Unsolicited Status Protocol (USP)

PESA Switching Systems, Inc.
330-A Wynn Drive Northwest
Huntsville, AL 35805-1961
http://www.pesa.com
(256) 726-9200

Document No. 81-9059-0409-0 Revision C

Service and Ordering Assistance

PESA Switching Systems, Inc.
330-A Wynn Drive Northwest
Huntsville, AL 35805-1961 USA
www.pesa.com

Main Office
(256) 726-9200 (Voice)
(256) 726-9271 (Fax)

Service Department
(256) 726-9222 (Voice) **(24 hours/day, 7 days/week)**
(256) 726-9268 (Fax)
service@pesa.com

National Sales Office

PESA Switching Systems, Inc.
35 Pinelawn Rd., Suite 99-E
Melville, NY 11747 USA
(800) 328-1008 (Voice)
(516) 845-5020 (Voice)
(516) 845-5023 (Fax)

© 1999 PESA Switching Systems, Inc. All Rights Reserved.

No part of this publication may be reproduced, stored in any retrieval system, or transmitted in any form or by any means, including but not limited to electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of PESA Switching Systems, Inc. This includes text, illustrations, tables and charts.

All information, illustrations and specifications contained in this publication are based on the latest product information available at the time of publication approval. The right is reserved to make changes at any time without notice.

Comments and/or suggestions concerning this document are welcome and should be directed to:
techpubs@pesa.com

Printed in the United States of America.

Table of Contents

Figures

Tables

Chapter 1 – Introduction

The Unsolicited Status Protocol (USP) is a PESA CPU link protocol that provides information to the device interfacing to the PESA controller in an unsolicited fashion. The interfacing device is not required to provide stimulus in order to get information back. The simple occurrence of an event in the control system causes information to be sent out the USP.

This document describes the structure of the protocol. The structure of the unsolicited response messages are detailed. The command structure of the USP is described as well.

This document does not describe the particular information that is controller dependent. Issues such as controller configuration are beyond the scope of this document.

Chapter 2 – Acronyms and Definitions

Interfacing Device

An external device that communicates over the USP to the PESA controller. The interfacing device issues commands over the USP and receives both unsolicited and solicited messages from the PESA controller.

Message

A single information packet that is sent either to the PESA controller from the interfacing device or vice-versa.

PESA Controller

A device that controls PESA routing matrices. The PESA controller provides one end of the USP protocol interface. It receives commands and issues solicited responses. It generates unsolicited responses in response to occurrence of controller events.

Chapter 3 – Communication Format

The primary medium for the USP is an asynchronous full-duplex connection that is usually run over RS-232 or RS-422 (this is dependent upon the control system). Data is transmitted over the bus asynchronously. The data is sent at user selectable baud rate (default to 38.4 K Baud) with 1 start bit, 8 data bits, no parity, and 1 stop bit. The default method of handshaking is RTS-CTS control though some controllers may allow for methods such as XON-XOFF.

Chapter 4 – Message Structure

The USP protocol is a point to point interface. The PESA controller assumes that it is communicating with only one device and makes no effort to distinguish multiple devices on a link.

The USP is an ASCII based protocol. Messages consist of all ASCII characters except for the terminating character that marks the end of the message. Messages are sent both to and from the PESA controller. The two message flows require no coordination other than conformance to low level serial handshake standards. The choice of serial flow control is dependent upon the PESA controller implementation.

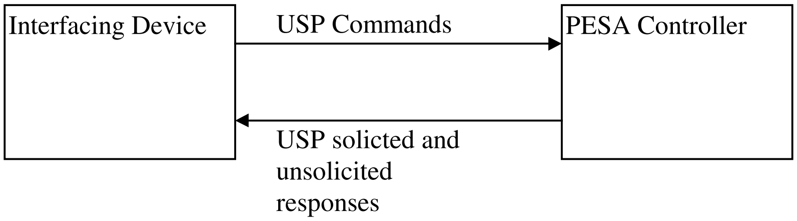

Figure 1. Message Flow

There are two types of messages that are sent from the PESA controller. The first is a solicited response which is response to a command sent to the PESA controller. All commands sent to the PESA controller with proper termination character receive a response from the PESA controller unless the PESA controller runs into an overflow condition (adherence to handshake flow control should prevent overflow).

The second response is an unsolicited response. These are generated in response to a predetermined set of events within the PESA controller. The set of event stimuli are determined within the scope of the controllers configuration.

All solicited responses are guaranteed to be sent in the order to which the commands were received. All unsolicited information is sent out as close to the recognition of the actual event as possible. Unsolicited responses are given priority over solicited responses in transmission. Therefore, time coherence is not guaranteed between solicited and unsolicited responses.

Message Format

Messages sent between the PESA controller and the interfacing device are in the following format:

- <Message Type > [<Message Extension>] <Message Extension Delimiter> <Message Indicator> [<Variable Data Delimiter> [<Variable Data>]] [<Checksum>] <Terminator>

Where:

- <Message Type> is a single character that identifies the type of message that is being sent to and from the PESA controller.
  - Commands are initiated with the pound sign(#) character
  - Solicited response are initiated with the asterisk (*) character.
  - Unsolicited messages are initiated with the tilde (~) character.
- <Message Extension> is an indicator to the message receiver that indicates that the message has special properties such as the addition of more data to be sent in a later packet. The response extension can be a maximum of eight characters in length.
- <Message Extension Delimiter> assists in parsing the response extension from remainder of the message. The delimiter is a single ASCII character “*”.

- <Message Indicator> indicates the contents of the message. It guarantees a uniqueness with other messages of the same type.
- <Variable Data Delimiter> assists in determining the variable data portions of the message. The delimiter is a single ASCII character "*".
- <Variable Data> is message data that may take many forms. It is delimited so that it does not cause response uniqueness conflicts in conjunction with the message indicator.
- <Checksum> is an optional two character field that calculates an 8 bit checksum of the command header and command body. The checksum can be of the standard PESA form (The standard PESA CPU link checksum) or a more common ASCII HEX format.
- <Terminator> is a carriage return, linefeed character or both a carriage return and linefeed character indicating the end of the message transmission. Other types of terminators may be used in the future.

Chapter 5 – Message Descriptions

Unsolicited Responses

The following are responses sent from the PESA controller to the interfacing device in response to an event occurring on the controller. Some controllers may not support all of the following events. Some controllers' configurations allow for the filtering of which events will be sent back.

Many of the following messages contain an address field. This field is used to identify the cause of the event being reported. The address uniquely identifies the device that caused the event. The address may correspond to a user panel address or CPU link port (the controller addresses 1-1023 correspond to RCP panels while addresses 1024 and 1025 correspond to CPU link ports 1 and 2 respectively).

CFG - Configuration Change

Configuration change is an event that indicates a full system configuration change. This event occurs when the system is brought out of a writers lock on the controller configuration.

Table 1. CFG - Configuration Change

Response	~ * C F G * <Address> , <Change Type > , <Cfg Id> [<Checksum>] <Terminator> Where: <Address> is a uniquely identified source of configuration change. <Change Type> indicates whether the change to configuration occurred or was aborted: A- Change was aborted. C - Change was committed to. <Cfg Id> is a text string used to identify the configuration (this may be 0 length).

CNF - Confidence Error

The Confidence Error response indicates that a confidence error was either detected or cleared by the system. The confidence error location is indicated by both the component name and physical address location. Confidence errors are always described in rectangular blocks of crosspoints.

Table 2. CNF - Confidence Error

Response	~ * C N F * <1/0 > , <Component Name> , <Matrix Type> , <Strobe> , <First Output> , <First Input> , <# Outputs> , <# Inputs> , <Error Code> [<Checksum>] <Terminator> Where: <1/0> 1 indicates that the error is active, 0 indicates that the error cleared. <Component Name> is the name of the component in which the error occurred. <Matrix Type> indicates the family of matrix where the error was detected (PRC or RM5). <Strobe> indicates the strobe on which the confidence error was found. <First Output> and <First Input> indicate the physical address origin of the block of crosspoints reporting the confidence error. <# Outputs> and <# Inputs> indicate the size of the confidence error block. <Error Code> is the error code returned by the confidence request response for PRC matrix types. RM5 matrix types will return either 0 (no error) or 0xFF (communications error). Error code is not present in the 3300 USP.

SWX - Switch

The Switch response is an indicator that a switch has been requested. This will indicate both blocked and unblocked switch requests. The Locked and Protected Destination Status is based on the initiator (Devices' Requester Code) of the configuration change.

Table 3. SWX - Switch

Response	~ * S W X * <Address> , <Destination Name> [, <Destination Status Codes>] [, [<Attempted Level 1 Source Name>] [, <Level 1 Status Code>] ...] [<Checksum>] <Terminator> Where: <Address> is a uniquely identified initiator of switch request. <Destination Name> - name of the destination on which the switch was attempted. <Destination Status Codes> - indicates properties of the destination. Any combination of the following: C - The destination is in Chop L - The destination is Locked P - The destination is Protected <Attempted Level x Source Name> - name of the source which was attempted on level x. <Level x Status Code> - indicates the reason for a blocked switch. Lack of a status code means that the switch was valid for the level. B - The source to destination specified is blocked. R - Reentry to Reentry switch was attempted. T - Tie Line unavailable (Tie Line Block). (3500 Only)

LCK - Lock/Protect

The Lock/Protect response is sent when a destination is either locked, protected, or unlocked/unprotected by an external device.

Table 4. LCK - Lock/ProtectSwitch

Response	~ * L C K * <Address> , <Destination> , <Lock Status Code> [<Checksum>] <Terminator> Where: <Address> - Uniquely identified initiator of lock/protect action. <Destination> is the destination on which the switch was attempted <Lock Status Code> indicates the result of the action. L - Lock action was successful. K - Lock action failed due to destination being locked/protected by same or higher priority. P - Protect action was successful. Q - Protect action failed due to destination being locked/protected by same or higher priority. U - Unlock/Unprotect action was successful. V - Unlock/Unprotect action failed due to destination being locked/protected by same or higher priority.

LOG - User Login/Logoff

The User log on and log off event indicates when a device is logged onto or off of by a named user. This is supported currently by the CPU link interface.

Table 5. LOG - User Login/Logoff

Response	~ * L O G * <Address> , <0/1> [, <User>] [<Checksum>] <Terminator> Where: <Address> - Uniquely identified initiator of Log on/off action. <0/1> - 1 indicates a log on, 0 indicates a log off <User> - An alphanumeric string indicating the user who performed the log on or off action. The user name can be a 0 length string.

PHY - Physical Switch

The Physical Switch response indicates when a physical switch is attempted on the controller. These are switches used for diagnostic purposes. Switches taken during normal operation do not generate a physical switch response.

Table 6. PHY - Physical Switch

Response	~ * P H Y * <Address> , <Component Name> , <Matrix Type> , <Strobe> , <Output #> , <Input #> [<Checksum>] <Terminator> Where: <Address> - Uniquely identified initiator of physical switch request. <Component Name> is the name of the component on which the switch is being taken. <Matrix Type> indicates whether the switch is being taken on a PRC or RM5 matrix. <Strobe>, <Output #>, and <Input #> indicate the crosspoint being activated.

RES - System Restart

The System Restart response indicates when the controller has had an event that causes the controller to go through a restart sequence. These events include when a system is rebooted, when a dual transition occurs, or when a new configuration is downloaded.

Table 7. RES - Physical Switch

Response	~ * R E S * <Source of Restart > , <Active Controller> [<Checksum>] <Terminator> Where: <Source of Restart>: DL - Dual Changeover RA - Cold Start (controller reset from power on or front switch). SS - Soft Start (controller reset from program control). Active Controller: A - A CPU is active B - B CPU is Active

SER - Status Error Event

The Status Error Event indicates when any destination change causes a READBACK or CONFIDENCE Error.

Table 8. SER - Status Error Event

Response	~ * S E R * <Destination Name> [, [<Level 1 Source Name>] [, <Level 1 Status Code>] ...] [<Checksum>] <Terminator> Where: <Destination Name> - name of the destination on which the switch was attempted. <Level x Source Name> - name of the source that is switched on level x. <Level x Source Status Code> - indicates properties for the level in question. Any Combination of the following: R - The source is reporting a read back error. C - The source is reporting a confidence error.

USR - User Account Change

The User Account Change response indicates the user that made changes to the user account information stored on the PESA controller.

Table 9. USR - User Account Change

Response	~ * U S R * <Address> [<Checksum>] <Terminator> Where: <Address> - Uniquely identified source of user account change

Commands and Command Responses

Commands and command responses are interface initiated transactions. The interfacing device sends a command to the PESA controller for which the PESA controller may perform an action on the router. In response, the PESA controller returns a message with either confirmation information indicating receipt of the command or requested data.

The following is a parse tree representation of the command structure for the USP commands.

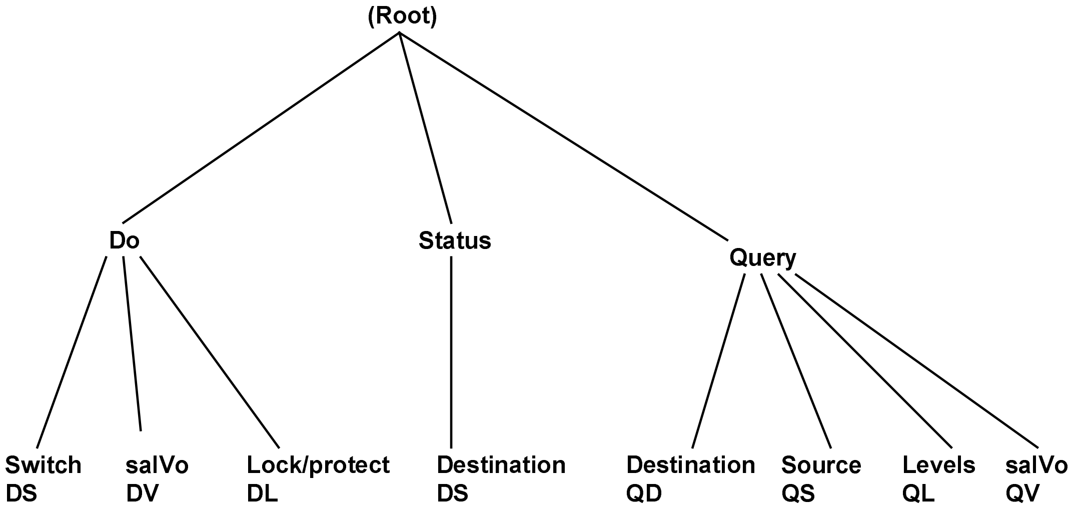

Figure 2. Parse Tree

Do Commands

The following set of commands perform an action on the PESA controller.

DL - Do Lock/Protect on Destination

This command causes the specified destination to be locked or unlocked by the CPU link device. A lock protect action will occur if the destination is presently not locked/protected, or if the CPU link's lock/protect priority is greater than the destination's existing lock/protect status or the CPU link has Master lock/protect priority.

Table 10. DL - Do Lock/Protect on Destination

Command	# * D L * <Destination Name> , <L|P|N> [<Checksum>] <Terminator> Where: <L|P|N>: L causes the destination to be locked P causes the destination to be protected. N causes the destination's lock/protect status to be cleared.
Response	* * D L [<Checksum>] <Terminator> - Action was taken. * * E [<Checksum>] <Terminator> - Error in Transmittal. * * L [<Checksum>] <Terminator> - Higher or Equal Priority Lock is in place. * * N [<Checksum>] <Terminator> - Invalid destination specification. * * P [<Checksum>] <Terminator>- Higher or Equal Priority Protect is in place.

DS - Do Switch

This command takes a switch on a single destination for all levels specified.

Commas are used to delimit the individual destination and source level fields.

If the command specifies less than the number defined levels, the command is accepted and unspecified levels are not switched.

Omitting a source definition from a level means that no switch is taken on that level.

Table 11. DS - Do Switch

Command	# * D S * <Destination Name> [, <Level 1 Source Name>] [, <Level 2 Source Name>] ... [, >] [<Checksum>] <Terminator> Where: > indicates that the last level specification applies for all following levels.
Response	* * B [<Checksum>] <Terminator>- Part or all of the requested switch was blocked/prevented due to controller configuration. Reasons for preventing a switch may include that the source to destination is blocked, the switch specified a reentry switched to a reentry, or a reentry switch specified a source to destination block. Portions of the switch request that did not have any configuration conflicts are attempted * * D S [<Checksum>] <Terminator> - Attempt to make full switch was performed. Confidence or readback errors may indicate that an error occurred during taking of the switch. * * E [<Checksum>] <Terminator> - Error in Transmittal. * * L [<Checksum>] <Terminator> - Destination was locked. * * N [<Checksum>] <Terminator> - Illegal specification of the switch (e.g., invalid destination, invalid source specification, too many levels specified). * * T [<Checksum>] <Terminator>- Part or all of the requested switch was blocked/prevented due to a unavailable Tie Line (Tie Line Block). (3500 Only)..

DV - Do Salvo

This command causes a salvo to be fired.

Table 12. DV - Do Salvo

Command	# * D V * <Salvo Name> [<Checksum>] <Terminator>
Response	* * D V [<Checksum>] <Terminator> - Salvo was fired * * E [<Checksum>] <Terminator> - Error in Transmittal. * * N [<Checksum>] <Terminator> - Invalid salvo specified.

Status Commands

The following set of commands request active status back from the PESA controller.

SD - Status Destination

This command requests that information be returned about the status of the specified destination.

Commas are used to delimit the individual destination and source fields.

If all defined levels are not specified in the response, it is assumed those levels have no defined status on them.

Table 13. SD - Status Destination

Command	# * S D * <Destination Name> [<Checksum>] <Terminator>
Response	* * E [<Checksum>] <Terminator> - Error in Transmittal. * * N [<Checksum>] <Terminator> - Invalid destination specified. * * S D * <Destination Information> CS @ Where: <Destination Information> is: <Destination Name> , [<Destination Status Codes>] [, [<Level 1 Source Name>] [, <Level 1 Source Status Codes>] ...] Where: [<Destination Status Codes>] indicate: C - The destination is in Chop L - The destination is Locked P - The destination is Protected <Level x Source Status Codes> R - The source is reporting a read back error C - The Level source is reporting a confidence error

SA - Status All

This command requests that information be returned about the status of the all defined destinations within a system. The nominal response to this command is a sequence of SD messages sent for all destinations.

If all defined levels are not specified in the response, it is assumed those levels have no defined status on them.

Note: This command only works on 3500Plus systems, V1.2 or greater.

Table14. SD - Status Destination

Command	# * S A [<Checksum>] <Terminator>
Response	* * E [<Checksum>] <Terminator> - Error in Transmittal. * * S D * <Destination Information> CS @ Where: <Destination Information> is: <Destination Name> , [<Destination Status Codes>] [, [<Level 1 Source Name>] [, <Level 1 Source Status Codes>] ...] Where: [<Destination Status Codes>] indicate: C - The destination is in Chop L - The destination is Locked P - The destination is Protected <Level x Source Status Codes> R - The source is reporting a read back error C - The Level source is reporting a confidence error

Query Commands

The following set of commands request configuration information be sent back from the PESA controller.

Given that query commands ask for information that may be represented in thousands of characters, the commands are structured to allow for multiple message transactions. Multiple messages handling a single command is handled by utilizing the message extension field.

QD - Query Destinations

This command requests that the controller destination names configuration be returned.

Given the ability to configure a large number of destinations, this command structure has the ability to send the destination information in multiple transactions. The initial transaction is sent as #*QD. It indicates that the controller is to send all the destination names starting from the first destination in the controller. If there are more destinations than can be sent in one message, the response comes back with as many destinations as can be sent in one message in addition to a “M” response extension. The interface device recognizes the response extension of M, and knows it needs to send a continuation command. The interface device sends #C*QD command indicating to the controller to return destinations starting where it left off on the previous query.

Table 15. QD - Query Destination

Command	# [C] * Q D [<Checksum>] <Terminator> Where: C indicates to the controller to continue where it left off from the last response.
Response	* * E [<Checksum>] <Terminator> - Error in Transmittal. * [M] * Q D * <Destination Name> , ... [<Checksum>] <Terminator>

QL - Query Levels

This command requests that the controller level configuration be returned. All level information is returned in proper level order.

Table 16. QL - Query Levels

Command	# * Q L [<Checksum>] <Terminator>
Response	* * E [<Checksum>] <Terminator> - Error in Transmittal. * * Q L * [<Level 1 Name> [, <Level 2 Name> ...]] [<Checksum>] <Terminator>

QS - Query Sources

This command requests that the controller source names configuration be returned.

Given the ability to configure a large number of sources, this command structure has the ability to send the source information in multiple transactions. The initial transaction is sent as #*QS. It indicates that the controller is to send all the source names starting from the first source in the controller. If there are more sources than can be sent in one message, the response comes back with as many sources as can be sent in one message in addition to a “M” response extension. The interface device recognizes the response extension of M, and knows it needs to send a continuation command. The interface device sends #C*QS command indicating to the controller to return sources starting where it left off on the previous query.

Table 17. QS - Query Sources

Command	# [C] * Q S [<Checksum>] <Terminator> Where: C indicates to the controller to continue where it left off from the last response.
Response	* * E [<Checksum>] <Terminator> - Error in Transmittal. * [M] * Q S * <Source Name> ... [<Checksum>] <Terminator>

QV - Query Salvos

This command requests that the controller salvo group configuration be returned.

Table 18. QV - Query Salvos

Command	# * Q V [<Checksum>] <Terminator>
Response	* * E [<Checksum>] <Terminator> - Error in Transmittal. * * Q V * <Salvo Group Name> , ... [<Checksum>] <Terminator>

Revision History

Rev.	Date	Description	By
1.0	11-12-96	Initial Release.	D. Bailey
1.1	09-04-97	Added changes to accommodate 3500 System Controller.	D. Bailey
A	10-20-99	Revised and updated per ECO-3247. Added SA command.	G. Tarlton
B	11-16-99	Changed security classification from Company Confidential to None per ECO-3499	G. Tarlton
C	03-06-01	Deleted Printing Specification per ECO CE00113.	G. Tarlton

PESA

Switching
Systems