PeriProducer 3.00 Release Notes

(Software Release 3.00)

Avaya Business Communications Manager
Release 6.0

Document Status: **Standard**

Document Number: **P0605081**

Document Version: **2.4**

Date: **June 2010**

Table of Contents

This page has been intentionally left blank.

Preface

Scope

The manual *PeriProducer 3.00 Release Notes* provides a summary of new features and enhancements to PeriProducer 2.30.

This manual does not describe PeriProducer 3.00 features and enhancements in detail. For a more complete description, see the *PeriProducer User's Guide*.

Intended Audience

This manual is intended for application developers of Avaya Media Processing Server (MPS) using the PeriProducer graphical application development tool. A basic understanding of the PeriProducer product as well as attendance in a PeriProducer training course is highly recommended.

How to Use This Manual

This manual uses many standard terms relating to computer system, software application functions, and the Internet. However, it contains some terminology that can only be explained in the context of the MPS system. Refer to the *Glossary of Avaya Media Processing Server Terminology* for definitions of MPS specific terms.

Initially, you should read this manual at least once, from start to finish. Later, you can use the Table of Contents to locate topics of interest for reference and review.

If you are reading this document online, use the cross-reference links (shown in blue) to quickly locate related topics. <LEFT> click once with your mouse while positioned with your cursor over the cross-reference link. Click on any point in a Table of Contents entry to move to that topic. Click on the page number of any Index entry to access that topic page.

To familiarize yourself with various specialized textual references within the manual, see *Conventions Used in This Manual* on page 9.

Periphonics is now part of Avaya. The name Periphonics, and variations thereof, appear in this manual only where it is referred to in a product. (For examples, a PeriProducer application, the PERImps package, the **perirev** command, etc.)

Organization of This Manual

The chapters are as follows:

Chapter 1 — PeriProducer 3.00 New Features and Enhancements

Discusses features and enhancements new to PeriProducer 3.00 and related materials.

Chapter 2 — Porting 2.30 Applications to 3.00

Outlines procedures and configurations for porting applications made with PeriProducer 2.30 and bringing them into PeriProducer 3.00.

Procedures for installing PeriProducer (as part of the complete MPS software package) are included in the *Installing MPS Software on the Solaris Platform* and *Installing MPS Software on the Windows Platform*.

Conventions Used in This Manual

This manual uses different fonts and symbols to differentiate between document elements and types of information. These conventions are summarized in the following table.

Conventions Used in This Manual Sheet 1 of 2

Notation	Description
Normal text	Normal text font is used for most of the document.
important term	The Italics font is used to introduce new terms, to highlight meaningful words or phrases, or to distinguish specific terms from nearby text.
system command	This font indicates a system command and/or its arguments. Such keywords are to be entered exactly as shown (i.e., users are not to fill in their own values).
command, **condition** and **alarm**	Command, Condition and Alarm references appear on the screen in magenta text and reference the *Command Reference Manual*, the *PeriProducer User's Guide*, or the *Alarm Reference Manual*, respectively. Refer to these documents for detailed information about **Commands**, **Conditions**, and **Alarms**.
file name / directory	This font is used for highlighting the names of disk directories, files, and extensions for file names. It is also used to show displays on text-based screens (e.g., to show the contents of a file.)
on-screen field	This font is used for field labels, on-screen menu buttons, and action buttons.
<KEY NAME>	A term that appears within angled brackets denotes a terminal keyboard key, a telephone keypad button, or a system mouse button.

Conventions Used in This Manual Sheet 2 of 2

Notation	Description
Book Reference	This font indicates the names of other publications referenced within the document.
cross reference	A cross reference is shown on the screen in blue. Click on the cross reference to access the referenced location. A cross reference that refers to a section name accesses the first page of that section.
	The Note icon identifies notes, important facts, and other keys to understanding.
	The Caution icon identifies procedures or events that require special attention. The icon indicates a warning that serious problems may arise if the stated instructions are improperly followed.
	The flying Window icon identifies procedures or events that apply to the Windows 2000 operating system only.[1]
	The Solaris icon identifies procedures or events that apply to the Solaris operating system only.[2]

1. Windows 2000 and the flying Window logo are either trademarks or registered trademarks of the Microsoft Corporation.
2. Solaris® is a registered trademark of The Open Group in the U.S. and other countries.

Solaris and Windows 2000 Conventions

This manual depicts examples (command line syntax, configuration files, and screen shots) in Solaris format. In certain instances Windows 2000 specific commands, procedures, or screen shots are shown where required. The following table lists examples of general operating system conventions to keep in mind when using this manual with either the Solaris or Windows operating system.

	Solaris	Windows 2000
Environment	`$MPSHOME`	`%MPSHOME%`
Paths	`$MPSHOME\common\etc`	`%MPSHOME%\common\etc`
Command	**`<command> &`**	**`start /b <command>`**

Two-Button (Windows 2000) vs. Three-Button (Solaris) Mouse

<SELECT>	Left button
<ADJUST>	Left and Right together
<MENU>	Right button

<SELECT>	Left button
<ADJUST>	Middle button
<MENU>	Right button
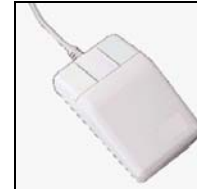
Trademark Conventions

The following trademark information is presented here and applies throughout for third party products discussed within this manual. Trademarking information is not repeated hereafter.

Solaris® and Motif® are registered trademarks of The Open Group in the U.S. and other countries.

Solaris, SunOS, OpenWindows, SPARC, and UltraSPARC are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Microsoft, MSSQL, Windows, Windows 2000, Internet Explorer, and the Flying Windows logo are either trademarks or registered trademarks of Microsoft Corporation.

Oracle® is a registered trademark of Oracle Corporation.

Sybase™ and SYBASE™ are trademarks of Sybase, Inc. or its subsidiaries.

Informix® and INFORMIX® are registered trademarks of Informix Corporation or its affiliates.

This page has been intentionally left blank.

PeriProducer 3.00 New Features and Enhancements

<u>This chapter covers:</u>

1. New Features and Enhancements

2. PeriProducer Block Changes

3. Discontinued Features/Functions

4. Other PeriProducer Changes

New Features and Enhancements

New Blocks

The following blocks are new to PeriProducer 3.00:

Block	Function
Abort	Abort input/output
Bridge	Perform bridging functions (moved from Originate block in PeriProducer 2.30)
Call Control	Send data to telephony protocol layer/perform a hookflash (moved from Originate block in PeriProducer 2.30)
Call Conferencing	Manage conference call functions
Call Progress Detection	Manage call progress detection functions
Edit Sequence	Manage touch tone input editing and user edit sequences
Line Operations	Perform phone line operations (e.g., offer call, accept call, get line/application resource from pool)
Media Operations	Perform operations on MultiMedia File elements or audio files
Select Input	* Not currently implemented

See the *PeriProducer 3.00 User's Guide* for full descriptions of the blocks.

Call Conferencing

Call Conferencing allows up to 16 full participants (or 15 full participants with 19 listen port callers) to connect to a call. Per-line talk and listen privileges can be set at the start of a conference and dynamically changed during the conference. Call Conferencing includes the "coaching" feature which allows a conference administrator (a "coach") to talk to another person in the conference (a "student") and have the speech output heard only by the student.

The actual number of ports available on a system is determined by the DSP configuration.

Block-level Access to Protocol Data

Telephone line function blocks (Answer, Disconnect, Call Control, Originate) can send/receive protocol data directly to/from the telephony protocol layer using built-in block functions. Specify data to send (or receive) using standard PeriProducer data cards or literals (send only).

Blocking/Non-blocking Execution

Many PeriProducer blocks allows the option of choosing blocking (execution waits at that block until it receives success/failure message) or non-blocking (execution continues to next block without waiting for message) execution. A "Wait" checkbox enables/disables blocking execution for the applicable blocks. See Blocking/Non-blocking Operation in the *PeriProducer 3.00 User's Guide* for details.

Resource Pool Support

The MPS allows resources (such as phone lines and applications) to be defined in the configuration files as shared pools. In the appropriate blocks (e.g., Line Operations), PeriProducer provides built-in block access to these resources.

Saving Partial Recorded Data

PeriProducer saves partially received Caller Message Recording messages and faxes in the event of a receive failure. The partial data is either stored in a new file (if receiving directly into a file) or the application receives a Media Storage token which references an element in a MultiMedia File. This behavior can be optionally enabled and disabled in a Record block.

Call Progress Detection

Using Call Progress Detection (CPD) is updated in PeriProducer 3.00. CPD is controlled by dynamically enabling and disabling specific tone/event detection in the Call Progress Detection block. All CPD events return to the application as the cpd condition with the specific event (Busy, Reorder, etc.) in the condition data. See the *PeriProducer 3.00 User's Guide* for details.

PeriProducer Block Changes

Connection IDs (Caller I/O blocks)

Phone line and resource numbers are replaced with Connection IDs (CIDs). Connection IDs indicate the component name and the line number of the connection. For example, the CID "mps24.1" indicates the connection is from line one on component "mps24". The default Connection ID for the current phone line is available in the System folder's DefaultCID data card. The CID for an operation is supplied by the system (e.g., when a resource is allocated).

Accessory Toolkit Blocks

The Table Search, Table Sort, Date Calculations, and Send email blocks moved from the main toolkit to the accessory toolkit.

Answer Block

Continue on Ring Detect Removed

The "Continue on Ring Detect" option is removed. In general, use the "Get Call" -> "Alert Call" states to accomplish the same result.

Get Phone Number Options Removed

The Get Dialed Phone Number and Get Caller's Phone Number options are removed. Depending on the specific telephony protocol, this information is automatically returned by the system for all Answer block actions (except "Reject Call").

Disconnect Block

Abandoned Call Counter Removed

The MPS does not automatically track system abandoned calls. The "Mark as abandoned" option is removed. To track abandoned calls, create application statistics and use PeriReporter to display the reports.

Speak Block

Speech Recognition/Text To Speech

All speech recognition and synthesis functions are performed by external resources. Refer to the appropriate resource guides for details on using the resource for optimal programming results.

Speak a URL

The element to speak can be specified by a valid URL address (e.g., http://server1/prompts/greeting1.wav).

Read Phone Block

The Read Phone Block now handles Speech Recognition Errors and provides a failure path for a recognition error. The following recognition errors cause the application to follow the "Recognition Error" path:

- Speech Too Early
- Too Much Speech
- Too Slow Recognition

System Block

Diagnostic Functions Removed

The diagnostics function is removed.

Condition Data

The "Get Condition Data" function is added. This replaces using the condition-data call function. "Get Condition Data" requires a folder with the same structure, data names, and data types expected with the condition (see *Enhanced Condition Data* on page 22).

Send Host Block

Virtual Terminal (VT) Allocation

VT allocation functions are available from the Send Host block. These replace the VT allocation Environment block options.

new VT options
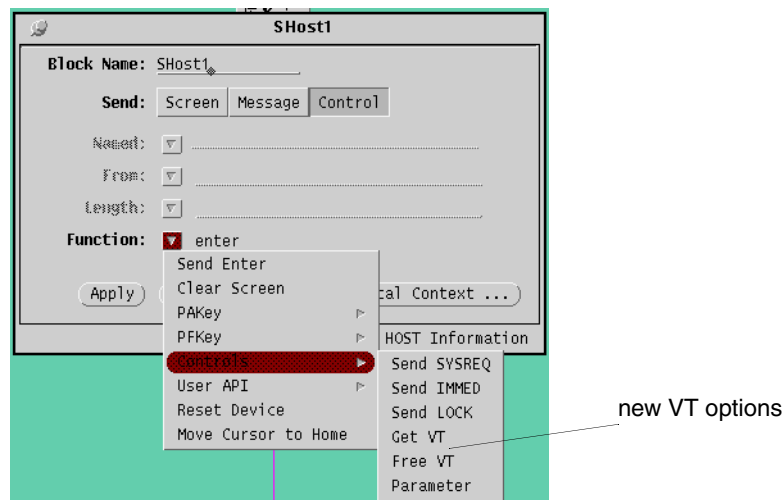
Setting Host Environment Parameters

Host environment parameters are sent from the Send Host block. This function replaces the Environment block "host" option.

parameter option
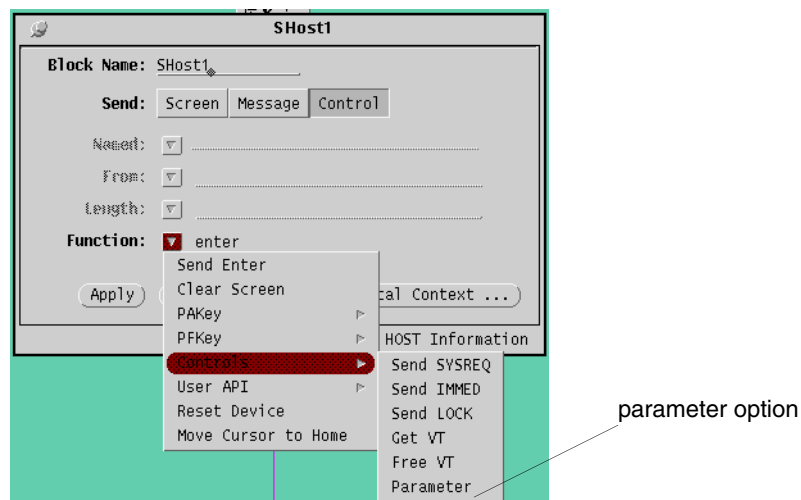
Receive Host Block

Asynchronous Operation

The "Asynchronous" checkbox is replaced by the "Wait" checkbox (see *Blocking/Non-blocking Execution* on page 15). Receive Host operates asynchronously when the "Wait" checkbox is unchecked.

Send Fax Block

Fax Composition Removed

The fax Composition feature is not supported. All controls/options used to create and store composed faxes are removed.

Send Fax from TIFF File

PeriProducer 3.0 supports sending faxes directly from TIFF format files. TIFF faxes no longer need to be imported into a MultiMedia File and accessed by a media storage token (this is still supported).

Fax Mode Removed

PeriProducer 3.0 supports only Group 3 faxes.

Receive Fax Block

Receive Fax Direct into TIFF File

Applications can save an incoming fax directly to a TIFF-format file. In previous versions of PeriProducer, faxes were stored in MultiMedia Files and had to be exported to individual TIFFs.

Saving to Named Element

A fax can be saved to a specific-named element in a MultiMedia File. If the application does not specify an element name, the fax is stored as a new element and can be accessed by the returned media storage token.

Local Station ID

The Local Station ID (which typically represents the phone number of the station receiving the fax) can be specified in the block. The Local Station ID is typically displayed on the transmitting fax machine.

Fax Mode Removed

PeriProducer 3.0 supports only Group 3 faxes.

Originate Block

Moved Functions

All bridging functions are moved to the new 3.00 Bridge Block. Hookflash is moved to the Call Control Block. The End Transfer function is deleted (the Disconnect block can be used for this purpose).

Record Block

Asynchronous Recording

Asynchronous recording allows for an unlimited duration message. Asynchronous recording is explicitly started and stopped by "Begin" and "End" functions of the Record block.

Full Duplex Recording

Full duplex recording saves all audio that occurs on a phone line, caller and system. For example, an application can record a full conversation between a caller and a referral agent. Recording can also be set to record either caller-only or system-only output.

Moved Functions

Element deletion functions moved to the Media Operations Block.

Recorded Element Destinations

PeriProducer 3.00 supports recording directly into a specified audio format file (typically WAV). Applications can also append recordings to existing MMF elements.

Abort Recording using Touch Tones

Callers can abort CMR recordings using a touch tone(s). Note that this differs from using a touch tone to terminate (and subsequently save) a recording.

Resource Block

Asynchronous Execution Changes

There is no longer an "Async" checkbox in the Resource block. To perform the resource receive asynchronously, uncheck the "Wait" box.

Resource Support

The following new resources are available in the Resource block: lvr-p, lvr-s, extts-r, sip, DTMF, Player, Fax, Recorder, FullDuplex Recorder. Many resources are no longer supported (see *Unsupported Resources*).

Discontinued Features/Functions

Fax Composition

Fax composition is not supported in PeriPro 3.00. Faxes to be sent must be created by external applications or received (and stored for later use) by the MPS.

Hardware Properties Window

The Hardware Properties Window is removed.

Unsupported Resources

The following resources are no longer supported in PeriProducer 3.00:

- mps
- abb
- asdi
- ast
- iwr
- ppd
- modem
- mts
- iscp
- ctx
- lcr

Obsoleted Functions

The Media Operations block in PeriProducer has obsoleted the file-to-message, message-to-item and message-to-file functions.

Other PeriProducer Changes

Enhanced Condition Data

Condition Data is now returned as a data structure consisting of different field/value pairs. To access the expanded condition data, use the System block's "Get Condition Data" function and specify the appropriate response folder (templates provided in $PPROHOME/sample/folders). To access all of the condition data, the folder must have data cards defined as the same name and type as the associated condition data field. The data for one field only can be obtained by specifying a datacard whose name matches the desired field.

The condition data for many conditions have a "Status" field. The Status field value is automatically copied into the System folder's ConditionData card.

Default Text to Speech Resource Specification

To specify a default external Text to Speech resource (used with implicit resource allocation for all "Speak as TTS" prompts), use the System folder's DefaultTTSRsrc datacard. The default value for DefaultTTSRsrc is "extts-b". This replaces the Text To Speech Resource setting in the Hardware properties window of PeriProducer 2.30.

System Folder Changes

- The Constant TTSvoice card is no longer available.

- The ConfidenceLevel card is no longer available.

- The following constants are added:
    - None
    - All
    - Any
    - Flush
    - On
    - Off
    - Speech ("S" to identify speech input)
    - DTMF ("D" to identify DTMF input)

- The following datacards are added:
    - AutoDetectTime
    - DefaultCID
    - DefaultOutPool
    - TransferCID
    - DefaultTTSRsc
    - ApplicationNumber (replaces LineNumber)

Environment Options

Many PeriProducer 2.30 Environment options have been obsoleted or replaced by new environment options and/or block built-in functions in PeriProducer 3.00. The following table lists the 2.30 Environment options and the analogous 3.00 option (if any). Comments are provided where appropriate.

Environment Options Conversions Sheet 1 of 7

2.30 Environment Option	3.00 Equivalent
Phone Environment	
answer	Unsupported as environment option. Use the Answer block (Answer function) to answer a call.
bargein	Supported for legacy applications only
bargeinevent	Unsupported (for legacy applications, this is an implied "start of speech" event)
edit	Unsupported as environment option. Use the Edit Sequence Block to enable/disable DTMF input editing.
ekeybd	Unsupported
emess	Unsupported
eopt	Unsupported
eretry	Unsupported
first	DtmfFirst (Application and System Environment options)
fkeybd	Unsupported
fmess	Unsupported
fopt	Unsupported
fretry	Unsupported
ikeybd	Unsupported
imess	Unsupported
inter	DtmfInter (Application and System Environment options)
iopt	Unsupported
iretry	Unsupported
keepterm	Unsupported as environment option. Use the "Retain" option (for the termination character) in Edit Sequence block.
termchar	Unsupported as environment option. Use the Edit Sequence block to enable/set a termination character.
timeout	Unsupported
tnabort	Unsupported
tnext	Unsupported

Environment Options Conversions Sheet 2 of 7

2.30 Environment Option	3.00 Equivalent
tnull	Unsupported
total	LineTotalCall (Application and System Environment options)
trepeat	Unsupported
typeahead	Unsupported
tzone	Unsupported
Phone Line Task Environment	
autoid	Supported for legacy applications only
backsp	Unsupported as environment option. Use the Edit Sequence block to enable/set a backspace edit sequence.
busy	LineStandbyMode, set to Busy (Application and System Environment options)
callref	Unsupported
delete	Unsupported
eXtext	Unsupported. Use the Edit Sequence block to enable/set user edit sequences.
getring	Unsupported as environment option. Use the Answer block "Get Call" and "Alert Call" functions to continue application processing without answering the line.
noanswer	LineStandbyMode, set to NoAnswer (Application and System Environment options)
repeat	Unsupported
vrepeat	Unsupported
xterm	Unsupported
Phone Resource Server Environment	
prs	Unsupported
abeep	Unsupported
asetup	Unsupported
cdtmf	Unsupported
echo	Unsupported
idle	Unsupported
mdeblock	Unsupported
rscconfig	RscConfig (Application and System Environment options)
smaxlen	Unsupported
smaxres	Unsupported as environment option

Environment Options Conversions Sheet 3 of 7

2.30 Environment Option	3.00 Equivalent
sminlen	Unsupported as environment option
srecord	Unsupported as environment option
tfield	Unsupported as environment option
tfinal	Unsupported as environment option
tfirst	Unsupported
tstop	Unsupported as environment option
Message Recording Environment	
firstsil	RecFirstSil (Application and System Environment options)
intersil	RecInterSil (Application and System Environment options)
maxreclength	Unsupported
silstrip	RecSilStrip (Application and System Environment options)
Call Origination/Referral Environment	
bctt	Unsupported
bcvoc	Unsupported
brtt	Unsupported
brvoc	Unsupported
hctt	Unsupported
hcvoc	Unsupported
hrtt	Unsupported
hrvoc	Unsupported
number	Unsupported
pdial	Unsupported
reftot	Unsupported
tpickup	Unsupported
vbridge	Unsupported
Call Progress Detection Environment	
cpansup	Unsupported
cpdial	Unsupported
cpdisc	Unsupported
decide	Unsupported
notone	Unsupported
pickup	Unsupported

Environment Options Conversions Sheet 4 of 7

2.30 Environment Option	3.00 Equivalent
rsilence	Unsupported
rvoice	Unsupported
silence	Unsupported
stop	Unsupported
tone	Unsupported
voice	Unsupported
window	Unsupported
Speech Management Environment	
clear	Supported for legacy applications only. Use the Abort block with the "Caller I/O Only" option enabled.
vioabort	Unsupported as environment option. Use the Abort block.
Host Environment	
er	er
freevt	Supported for legacy applications only. Use the Send Host block "Free VT" (in Control > Function > Controls) option.
getvt	Supported for legacy applications only. Use the Send Host block "Get VT" (in Control > Function > Controls) option.
headermode	headermode
hostctl	hostctl
intime	intime
parameter	Supported for legacy applications only. Use the Send Host block "parameter" (in Function > Controls) option.
refer	refer
rfno	rfno
session	session
setaid	setaid
unlocks	unlocks
usepool	usepool
Advanced Phone Line Management	
phone	Unsupported as environment option
capdtmf	Unsupported as environment option
capmf	Unsupported as environment option
captdd	Unsupported as environment option
detect	Supported for legacy applications only

Environment Options Conversions Sheet 5 of 7

2.30 Environment Option	3.00 Equivalent
dfirst	Unsupported as environment option
dinter	Unsupported as environment option
disc	Unsupported
dtmf	DtmfToneDur (Application and System Environment options)
eline	Unsupported
endtone	Unsupported
fconnect	Supported for legacy applications only. Use the Answer block "Answer Call" function.
finish	Unsupported
flash	Unsupported
gobusy	Unsupported
guard	DtmfGuard (Application and System Environment options)
indial	Unsupported
keyin	Unsupported
keyout	Unsupported
mftone	Unsupported
nansup	Unsupported
nofhook	Unsupported as environment option. Use the Answer block "Get Call" and "Alert Call" functions.
rend	Unsupported
reortext	Unsupported
ringback	Unsupported
ringtext	Unsupported
rline	Unsupported
rvalid	Unsupported
sendcp	Unsupported
ISDN Environment	
cats	Unsupported as environment option. Use the "Protocol Data" fields in the Answer, Disconnect, Originate, and Call Control blocks (as appropriate).
ccode	Unsupported as environment option. Use the "Protocol Data" fields in the Answer, Disconnect, Originate, and Call Control blocks (as appropriate).

Environment Options Conversions Sheet 6 of 7

2.30 Environment Option	3.00 Equivalent
cproc	Unsupported as environment option. Use either the "Protocol Data" fields in the Answer, Disconnect, Originate, and Call Control blocks (as appropriate), or the Answer block "Proceeding Call", function. Refer to the protocol-specific documentation for details on sending call state information.
smrt	Unsupported as environment option. Use the "Protocol Data" fields in the Answer, Disconnect, Originate, and Call Control blocks (as appropriate).
udata	Unsupported as environment option. Use the "Protocol Data" fields in the Answer, Disconnect, Originate, and Call Control blocks (as appropriate).
Optional Exception Conditions	
crefer	Unsupported as environment option
crepeat	Unsupported as environment option
rngback	Unsupported as environment option
uedit3	Unsupported as environment option (in the context of taking a digital system out of wait for an outdial complete message)
Generic Environment Options	
Phone Line Manager Options	Superseded by "Application and System Options"
Speech Manager Options	Superseded by "Application and System Options"
Host Manager Options	Superseded by "Host Environment"
Vengine Options	Vengine Options
PeriWeb Options	PeriWeb Options
VENGINE Environment	
alarmdbtask	alarmdbtask
apprestart	apprestart
centurymark	centurymark
debug	debug
deltimedcall	deltimedcall
hnowait	Unsupported (a corresponding VENGINE runtime option is available)
intermsg	intermsg
maxmessage	Unsupported
mode	mode
notice	notice

Environment Options Conversions Sheet 7 of 7

2.30 Environment Option	3.00 Equivalent
numset	numset
rscertime	rscertime
rscintime	rscintime
setvpsline	setvpsline
softterm	softterm
speak	speak
timedcall	timedcall
unnotice	unnotice
vmstimedcall	vmstimedcall
vpsrcvtime	vpsrcvtime
webtimeout	webtimeout

Conditions

When a PeriProducer 2.30 application is opened in 3.00, PeriProducer attempts to automatically convert handle conditions to their counterparts in 3.00. Condition conversions shown with a preceding dollar sign ("$") are displayed in the application as the 2.30 condition but are automatically converted to the appropriate 3.00 condition at runtime.

The following table lists the 2.30 condition, the corresponding 3.00 condition (if any), and how PeriProducer 3.00 converts the condition when a 2.30 application is ported to 3.00.

Conditions Conversion Sheet 1 of 10

2.30 Condition	3.00 Condition	Conversion
abend	abend	abend
addfail	importfail w/Status "ErrInUse" in condition data	$addfail
addsucc	importcmp	importcmp
altlinkdown	altlinkdown	altlinkdown
ansfail	answerfail	answerfail
asrdet	asrdet	asrdet
asyncdata	Unsupported	

Conditions Conversion Sheet 2 of 10

2.30 Condition	3.00 Condition	Conversion
asyncfail	Unsupported	
autofail	detinputfail	detinputfail
autotim	autotim	autotim
avserr	avserr	avserr
badoperation	badoperation	badoperation
badparameter	badparameter	badparameter
calltim	calltim	calltim
carloss	Unsupported	
ccs7cc	ccs7cc	ccs7cc
chartim	getinputfail w/Status "ErrInter" field in condition data	getinputfail
cmrhigh	Unsupported	
cmrlow	Unsupported	
comfail	comfail	comfail
conn	answercmp	answercmp
crefer	Unsupported	
crepeat	Unsupported	
crepmax	Unsupported	
cticond	cticond	cticond
ctidown	ctidown	ctidown
ctifail	ctifail	ctifail
ctiup	ctiup	ctiup
ctxcc	Unsupported	
ctxfailcc	Unsupported	
ctxokcc	Unsupported	
cvoice	Unsupported	
dcdown	Unsupported	
dcup	Unsupported	
deadlock	deadlock	deadlock
delcomp	delcmp	delcmp
delfail	delfail	delfail
dialtn	Unsupported	

Conditions Conversion Sheet 3 of 10

2.30 Condition	3.00 Condition	Conversion
disable	Unsupported	
disc	Dependent upon event which caused disc condition.	
	disc (caller hangup)	$disc
	disccmp (system-initiated disconnect)	$disc
discfail	discfail w/Status field in condition data	$discfail
dtmfzl	recordfail w/Status "ErrZeroLengthDTMF" in condition data	recordfail
dupkey	dupkey	dupkey
dupvalidx	dupvalidx	dupvalidx
enable	Unsupported	
endfail	discfail	$endfail
endfile	endfile	endfile
error	error	error
ertimeout	Dependent upon event which caused ertimeout condition	
	ertimeout (gen'd by VENGINE)	$ertimeout
	hrcvmapfail w/Status "ErrTimeout" in condition data (failed receive map)	$ertimeout
	hrcvtxtfail w/Status "ErrTimeout" in condition data (failed receive text)	$ertimeout
expired	expired	expired
faxdet	faxdet	faxdet
forcefree	forcefree	forcefree
forward	Unsupported	
fromphone	fromphone	fromphone
frstim	getinputfail w/Status "ErrFirst" in condition data	getinputfail
ftomfail	Unsupported	
ftomsucc	Unsupported	

Conditions Conversion Sheet 4 of 10

2.30 Condition	3.00 Condition	Conversion
getfail	Dependent upon event which caused getfail condition	
	getfail (gen'd by VENGINE)	$getfail
	getrsrcfail (failed resource get request)	$getfail
	rcvfaxfail w/Status "ErrNoFaxAvail" in condition data (failed receive fax when fax not available)	$getfail
	sndfaxfail w/Status "ErrNoFaxAvail" in condition data (failed send fax when fax not available)	$getfail
getvtfail	hgetvtfail	hgetvtfail
getvtpass	hgetvtcmp	hgetvtcmp
gotres	Dependent upon event which caused gotres condition	
	getrsrccmp	$gotres
	rcvfaxcmp	$gotres
	sndfaxcmp	$gotres
green	Unsupported	
hctloff	hctloff	hctloff
hctlon	hctlon	hctlon
heldres	Unsupported	
hkfcomp	hookflshcmp	hookflshcmp
hkffail	hookflshfail	hookflshfail
hostasyncevt	hostasyncevt	hostasyncevt
hostdown	hostdown	hostdown

Conditions Conversion Sheet 5 of 10

2.30 Condition	3.00 Condition	Conversion
hostfail	Dependent upon event which caused hostfail condition	
	hrcvmapfail w/Status "ErrNoData" or "ErrTimeout" in condition data (failed to receive map)	$hostfail
	hrcvtxtfail w/Status "ErrNoData" or "ErrTimeout" in condition data (failed to receive text)	$hostfail
	hsndmapfail (failed to send map)	$hostfail
	hsndtxtfail (failed to send text)	$hostfail
	hsndaidfail (failed to send AID key)	$hostfail
hostup	hostup	hostup
hstatdata	hstatdata	hstatdata
idle	Unsupported	
inf	Dependent upon event which caused inf condition	
	inf (gen'd by VENGINE)	$inf
	rcvfaxfail w/Status "ErrNoFaxAvail" in condition data (failure to receive fax)	$inf
	recordfail w/Status "ErrZeroLengthDTMF" or "ErrZeroLengthSilence" in condition data (failure to record CMR message)	$inf
intertimeout	intertimeout (gend' by VENGINE)	$intertimeout
invreq	invreq	invreq
ioerr	ioerr	ioerr
iscpf	iscpf	iscpf
isdncc	Unsupported	
iupdbusy	Unsupported	
iupdcomp	Unsupported	
iupdfail	Unsupported	
lengerr	lengerr	lengerr
linkdown	linkdown	linkdown

Conditions Conversion Sheet 6 of 10

2.30 Condition	3.00 Condition	Conversion
lockfail	lockfail	lockfail
logdeny	logdeny	logdeny
lost	Unsupported	
marshall	mailshall	marshall
mmfhigh	mmfhigh	mmfhigh
mmflow	mmflow	mmflow
modvar	modvar	modvar
mpscc	Unsupported	
mpsinfo	Unsupported	
mpsoc	Unsupported	
mpsof	Unsupported	
mtoffail	exportfail	exportfail
mtofsucc	exportcmp	exportcmp
nilobjref	nilobjref	nilobjref
nonexistobj	nonexistobj	nonexistobj
norecfound	norecfound	norecfound
norestart	norestart	norestart
nospace	nospace	nospace
notfnd	notfnd	notfnd
notimpl	notimpl	notimpl
notlogon	notlogon	notlogon
notopen	notopen	notopen
oa	speakcmp w/Status "Abort" in condition data	speakcmp
oc	Dependent upon event which caused oc condition	
	speakcmp w/Status "Done" in condition data (completed speak request)	$oc
	sndfaxcmp (completed send fax)	$oc
	sndrsrccmp (completed send resource)	$oc

Conditions Conversion Sheet 7 of 10

2.30 Condition	3.00 Condition	Conversion
of	Dependent upon event which caused of condition	
	of (gen'd by VENGINE)	$of
	sndfaxfail w/Status "ErrNoFaxAvail" in condition data (failed send fax)	$of
	sndrsrcfail (failed send resource)	$of
	speakfail (failed speak request)	$of
ofaxdet	Unsupported	
orberr	orberr	orberr
oscoc	sndrsrccmp	
oscof	sndrsrcfail	
outbad	origfail w/Status "ErrInvalidLineState" in condition data	origfail
outbsy	origfail w/Status "ErrBusy" in condition data	origfail
outcomp	origcmp	origcmp
outfail	origfail w/Status "ErrBusy", "ErrInvalidLineState", ErrNoAnswer", or "ErrRejected" in condition data	$outfail
outnoa	origfail w/Status "ErrNoAnswer" in condition data	origfail
outrej	origfail w/Status "ErrRejected" in condition data	origfail
outvoa	Unsupported	
pgid	pgid	pgid
pgml	pgml	pgml
pgun	pgun	pgun
prsfree	Dependent upon event which caused prsfree condition	
	prsfree (gen'd by VENGINE)	$prsfree
	freersrccmp (completed free resource)	$prsfree
qiderr	qiderr	qiderr

Conditions Conversion Sheet 8 of 10

2.30 Condition	3.00 Condition	Conversion
qzero	qzero	qzero
rcverr	rcvrsrcfail	$rcverr
rcvnull	Dependent upon event which caused rcvnull condition	
	rcvnull (gen'd by VENGINE)	$rcvnull
	hrcvmapfail w/Status "ErrNoData" in condition data (failed receive map)	$rcvnull
	hrcvtxtfail w/Status "ErrNoData" in condition data (failed receive text)	$rcvnull
rcvoice	Unsupported	
rdcdown	Unsupported	
rdcup	Unsupported	
rdisable	Unsupported	
red	Unsupported	
refbad	transferfail w/Status "ErrInvalidLineState" in condition data	transferfail
refbeg	transfercmp	transfercmp
refcan	disccmp	$refcan
reffail	transferfail w/Status "ErrInvalidLineState" or "ErrRejected" in condition data	$reffail
refrej	transferfail w/Status "ErrRejected" in condition data	transferfail
reftim	calltim	$reftim
renable	Unsupported	
reorder	Unsupported	
resumefail	resumefail	resumefail
rgreen	Unsupported	
ring	alertcmp	alertcmp
rinfail	alertfail	alertfail
rlost	Unsupported	
rngback	Unsupported	

Conditions Conversion Sheet 9 of 10

2.30 Condition	3.00 Condition	Conversion
rred	Unsupported	
rscoc	rscoc	rscoc
rscof	rscof	rscof
rsilence	Unsupported	
rvoice	Unsupported	
rxdcdown	Unsupported	
rxdcup	Unsupported	
rxdisable	Unsupported	
ryellow	Unsupported	
sentcp	sndtonecmp	sndtonecmp
serverdown	serverdown	serverdown
setfail	ctrlrsrcfail	$setfail
setres	ctrlrsrccmp	ctrlrsrccmp
silence	Unsupported	
silzl	recordfail w/Status "ErrZeroLengthSilence" in condition data	recordfail
softterm	softterm	softterm
sqlerr	sqlerr	sqlerr
sslfail	sslfail	sslfail
stoptim	Unsupported	
stpring	Unsupported	
tcapcc	tcapcc	tcapcc
tftprecfail	tftprecfail	tftprecfail
tftprecsucc	tftprecsucc	tftprecsucc
tftpsendfail	tftpsendfail	tftpsendfail
tftpsendsucc	tftpsendsucc	tftpsendsucc
timeres	timeres	timeres
toomanyrows	toomanyrows	toomanyrows
tophonefail	tophonefail	tophonefail
ttdata	getinputcmp	getinputcmp
ttdet	ttdet	ttdet

Conditions Conversion Sheet 10 of 10

2.30 Condition	3.00 Condition	Conversion
ttfail	getinputfail w/Status "ErrFirst" in condition data	$ttfail
uedit0 - uedit3	uedit0 - uedit3	uedit0 - uedit3
unexdata	unexdata	unexdata
unexhost	unexhost	unexhost
unexphone	unexphone	unexphone
valueperr	valueperr	valueperr
voice	Unsupported	
vrto	vrto	vrto
webfail	webfail	webfail
webtimeout	webtimeout	webtimeout
xdisable	Unsupported	
xmtcomp	Unsupported	
xmtfail	Unsupported	
yellow	Unsupported	

Using MultiMedia Elements

Deleting Elements

In PeriProducer 3.00, delete MMF elements using the Media Operations block "Delete" function. The System Block "Delete MS Token or Vocabulary Item" function is still available, but does not provide a failure path connector as does the Media Operations block.

Importing and Exporting Elements

In PeriProducer 3.00, import and export MMF elements using the Media Operations block "Import" and "Export" functions. The associated call functions are still available, but using the System block does not provide a failure path connector as does the Media Operations block.

Receiving Speech Recognition Resource Results

If the system uses DTMF detection and speech recognition, the application does not need to start **autoid** and handle detection conditions (**asrdet/ttdet**). The Read Phone operation passes back a field to the application which indicates the source of the input (speech or DTMF, accordingly).

Play the initial prompts with "Accept **Input**..." to allow for both DTMF and speech input. To allow speech recognition bargein, enable the Prompt Interrupt feature (click the ▣ button).

Miscellaneous

VRAM Language

VRAM is no longer supported.

Downward Porting of .ppr Source Code

Source code cannot be ported to previous versions of PeriPro.

MPS Release Level

You must use PeriPro 3.0 on MPS 2.1 or above.

This page has been intentionally left blank.

2

Porting 2.30 Applications to 3.00

This chapter covers:

1. Porting PeriProducer 2.30 applications to the 3.00 environment

2. Porting Standard 2.30 Toolkit Applications to 3.00

3. Porting MPS Toolkit Applications to 3.00

4. Known Conversion Issues

Porting 2.30 Applications to 3.00

PeriProducer 3.00 supports legacy (PeriProducer 2.30) applications. Load the application into PeriProducer and PeriProducer converts the application automatically.

Before porting the application, read through the entire Release Notes to find out what features are not supported/have changed and other issues.

Application Porting Considerations

Consider the following when porting the legacy application:

- Make a copy of the source file and port the copy.
- Back up the source file on another system and/or external media
- Save the ported copy as a different name (like <original_name>300)

Porting Standard 2.30 Toolkit Applications to 3.00

Applications that used the standard PeriProducer 2.30 toolkit are automatically converted by PeriProducer 3.00. No other setup or configuration procedures are required to convert the 2.30 legacy applications.

Load into PeriProducer 3.00

Load the legacy application into PeriProducer as you would any other application. The conversion takes place automatically. The PeriProducer logo will indicate that the application has changed and needs to be saved.

Use the PeriProducer Command Line

Use the following command line syntax to convert a legacy application:

peripro -c *<application>*

For example, to convert `numdemo.ppr`, enter **peripro -c numdemo**. You can also use filename wildcards (e.g., **peripro -c num***) to batch convert applications. The example converts all applications in the current directory whose names start with "`num`".

Porting Events

During application porting, the following events occur:

- Conditions and environment options are converted when possible. Refer to Conditions on page 29 and Environment Options on page 23 for conversions.
- Blocks are converted to their nearest possible counterparts in 3.00.
- Resources are obsoleted.
- The original file is renamed by changing its extension to `.p2X`.
- The "PeriPro 2.30 Runtime Behavior" option is enabled by default. See the *PeriProducer User's Guide* for information about the 2.30 Runtime Behavior option.
- A file with name beginning with `conversionlog_` followed by the application name and with extension `.txt` is created and contains messages pertaining to the conversion process. A similar file is created with the extension `.html` and is in HTML format.

Porting MPS Toolkit Applications to 3.00

Applications that use the MPS Toolkit require procedures to replace the old MPS toolkit with a special conversion toolkit for porting to the PeriProducer 3.00 toolkit. These special conversion toolkits are distributed with PeriProducer 3.00 by default.

MPS Toolkit Replacement

There are two replacement versions of MPS Toolkit: 1.30 and 2.00. The files are located in `$PPROHOME/conv` as follows:

`conv/`
- `IMpsfolder1.3` (`IMps.folder` for MPS Toolkit 1.3)
- `IMpsfolder2.0` (`IMps.folder` for MPS Toolkit 2.0)
- `mpsconvkit1.3` (`mps.pprotoolkit` for MPS Toolkit 1.3)
- `mpsconvkit2.0` (`mps.pprotoolkit` for MPS Toolkit 2.0, including conference toolkit components)
- `IMps.folder` (working file)
- `mps.pprotoolkit` (working file)
- `mpsContainer.ppr` (for debugging only)

`conv/icons`
- `*.icon` (i.e. all component icons)

`conv/pprs`
- `*.ppr` (i.e. all component pprs, including the ones below)
- `mpsSpeak1_3.ppr` (Speak for MPS Toolkit 1.3)
- `mpsSpeak2_0.ppr` (Speak for MPS Toolkit 2.0)
- `mpsMoveConditionData1_3.ppr` (MoveConditionData for MPS Toolkit 1.3, directly specified in `mpsconvkit1.3`)
- `mpsMoveConditionData2_0.ppr` (MoveConditionData for MPS Toolkit 2.0, directly specified in `mpsconvkit2.0`)

Replace Previous MPS Toolkit

Before converting a legacy MPS Toolkit application, replace the previous MPS toolkit with a special conversion toolkit using the **pproinfo -m** [**1.3** (for MPS Toolkit 1.3) | **2.0** (for MPS Toolkit 2.0)] command. **pproinfo -m** copies the appropriate conversion toolkit and system folder files to the proper PeriProducer areas.

Additionally, include `$PPROHOME/conv` in the `$PPROPATH` so that the replacement MPS toolkit can be loaded under PeriProducer 3.00. `$PPROPATH` should not include a path to the old MPS Toolkit.

Load into PeriProducer 3.00

Perform the following to convert a legacy PeriProducer 2.30/MPS Toolkit 1.3/2.00 application by loading the legacy application into PeriProducer 3.00:

1. Identify which version of the MPS Toolkit was used by the PeriPro 2.30 application.

2. Replace the old MPS Toolkit (see Replace Previous MPS Toolkit on page 44).

3. Start PeriProducer with the **-m** option (**peripro -m**). This option causes the conversion of certain conditions (generated by the MPS resource) into their equivalent PeriProducer 3.00 conditions.

4. Load and save the PeriProducer 2.30 application.

Use the PeriProducer Command Line

Use the following command line syntax to convert a legacy application:

peripro -c -m *<application>*

For example, to convert numdemo.ppr, enter **peripro -c -m mpsnumdemo**. You can also use filename wildcards (e.g., **peripro -c mpsnum***) to batch convert applications. The example converts all applications in the current directory whose names start with "mpsnum".

Conversion Logs

The conversion logs are detailed lists of issues encountered during conversion of PeriProducer 2.30 applications to PeriProducer 3.00. The conversion logs are written to the same directory as the source (2.30) application. The logs are named `conversion_`*appname*`.txt` (in plain text format) and `conversion_`*appname*`.html` (in HTML format).

Each log entry indicates an application function which required some translation to become PeriProducer 3.00 compatible. A conversion is divided into three categories:

1. **Information**. This indicates something which should not affect the functionality of the application. A typical information message is the conversion of a 2.30 condition name to a new 3.00 condition name. The application will react the same way in PeriProducer 3.00, just using a different condition name.

2. **Warning**. This indicates something which successfully converted but may not work as expected in 3.00. A typical warning message is for an obsoleted block setting which should not affect application function.

3. **Severe**. This indicates something which requires manual application editing in order for the application to run under PeriProducer 3.00. Severe messages can include

   - using features/resources no longer supported (such as fax composition)
   - obsoleted environment settings, conditions, system datacards which do not have an equivalent in PeriProducer 3.00

Known Conversion Issues

Set Resource Label in 2.30

PeriProducer 2.30 uses a Resource block Set operation to change the external resource (e.g. OSCAR) label. PeriProducer 3.00 converts the Resource (Set) block to an Environment block which sets the RscLabel parameter and enables the Wait option.

If the Resource (set) block had the failure connector enabled, the Environment block has the failure connector enabled and connects to the same failure path as in the 2.30 application.

If the 2.30 application explicitly handled the **setfail** (or **setres**) condition for a Resource set (instead of using the failure connector), you must manually edit the converted application to explicitly handle the **envfail** (or **envcmp**) condition. The **setfail** (or **setres**) condition in 2.30 automatically converts to **ctrlrsrcfail** (or **ctrlrsrccmp**) in 3.00. If the resource label change fails in 3.00, the **envfail** condition occurs, not the **ctrlrsrcfail** condition.

Unsupported 2.30 Resources

Several resources from PeriProducer 2.30 are not supported in PeriProducer 3.00 (see). When an unsupported resource is encountered during conversion, the unsupported resource is reported in the conversion log file. In the application, the resource is changed from the resource name to "Unsupported" in the associated Resource block. Depending on the resource type, you may not be able to use the application with PeriProducer 3.00.

Resource Block Conversions

The following conversions occur automatically for 2.30 applications which use external resources (Resources block):

- Receive Resource with the resource "phonePRS" is converted into a Read Phone block.
- Receive Resource with the resource "lvr-X" (where X is any letter of a valid 2.30 resource) is converted into a Read Phone block.
- Send Resource with the resource "phonePRS" is converted into a Speak block (with the prompt spoken as TTS Literal).
- Send Resource with the resource "extts-X" (where X is any letter of a valid 2.30 resource) is converted into a Speak block (with the prompt spoken as TTS Literal).
- If Wait is enabled or a failure connector is available on a 2.30 Resource block, Wait is enabled on the 3.00 converted block.

Flushing the Speak Prompt Buffer

The PeriProducer 2.30 System block "Start Reprompt List" function flushes the list of previously spoken output. This function is still supported in PeriProducer 2.30. However, it is recommended that to flush the output list buffer, use a Resource block with the Resource field set to "Player", the CID field set to System.DefaultCID, the Operation set to "Control", and the Send From field set to System.Constants.Flush. This suggestion action appears in a 2.30 application's conversion log file.

System Transfer Connection ID Datacard

PeriProducer 3.00 introduces the system datacard "TransferCID". The system returns the Connection ID (CID) of line which is used for a transfer operation back to the TransferCID datacard. TransferCID should **not** be used in PeriProducer 3.00-native applications to specify a CID for a transfer operation.

TransferCID is populated only if the transfer operation is performed synchronously (Originate block with Wait enabled). If Wait is not enabled, TransferCID stays empty.

During 2.30 to 3.00 conversion, PeriProducer may use TransferCID in this capacity (to specify a CID instead of receiving a value from the system). However, this is done only when converting applications.

Automatic Detection of Speech or DTMF Input (AUTOID)

When converting applications that perform AUTOID (automatic detection of speech or DTMF input) it is necessary to make sure that the prompt(s) prior to the environment block(s) that setup autoid (detect timer and enable autoid) have, at minimal, the DTMF and LVR resources on after the prompt completes.