trip status, and e) exports the last trip information.
- Efficient: Each scan behaves similar to a PLC. On each scan cycle the data needed by all its elements is acquired, each element tests the data (no data re-acquisition), and takes action(s).
- Scripts are easy to read: The scripts are in ASCII. Thus they are relatively easy to understand and modify.
- Scalable: Elements can be added to any scan and new scans can be introduced with minimal disruption. The Xscan can be displayed in any platform capable of displaying X Window applications.
- Temporary variables: Variables can be used in The Scans to give it more flexibility.

PERFORMANCE

The master process and child scans run in one server with access to all CCS devices. In a typical operational day, over 50 scans are active which translates into more than 4500 scan elements altogether. About 15 scans run at 5 Hz. Typical CPU load on the server is about 2 percent.

The only noticeable load on the system occurs when the child scans go into a state referred to as a “message storm”. This occurs when, for example, there is a power bump that knocks out a significant number of hardware devices, which in turn forces the child scans to send a large volume of error messages to the logs. Smart message filters have been added to the message handling to minimize the impact to the system load.

OPERATIONAL EXPERIENCE

The Scans has become essential for running beam and it is one of the primary mechanisms providing machine protection through software. Changes to scan elements occur almost every week which is indicative to how useful and reliable it is to Operations.

The messages generated by The Scans are useful in forensic analysis. These messages often give a good view of the activities and timing of events that occurred and are used daily in this fashion. The message log can be scanned back more than 10 years, which provides long term information about event frequency. A special graphical user application called the X Window Message Browser (XMB) has been developed to aid in searching the message log.

ISBN 978-3-95450-139-7

PLANS FOR ENHANCEMENTS

One outstanding request is to allow users (primarily Operations) to define their own scans from the XTpage application. This request would provide a quick, simple, short term, functionality. The need is to have the ability to put an alarm/watchdog on a signal (or set of conditions) and to have an audible alarm and an entry in the message log.

There has also been a request for additional Xscan display features like a sortable, tree-like display of individual scans and scan elements.

Another notable request is to have the ability to save and restore the enable/disable state of individual scan elements. This feature would be useful in cases when the disable state of all scan elements need to be restored after a server reboot or to disable the elements in a specific scan when the scan is reactivated.

SUMMARY

In summary, The Scans has proven to be a useful, easy-to-modify, scalable, robust application. This software has evolved over more than three decades to be an essential part of the Cyclotron's Central Control System. The Scans runs reliably and efficiently with very minimal load on the system. There is sufficient functionality to be useful in a variety of areas such as machine protection (warnings, hard trips, and soft trips), event logging, diagnostics, and device initialization.

These functions have gradually been expanded from providing basic machine interlock protection and message logging to finer actions like setting the proper gain on devices and iteratively/adaptively reducing beam intensity (soft trips).

REFERENCES

[1] M.M. Mouat et al., "Update on the Central Control System of TRIUMF's 500 MeV Cyclotron", ICALEPCS2011, Grenoble, France, Oct 2011, p.469 (2011).

[2] J.J. Pon et al., "Recent Changes in the 500 MeV Cyclotron's Central Controls System to Reduce Beam Downtime and Beam On/Off Transitions", ICALEPCS2009, Kobe, Japan, Oct 2009, TUP043, p. 179 (2009); http://www.JACoW.org/

[3] M. Trinczek, M. M. Mouat, J. J. Pon, "Increasing Beam Delivery through Soft Trips", Accelerator Reliability Workshop (ARW2009), Vancouver, Canada, 25-30 Jan 2009.