



Pulse Connect Secure Pulse Policy Secure

Custom Sign-In Pages Developer Reference

Product Release 9.1R4

Document Revision 1.0
Published: June 2020

Pulse Secure, LLC
2700 Zanker Road, Suite 200
San Jose, CA 95134
<http://www.pulsesecure.net>

© 2020 by Pulse Secure, LLC. All rights reserved

Pulse Secure and the Pulse Secure logo are trademarks of Pulse Secure, LLC in the United States. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Pulse Secure, LLC assumes no responsibility for any inaccuracies in this document. Pulse Secure, LLC reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Pulse Custom Sign-In Pages Developer Reference

The information in this document is current as of the date on the title page.

END USER LICENSE AGREEMENT

The Pulse Secure product that is the subject of this technical documentation consists of (or is intended for use with) Pulse Secure software. Use of such software is subject to the terms and conditions of the End User License Agreement (“EULA”) posted at <http://www.pulsesecure.net/support/eula>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.”

Revision History

June 2020 – Updated with confirmation-new-ux.html, confirmation_opensessions-new-ux.html, LoginPage-new-ux.html, NextToken-new-ux.html, NewPin-new-ux.html, PleaseWait-new-ux.html, psaldownloadmacapp.html, psaldownloadmacapp-new-ux.html, psaldownloadwindowsapp-new-ux.html, psalinstall-new-ux.html, psallaunch.html, psallaunch-new-ux.html, psalwait-new-ux.html, Remediate-new-ux.html, SigninNotifPostAuth-new-ux.html, SigninNotifPreAuth-new-ux.html, SSL-new-ux.html, TotpAuthRegister-new-ux.html, TotpAuthTokenEntry-new-ux.html

May 2017- Removed UserAdmin-add-user.html, UserAdmin-add-bulk-users.html, UserAdmin-update-user.html, UsersPrintInfo as customization is not supported for GUAM beginning with PPS Release 5.2.

Abbreviated Table of Contents

Table of Contents.....	ii
List of Tables.....	ix
About This Guide	x
CHAPTER 1 Custom sign-in pages.....	12
Index	161

Table of Contents

Table of Contents	ii
List of Tables	ix
About This Guide	x
Objectives	x
Document Conventions	x
Requesting Technical Support	xi
<i>Self-Help Online Tools and Resources</i>	xi
<i>Opening a Case with PSGSC</i>	xi
CHAPTER 1 Custom sign-in pages	12
Downloadable Zip Files	19
Adding Custom Help Files	21
Using the Template Toolkit.....	22
<i>Understanding the Template Toolkit Language</i>	24
<i>Template Comments</i>	25
<i>Accessing and Updating Variables and Files</i>	26
<i>Creating Conditional Statements</i>	29
<i>Creating Looping Constructs</i>	31
<i>Unsupported Directives</i>	31
<i>Getting Familiar with the Template Page Structure</i>	31
Using Templates from the samples.zip File	32
<i>Customizing Look-and-Feel</i>	33
<i>Simplifying the Code in the Sample Templates</i>	34
<i>Using Cascading Style Sheets</i>	38
<i>Working with the Pulse Secure Desktop Client or Standalone Network Connect</i>	45
Customizing Error Handling and Navigation	46
<i>Redirecting Based on any Errors</i>	46
<i>Redirecting Based on Specific Errors</i>	46
<i>Defining Error Message Locations</i>	46
<i>Customizing Error Messages</i>	46
Localizing Custom Sign-In Pages.....	47
<i>Creating a Localized Set of Custom Sign-In Pages</i>	48
Upgrade Considerations.....	48
Custom Templates Reference.....	48
Mobile Device Pages.....	49
iPad Pages.....	50
<failedPolicy>	50
<failedPolicy>.name	50
<failedPolicy>.remediation	50
action	50
activex_ini	51
AgentInstall.html.....	51
AgentlessInstall.html	52
AnonymousAuthentication.....	52
applet_ini.....	53
appstr	53
authenticate()	54

autocomplete.....	56
Cancel.html / Cancel-mobile-webkit.thtml / Cancel-ipad.thtml	56
cboxuser.....	56
ccInAcTimeout.....	57
ccRunning	57
cert_md5	57
CertificateAuthentication.....	58
changePasswordTitle	58
check.....	59
checkActiveX()	59
color	61
confirmation-new-ux.thtml	61
confirmation_opensessions-new-ux.thtml.....	61
Continue.....	62
secid_key	62
Defender.html / Defender-mobile-webkit.thtml / Defender-ipad.thtml.....	62
delivery_mode.....	63
DoUnload().....	63
ds_mobile_common.css	63
ExceededConcurrent.html / ExceededConcurrent-mobile-webkit.thtml / ExceededConcurrent-ipad.thtml	64
expired	64
fieldsArr.....	65
fallback_setup_applet_vitals.....	65
FILTER verbatim	65
FinishLoad()	66
focus	67
FriendlyTime	67
frmCasque.....	67
frmLogin	68
frmNextToken.....	68
frmSelectRoles.....	68
g_time	69
gCancelNewPinMode.....	69
gCancelNextTokenMode.....	69
geckoBrowser	70
GeneratePin.html / GeneratePin-mobile-webkit.thtml / GeneratePin-ipad.thtml.....	70
GetCookieValue().....	71
grab()	72
GraceLoginsRemaining.....	72
GraceLoginUsed.html / GraceLoginUsed-mobile-webkit.thtml / GraceLoginUsed-ipad.thtml.....	73
guest-user-print-page.thtml	73
guest-user-print-page-batch.thtml.....	73
guest-user-email-page.thtml.....	73
HC_REMED_POLICIES_CHECK	73
HC_REMED_SHOW_ADV_PREFS	74
hcInAcTimeout	74
hcRunning.....	75
heading	75
help	75
help_on	76
HideRemed().....	76
Home	77
I18N_UAC_DOWNLOAD_AND_INSTALL_64_BIT	77
ie	77
IEonWindows	78

ie_winxp	78
install.....	78
instanceId.....	79
instructions	79
isLinux	79
isMac.....	79
isSAMEnabled.....	80
isUAC.....	80
keyboard.js.....	80
labelInstructions	81
listFailedPolicies.....	81
loading.....	82
locale.....	82
loginmode.....	82
Login().....	83
LoginMeeting.shtml	83
LoginPage.shtml / LoginPage-mobile-webkit.shtml / LoginPage-ipad.shtml.....	84
LoginPageErrorArgs.....	85
LoginPageErrorCode.....	85
LoginPageErrorMessage.....	88
LoginPage-new-ux.shtml.....	89
login_required	89
Logout.shtml / Logout-mobile-webkit.shtml / Logout-ipad.shtml	90
meeting_ini.....	91
MeetingAppletInstaller.shtml	91
MeetingRun.shtml	92
MeetingRunJava.shtml.....	93
meetings.....	93
MeetingSelect.shtml.....	93
MeetingTestJava.shtml	94
MeetingTestJS.shtml.....	94
MeetingTestMSJava.shtml	95
MeetingTestResult.shtml.....	95
MeetingTrouble.shtml.....	95
message.....	96
mid	96
mid_param	97
midStr.....	97
MinimumPasswordLength	98
msg	98
nc_logging.....	98
ncp_read_timeout.....	99
netacekey.....	99
netesecid_key	99
netesecid_loginmode	100
netesecid_pinerr.....	100
netesecid_realm.....	100
netesecid_username	101
netesecidactionCancel	101
netesecidactionEnter.....	102
Netscape4xx	102
NextToken.shtml / NextToken-mobile-webkit.shtml / NextToken-ipad.shtml	102
NextToken-new-ux.shtml.....	102
NewPin.shtml / NewPin-mobile-webkit.shtml / NewPin-ipad.shtml.....	103
NewPin-new-ux.shtml	103

occurrence	104
onsubmit.....	104
p	104
PageErrorArgs	104
PageErrorCode	105
params	106
password.....	106
password.....	107
password.....	107
password2.....	108
PasswordChange.html / PasswordChange-mobile-webkit.html / PasswordChange-ipad.html	108
PasswordComplexityMessage	109
PasswordExpiration.html / PasswordExpiration-mobile-webkit.html / PasswordExpiration-ipad.html	109
PasswordExpirationFriendlyTime	109
PasswordExpirationSeconds.....	110
PasswordHistoryLength.....	110
pleasewait	111
PleaseWait-mtg.html	111
PleaseWait.html / PleaseWait-mobile-webkit.html / PleaseWait-ipad.html.....	111
pleasewait-ipad.html.....	112
pleasewaitLogoutJSCode.....	112
PleaseWait-new-ux.html.....	113
PleaseWaitObjectCC.....	114
pleaseWaitObjectEP	114
PleaseWaitObjectHC.....	114
pleasewaitObjectJAM.....	115
pleasewaitTO_msec.....	115
pleasewaitWin32	116
plugintype.....	116
portal	117
postAuthSNDecline	117
postAuthSNProceed.....	117
postAuthSNTText	117
postAuthSNTitle	117
preAuthSNDecline.....	118
preAuthSNProceed	118
preAuthSNTText.....	118
preAuthSNTitle	118
prompts	119
psaldownloadmacapp.html	119
psaldownloadmacapp-new-ux.html	120
psaldownloadwindowsapp-new-ux.html.....	120
psalinstall-new-ux.html	120
psallaunch.html	121
psallaunch-new-ux.html.....	121
psalwait-new-ux.html	121
realm	122
realm	123
RealmList	123
recallLastRealmUsed().....	123
redirect().....	124
Remediate.html / Remediate-mobile-webkit.html / Remediate-ipad.html	125
Remediate-new-ux.html.....	125
remedy1	126
remedy2	126

RoleList	127
rows	127
runOnLoad	128
safari	128
secid_challenge.....	128
secidcontextid	128
secid_contextid.....	129
secid_loginmode	129
secid_pinerr.....	129
secid_pinformat.....	130
secid_pinselectmode.....	130
secid_systempin.....	131
secid_username.....	131
secidactionSavePin	131
secidpinformat.....	132
secidpinselectmode.....	132
SecondaryLoginPage.html / SecondaryLoginPage-mobile-webkit.html / SecondaryLoginPage-ipad.html.....	132
SelectRole.html / SelectRole-mobile-webkit.html / SelectRole-ipad.html	133
serverdetails	133
setFailed()	133
setFinished.....	134
SetLastRealm()	134
setStarted()	135
setSucceeded()	135
setthankyou()	135
setup_classid	136
setup_codebase.....	136
setup_ini.....	136
Setupappversion	137
setup_codebase.....	137
showButtons.....	138
showChangePasswordTitle	138
showClose.....	139
showContinue	139
showHeading	140
showLoading	140
showReinstall	140
showRemedOption.....	141
showPolicies	141
ShowSystemPin.html / ShowSystemPin-mobile-webkit.html / ShowSystemPin-ipad.html	141
showTryAgain	142
signinAgain.....	142
signinAgain.....	142
signinAgainUrl.....	143
SigninNotifPostAuth.html / SigninNotifPreAuth-mobile-webkit.html / SigninNotifPostAuth-ipad.html	143
SigninNotifPostAuth-new-ux.html	143
SigninNotifPreAuth.html / SigninNotifPreAuth-ipad.html	143
SigninNotifPreAuth-new-ux.html.....	144
SM-NewPinSelect.html / SM-NewPinSelect-mobile-webkit.html / SM-NewPinSelect-ipad.html.....	144
SM-NewPinSystem.html / SM-NewPinSystem-mobile-webkit.html / SM-NewPinSystem-ipad.html.....	144
SM-NewUserPin.html	144
SM-NextToken.html / SM-NextToken-mobile-webkit.html / SM-NextToken-ipad.html.....	145
softid_error	145
softid_time.....	146
SSL.html / SSL-mobile-webkit.html / SSL-ipad.html	146

SSL-new-ux.thtml	146
Start_onclick()	146
Start_onclick_java()	147
start_status	147
StartCC()	148
StartHC()	148
started	148
startPageLink	149
startSessionReDir	149
stopComponents()	149
SubmitClicked()	150
submitFrmCasque()	151
testjava_ini	151
textClose	151
textContinue	152
textRemedOption	152
textTryAgain	153
thankyou()	153
TotpAuthRegister-new-ux.thtml	154
TotpAuthTokenEntry-new-ux.thtml	154
troubleshooting	154
TryAgain	155
type	155
tz_offset	156
uninstall_action	156
Uninstall_applet()	156
Uninstall_onclick()	157
uninstallReDir	157
UninstallCC	158
upAndRunning()	158
userAgentTypeAndroid	158
username	159
warnings	159
welcome	159
win32	160
Index	161
A	162
B	162
C	162
D	162
E	162
F	162
G	162
H	162
I	162
J	162
K	162
L	162
M	162
N	163
P	163
R	163
S	163

T..... 163
U..... 163
W..... 163

List of Tables

Table1: Notice IconsX
Table2: Required Templates in Each Zip 20
Table3: English Values of Common Variables 26
Table4: Conditional Operators 30
Table5: PIN usage indicators 71
Table6: Login Error Messages, Codes, and Related Notes..... 86
Table7: Additional Login Error Messages and Codes..... 90
Table8: PIN Error Messages 103
Table9: Additional PIN Error Messages..... 103
Table10: Page Error Codes..... 105
Table11: New PIN Assignment Messages 144

About This Guide

- Objectives
- Document Conventions
- Requesting Technical Support







Objectives

This guide provides detailed developer reference information on custom sign-in pages.

Document Conventions

Table 1 defines notice icons used in this guide.

Table1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Requesting Technical Support

Technical product support is available through the Pulse Secure Global Support Center (PSGSC). If you are a customer with an active support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with PSGSC.

- Product warranties—For product warranty information, visit <http://www.pulsesecure.net/support>

Self-Help Online Tools and Resources

For quick and easy problem resolution, Pulse Secure has designed an online self-service portal called the Pulse Secure Global Support Center (PSGSC) that provides you with the following features:

- Find CSC offerings: <http://www.pulsesecure.net/support>
- Search for known bugs: <http://www.pulsesecure.net/support>
- Find product documentation: <http://www.pulsesecure.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://www.pulsesecure.net/support>
- Download the latest versions of software and review release notes: <http://www.pulsesecure.net/support>
- Search technical bulletins for relevant hardware and software notifications: <http://www.pulsesecure.net/support>
- Join and participate in the Community Forum: <http://www.pulsesecure.net/support>
- Open a case online in the CSC Case Management tool: <http://www.pulsesecure.net/support>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <http://www.pulsesecure.net/support>

Opening a Case with PSGSC

You can open a case with PSGSC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.pulsesecure.net/support>.
- Call 1-844 751 7629 (Toll Free, US).

For international or direct-dial options in countries without toll-free numbers, see <http://www.pulsesecure.net/support>.

CHAPTER 1 Custom sign-in pages

The customizable sign-in pages feature enables you to personalize the look-and-feel of the pre- and post-authentication, password management, and Pulse Secure Collaboration pages that Pulse Connect Secure presents to administrators and end-users. The tasks you can accomplish include:

- Customizing the presentation style using CSS or table-based design
- Customizing error message text
- Redirecting traffic based on error codes
- Localizing custom sign-in pages in supported languages
- Adding custom HTML help

The system ships with several complete sets of sample custom sign-in pages that you can modify to suit your look-and-feel and behavior. You can download these files from Connect Secure. This document describes the files, their uses, and how to modify them. Connect Secure contains these files in .zip file format. You can download the zip files from the **Authentication > Signing In > Sign-in pages > Upload custom sign-in pages** page.

You can achieve a good understanding of the process of modifying and uploading your custom sign-in pages by reading through this document completely before attempting to perform the customization. When you feel you have a comprehensive understanding of what is involved, you should consider downloading the **samples.zip** file and modifying one or two pages first, just to test your understanding. There is a strong likelihood that you will want to apply a similar look and feel to all of the pages, which you can accomplish fairly simply, if you follow the directions provided in this document. Other tasks, such as redirecting traffic based on errors may require more skill and time.

Custom sign-in pages can potentially require a large amount of memory and disk space. To provide custom sign-in pages, we recommend you customize the sample custom sign-in pages provided.

You can find the following topics in this document:

- [Downloadable Zip Files](#)
- [Adding Custom Help Files](#)
- [Using the Template Toolkit](#)
- [Using Templates from the samples.zip File](#)
- [Customizing Error Handling and Navigation](#)
- [Localizing Custom Sign-In Pages](#)
- [Upgrade Considerations](#)
- [Custom Templates Reference](#)
- [Mobile Device Pages](#)
- [iPad Pages](#)
- [<failedPolicy>](#)

-
- <failedPolicy>.name
 - <failedPolicy>.remediation
 - action
 - activex_ini
 - AgentInstall.html
 - AgentlessInstall.html
 - AnonymousAuthentication
 - applet_ini
 - appstr
 - authenticate()
 - autocomplete
 - Cancel.html / Cancel-mobile-webkit.html / Cancel-ipad.html
 - cboxuser
 - cclnAcTimeout
 - ccRunning
 - cert_md5
 - CertificateAuthentication
 - changePasswordTitle
 - check
 - checkActiveX()
 - color
 - Continue
 - secid_key
 - Defender.html / Defender-mobile-webkit.html / Defender-ipad.html
 - delivery_mode
 - DoUnload()
 - ds_mobile_common.css
 - ExceededConcurrent.html / ExceededConcurrent-mobile-webkit.html / ExceededConcurrent-ipad.html
 - expired
 - fieldsArr
 - fallback_setup_applet_vitals
 - FILTER verbatim
 - FinishLoad()
 - focus
 - FriendlyTime
 - frmCasque
 - frmLogin

- frmNextToken
- frmSelectRoles
- g_time
- gCancelNewPinMode
- gCancelNextTokenMode
- geckoBrowser
- GeneratePin.html / GeneratePin-mobile-webkit.html / GeneratePin-ipad.html
- GetCookieValue()
- grab()
- GraceLoginsRemaining
- GraceLoginUsed.html / GraceLoginUsed-mobile-webkit.html / GraceLoginUsed-ipad.html
- guest-user-print-page.html
- guest-user-print-page-batch.html
- guest-user-email-page.html
- HC_REMED_POLICIES_CHECK
- HC_REMED_SHOW_ADV_PREFS
- hcInAcTimeout
- hcRunning
- heading
- help
- help_on
- HideRemed()
- Home
- I18N_UAC_DOWNLOAD_AND_INSTALL_64_BIT
- ie
- IEonWindows
- ie_winxp
- install
- instanceId
- instructions
- isLinux
- isMac
- isSAMEnabled
- isUAC
- keyboard.js
- labelInstructions
- listFailedPolicies

-
- loading
 - locale
 - loginmode
 - Login()
 - LoginMeeting.shtml
 - LoginPage.shtml / LoginPage-mobile-webkit.shtml / LoginPage-ipad.shtml
 - LoginPageErrorArgs
 - LoginPageErrorCode
 - LoginPageErrorMessage
 - login_required
 - Logout.shtml / Logout-mobile-webkit.shtml / Logout-ipad.shtml
 - meeting_ini
 - MeetingAppletInstaller.shtml
 - MeetingRun.shtml
 - MeetingRunJava.shtml
 - meetings
 - MeetingSelect.shtml
 - MeetingTestJava.shtml
 - MeetingTestJS.shtml
 - MeetingTestMSJava.shtml
 - MeetingTestResult.shtml
 - MeetingTrouble.shtml
 - message
 - mid
 - mid_param
 - midStr
 - MinimumPasswordLength
 - msg
 - nc_logging
 - ncp_read_timeout
 - netacekey
 - netesecid_key
 - netesecid_loginmode
 - netesecid_pinerr
 - netesecid_realm
 - netesecid_username
 - netesecidactionCancel

- netesecidactionEnter
- Netscape4xx
- NewPin.html / NewPin-mobile-webkit.html / NewPin-ipad.html
- NextToken.html / NextToken-mobile-webkit.html / NextToken-ipad.html
- occurrence
- onsubmit
- p
- PageErrorArgs
- PageErrorCode
- params
- password
- password
- password
- password2
- PasswordChange.html / PasswordChange-mobile-webkit.html / PasswordChange-ipad.html
- PasswordComplexityMessage
- PasswordExpiration.html / PasswordExpiration-mobile-webkit.html / PasswordExpiration-ipad.html
- PasswordExpirationFriendlyTime
- PasswordExpirationSeconds
- PasswordHistoryLength
- pleasewait
- PleaseWait-mtg.html
- PleaseWait.html / PleaseWait-mobile-webkit.html / PleaseWait-ipad.html
- pleasewait-ipad.html
- pleasewaitLogoutJSCode
- PleaseWaitObjectCC
- pleaseWaitObjectEP
- PleaseWaitObjectHC
- pleasewaitObjectJAM
- pleasewaitTO_msec
- pleasewaitWin32
- plugintype
- portal
- postAuthSNDecline
- postAuthSNProceed
- postAuthSNText
- postAuthSNTitle

-
- [preAuthSNDecline](#)
 - [preAuthSNProceed](#)
 - [preAuthSNText](#)
 - [preAuthSNTitle](#)
 - [prompts](#)
 - [realm](#)
 - [realm](#)
 - [RealmList](#)
 - [recallLastRealmUsed\(\)](#)
 - [redirect\(\)](#)
 - [Remediate.html / Remediate-mobile-webkit.html / Remediate-ipad.html](#)
 - [remedy1](#)
 - [remedy2](#)
 - [RoleList](#)
 - [rows](#)
 - [runOnLoad](#)
 - [safari](#)
 - [secid_challenge](#)
 - [secidcontextid](#)
 - [secid_contextid](#)
 - [secid_loginmode](#)
 - [secid_pinerr](#)
 - [secid_pinformat](#)
 - [secid_pinselectmode](#)
 - [secid_systempin](#)
 - [secid_username](#)
 - [secidactionSavePin](#)
 - [secidpinformat](#)
 - [secidpinselectmode](#)
 - [SecondaryLoginPage.html / SecondaryLoginPage-mobile-webkit.html / SecondaryLoginPage-ipad.html](#)
 - [SelectRole.html / SelectRole-mobile-webkit.html / SelectRole-ipad.html](#)
 - [serverdetails](#)
 - [setFailed\(\)](#)
 - [setFinished](#)
 - [SetLastRealm\(\)](#)
 - [setStarted\(\)](#)
 - [setSucceeded\(\)](#)

- setthankyou()
- setup_classid
- setup_codebase
- setup_ini
- Setupappversion
- setup_codebase
- showButtons
- showChangePasswordTitle
- showClose
- showContinue
- showHeading
- showLoading
- showReinstall
- showRemedOption
- showPolicies
- ShowSystemPin.shtml / ShowSystemPin-mobile-webkit.shtml / ShowSystemPin-ipad.shtml
- showTryAgain
- signinAgain
- signinAgain
- signinAgainUrl
- SigninNotifPostAuth.shtml / SigninNotifPreAuth-mobile-webkit.shtml / SigninNotifPostAuth-ipad.shtml
- SigninNotifPreAuth.shtml / SigninNotifPreAuth-ipad.shtml
- SM-NewPinSelect.shtml / SM-NewPinSelect-mobile-webkit.shtml / SM-NewPinSelect-ipad.shtml
- SM-NewPinSystem.shtml / SM-NewPinSystem-mobile-webkit.shtml / SM-NewPinSystem-ipad.shtml
- SM-NewUserPin.shtml
- SM-NextToken.shtml / SM-NextToken-mobile-webkit.shtml / SM-NextToken-ipad.shtml
- softid_error
- softid_time
- SSL.shtml / SSL-mobile-webkit.shtml / SSL-ipad.shtml
- Start_onclick()
- Start_onclick_java()
- start_status
- StartCC()
- StartHC()
- started
- startPageLink
- startSessionReDir

-
- stopComponents()
 - SubmitClicked()
 - submitFrmCasque()
 - testjava_ini
 - textClose
 - textContinue
 - textRemedOption
 - textTryAgain
 - thankyou()
 - troubleshooting
 - TryAgain
 - type
 - tz_offset
 - uninstall_action
 - Uninstall_applet()
 - Uninstall_onclick()
 - uninstallReDir
 - UninstallCC
 - upAndRunning()
 - UserAdmin-add-user.shtml
 - UserAdmin-add-bulk-users.shtml
 - UserAdmin-update-user.shtml
 - UsersPrintInfo
 - userAgentTypeAndroid
 - username
 - warnings
 - welcome
 - win32

Downloadable Zip Files

Although each zip file contains all the templates, each zip file is for a particular set of features:

- **Sample.zip**—This zip file is for standard PPS/PCS pages, including standard pre- and post-authentication pages, ACE pre-authentication pages, ACE pre-authentication pages for use with eTrust SiteMinder, and password management pages.
- **SoftID.zip**—This zip file is for use with the RSA Soft ID client.
- **Kiosk.zip**—This zip file is for use by kiosk users or for any system in which you want to lock out keyboard-based login.

- **Meeting.zip**—This zip file is for pre-authentication and post-authentication pages for use by Pulse Secure Collaboration attendees.



NOTE: When applicable, the zip files contain templates for mobile devices, including templates for mobile webkit and specific mobile devices, such as Windows Mobile and Apple iPad. These templates are identical to the other templates contained in the zip file, except that they are customized for a smaller viewing area. You are not required to add any of them to the zip file. The device templates use the same names as their full screen counterparts, except that **mobile-webkit**, **ppc**, and **ipad** are added to their filenames. For example, the full screen sign-in page template is named **LoginPage.thtml** and its Mobile counterpart is named **LoginPage-ppc.thtml**.

Certain templates are required when uploading the zip files. If you do not include these required templates, a warning message appears. “[Custom sign-in pages](#)” lists the required templates for each zip file.

Table2: Required Templates in Each Zip

	Samples.zip	Kiosk.zip	SoftID.zip (ACE)	Meeting.zip
Cancel.thtml	R	R	R	
confirmation.thtml				
confirmation_opensessions.thtml	R			
Defender.thtml	R			
ds.css				R
ExceededConcurrent.thtml	R	R	R	
GeneratePin.thtml	R	R	R	
GraceLoginUsed.thtml	R			
keyboard.js		R		
LoginMeeting.thtml				R
LoginPage.thtml	R	R	R	
Logout.thtml	R	R	R	
MeetingAppletInstaller.thtml				R
MeetingRun.thtml				R
MeetingRunJava.thtml				R
MeetingSelect.thtml				R
MeetingTestJava.thtml				R
MeetingTestJS.thtml				R
MeetingTestMSJava.thtml				R
MeetingTestResult.thtml				R
MeetingTrouble.thtml				R

NewPin.thtml	R	R	R
NextToken.thtml	R	R	R
PasswordChange.thtml	R		
PasswordExpiration.thtml	R		
PleaseWait.thtml	R	R	R
PleaseWait-mtg.thtml			R
Remediate.thtml	R	R	
SecondaryLoginPage.thtml	R		
SelectRole.thtml	R		R
ShowSystemPin.thtml	R	R	
SigninNotifPostAuth.thtml			
SigninNotifPreAuth.thtml			
SM-NewPinSelect.thtml	R		
SM-NewPinSystem.thtml	R		
SM-NewUserPin.thtml	R		
SM-NextToken.thtml	R		
SSL.thtml	R	R	R
UserAdmin-add-user.thtml	R		
UserAdmin-add-bulk-users.thtml	R		
UserAdmin-update-user.thtml	R		

To customize the pages, you must use the Template Toolkit language. For more information, see [“Understanding the Template Toolkit Language”](#).

For information on how to internationalize custom sign-in pages, see [“Localizing Custom Sign-In Pages”](#).

Adding Custom Help Files

You can add custom help files by creating your own HTML help files and linking them to the custom sign-in pages. You can easily add links to the basic HTML in the custom sign-in pages.

If you add customized online Help to your custom sign-in pages, you must set the destination URL to point to a **.html** file, not a **.thtml** file. For example, in the following input statement, the help file, **help.html**, should not have a **.thtml** extension:

```
<input type='submit' name='help' value="Help"
onclick='window.open("<% Home%>/imgs/help.html",
"wndHelp","height=400,width=500,resizeable=yes,scrollbars=yes");
return false;'>
```

If you use the **.thtml** extension, the **Help** link on the custom sign-in page returns an error.

Using the Template Toolkit

The Template Toolkit is an open source software package written in Perl, that enables you to build and replicate Web pages using a special-purpose language. The Template Toolkit language is simple and easy to learn.

The system software recognizes the Template Toolkit directives and is pre-configured to process templates. The PPS/PCS processes templates into **.html** files, which it then displays as administrator Web console pages or as end-user Web pages. You cannot change all of the generated Web pages, but you can customize the ones noted in this topic. You can also modify a variety of presentation items on the **UI Options** page in the administrative Web console.



NOTE: The **.html** files and many **.css** and **.js** files in the **.html** files include `<%DANA_VERSION_HASH%>` in the filename. This placeholder is replaced by a hash value during runtime to verify the files have not been modified.

The Template Toolkit processing utilities transform template source files into complete HTML files. The source files are a combination of HTML and Template Toolkit directives. The source files carry a user-defined extension, but Connect Secure recognizes the file extension of **.html**. Processed files typically carry a **.html** file extension.



NOTE: When creating customized sign-in pages, save them as UTF-**i**.

Testing Your Custom Pages

There are two ways to test your custom pages:

- Using the Template Toolkit utilities you have installed locally.
- Uploading customized files to Connect Secure.

If you are customizing just one or two of the sample template files, you might find it easiest to upload the files to PPS/PCS. If you do not have access to a test system, or if you are customizing many of the sample template files or making significant changes to those files, you may want to install and configure the Template Toolkit on your local machine.

Setting up the Template Toolkit Locally

You can install the Template Toolkit on your local system. Because it is a Perl application, you need a Perl distribution on your system before you install the Template Toolkit. If you run a Linux or Mac OS X system, you probably already have a Perl distribution on your system. If you run a Windows system, you may need to install the ActiveState ActivePerl distribution, which you can find at <http://www.activestate.com/>.

If you prefer, you can install Cygwin, which allows you to run UNIX programs in a virtual environment on your Windows system. You can download Cygwin from <http://sourceware.org/cygwin/>.

It is likely that you will need to download the latest version of the Perl AppConfig module, which allows your Perl applications to read and process configuration files. You can find the latest version of AppConfig at <http://www.cpan.org/modules/by-module/AppConfig/>.

You can find a **gzip-tar** archive of the latest version of the Template Toolkit at <http://www.cpan.org/modules/by-module/Template/>.

Follow these steps:

1. Download, install, and configure Perl, ActivePerl, or Cygwin, depending on the type of operating system you are running, and your personal requirements.
2. Download, install, and configure AppConfig.

3. Download, install, and configure the Template Toolkit.



NOTE: The Template Toolkit distribution includes a number of useful documents. You should print the INSTALL document and read it closely. You must perform a number of configuration steps as detailed in the INSTALL document. If you try to perform the steps as described in INSTALL, but have not performed the requisite configuration, the installation fails. Fortunately, you can go back and perform the configuration steps you missed or performed incorrectly, and try to install again, until the installation succeeds.

After you have successfully installed and configured the Template Toolkit, you can generate HTML files from the sample templates that are available on PPS/PCS. PPS/PCS is pre-configured to process Template Toolkit files and many of the components are not available on your local system. Therefore, you are unable to run the Template Toolkit in the same way or to obtain the same exact results. However, you can simulate the processing closely enough to test the presentation. You will encounter the following limitations:

- You can only test the presentation of the sample templates on your local system. You cannot test the actual pre-authentication logic unless you upload the files to a PPS/PCS system.
- You must do one of the following:
 - Use the Template Toolkit [% TAGS %] directive to change any new directives so that they use the angle bracket tokens (<% %>) used in the Pulse Secure sample templates.
 - Use the standard Template Toolkit tokens ([% %]) and when satisfied with the presentation, modify them to angle bracket tokens (<% %>) before uploading.
 - The Template Toolkit provides two command-line utilities for processing template files: **tpage**, which enables you to process a single template at a time, and **tree**, which enables you to process an entire directory of template files. Although you can use the **tree** utility, The system uses a proprietary process that simulates the **tpage** utility, and so, you may find your final results more effective by using **tpage** only.

Downloading Sample Templates and Uploading Files

Your other option for testing customized template files is to upload the files to the system. Although this may be a bit tedious, if you are only customizing a handful of pages, you might find it easier than setting up the Template Toolkit environment on your local system.



NOTE: The total combined size of all uploaded customizable zip files cannot exceed 12MB.

To make sure you can view the customized pages, you must perform these procedures.

To download the sample templates:

1. Select **Authentication > Signing In > Sign-in Pages**.
2. Click **Upload Custom Pages**.
3. Download one or more of the sample template files.
4. Unzip the files on your local system.

To customize and upload the new templates:

1. Customize the files, as you like.

2. Add the files to a new zip file.



NOTE: When creating your zip file, zip only the template files. Do not include the folder that contains the templates in the zip file.

3. On the **Upload Custom Sign-In Pages** page, enter a name for the template files. This is a label that is used to identify the pages when you associate them with a specific URL.
4. Select the page type.
5. Click **Browse** to locate the new zip file containing your customized files.
6. Click **Upload Custom Pages**.

If you want your end-users to be able to see the new templates, you need to associate the templates, using the label you assigned to them, with a URL that you specify for a given role.

To associate the templates with a role-specific URL:

1. Select **Authentication > Signing In > Sign-in Policies**.
2. Click **New URL**.
3. Configure the new policy and URL, using the sign-in policies instructions in the *Pulse Policy Secure Administration Guide*. Select the label you assigned to the templates from the Sign-in page drop down menu.
4. Click **Save Changes**.

To test your customized pages iteratively, you need to cycle through the preceding procedures to upload and associate a URL. You only need to download the templates once.

Understanding the Template Toolkit Language

This section includes an abbreviated description of the Template Toolkit language for PPS/PCS users. The section describes the most common directives and operators that designers may want to use when creating a customized page. Although the Template Toolkit supports many powerful constructs, such as simple and complex datatypes, virtual methods, and interfaces to database systems, the custom sign-in pages use only a limited set of the Toolkit capabilities.

You can use the Template Toolkit for many applications that are far beyond the scope of this discussion. For complete information about the Template Toolkit, go to <http://www.template-toolkit.org>.

A Web page created with the Template Toolkit resembles a standard Web page. It can include HTML, XML, and JavaScript. Unlike a standard Web page, however, it can also include Template Toolkit directives, which you can use to add dynamic behavior to your pages.

A *directive* is a simple instruction that tells the template processor to perform an action and substitute the result into the document in place of the original directive. You can use directives for many purposes, such as iterating through a list of values (**FOREACH**), creating conditional statements (**IF/UNLESS/ELSE**), or including and processing another template file (**PROCESS/INCLUDE**).

When using directives in your code, note that:

- Directives are case-sensitive and all UPPER CASE.
- You must insert directives within the markup tags '<%>' and '%>'.



NOTE: The Template Toolkit documentation includes examples of using square bracket tags (`[% %]`) to mark directives. PPS/PCS only supports angle bracket markup tags (`<% %>`).

- You can embed directives anywhere in a line of text.
- You can split directives across several lines.
- You can use the `#` character to indicate comments within a directive. The Template Toolkit language ignores characters after the `#` character. For more information on different types of comments, see “[Template Comments](#)”.
- The Template Toolkit language generally ignores insignificant white space within the directive.

In addition to directives, you can also use the Template Toolkit language to include loops, conditionals, variable substitutions, and other template files in your page.

The following sections describe common Template Toolkit tasks, directives, and operations you may want to use in your customized pages:

- “[Accessing and Updating Variables and Files](#)”
 - “[GET Directive](#)”
 - “[SET Directive](#)”
- “[Creating Conditional Statements](#)”
 - “[Conditional Operators](#)”
 - “[IF and ELSIF Directives](#)”
 - “[SWITCH and CASE Directives](#)”
- “[Creating Looping Constructs](#)”
- “[Unsupported Directives](#)”

For information about additional directives that you may use within the Template, Toolkit language, such as **PROCESS**, **INCLUDE**, and **INSERT** see the Template Toolkit documentation, available at <http://www.template-toolkit.org>.

Template Comments

You can include template comments by adding a `#` sign to the directive tag, as follows:

```
<%# This is a comment
    that comments out all
    of the lines that follow.
%>
```

You can include a single-line comment by leaving a space between the tag and the `#` sign, as follows:

```
<% # This is the first line comment, but the next line is not commented.
    IF foo...
    # This is another single-line comment.
%>
```

Accessing and Updating Variables and Files

GET Directive

The **GET** directive retrieves and outputs the value of the named variable.

```
<% GET foo %>
```

The **GET** keyword is optional. A variable can be specified in a directive tag by itself.

```
<% foo %>
```

SET Directive

The **SET** directive allows you to assign a value to existing variables or create new temporary variables.

```
<% SET title = 'Hello World' %>
```

The **SET** keyword is optional.

```
<% title = 'Hello World' %>
```

getText Directive

Used for multi-lingual support, the **getText** directive retrieves and outputs the localized value of the named variable.

```
<% getText foo %>
```

Table 3 lists some of common variables used in the templates and their English values.

Table3: English Values of Common Variables

Variable	English Definition
I18N_ACE_CONFIRM_PIN_COLON	Confirm PIN:
I18N_ACE_GEN_NEW_PIN	Generate New PIN?
I18N_ACE_GEN_PIN	Generate PIN
I18N_ACE_GEN_PIN_MESSAGE1	The system will now generate a PIN for you. Make sure that no one else can see your screen and then click Generate PIN to continue.
I18N_ACE_INVALID_PIN_FORMAT	Invalid PIN Format
I18N_ACE_INVALID_PIN_LENGTH	Invalid PIN Length

I18N_ACE_NEW_PIN_COLON	New PIN:
I18N_ACE_NEW_PIN_GENERATED	New PIN Generated
I18N_ACE_NEW_PIN_GENERATED_MESSAGE1	Your new PIN is
I18N_ACE_NEW_PIN_GENERATED_MESSAGE2	Be sure to remember it, because you need your PIN each time you sign in. When you have memorized it, click Continue to return to the sign-in page.
I18N_ACE_NEW_PIN_MESSAGE0	You must create a new Personal Identification Number (PIN) before you can sign in.
I18N_ACE_NEW_PIN_MESSAGE1	Your PIN should be
I18N_ACE_NEW_PIN_MESSAGE2	long.
I18N_ACE_NEW_PIN_MESSAGE3	Be sure to remember your PIN, because you need it to sign in.
I18N_ACE_NEW_PIN_MESSAGE4	If you prefer, the system can
I18N_ACE_NEW_PIN_MESSAGE5	generate a PIN
I18N_ACE_NEW_PIN_MESSAGE6	for you. Generated PINs are typically more secure.
I18N_ACE_NEW_PIN_MESSAGE7	If you decide not to create a new PIN now, click Cancel.
I18N_ACE_NEW_PIN_REQUIRED	New PIN Required
I18N_ACE_NEW_PIN_SIZE_MSG	Your PIN should be %1 to %2 digits long.
I18N_ACE_NEXT_TOKEN_MESSAGE1	Please enter an additional token code to continue.
I18N_ACE_NEXT_TOKEN_MESSAGE2	The server requires that you enter an additional token code to verify that your credentials are valid. To continue, wait for the token code to change and then enter the new code in the SecurID Token Code field.
I18N_ACE_NEXT_TOKEN_REQUIRED	Token Resync Required
I18N_ACE_PIN_MISMATCH	The Two PINs Entered Do Not Match
I18N_ACE_SAVE_PIN	Save PIN
I18N_ACE_TOKEN_CODE_COLON	SecurID Token Code:
I18N_CANCEL	Cancel
I18N_CLICK	Click
I18N_ENTER	Enter
I18N_HERE	Here
I18N_NETEACE_NEW_PIN_MESSAGE1	You must create a new Personal Identification Number (PIN) before you can sign in.

I18N_NETEACE_NEW_PIN_MESSA GE2	Wait for the token code to change, and then enter your current PIN* followed by the new token code in the SecurID Pass Code field. (If you want the system to generate a new PIN for you, select the System PIN checkbox.) Click Enter to advance to the new PIN assignment page. *If your administrator tells you that you no longer have a PIN, enter only the new token code in the Pass Code field.
I18N_NETEACE_NEW_SYSTEM_PI N	New System PIN Assignment
I18N_NETEACE_NEW_SYSTEM_PI N_MESSAGE1	The system will now generate a Personal Identification Number (PIN) for you.
I18N_NETEACE_NEW_SYSTEM_PI N_MESSAGE2	Wait for the token code to change, and then enter your current PIN* followed by the new token code in the SecurID Pass Code field. Make sure that no one else can see your screen and then click Generate PIN to continue. *If your administrator tells you that you no longer have a PIN, enter only the new token code in the Pass Code field.
I18N_NETEACE_NEW_USER_PIN	New PIN Assignment
I18N_NETEACE_NEW_USER_PIN_ MESSAGE1	Create your new Personal Identification Number (PIN)
I18N_NETEACE_NEW_USER_PIN_ MESSAGE2	Wait for the token code to change, and then enter your current PIN* followed by the new token code in the SecurID Pass Code field. Then, enter your new PIN. *If your administrator tells you that you no longer have a PIN, enter only the new token code in the Pass Code field.
I18N_NETEACE_NEXT_TOKEN_ME SSAGE1	Please enter additional token codes to continue.
I18N_NETEACE_NEXT_TOKEN_ME SSAGE2	The server requires that you enter additional token codes to verify that your credentials are valid. To continue, wait for the token code to change, and then enter your PIN followed by the new token code in the SecurID Pass Code field. Then, wait for the token code to change again and enter only the token code in the SecurID Token Code field.
I18N_NETEACE_NEXT_TOKEN_RE QUIRED	Token Resync Required
I18N_NETEACE_PASS_CODE_COL ON	SecurID Pass Code:
I18N_NETEACE_SYSTEM_PIN_REQ UEST	System PIN
I18N_NETEACE_TOKEN_CODE_CO LON	SecurID Token Code:
I18N_NEW_PIN_INVALID	New PIN entered is invalid. Please wait for the token to change and try login again.
I18N_TO_CONTINUE	to continue
I18N_USERNAME_COLON	Username:

I18N_AUTH_SERVER_COLON	Name of local authentication server.
I18N_FULL_NAME_COLON	Full name
I18N_PASSWORD_COLON	Password
I18N_USER_ADMIN_EMAIL_COLON	email ID
I18N_USER_EXPIRATION_START_COLON	Account start time
I18N_USER_EXPIRATION_END_COLON	Account end time
I18N_USER_ADMIN_TIME_ZONE_COLON	Time zone

If you want to customize a page to add your own message, edit the appropriate template file and enter your custom message using HTML tags. Make sure you do not change any of the surrounding logic. You cannot edit the strings file(s) and change the value of the **I18N_*** variables. For example, suppose the template file has the following code:

```
<TABLE border="0" cellspacing="0" cellpadding="0">
  <TR>
    <TD><IMG border='0' src='/dana-na/imgs/logininfo.gif'
alt='Info' width='21' height='20'></TD>
    <TD><B><% GetText('I18N_ACE_NEW_PIN_REQUIRED') %></B></TD>
  </TR>
</TABLE>
```

To change the displayed message to “You must enter a new PIN number” instead of the default message “ New PIN Required”, change the code as follows:

```
<TABLE border="0" cellspacing="0" cellpadding="0">
  <TR>
    <TD><IMG border='0' src='/dana-na/imgs/logininfo.gif'
alt='Info' width='21' height='20'></TD>
    <!--<TD><B><% GetText('I18N_ACE_NEW_PIN_GENERATED')
%></B></TD>-->
    <TD><B>You must enter a new PIN number</B></TD>
  </TR>
</TABLE>
```

Creating Conditional Statements

You can create conditional statements in the Template Toolkit language. Review the sample templates to find some example conditional statements. One of the simplest and most effective uses of conditional statements is in trapping error conditions.

For example, the following code snippet from the **LoginPage.html** template verifies the current user’s account status:

```
<% IF AccountDisabled %>
  <% LoginPageErrorMessage %>
<% ELSE %>
  <% IF LoginPageErrorMessage %>
    <% LoginPageErrorMessage %>
```

```
<% END %>
<% END %>
```

Conditional Operators

You can use the following conditional operators:

Table4: Conditional Operators

Operator	Description
==	is equal to
!=	is not equal to
<	is less than
<=	is equal to or less than
>	is greater than
>=	is equal to or greater than
&&	and
	or
!	not
and	and
or	or
not	not

IF and ELSIF Directives

You can use the **IF** and **ELSIF** directives to construct conditional behavior. Note that these are block directives, which means that you must use the **END** directive to indicate where each block ends. You may nest blocks indefinitely, provided you include an end statement for each nested block.

```
<% IF LoginPageErrorCode == 1002 %>
  <b> Your username or password is incorrect. Please reenter your credentials.
</b>
<%ELSIF LoginPageErrorCode == 1006 %>
  <b> The system is undergoing maintenance. Only administrators are allowed
to sign in at this time.</b>
<% END %>
```

SWITCH and CASE Directives

You may use the **SWITCH** and **CASE** directives to construct a multi-way conditional test. Note that these are block directives, which means that you must use the **END** directive to indicate where each block ends. You may nest blocks indefinitely, provided you include an end statement for each nested block.

```
<% SWITCH loginPageErrorCode %>
  <% CASE 1001 %>
    <b> Your session time has expired. </b>
  <% CASE 1006 %>
    <b> The system is undergoing maintenance. Only administrators are
permitted to sign in at this time. </b>
  <% CASE %> # default ...
<% END %>
```

Creating Looping Constructs

You may use directives such as **FOREACH** and **WHILE** to loop through blocks of code. Note that these are block directives, which means that you must use the **END** directive to indicate where each block ends. You may nest blocks indefinitely, provided you include an end statement for each nested block.

For example, the following sign-in page code loops through all of the authentication realms and displays them in a select list. The sample also uses the **RealmList** and **realm** predefined values.

```
<select size="1" name="realm">
  <% FOREACH r = RealmList %>
    <option value="<% r %>" ><% r %> </option>
  <% END %>
</select>
```

Unsupported Directives

We do not support the **USE**, **INTERPOLATE**, **TAGS**, **PERL**, or **RAWPERL** directives when creating custom pages. Additionally, we do not recommend using the **CALL** or **FILTER** directives.

Getting Familiar with the Template Page Structure

The sample templates were created using the common method of designing with HTML tables. One of the easiest ways to see how the layout works is to set all of the table borders to a visible state. For example, change **<TABLE border=" 0" >** to **<TABLE border=" 1" >** wherever you find it in the **LoginPage.html** template.

The table layout affords you a columnar layout. If you choose to maintain your customized design within the table layout, you may need to experiment with additions or deletions of table rows or cells. You can also change colors of cells, modify the locations of required fields, messages, or UI components.

You must always maintain the Template Toolkit directives that are included in each of the templates. These directives provide required information to PPS/PCS and enable to communicate system state back to the end-user, appropriately. Deleting or modifying any of the provided directives can result in unexpected behavior or data corruption.

CGI Files

The system generates a number of CGI files, which are called by the custom sign-in pages. Each file performs specific functions that are required by PPS/PCS. You should not modify any of these calls nor should you attempt to create your own CGI files to take the place of the supplied CGIs. You may see several references in the custom sign-in pages to the three following CGI files:

- **login.cgi**—Handles authentication request between custom pages, PPS/PCS, and any defined authentication servers.
- **welcome.cgi**—Handles redirection following a secondary authentication request, a password change, or new PIN generation.
- **starter.cgi**—Verifies that client applications are running and allows the user to attempt another login, typically when too many users are concurrently signed in to PPS/PCS.

Typically, in standard sign-in pages and sample custom pages the order of cgi usage is **welcome.cgi**, **login.cgi**, and then **welcome.cgi** again. For Host Checker pre-authorization, policies are evaluated in the first invocation of **welcome.cgi** and are passed to **login.cgi**. **login.cgi** takes the username and password parameters and determines the roles to assign. For Host Checker post-authorization, policies are evaluated only in the second invocation of **welcome.cgi**.



NOTE: When Host Checker and Cache Cleaner are configured, using **login.cgi** as the first URL to log in to Pulse Connect Secure is not supported.

Using Templates from the samples.zip File

The **samples.zip** file contains the following templates which you can customize for use in your own environment. **Samples.zip** contains the most general set of templates. Some of the templates in **samples.zip** are also contained in **kiosk.zip** and **softid.zip**. All files in **meeting.zip** are specific to the Connect Secure Meeting product.

- The environment Connect Secure pre-authentication pages
 - [“Logout.shtml / Logout-mobile-webkit.shtml / Logout-ipad.shtml”](#)
 - [“ExceededConcurrent.shtml / ExceededConcurrent-mobile-webkit.shtml / ExceededConcurrent-ipad.shtml”](#)
 - [“SSL.shtml / SSL-mobile-webkit.shtml / SSL-ipad.shtml”](#)
 - [“PleaseWait.shtml / PleaseWait-mobile-webkit.shtml / PleaseWait-ipad.shtml”](#)
 - [“SelectRole.shtml / SelectRole-mobile-webkit.shtml / SelectRole-ipad.shtml”](#)
- ACE pre-authentication pages
 - [“NewPin.shtml / NewPin-mobile-webkit.shtml / NewPin-ipad.shtml”](#)
 - [“NextToken.shtml / NextToken-mobile-webkit.shtml / NextToken-ipad.shtml”](#)
 - [“GeneratePin.shtml / GeneratePin-mobile-webkit.shtml / GeneratePin-ipad.shtml”](#)
 - [“ShowSystemPin.shtml / ShowSystemPin-mobile-webkit.shtml / ShowSystemPin-ipad.shtml”](#)
- ACE with eTrust pre-authentication pages
 - [“SM-NewPinSelect.shtml / SM-NewPinSelect-mobile-webkit.shtml / SM-NewPinSelect-ipad.shtml”](#)
 - [“SM-NewPinSystem.shtml / SM-NewPinSystem-mobile-webkit.shtml / SM-NewPinSystem-ipad.shtml”](#)
 - [“SM-NewUserPin.shtml”](#)
 - [“SM-NextToken.shtml / SM-NextToken-mobile-webkit.shtml / SM-NextToken-ipad.shtml”](#)
- Guest User Account Manager pages
 - [UserAdmin-add-user.shtml](#)
 - [UserAdmin-add-bulk-users.shtml](#)

-
- [UserAdmin-update-user.shtml](#)
 - [guest-user-print-page.shtml](#)
 - [guest-user-print-page-batch.shtml](#)
 - Password management pages
 - “[Defender.shtml](#) / [Defender-mobile-webkit.shtml](#) / [Defender-ipad.shtml](#)”
 - “[GraceLoginUsed.shtml](#) / [GraceLoginUsed-mobile-webkit.shtml](#) / [GraceLoginUsed-ipad.shtml](#)”
 - “[PasswordChange.shtml](#) / [PasswordChange-mobile-webkit.shtml](#) / [PasswordChange-ipad.shtml](#)”
 - “[PasswordExpiration.shtml](#) / [PasswordExpiration-mobile-webkit.shtml](#) / [PasswordExpiration-ipad.shtml](#)”

Customizing Look-and-Feel

You can customize the look-and-feel of your custom sign-in pages. This enables you to add your own corporate design, including logos, colors, and layout.

Required Components

You must include:

- Any template directives you find in the template files. This includes any code surrounded with template tags `<% %>`.
- Required HTML tags, such as the `<html>`, `<head>`, and `<body>` tags, although these may be consolidated into a separate template file which you can include using either the **PROCESS** or **INCLUDE** directives.
- Any `<form>` definitions.
- Any `<input>` statements.
- Any `<script>` statements and included scripts, typically JavaScript.
- Any `<object>` statements.
- Any `<embed>` statements.
- Any **onload**, **onUnload**, or other event triggers in the `<body>` tag.
- Any of the required variables, directives, or other items as noted in the commented section at the beginning of each template file.

You can exclude:

- Any HTML presentation code that is not a template directive, such as table tags, images, or style tags.

Displaying Custom Logos

You can change the templates to display your company’s logo by following these steps:

1. Open the template in a text editor or HTML editor.
2. Search for either of the two strings:

```
src="welcome.cgi?p=logo&signinId=<%signinId%>"  
src="welcome.cgi?=logo"
```

3. Replace the string with the following:

```
src="<% Home %>/imgs/custom-logo.gif
```

where *custom-logo.gif* is the new logo to be displayed. *custom-logo.gif* must be uploaded to PPS/PCS as part of your zip file. For more information, see ["Downloading Sample Templates and Uploading Files"](#).



NOTE: For **Remediate.html**, use `<%signinId%>` instead. For example:

```
src="<% signinID %>/imgs/custom-logo.gif"
```

Simplifying the Code in the Sample Templates

One of the first things you can do is simplify the code in the templates, by using the Template Toolkit **PROCESS** directive to extract common code into an **include** file. By moving common code to separate files, you can eliminate a certain amount of redundant code, thereby decreasing the size of the template files, and improving the readability of the files.

Let's take a look at one of the template files, **LoginPage.html**. Almost every designer who wants to customize the custom sign-in pages will likely want to customize **LoginPage.html**, which is the first screen that end-users see. If you have not downloaded the **Samples.zip** file, do so now, following the instructions in ["Downloading Sample Templates and Uploading Files"](#).

Open the **LoginPage.html** file in a text editor or HTML editor.

Version Directive

Each template file starts with a version directive, similar to the following:

```
<## NetScreen Page Version 1001 %>
```

Do not modify this directive.

Comment Section

Each template file then contains a set of comments that describe template variables, JavaScript, and HTML code required in that particular template. These comments can help you understand what directives and other elements of the file you must maintain, and give hints about other directives, variables, and methods that you can use to further customize the template.



NOTE: When you have completed your customization, you can consider removing the comment section from your files. The removal of this section decreases the size of the template files, and can possibly speed up page download times.

Header Tags

With the exception of the **Remediate.thtml** file, all of the template files contain the following HTML tags:

```
<html>
<head>
  <meta http-equiv="Content-Language">
  <meta http-equiv="Content-Type" content="text/html">
  <meta name="robots" content="none">
```

You can extract these tags into a separate **.thtml** file. If you ever need to change any of these tags, for example, to modify the meta tag definitions, you can then modify one file and the changes are propagated throughout all the files.

To create a simple header template file:

1. Open a new text file in your favorite text editor or HTML editor.
2. Cut and paste the preceding header lines into the file. You can cut these from one of the template files.
3. Save the file into the directory where the other template files are saved, naming the file **customhead.thtml**.



NOTE: Do not name the file **header.thtml**. Along with **config.thtml** and **footer.thtml**, this name may already be used by the PPS/PCS system software. You can call the files whatever you want, as long as they do not use these reserved names. They must end in the **.thtml** file extension.

4. Open each template file and delete the header lines.
5. In the place of the header lines, insert the following directive:

```
<% PROCESS customhead.thtml %>
```

The command **PROCESS** must be in all UPPER CASE. If you name the header file something other than **customhead.thtml**, then use that name in the directive instead.

6. When you zip your template files up, make sure you include the **customhead.thtml** file.

Copyright and Footer Tags

You might want to perform a similar task with the footer tags. This topic shows how to create a copyright notice in a separate include file.

1. Open a new text file in your favorite text editor or HTML editor.
2. Copy the following directive into the file:

```
<% year = 2005;
company = "Pulse Secure";
copyr = "Copyright $year $company" -%>
```

You can change the company name to your own company name.

3. Save the file into the directory where the other template files are saved, naming the file **customconfig.thtml**.

4. Open another new text file in your favorite text editor or HTML editor.
5. Copy the following directive and HTML to the file:

```
<div id="footer">
  &copy; <% copyr -%>
</div>
</body>
</html>
```

6. Save the file into the directory where the other template files are saved, naming the file **customfoot.thtml**.
7. In your **LoginPage.thtml** file, add the following directive just prior to the **PROCESS** directive for **customhead.thtml**. The lines should look like:

```
<% PROCESS customconfig.thtml %>
<% PROCESS customhead.thtml %>
```

8. In your **LoginPage.thtml** file, scroll to the end of the file.
9. Replace the last two lines that contain the closing **</body>** and closing **</html>** tags with the following directive:

```
<% PROCESS customfoot.thtml %>
```

10. Save the **LoginPage.thtml** file.

Page Banner

You can also create a common banner for your pages, eliminating the repetitious use of images and HTML code in every file. The following sample code creates a banner using an H1 header for the text displayed in the banner. Note the commented image tag, which you could use instead of the H1, if you prefer to use an image, for example, a logo.

1. Open a new text file in your favorite text editor or HTML editor.
2. Copy the following HTML into the file:

```
<div id="banner">
  <h1 class="banner">Sample</h1>
  <!-- 
</div>
```

3. Save the file into the directory where the other template files are saved, naming the file **custombanner.thtml**.
4. In your **LoginPage.thtml** file, add the following directive just after the **<body>** tag.

```
<body id=" body" onload="FinishLoad()">
  <% PROCESS custombanner.thtml %>
```

5. Save the **LoginPage.thtml** file.

Processing LoginPage.thtml

Now you can process the template to see the results of making the changes. Keep in mind the following requirements:

- You must have successfully installed and configured the Template Toolkit on your local system.
- All of the template files, including the custom header, custom footer, and custom config files must be saved into the same directory along with **LoginPage.thtml**.
- If you intend to upload your files to PPS/PCS, instead of processing them locally first, you can ignore this instruction. You must begin each file with the directive [% TAGS <% %> %] so that you can use angle brackets on any directives you add to the templates. By default, the Template Toolkit recognizes square brackets ([]) instead of angle brackets (<>) as delimiters for directives. PPS/PCS recognizes angle brackets only. Adding the **TAGS** directive tells the local Template Toolkit utilities that you are using angle brackets.

To process **LoginPage.thtml**:

1. Open a command window.
2. Navigate to the directory in which you have saved the **LoginPage.thtml** file, **customhead.thtml**, **customconfig.thtml**, and **customfoot.thtml**. For example, `/Templates/Sample/`.
3. Enter the following command:

```
tpage LoginPage.thtml > index.html
```

If you get no errors and the system returns a blank directory prompt, the process completed successfully.

4. Open a browser window.
5. Select **File > Open** and navigate to the directory where you have saved **LoginPage.thtml**.
6. Select **index.html**.

You should be able to see a copyright statement at the bottom of the page. The directive in **customfoot.thtml** pulled the data from **customconfig.thtml** and substituted the year and company name into the substitution variables **\$date** and **\$company**. Then, during the processing of **LoginPage.thtml**, **customfoot.thtml** was included based on the use of the **PROCESS** directive.



NOTE: Two warnings:

- If you received errors or if you were unable to run **tpage**, you need to check your Template Toolkit installation and documentation. You may not have the proper environment variable set to point to the Template Toolkit installation location or to your Perl distribution.
- Once you create a file, such as **index.html** in the preceding example, you must delete or move the file if you intend to create another version of the same file. The Template Toolkit **tpage** utility does not overwrite a file of the same name.

Changing Colors in the Table-Based Layout

Changing colors in the table-based layout is a very simple operation. All you need to do is replace color codes in the templates with the color codes representing the colors you prefer. Color codes are expressed in the hexadecimal format (#000000). If you want to change everything that is white to black, replace all occurrences of #FFFFFF to #000000.

In the **UI Options** page of a given role, you can also change banner, background, and link colors.

Changing the Layout

One of the simplest ways to customize the presentation of the pages is to modify the banner, colors, and table format. You can add table rows (`<tr>`) and table cells (`<td>`), graphics, change the **custom-logo.gif** to your own company logo, and change the background color of one or more table cells. You can also move pre-configured objects, such as the username field, password field, realm drop down menu, and submit button to other locations within the table. We do not go into detail about how to accomplish this. You should be familiar with HTML tables and be prepared to go through a trial and error process.

If you prefer to separate the presentation from the content, consider using cascading style sheets (CSS).

Using Cascading Style Sheets

One of the most dramatic changes you can make to the presentation style of the templates is to convert them fully to use cascading style sheets (CSS). In some templates you can already find brief examples of CSS codes, such as the use of `<DIV>` and `` tags. In those cases, it is best to leave those tags in place. You can, however, replace much of the HTML code around those tags, as well as around Template Toolkit directives.

As mentioned before, the sample templates use a traditional table-based design model. In effect, the design uses HTML tables as containers for all other objects on the page. Table rows and cells contain all graphics, error messages, fields, and buttons.

Although tables can simplify some elements of HTML design, they can complicate others, not to mention that designing with tables means that you must use tables on every page, if you intend to maintain the same basic layout. This adds complexity and decreases download times, as the end-user's browser must parse each table in each page every time it loads a page.

CSS provides an alternative. CSS consists of a set of styles that can be embedded in each HTML page or that can be stored in a separate `.css` file and referenced in the HTML files with a `<link>` tag. This topic is not meant to be a comprehensive CSS tutorial. It assumes you already have some familiarity with CSS. If you do not, you can find references on the Web, starting with the CSS specification at <http://www.w3c.org>.

For the purposes of this discussion, we will describe how to use CSS to eliminate redundant items, presentation information, and tables from your design. By doing so, you can improve the readability of your template files, and can consolidate almost all presentation instructions into one file.

Embedding CSS

You can embed CSS styles into an individual HTML page. If you plan on customizing only one file, you might find this approach easiest. Otherwise, you gain the most benefit by using a standalone CSS file and linking it into your HTML files.

Attaching a Standalone CSS

For the purposes of this discussion, we will introduce a simple CSS that will serve as an example method for replacing tables as a design element.

You can copy the following CSS into a text file and save it into the same directory with **LoginPage.shtml**. Save it as **sample.css**.

```
body { // Configures body margins
    margin:10px 10px 0px 10px;
    padding:0px;
}
```

```
span.error ( // Formats errors with bold and background
font-weight: bold;
background: silver;
}
```

```
#leftcontent { // Styles left column
position: absolute;
left:10px;
top:69px;
width:199px;
background:#809FB0;
border:1px solid #000000;
text-align:center
}
```

```
#centercontent { // Styles center column
background:#EFEFC9;
margin-left: 175px;
margin-right:11px;
border:1px solid #000000;
voice-family: "\"}\"";
voice-family: inherit;
margin-left: 199px;
margin-right: 0px;
```

```
#banner { // Styles the top banner
background:#C0C0B0;
height:60px;
border-top:1px solid #000000;
border-left:1px solid #000000;
border-right:1px solid #000000;
voice-family: "\"}\"";
voice-family: inherit;
height:59px;
}
```

```
p,h1,h2,h3,pre { // Sets common properties of these
margin:0px 10px 10px 10px;// paragraph elements
font-family:Arial, Verdana, Helvetica, sans-serif;
}
```

```
p{ // Styles paragraph element
font-size:12px;
text-align:left
}
```

```
h1,h2,h3 ( // Sets common elements for
```

```
padding-top:10px;// headings.
font-weight:bold;
}
h1 { // Sets size of H1 heading
font-size:18px;
}

h2 { // Sets size of H2 heading
font-size:14px;
}

h3{ // Sets size of H3 heading
font-size:12px;
}

#banner h1 { // Sets heading selector for banner
font-size:36px;
padding:10px 10px 0px 10px;
margin:0px;
font-weight: bolder
}

#footer { // Sets footer selector
border-top:3px solid #000000;
text-align:center;
font-family:sans-serif;
font-size:10px
}
a { // Sets common styles for anchor
font-family:sans-serif;
font-size:14px;
font-weight:bold;
}
a:link { // Sets link style
text-decoration: underline;
text-align:right;
color: black;
}
a:hover { // Sets hover style for links
color:white;
}
```

To make the connection between the CSS stylesheet and the page, you must include a `<link>` tag prior to the closing head tag `</head>`, as follows:

```
<link rel="stylesheet" type="text/css" href="sample.css" />
</head>
```

If you store the stylesheet in a directory other than the root directory where you keep your template files, make sure you express the path correctly in the href attribute of the link tag. For example, if you keep all of your stylesheets in a directory one level above the template file, you might indicate it as follows:

```
<link rel="stylesheet" type="text/css" href="../sample.css" />
```

or

```
<link rel="stylesheet" type="text/css" href="css/sample.css" />
```

This CSS defines the presentation style of all of the major components of the page. Additionally, if you want to change the look-and-feel of the page, while keeping the basic structure intact, you can create another CSS that uses the same style names, but that modifies the definitions. Without making any changes to your template file, you can apply an entirely unique style.

Here is a copy of the **LoginPage.shtml** file (excluding the comments section) after it has been converted to a CSS-based page. Compare it with the template file that you download. You might notice that virtually all of the presentation codes, things such as color settings, font settings, and border settings are missing from the following code. All of the presentation code has been extracted to the CSS stylesheet.

```
<%# NetScreen Page Version 1001 %>
```

```
<!-- The following line, when uncommented, allows you to process the .shtml file locally using the tpage utility, without having to replace the angle brackets on all of the Connect Secure template directives. The TAGS directive tells the Template Toolkit utilities to substitute angle brackets for the default square brackets. This line should be on the first line of the file when you are testing locally. Prior to uploading to Connect Secure, move and comment the line, or remove it from the file completely.
```

```
-->
<!--[% TAGS <% %> %]-->
```

```
<% PROCESS customconfig.shtml %>
```

```
<% PROCESS customhead.shtml %>
```

```
<title>Instant Virtual Extranet</title>
```

```
<link rel="stylesheet" type="text/css" href="sample_two.css" />
```

```

<script>
<!--
    JavaScript removed for readability. See template file for actual JavaScript.
-->
</script>
</head>
<body id="body" onload="FinishLoad()">

```

```

<% PROCESS custombanner.shtml %>

```

```

<!-- //////////////// -->
<!-- BEGIN LEFT COLUMN -->
<!-- //////////////// -->

```

```

<div id="leftcontent">

```

```

<!-- //////////////////////////////////////////////////////////////////// -->
<!-- BEGIN CUSTOM CONTENT LEFT COLUMN (Optional) -->
<!-- //////////////////////////////////////////////////////////////////// -->

```

```

<!-- Add more content here, if needed. -->

```

```

<!-- //////////////////////////////////////////////////////////////////// -->
<!-- END CUSTOM CONTENT LEFT COLUMN -->
<!-- //////////////////////////////////////////////////////////////////// -->

```

```

<!-- //////////////////////////////////////////////////////////////////// -->
<!-- BEGIN AUTHENTICATION TEMPLATE DIRECTIVES -->
<!-- and FORM USERNAME/PASSWORD/REALM PROMPTS -->
<!-- DO NOT DELETE. Must be included in page. -->
<!-- //////////////////////////////////////////////////////////////////// -->

```

```

<form name="frmLogin" action="login.cgi" method="post" autocomplete="off"
onsubmit="return Login()">

```

```

<input type="hidden" name="tz_offset">
<%IF !AnonymousAuthentication && !CertificateAuthentication%>
    <% FOREACH prompt = prompts %>
        <%NEXT IF !prompt.required %><br>
        <p><% prompt.promptText %></p>
<input type="<% prompt.type %>" name="<% prompt.name %>" size="20">
<% END %>

```

```
<% IF RealmList.size == 0 %>
<p>LoginRealm</p>
<input type="text" name="realm" size="20">
<% ELSIF RealmList.size == 1 %>
<input type="hidden" name="realm" value="<% RealmList.0 %>">
<% ELSE %>
<p>LoginRealm</p>
<select size="1" name="realm">
<% FOREACH r = RealmList %>
<option value="<% r %>" ><% r %></option>
<% END %>
</select>
<% END %>
<%ELSE%>
<input type="hidden" name="realm" value="<% RealmList.0 %>">
<%END%>
<input type="submit" value="Sign In" name="btnSubmit">
</form>
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
<!-- BEGIN CUSTOM CONTENT LEFT COLUMN -->
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<p>You are not yet authorized!</p>
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
<!-- END CUSTOM CONTENT LEFT COLUMN -->
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
</div>
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
<!-- END LEFT COLUMN -->
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
<!-- BEGIN CENTER COLUMN -->
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<div id="centercontent">
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
<!-- BEGIN PASSWORD MANAGEMENT TEMPLATE DIRECTIVES -->
<!-- DO NOT DELETE. Must be included in page. -->
<!-- Added for Account Disabled Case for Password Managemnt -->
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<% IF AccountDisabled %>
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<!-- Account Disabled Case -->
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<span class="error"><% LoginPageErrorMessage %></span>
```

```
<% ELSE %>
```

```
<% IF LoginPageErrorMessage %>
```

```
<span class="error"><% LoginPageErrorMessage %></span>
```

```
<% END %>
```

```
<% END %>
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<!-- END PASSWORD MANAGEMENT DIRECTIVES -->
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<!-- Optional error handling codes -->
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<span class="error">
```

```
<% IF PageErrorCode == 1020 %>
```

```
You have successfully changed your password.
```

```
<% END %>
```

```
</span>
```

```
<span class="error">
```

```
<% IF PageErrorCode == 1021 %>
```

```
Account Disabled. Please contact Administrator for help.
```

```
<% END %>
```

```
</span>
```

```
<span class="error">
```

```
<% IF PageErrorCode == 1023 %>
```

```
Account Expired. Please contact Administrator for help.
```

```
<% END %>
```

```
</span>
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<!-- END optional error handling codes -->
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
<!-- BEGIN CUSTOM CONTENT CENTER COLUMN -->
<!-- Add any content below that you want to appear in the center column. -->
<!-- You can add a mix of headings, paragraphs, graphics or any other HTML.-->
<!--//////////////////////////////////////////////////////////////////// -->
```

```
<h1>Welcome to XYZ Company!</h1>
```

```
<h2>New Employee Benefits!</h2>
<p>Per munere latine officiis ad. Quo te elit vivendum, mea aperiri integre
periculis ex, ea sea corpora consecetuer. </p>
<h2>Company Party!</h2>
<p>Dolore efficiantur vim ea, alia putant vivendo at pri. Odio cetero tibi que no vis.
Mea vero delenit ad. </p>
<h2>Corporate Travel Advisory</h2>
<p>Cu zzril expetenda has, ludus utinam instructor nam an, id eam fugit putent
possit. Mazim epicuri contentiones mel ut, pri atqui maiestatis ei. </p>
<p>Tempor oportere usu ne. Ea aperiam sanctus oportere vel.</p>
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
<!-- END CUSTOM CONTENT CENTER COLUMN -->
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
</div>
```

```
<!-- //////////////////////////////////////////////////////////////////// -->
<!-- END CENTER COLUMN -->
<!-- //////////////////////////////////////////////////////////////////// -->
```

```
<% PROCESS customfoot.thtml %>
```

Working with the Pulse Secure Desktop Client or Standalone Network Connect

When you manually launch Network Connect (from the Start menu) or initiate a connection to the PCS gateway from the Pulse Secure desktop client user interface, a different window appears compared to launching the client from a browser window.

To modify the standalone version of the login page, create a **LoginPage-stdaln.thtml** file (you can copy the existing **LoginPage.thtml** file), customize it according to your needs, and add it to your zip file.

Generally speaking, standalone pages follow the convention name-**stdaln.thtml** where name is the equivalent name.**thtml** file (similar to the method used for the Mobile files, name-**ppc.thtml**).

Customizing Error Handling and Navigation

The Template Toolkit provides a rudimentary level of error handling capability. In most cases, errors are handled sufficiently by the included logic. However, you might want to modify the behavior of the pages, given a particular error. For example, you might want to redirect a user to your own specific custom error page on an “Account Disabled” error.

You can redirect based on specific individual errors or based on any error occurring at all. The following samples illustrate both methods.

Redirecting Based on any Errors

If you want to redirect to your own Web page based on the occurrence of any errors, you can implement logic, as follows:

```
<head>
  <% IF LoginPageErrorMessage %>
    <meta http-equiv=" refresh" content=" 0;url=DisplayErrorPage.html" >
    <title>Redirect</title>
</head>
```

Redirecting Based on Specific Errors

If you want to redirect to your own Web page based on the occurrence of a specific error, such as the “Account Disabled” error, you can implement logic, as follows:

```
<head>
  <% IF LoginPageErrorCode == 1021%>
    <meta http-equiv=" refresh" content=" 0;url=AcctDisabledError.html" >
  <% ELIF PageErrorCode == 1023 %>
    <meta http-equiv=" refresh" content=" 0;url=AcctExpiredError.html" >
  <% END %>
  <title>Redirect</title>
</head>
```

For more information on error codes, see “[LoginPageErrorCode](#)”.

Defining Error Message Locations

You can locate error messages wherever you want on the page. Keep in mind, however, that you cannot move the error handling code relative to its position within the `<form/>` tags. For example, do not reverse the positions of two directives appearing within the body of `<form/>` tags or move the error handling code outside of the `<form/>` tags.

Customizing Error Messages

You can customize some of the error and confirmation messages your end-users might see. The system produces some error messages that you cannot change. [Table 6](#) lists those that you can change.

Although you cannot change any messages on PCS/PPS itself, you can customize error messages that will appear instead of the messages originating. Using the Template Toolkit `IF` directive, you can test for a message based on an error’s unique

error code, and expressed by the predefined template variable **PageErrorCode**. In the directive block, you include the customized message, as shown in the following example:

```
<% IF PageErrorCode == 1020 %>  
  You have successfully changed your password.  
<% END %>
```

```
<% IF PageErrorCode == 1021 %>  
  Account Disabled. Please contact administrator at X5540 for help.  
<% END %>
```

```
<% IF PageErrorCode == 1023 %>  
  Account Expired. Please contact administrator at X5541 for help.  
<% END %>
```

You might include the preceding code in your **LoginPage.html** file following the form (**<form></form>**) that prompts the user for login credentials. You can see that the example messages in the preceding sample include phone extension numbers, to help end-users identify the correct administrator to call after encountering specific problems.

Localizing Custom Sign-In Pages

By default, text strings are in English. Administrators supporting non-English users may need to configure the sign-in pages to provide localized text labels. This can only be done on a per-sign-in page basis. For multi-language support, Administrators must configure different sign-in pages for different locales. For further customizations, you can upload customized sign-in pages using the Template Toolkit. Contact Pulse Secure Global Support Centre for details.

You can create localized custom sign-in pages, using the downloadable sample template files. You can download the samples from **Signing In > Sign-in pages > Upload Custom Pages > Upload Custom Sign-In Pages** page.

You include the localized version of the template files in separate directories, one directory per language, in your uploaded zip file. PPS/PCS recognizes all of the **.html** files in the root directory of the zip file as default pages. When creating the subdirectory for a set of given language files, name the subdirectory with the Accept-Language header 2-char abbreviation. Retain the default names for the localized file names.

At runtime, PPS/PCS notes the default language of the browser, then searches the subdirectory corresponding to the 2-char **Accept-Language** abbreviation to find the specific **.html** file. If the system locates the file, PPS/PCS uses that **.html** file. If the file cannot be located the default page in the root directory is used.



NOTE: Use the Accept-Language abbreviation **zh-cn** if you plan to use either of the abbreviations **zh-cn** or **zh-sg**.

The directories should contain full sets of template files, as required to implement custom sign-in pages.



NOTE: For a list of acceptable Accept-Language header abbreviations, see <http://www.oasis-open.org/cover/iso639a.html>.

Creating a Localized Set of Custom Sign-In Pages

To create a localized set of custom sign-in pages, perform the following procedure.

1. Select **Signing In > Sign in > Sign in pages**.
2. Click **Upload Custom Pages**.
3. Click the link for the sample templates you want to localize.
4. Once downloaded, unzip the files.
5. Create a subdirectory in the sample templates directory and name it with the 2-character identifier that specifies the language into which you intend to translate the template files. For example, if you are translating into French, name the directory **fr**. Create subdirectories for each language into which you want to translate the templates, and copy all template files and the images directory into each subdirectory.
6. Copy all of the templates and the images directory (**/imgs**) and paste them into the new folder.
7. Translate the template files.
8. Change the paths in the translated files to point to the correct path. For example, you need to update the paths to images in the **/imgs** subdirectory, otherwise, they will continue to point to the default directory.
9. When translation is completed, zip the files and upload to PPS/PCS.

To test the languages, make sure you update the language option in your Web browser, first.

Upgrade Considerations

Consider the following when upgrading to the latest version of the custom sign-in pages:

- You must update the version number in your templates to match the current version. If you do not update your version number, the default page appears instead of your custom page. Alternatively, you can copy your custom content to the new templates.
- Old variable names may change and new variables may be added. It is recommended that you convert old variable names to their new counterparts as the default values for the old variables may no longer exist. If you do not want to update your variable names, you can select the **Skip validation checks** during **upload** option in the **Upload Custom Sign-In Pages** page. If you select this option, you should review all your custom pages to ensure that they are still functioning correctly.
- Some formatting, such as text strings and color, change from prior versions. Review your page to ensure that the changes are acceptable.

Custom Templates Reference

The following topics describe each of the template files, JavaScript, template variables, some form variables, their uses, functionality, and customizability. The topics are organized alphabetically. Within the reference, you can find definitions for:

- Pre-authentication pages

You may customize a variety of standard pre-authentication pages, including the standard sign-in page (**LoginPage.thtml**), the failed authentication page (**SSL.thtml**), the sign out page (**Logout.thtml**), the sign-in warning page (**ExceededConcurrent.thtml**), the Host Checker and Cache Cleaner start page (**PleaseWait.thtml**), and the select from multiple roles page (**SelectRole.thtml**).

-
- ACE pre-authentication pages

The **SoftID.zip** file enables you to customize the pages for use with the RSA Soft ID client.

When the end-user browses to the URL of the SoftID custom sign-in page, the page prompts the user to enter a PIN by way of the SecurID application on his local machine. If the application accepts the PIN, the end-user is logged in.

To configure the SoftID custom sign-in page, download the **Samples.zip** and **SoftID.zip** files. Unzip Samples.zip to a directory first, and then unzip the **SoftID.zip** file to the same location, overwriting any duplicates. Once you have customized the files to your liking, zip the entire set of files and upload them.

To ensure that the New Pin and Next Token pages are customized for SoftID authentication, copy **NewPin.shtml** to **GeneratePin.shtml** in the **SoftID.zip** file and upload the modified zip file to system for the custom sign-in page.

- ACE with eTrust pre-authentication pages

The ACE with eTrust pre-authentication pages are named with the SM prefix, for SiteMinder.

- Password management pages

These include pages such as the secondary authentication server login page (**SecondaryLoginPage.shtml**).

- Template variables

- Kiosk pages

The **Kiosk.zip** file contains templates that enable you to customize the pages for use by Kiosk users. This sample shows how you can implement a custom UI to circumvent keystroke loggers. When you run the sample **LoginPage.shtml** (after it has been uploaded) you see a Virtual Keyboard. The password field is disabled, so an end-user must use the Virtual Keyboard to enter a password. Because the normal keyboard is disabled, a keystroke logger cannot capture the password.

- Meeting pages

The **Meeting.zip** file enables you to customize a variety of pre-authentication and post-authentication pages used by Pulse Secure Collaboration attendees.



NOTE: When uploading templates, you must include all of the THTML pages included in the original zip files for PPS/PCS to accept the upload.

- JavaScript

Many of the templates include optional JavaScript functions that you can use to perform a variety of tasks on that particular page. For example, the header in **LoginPage.shtml** contains several JavaScript functions. Most of these functions handle cases where you associate multiple authentication realms with a single sign-in URL. You can also add your own JavaScript functions to perform presentation effects, or to modify the default navigation by redirecting to your own Web pages, in some cases.

Mobile Device Pages

The .html pages for mobile devices such as the Android and iPhone are identified by the **-mobile-webkit** string in the page name. For example, Cancel-mobile-webkit.html. Requirements for these pages are the same as their non-mobile counterparts except as noted in the following topics. For example, Cancel-mobile-webkit.html has the same requirements as cancel.html.

iPad Pages

The .html pages for the iPad are identified by the **-ipad** string in the page name. For example, Cancel-ipad.html. Requirements for these pages are the same as their non-mobile counterparts except as noted in the following topics. For example, Cancel-ipad.html has the same requirements as cancel.html.

<failedPolicy>

	Tag used to go through list of failed policies.
Object Type	Template variable
Included In	Remediate.html
Usage	Required. Do not modify.
Example	None
Related Documentation	Remediate.html / Remediate-mobile-webkit.html / Remediate-ipad.html

<failedPolicy>.name

	Name of the policy that fails during remediation.
Object Type	Template variable
Included In	Remediate.html
Usage	Required. Do not modify.
Example	None
Related Documentation	Remediate.html / Remediate-mobile-webkit.html / Remediate-ipad.html

<failedPolicy>.remediation

	Custom instructions as defined in the remediation policy.
Object Type	Template variable
Included In	Remediate.html
Usage	Optional
Example	None
Related Documentation	Remediate.html / Remediate-mobile-webkit.html / Remediate-ipad.html

action

	An HTML form element that provides the action that is meant to occur, usually on a submit button click or POST .
Object Type	HTML form element
Included In	All templates with forms
Usage	Required

Example The following fragment from a `<form>` field runs the `login.cgi` Perl script:

```
action="login.cgi"
```

Related Documentation [autocomplete](#)

activex_ini

Contains the name of an ActiveX control.

Object Type Template variable

Included In `LoginMeeting.thtml`, `MeetingTestJS.thtml`

Usage Required if running Internet Explorer on a Windows system. This variable contains the name of the ActiveX control that must be loaded to run Meeting in the browser.

Example The following sample checks to see if the browser is Internet Explorer. If so, it passes the name of an ActiveX object to initialize.

```
<% IF ie %>
  <P>
  <OBJECT>
  <% activex_ini FILTER verbatim%>
  </OBJECT>
  </P>
<% END %>
```

Related Documentation

- [applet_ini](#)
- [FILTER verbatim](#)

AgentInstall.thtml

The page after downloading the UAC Agent.

Object Type Template variable

Usage Optional (Valid for Policy Secure).

Required Variables

- DSLlaunchURL
- pleasewaitObjectHC
- IEonWindows
- delivery_mode
- setup_codebase
- setup_classid
- setup_applet_vitals

isMac
 pleasewaitObjectJAM
 I18N_UAC_DOWNLOAD_AND_INSTALL_64_BIT

Related Documentation

-

AgentlessInstall.shtml

The page after downloading the UAC Agent.

Object Type Template variable

Usage Optional (Valid for Policy Secure).

Required Variables

 DSLlaunchURL

 pleaseWaitObjectEP

 setup_classid

 setup_codebase

 isLinux

 IEonWindows

Related Documentation

-
-

AnonymousAuthentication

Use this variable to specify that users signing into an anonymous realm should not see the username and password fields in the sign-in page.

Object Type Template variable

Included In **LoginPage.shtml**

Usage Optional. Pulse Connect Secure sets this value to **true** if the authentication realm is set to an anonymous server. If you use this variable, Pulse Connect Secure only displays the sign-in page to users when they encounter one of the errors described in “[LoginPageErrorCode](#)”. If you set **AnonymousAuthentication**, the realm is available in **RealmList.0**. You need to send the realm in the form as a hidden variable.

Example The following sample checks to see if **AnonymousAuthentication** and **CertificateAuthentication** are set to false. If the variables are set to false, the following directives generate authentication fields, setting the size of the input fields to 20 characters.

```
<%IF !AnonymousAuthentication && !CertificateAuthentication%>
<% FOREACH prompt = prompts %>
<%NEXT IF !prompt.required %>
```

```
<% prompt.promptText %>
<input type="<% prompt.type %>" name="<% prompt.name %>" size="20">
<% END %>
```

Related Documentation

- [CertificateAuthentication](#)
- [RealmList](#)

applet_ini

Contains instructions that initialize a meeting applet.

Object Type

Template variable

Included In

MeetingAppletInstaller.shtml, **MeetingTestMSJava.shtml**

Usage

Required

Example

The following code fragment shows how the **applet_ini** variable passes an applet to the page, and how the additional code passes parameters to the applet:

```
<% IF install %>
  <P>
  <APPLET
  <% applet_ini FILTER verbatim %>
  <PARAM NAME="Parameter0"
  VALUE="meeting_id=<%instanceId%>;user_name=<%username%>;cert_md5=<%c
ert_md5%>;ncp_read_timeout=<%ncp_read_timeout%>">
  <PARAM NAME="StartSessionReDir" VALUE="<%startSessionReDir%>">
  <PARAM NAME="UninstallReDir" VALUE="<%uninstallReDir%>">
  <PARAM NAME="locale" VALUE="<%locale%>">
  </APPLET>
```

Related Documentation [MeetingAppletInstaller.shtml](#)

appstr

Contains the applet tag for the Pulse Secure Collaboration client.

Object Type

Template variable

Included In

MeetingRunJava.shtml

Usage

Required

Example

The following example checks to see if a login is required. If not, Pulse Connect Secure loads the applet indicated in the **appstr** variable.

```
<% IF !login_required %>
<% appstr FILTER verbatim %>
<% END %>
```

Related Documentation [FILTER verbatim](#)

authenticate()

	Authenticates a user.
Object Type	JavaScript function
Included In	GeneratePin.shtml (ACE/SoftID), LoginPage.shtml (ACE/SoftID), NewPin.shtml (ACE/SoftID), NextToken.shtml (ACE/SoftID)
Usage	Required
Example	The following examples illustrate the use of the authenticate() function.

From ACE/SoftID **LoginPage.shtml**. This function checks to see whether or not the user is authenticated. If the user is not authenticated, the function displays an error from the authentication server. Otherwise, it passes the authentication credentials from the auth server to the **frmLogin HTML** form in **LoginPage.shtml**:

```
function authenticate()
{
    var rc = -1;
    var time = <% softid_time %> ;
    var user = document.frmLogin.username.value;
    var state = 0;

    if(typeof(document.sdui.sdAuth) == "undefined")
    {
        rc = -1;
    }
    else
    {
        if (error != "")
        {
            alert(error);
        }
        rc = document.sdui.sdAuth(time, user, state);
    }
    if (rc == 1)
    {
        document.frmLogin.username.value = document.sdui.getUsername();
        document.frmLogin.password.value = document.sdui.getPasscode();
        document.frmLogin.submit();
    }
    else
    {
        document.cancelForm.submit();
    }
}
```

From ACE/SoftID **GeneratePin.thtml**. The following function enables the user to generate a new PIN for signing in to an authentication server:

```
function authenticate()
{
    var min = parseInt(document.minmaxForm.minlen.value);
    var max = parseInt(document.minmaxForm.maxlen.value);
    var alp = parseInt(document.minmaxForm.alphanumeric.value);
    var sys = document.minmaxForm.systempin.value;

    if(typeof(document.sdui.sdPin) == "undefined")
    {
        rc = -1;
    }
    else
    {
        if (error != "")
            alert(error);

        rc = document.sdui.sdPin(min, max, <% secid_pinselectmode %>,
alp, sys);
    }
    if (rc == 1)
    {
        document.frmNewPin.password.value = document.sdui.getPin();
        document.frmNewPin.password2.value =
document.frmNewPin.password.value;
        document.frmNewPin.secidaction.value = "Save PIN";
        document.frmNewPin.secidactionSavePin.value = "Save PIN";
        document.frmNewPin.submit();
    }
    else if(rc == 2)
    {
        document.frmNewPin.password.value = "";
        document.frmNewPin.password2.value= "";
        document.frmNewPin.secidactionCancel.value = "Cancel";
        document.frmNewPin.submit();
    }
    else
    {
        document.cancelForm.submit();
    }
}
}
```

Related Documentation

- [GeneratePin.thtml](#) / [GeneratePin-mobile-webkit.thtml](#) / [GeneratePin-ipad.thtml](#)
- [LoginPage.thtml](#) / [LoginPage-mobile-webkit.thtml](#) / [LoginPage-ipad.thtml](#)

autocomplete

	Prevents the form from auto-filling username or authentication realm field entries using cached values.
Object Type	HTML form element
Included In	All templates with forms
Usage	Optional
Example	<pre><form name="frmLogin" action="login.cgi" method="post" autocomplete="off" onsubmit="return Login()"></pre>
Related Documentation	None

Cancel.shtml / Cancel-mobile-webkit.shtml / Cancel-ipad.shtml

	Displays a message to the user informing him that the sign-in request has been cancelled.
Object Type	Template
Usage	Required
Required Variables	None
Example	None

cboxuser

	Indicates whether or not the attendee is a PCS user.
Object Type	Template variable
Included In	LoginMeeting.shtml
Usage	Required
	If cboxuser = 1 , this user is not signed in as a PCS user, and the page displays a text box labeled Your Name . If cboxuser = 0 , this user is signed in as a Pulse Connect Secure user. In this case, the page does not display the text box labeled Your Name but instead displays the user's sign-in name.
Example	The following example displays the text box and label in an HTML table. <pre><% IF cboxuser %> <tr><td align="right">Your Name:</td> <td><input type="text" name="username" value="<% username %%" maxlength="40" size="20"></td> <% END %></pre>
Related Documentation	LoginMeeting.shtml

cclnAcTimeout

Specifies the Cache Cleaner login inactivity in minutes.

Object Type Template variable

Included In **LoginPage.shtml**

Usage Required

Example The following code sets the timeout interval to 60000 milliseconds times the smaller value of **cclnAcTimeout** or **hclnAcTimeout**.

```
<% IF hclnAcTimeout > cclnAcTimeout %>
window.setTimeout("deletepreauth()", 60000 * <% cclnAcTimeout %>);
<% ELSE %>
window.setTimeout("deletepreauth()", 60000 * <% hclnAcTimeout %>);
<% END %>
```

Related Documentation [LoginPage.shtml](#) / [LoginPage-mobile-webkit.shtml](#) / [LoginPage-ipad.shtml](#)

ccRunning

Specifies whether or not Cache Cleaner is currently running.

Object Type Template variable

Included In **LoginPage.shtml**

Usage Required

Example The following code sets the timeout interval based on whether Host Checker and Cache Cleaner are both running or only Host Checker.

```
<% IF hcRunning && ccRunning %>
<% IF hclnAcTimeout > cclnAcTimeout %>
window.setTimeout("deletepreauth()", 60000 * <% cclnAcTimeout %>);
<% ELSE %>
window.setTimeout("deletepreauth()", 60000 * <% hclnAcTimeout %>);
<% END %>
<% ELSIF hcRunning %>
window.setTimeout("deletepreauth()", 60000 * <% hclnAcTimeout %>);
<% END %>
```

Related Documentation [LoginPage.shtml](#) / [LoginPage-mobile-webkit.shtml](#) / [LoginPage-ipad.shtml](#)

cert_md5

Contains the **cert-md5** statement (the MD5 fingerprint of the SSL peer's certificate) that enables the user to install an applet in a meeting session.

Object Type Template variable

Included In	MeetingAppletInstaller.thtml
Usage	Required
Example	This fragment, which is contained in MeetingAppletInstaller.thtml , shows the use of the cert_md5 variable:

```
<% IF install %>
<APPLET
  <% applet_ini FILTER verbatim %>
  <PARAM NAME="Parameter0"
  VALUE="meeting_id=<%instanceId%>;user_name=<%username%>;
  cert_md5=<% cert_md5 %>;ncp_read_timeout=<%ncp_read_timeout%>">
  <PARAM NAME="StartSessionReDir" VALUE="<%startSessionReDir%>">
```

Related Documentation [MeetingAppletInstaller.thtml](#)

CertificateAuthentication

Use this variable if you want to specify that users signing into a certificate realm should not see the username and password fields in the sign-in page.

Object Type	Template variable
Included In	LoginPage.thtml
Usage	Pulse Connect Secure sets this value to true if the authentication realm is set to a certificate server. If you use this variable, Pulse Connect Secure only displays the sign-in page to users when they encounter one of the errors described in " LoginPageErrorCode ". (For information about hiding this page from users authenticated against a eTrust SiteMinder server using client-side certificate authentication, see " Login() ".) If you set CertificateAuthentication , the realm is available in RealmList.0 . You need to send the realm in the form as a hidden variable.
Example	The following code tests for AnonymousAuthentication and CertificateAuthentication :

```
<%IF !AnonymousAuthentication && !CertificateAuthentication%>
  <% FOREACH prompt = prompts %>
    <%NEXT IF !prompt.required %>
    <% prompt.promptText %>
    <input type="<% prompt.type %>" name="<% prompt.name %>" size="20">
  <% END %>
```

Related Documentation

- [AnonymousAuthentication](#)
- [RealmList](#)

changePasswordTitle

Argument to display password change title text.

Object Type	Template variable
--------------------	-------------------

Included In	PasswordChange.shtml
Usage	Optional
Example	The following code tests the boolean showChangePasswordTitle and, if true, passes the argument to the page, as is.

```
<%IF showChangePasswordTitle %>
  <% changePasswordTitle FILTER verbatim %>
<%END%>
```

Related Documentation [showChangePasswordTitle](#)

check

	Contains the name of the GIF file that Pulse Connect Secure displays on the MeetingTestResult page.
Object Type	Template variable
Included In	MeetingTestResult.shtml
Usage	Required
Example	The following code displays the GIF that is referenced in the check template variable:

```
<table cellpadding="2" cellspacing="2" border="0">
<tr valign="top">
  <td>
    
  </td>
  <td>
    
  </td>
  <td width="100%" nowrap>
    <% message FILTER verbatim %>
  </td>
</tr>
</table>
```

Related Documentation [MeetingTestResult.shtml](#)

checkActiveX()

	Checks to see if ActiveX is enabled.
Object Type	JavaScript function
Included In	MeetingAppletInstaller.shtml , MeetingTestMSJava.shtml , MeetingTestJS.shtml , MeetingTestJava.shtml
Usage	Optional
Example	The following examples illustrate the verification of ActiveX.

From **MeetingTestMSJava.thtml**:

```
function checkActiveX () {
  if ( typeof(SecureMeetingApplet) == "undefined") {
    redirect();
    return true;
  }
  setTimeout('redirect()', 5000);
  return true;
}
```

From **MeetingRun.thtml**:

```
function checkActiveX () {
  var f = document.meetingrun;
  if (typeof(f) == "undefined" ||
    typeof(f.button1) == "undefined")
    return;
  try {
    NeoterisSetup.isValid();
    f.button1.disabled = false;
  } catch (e) {
    ;
  }
}
```

From **MeetingTestJS.thtml**:

```
function checkActiveX () {
  var win = <% win32 %>;
  try {
    NeoterisSetup.isValid();
    document.form1.submit();
    return true;
  } catch (e) {
    var url = win ? "/dana-
na/meeting/<%signin%>meeting_testmsjava.cgi?<%mid_param%>" : "/dana-
na/meeting/<%signin%>meeting_testjava.cgi?<%mid_param%>";
    document.location.replace(url);
  }
}
```

From **MeetingTestJava.thtml**:

```
function checkActiveX () {
  setTimeout('redirect()', 30000);
  return true;
}
```

Related Documentation

- [MeetingAppletInstaller.shtml](#)
- [MeetingRun.shtml](#)
- [MeetingTestMSJava.shtml](#)
- [MeetingTestJava.shtml](#)
- [MeetingTestJS.shtml](#)

color

	Contains the color definition.
Object Type	Template variable
Default value	None
Included In	, Cancel.shtml , Defender.shtml , ExceededConcurrent.shtml , GeneratePin.shtml , GraceLoginUsed.shtml , LoginPage.shtml , Logout.shtml , NewPin.shtml , NextToken.shtml , PasswordChange.shtml , PasswordExpiration.shtml , PleaseWait.shtml , Remediate.shtml , SM-SM-NewPintSelect.shtml , SM-NewPinSystem.shtml , SM-NewUserPin.shtml , SM-NextToken.shtml , SSL.shtml , SecondaryLoginpage.shtml , SelectRole.shtml , ShowSystemPin.shtml
Cancel.shtmlNewPinSystem.shtmlUsage	Optional
Example	This example sets the background color of the table cell to the value of color . <pre><td bgcolor="<% color %>"></td></pre>
Related Documentation	None

confirmation-new-ux.shtml

	Prompts the user for confirmation before proceeding.
Object Type	Template
Usage	Required. Used for displaying confirmation page capturing opened sessions.
Required Variables	isIMODE mobile_browser_login isPhones msgType listButtons formDataStr

confirmation_opensessions-new-ux.shtml

	Prompts the user to confirm the already existing sessions for the same user.
Object Type	Template

Usage	Required. Used for displaying confirmation page capturing opened session.
Required Variables	mobile_browser_login sessionInfos listButtons formDataStr

Continue

	Submits the hidden form when an end-user clicks the Continue button.
Object Type	JavaScript function
Included In	Remediate.thtml
Usage	Optional
Example	The following code fragment defines the Continue button and passes control to the Continue() JavaScript function.

```
<input name="btnContinue" type="button" value="<% textContinue %>"  
onClick="Continue()"/>
```

Related Documentation [Remediate.thtml](#) / [Remediate-mobile-webkit.thtml](#) / [Remediate-ipad.thtml](#)

secid_key

	Contains the transaction state.
Object Type	Template variable
Included In	Defender.thtml , Defender-mobile-webkit.thtml , Defender-ipad.thtml
Usage	Required
Example	The following example uses secid_key in the INPUT statement.

```
<input type=hidden name="key" value="<% secid_key %>">
```

Related Documentation [Defender.thtml](#) / [Defender-mobile-webkit.thtml](#) / [Defender-ipad.thtml](#)

Defender.thtml / Defender-mobile-webkit.thtml / Defender-ipad.thtml

	Prompts the Radius user to enter his PIN and provides the appropriate challenge values from the server.
Object Type	Template
In Usage	Password Management
Required Variables	"secid_challenge"
Example	None

delivery_mode

Indicates the delivery mode of an applet, whether Java or ActiveX.

Object Type	Template variable
Included In	AgentInstall.shtml, Logout-ipad.shtml, Logout-mobile-webkit.shtml, Logout-ppc.shtml, Logout.shtml, PleaseWait-ppc.shtml
Usage	Required
Example	The following fragment checks to see if the delivery mode is Java when Cache Cleaner, Host Checker, or Win32 applets are running during a user logout. If Java is the delivery mode, the code indicates that the page should call the stopComponents function when loading.

```
<% IF (pleasewaitObjectCC || pleasewaitObjectHC || pleasewaitWin32) &&
    delivery_mode == 'java'%>
    onload="javascript:stopComponents()"
<%END%>
```

Related Documentation [stopComponents\(\)](#)

DoUnload()

Forces Pulse Connect Secure to cancel the current mode.

Object Type	JavaScript function
Included In	GeneratePin.shtml, NewPin.shtml
GeneratePin.shtml Usage	Required. You must include this function in the header of NewPin.shtml and call it from the body.
Example	The following code shows how the DoUnload() function appears in the NewPin.shtml template. The code serves a similar purpose in other templates, as well.

```
function DoUnload() {
    if (gCancelNewPinMode) {
        window.open("secid_cancelpin.cgi", "newpincancel",
            "resizable=1,height=250,width=200,status=0");
    }
}
```

Related Documentation

- [NewPin.shtml / NewPin-mobile-webkit.shtml / NewPin-ipad.shtml](#)
- [NextToken.shtml / NextToken-mobile-webkit.shtml / NextToken-ipad.shtml](#)

ds_mobile_common.css

Cascading style sheet for .shtml mobile pages such as the Android and iPhone.

Object Type	Text file
--------------------	-----------

Included In	PageHeader-mobile-webkit.shtml
Example	The following code shows how the ds_mobile_common.css style sheet is called in the PageHeader-mobile-webkit.shtml template.

```
<head>
<meta http-equiv="Content-Language"/>
<meta http-equiv="Content-Type" content="text/html"/>
<title><%title FILTER verbatim %></title>
<link href="dana-na/css/ds_mobile_common.css" type="text/css" rel="stylesheet"/>
```

ExceededConcurrent.shtml / ExceededConcurrent-mobile-webkit.shtml / ExceededConcurrent-ipad.shtml

	Displays a performance warning to the user when too many concurrent users are signed into Pulse Connect Secure.
Object Type	Template
Usage	Required. You must always include this template in your zip file. This is a standard Pulse Connect Secure page that displays a performance warning to the user when too many concurrent users are signed into Pulse Connect Secure. This template does not contain any JavaScript or forms. It does contain an optional link to starter.cgi , however, that allows users to sign into Pulse Connect Secure as well as error message variables. For more information, see comments in the template.
Required Variables	None
Example	None

expired

	Boolean indicating whether or not a meeting session has expired.
Object Type	Template variable
Included In	MeetingRun.shtml
Usage	Required
Example	The following code checks to see if ActiveX is enabled and that the session is not expired. If both conditions are true, the meeting is initialized.

```
<% IF type == "activex" && !expired %>
<% meeting_ini FILTER verbatim %>
<% END %>
```

Related Documentation	MeetingRun.shtml
------------------------------	----------------------------------

fieldsArr

Template Toolkit hash value of additional fields for guest user information (keys: field and fieldVal).

Object Type	Template variable
Included In	guest-user-email-page.thtml, guest-user-print-page-batch.thtml, guest-user-print-page.thtml
Usage	Optional (Required for Pulse Policy Secure)
Example	The following code displays the additional fields of guest user.

```
<%FOREACH f = fieldsArr%>
  <% f.field %>
  <% f.fieldVal %>
<% END %>
```

Related Documentation

fallback_setup_applet_vitals

Setup client Java applet.

Object Type	Template variable
Included In	Logout.thtml
Usage	Required
Example	<pre><APPLET id=NeoterisSignoff <% fallback_setup_applet_vitals FILTER verbatim %></pre>

Related Documentation

FILTER verbatim

Indicates that the page should process the preceding variable raw, without performing any parsing or other actions on the variable contents.

Object Type	Template directive
Included In	LoginMeeting.thtml, MeetingAppletInstaller.thtml, MeetingRun.thtml, MeetingRunJava.thtml, MeetingTestJava.thtml, MeetingTestJS.thtml, MeetingTestMSJava.thtml, MeetingTestResult.thtml, MeetingTrouble.thtml
Usage	Optional. Most often used when loading applets or ActiveX controls, which do not need to be processed by the Template Toolkit utilities in any way, but which should be passed through to the page as is. Can also be used to pass text to the page, unfiltered.
Example	The following code displays a label in a meeting page and loads the teleconference information contained in the template variable, regardless of its content or format:

```
<span class="cssSmall"><b>Teleconference:</b></span>
<span class="cssSmall"><% teleconference_info FILTER verbatim %></span>
```

Related Documentation

- [activex_ini](#)
- [applet_ini](#)
- [appstr](#)

FinishLoad()

Use this function if you want to allow users to select from multiple realms in the sign-in page.

Object Type	JavaScript function
Included In	GeneratePin.shtml, LoginPage.shtml, NewPin.shtml, NextToken.shtml, ShowSystemPin.shtml, Defender.shtml, SecondaryLoginPage.shtml, SM-NewPinSelect.shtml, SM-NewPinSystem.shtml, SM-NewUserPin.shtml, SM-NextToken.shtml
Usage	Optional. If you are authenticating users through a eTrust SiteMinder server using client-side certificate authentication, you may want to hide the standard Pulse Connect Secure sign-in page from users. If you are authenticating users through an anonymous authentication server or certificate authentication server and want to bypass the standard Pulse Connect Secure sign-in page, see " AnonymousAuthentication " or " CertificateAuthentication ".
Example	To post data to Pulse Connect Secure without prompting users for any credentials, you may use the following modified version of the FinishLoad() function. Note that this function only bypasses the sign-in page if no errors occur while loading the page:

```
function FinishLoad() {
  recallLastRealmUsed();
  <% IF !LoginPageErrorCode %>
  Login();
  document.frmLogin.submit();
  <% END %>
}
```

Related Documentation

- [GeneratePin.shtml / GeneratePin-mobile-webkit.shtml / GeneratePin-ipad.shtml](#)
- [LoginPage.shtml / LoginPage-mobile-webkit.shtml / LoginPage-ipad.shtml](#)
- [SM-NewPinSelect.shtml / SM-NewPinSelect-mobile-webkit.shtml / SM-NewPinSelect-ipad.shtml](#)
- [SM-NewPinSystem.shtml / SM-NewPinSystem-mobile-webkit.shtml / SM-NewPinSystem-ipad.shtml](#)
- [SM-NewUserPin.shtml](#)
- [SM-NextToken.shtml / SM-NextToken-mobile-webkit.shtml / SM-NextToken-ipad.shtml](#)

focus

Sets the focus on given component on a form.

Object Type	Template variable
Included In	Cancel.shtml, LoginPage.shtml, NewPin.shtml, LoginMeeting.shtml, NextToken.shtml, MeetingRun.shtml, Defender.shtml, SecondaryLoginPage.shtml, SM-PinSelect.shtml, SM- SM-NewPinSystem.shtml, SM-NewUserPin.shtml , SM-NextToken.shtml
Usage	Required
Example	The following code sets the focus (by way of a JavaScript function also called focus()) to the form object indicated by the template variable focus :

```
function onLoad() {  
    grab();  
    document.frmLogin.<% focus %>.focus();  
}
```

Related Documentation	None
------------------------------	------

FriendlyTime

Argument in number of seconds.

Object Type	Template variable
Included In	PasswordChange.shtml, PasswordExpiration.shtml
Usage	Argument for error code 2004, Password Too New.
Example	None
Related Documentation	

- [PageErrorArgs](#)
- [PageErrorCode](#)

frmCasque

HTML form that enables an end-user to initiate the CASQUE RADIUS server.

Object Type	HTML form
Included In	Defender.shtml
Usage	Required
Example	The following code illustrates the use of the frmCasque form. If the plug-in type is the CASQUE RADIUS server plug-in, then the page loads the form:

```
<% IF plugintype == 'CASQUE' %>  
    <form name="frmCasque" method="POST" autocomplete=off>  
        <input type=hidden name="key" value="<% secid_key %>">
```

```
<INPUT type=hidden name="p" value="casqueapp">
<a href="#" onClick="submitFrmCasque(); return false;">
  LAUNCH CASQUE PLAYER
</form>
```

Related Documentation [submitFrmCasque\(\)](#)

frmLogin

HTML form that enables the template page to trigger a call to the **Login()** function and begins the sign in process for a user.

Object Type	HTML Form
Included In	LoginPage.thtml , LoginMeeting.thtml
LoginPage.thtmlUsage	Required
Example	The following code is the form definition statement:

```
<form name="frmLogin" action="login.cgi" method="post" autocomplete="off"
onsubmit="return Login()">
```

Related Documentation [autocomplete](#)

frmNextToken

Next Token HTML form.

Object Type	HTML Form
Included In	NextToken.thtml , SM-NextToken.thtml
Usage	Required
Example	The following code illustrates the suggested form definition:

```
<FORM name="frmNextToken" action="login.cgi" method="POST"
autocomplete=off onsubmit="return SubmitClicked();">
```

Related Documentation [NextToken.thtml](#) / [NextToken-mobile-webkit.thtml](#) / [NextToken-ipad.thtml](#)

frmSelectRoles

Select Roles HTML form

Object Type	HTML Form
Included In	SelectRole.thtml
SelectRole.thtmlUsage	Required
Example	The following code illustrates the suggested form definition:

<FORM name="frmSelectRoles" autocomplete=off>

Related Documentation [SelectRole.shtml](#) / [SelectRole-mobile-webkit.shtml](#) / [SelectRole-ipad.shtml](#)

g_time

The time when the browser first loads the page.

Object Type JavaScript variable

Included In **PleaseWait.shtml**

Usage Optional

Example None

Related Documentation [PleaseWait.shtml](#) / [PleaseWait-mobile-webkit.shtml](#) / [PleaseWait-ipad.shtml](#)

gCancelNewPinMode

Prevents Pulse Connect Secure from cancelling Generate PIN mode while the user is trying to save a PIN or generate a new one.

Object Type JavaScript variable/flag

Included In **GeneratePin.shtml**, **NewPin.shtml**

Usage Required. You must include this flag in the header of **GeneratePin.shtml**. Works together with the JavaScript **DoUnload()** and **SubmitClicked()** functions to call the appropriate CGI on Pulse Connect Secure if the user chooses to close the browser window without cancelling his session first. If you do not include this code, the user may be locked out of Pulse Connect Secure when he closes his browser. In addition to including this JavaScript in your header, you must also call the **functionDoUnload()** function from the body of **GeneratePin.shtml**.

Example You include the following code in the header of the **NewPin.shtml** template. This function saves or cancels a PIN per the user's request. Note that in this example, the **gCancelNewPinMode** is set to false though it was initialized to true when the page loaded.

```
// Save / Cancel a PIN
function SubmitClicked() {
    gCancelNewPinMode = false;
    return true;
}
```

Related Documentation [SubmitClicked\(\)](#)

gCancelNextTokenMode

Prevents Pulse Connect Secure from cancelling Next Token mode while the user is trying to enter his token.

Object Type JavaScript function

Included In	NextToken.thtml
Usage	Required. The header in NextToken.thtml contains two required JavaScript functions, DoUnload() and SubmitClicked() , as well as the gCancelNextTokenMode flag. These work together to call the appropriate CGI on Pulse Connect Secure if the user chooses to close the browser window without cancelling his session first. If you do not include this code, the user may be locked out of Pulse Connect Secure when he closes his browser. In addition to including this JavaScript in your header, you must also call the DoUnload() function from the body of NextToken.thtml .
Example	You must include this flag in the header of NextToken.thtml :

```
var gCancelNextTokenMode = true;
```

Related Documentation [NextToken.thtml / NextToken-mobile-webkit.thtml / NextToken-ipad.thtml](#)

geckoBrowser

	Boolean set to 1 if the browser is gecko-based (Firefox).
Object Type	Template variable
Included In	Logout.thtml, PleaseWait.thtml
Usage	Required
Example	The following code checks to see if the browser is Firefox. If yes, then the page loads the proxy form in an iFrame (inline frame).
	<pre><% IF geckoBrowser %> <IFRAME ID="proxy_frm" src="/dana-na/help/blank.html" WIDTH=0 HEIGHT=0 FRAMEBORDER=0 SCROLLING=NO></IFRAME> <% END %></pre>

Related Documentation [safari](#)

GeneratePin.thtml / GeneratePin-mobile-webkit.thtml / GeneratePin-ipad.thtml

	Enables the user to create a system-generated PIN.
Object Type	Template
Usage	Required
Required Variables	None
Example	When configuring the secid_pinselectmode variable (which sets the user's PIN select mode), note that possible values include:

Table5: PIN usage indicators

Mode	Description
0	User must use the system-generated PIN. He cannot enter his own PIN.
1	User may enter his own PIN or use the system-generated PIN.
2	User must enter his own PIN. He cannot use the system-generated PIN.

GetCookieValue()

Generic helper function that retrieves a value from a cookie, creates a string the length of the value, and populates the string with the value.

Object Type

JavaScript function

Included In

LoginPage.shtml, PleaseWait.shtml

Usage

Use this function if you want to allow users to select from multiple realms in the sign-in page.

sName—Cookie name.

Example

The function definition follows:

```
function GetCookieValue(sName) {
  var s = document.cookie;
  sName += "=";

  // where nv pair starts
  var nStart = s.indexOf(sName);
  if (nStart == -1)
    return "";
  else
    nStart += sName.length;
  // if more values, clip, otherwise just get rest of string
  var nEnd = document.cookie.indexOf(";", nStart);
  if (nEnd == -1)
    s = unescape(s.substring(nStart));
  else
    s = unescape(s.substring(nStart, nEnd));
  return s;
}
```

Related Documentation

- [LoginPage.shtml / LoginPage-mobile-webkit.shtml / LoginPage-ipad.shtml](#)
- [PleaseWait.shtml / PleaseWait-mobile-webkit.shtml / PleaseWait-ipad.shtml](#)

grab()

Checks for browser type and grabs a cookie when a user tries to sign in to a meeting.

Object Type	JavaScript function
Included In	LoginMeeting.html
Usage	Required
Example	The following code shows the grab() function:

```
function grab() {
var f = document.frmLogin;
<% IF ie %>
try {
var result = NeoterisSetup.isValid();
factivex_enabled.value = 1;
if (result == 1) f.power_user.value = 1;
} catch (e) {
factivex_enabled.value = 0;
}
<% END %>
f.java_enabled.value = navigator ? (navigator.javaEnabled() ? 1 : 0) : 0;
}
if (document.cookie.indexOf("DSMTCookie") < 0) {
alert("You must enable cookies through your Web browser in order
to sign into the meeting.");
}
```

Related Documentation [LoginMeeting.html](#)

GraceLoginsRemaining

Number of remaining grace logins.

Object Type	Template variable
Included In	GraceLoginUsed.html
Usage	Optional. Argument for error code 2016.
Example	The following code shows an example of how to display a message for the number of grace logins available to the user:

```
<% IF LoginPageErrorCode == 2016 %>
You have <% LoginPageErrorArgs.GraceLoginsRemaining %> grace logins.
<% END %>
```

Related Documentation

- [GraceLoginUsed.html / GraceLoginUsed-mobile-webkit.html / GraceLoginUsed-ipad.html](#)
- [PageErrorArgs](#)

-
- [PageErrorCode](#)

GraceLoginUsed.thtml / GraceLoginUsed-mobile-webkit.thtml / GraceLoginUsed-ipad.thtml

	Tells the user how many more times he can sign in using his current username and password.
Object Type	Template
Usage	Required
Required Variables	“startPageLink”
Example	None

guest-user-print-page.thtml

	Page for printing single users by the Guest User Account Manager.
Object Type	Template
Usage	Optional (Valid for Pulse Policy Secure)
Required Variables	fieldsArr

guest-user-print-page-batch.thtml

	Page for printing batch users by the Guest User Account Manager .
Object Type	Template
Usage	Optional (Valid for Pulse Policy Secure)
Required Variables	UsersPrintInfo

guest-user-email-page.thtml

	Page for emailing user details by the Guest User Account Manager.
Object Type	Template
Usage	Optional (Valid for Pulse Policy Secure)
Required Variables	fieldsArr

HC_REMED_POLICIES_CHECK

	Default text to suggest that policies are being evaluated.
Object Type	Template variable
Included In	Remediate.thtml
Usage	Optional
Example	The following code checks to see if the page is loading. If yes, then the page displays several messages, including the default text to suggest that Pulse Connect Secure is evaluating policies:

```
<% IF showLoading %>
<% loading %>
<% pleasewait %>
<% HC_REMED_POLICIES_CHECK %>
<% END %>
```

Related Documentation

- [HC_REMED_SHOW_ADV_PREFS](#)
- [Remediate.shtml](#) / [Remediate-mobile-webkit.shtml](#) / [Remediate-ipad.shtml](#)

HC_REMED_SHOW_ADV_PREFS

	Default text instructing how to re-enable remediation.
Object Type	Template variable
Included In	Remediate.shtml
Usage	Optional
Example	The following code illustrates how to call the variable to display instructions to re-enable remediation:

```
<% IF showRemedOption %>
<input name="btnHideRemed" type="button"
value="<% textRemedOption %>" onClick="HideRemed();
<% disabled %>">
<% HC_REMED_SHOW_ADV_PREFS %>
<% END %>
```

Related Documentation

- [HC_REMED_SHOW_ADV_PREFS](#)
- [Remediate.shtml](#) / [Remediate-mobile-webkit.shtml](#) / [Remediate-ipad.shtml](#)

hclnAcTimeout

	Specifies the Host Checker login inactivity in minutes.
Object Type	Template variable
Included In	LoginPage.shtml
Usage	Required
Example	The following code sets the timeout interval to 60000 milliseconds times the smaller value of cclnAcTimeout or hclnAcTimeout .

```
<% IF hclnAcTimeout > cclnAcTimeout %>
window.setTimeout("deletepreauth();", 60000 * <% cclnAcTimeout %>);
<% ELSE %>
```

```
window.setTimeout("deletepreauth()", 60000 * <% hclnAcTimeout %>);
<% END %>
```

Related Documentation [LoginPage.thtml](#) / [LoginPage-mobile-webkit.thtml](#) / [LoginPage-ipad.thtml](#)

hcRunning

Specifies whether or not Host Checker is currently running.

Object Type Template variable

Included In **LoginPage.thtml**

Usage Required

Example The following code sets the timeout interval based on whether Host Checker and Cache Cleaner are both running or only Host Checker.

```
<% IF hcRunning && ccRunning %>
<% IF hclnAcTimeout > cclnAcTimeout %>
  window.setTimeout("deletepreauth()", 60000 * <% cclnAcTimeout %>);
<% ELSE %>
  window.setTimeout("deletepreauth()", 60000 * <% hclnAcTimeout %>);
<% END %>
<% ELSIF hcRunning %>
  window.setTimeout("deletepreauth()", 60000 * <% hclnAcTimeout %>);
<% END %>
```

Related Documentation [LoginPage.thtml](#) / [LoginPage-mobile-webkit.thtml](#) / [LoginPage-ipad.thtml](#)

heading

Contains default text for the different remediation headings.

Object Type Template variable

Included In **Remediate.thtml**

Usage Required

Example The following code shows a heading and a message if the **showHeading** variable is true:

```
<% IF showHeading %>
<% heading %>
<% msg FILTER verbatim %>
<% END %>
```

Related Documentation [showHeading](#)

help

String to use as the button name.

Object Type	Template variable
Default value	"Help"
Included In	LoginPage.thtml, SecondaryLoginPage.thtml
LoginPage.thtmlUsage	Optional
Example	The string "Help" (the default value of the help variable) is displayed on the button. <pre><input type='submit' name='help' value="<% help %>"</pre>
Related Documentation	Adding Custom Help Files

help_on

	Flag that determines whether the help button should be displayed.
Object Type	Template variable
Default value	0 (integer)
Included In	Loginpage.thtml, SecondaryLoginPage.thtml
Loginpage.thtmlUsage	Optional
Example	The following code displays a help button if the help_on variable is true. <pre><% IF help_on %> <input type='submit' name='help' value="<% help %>" onclick='window.open("welcome.cgi?p=help", "wndHelp", "height=400,width=500,resizeable=yes,scrollbars=yes"); return false;'</pre>
Related Documentation	Adding Custom Help Files

HideRemed()

	Submits the hidden form when the end-user clicks the Hide Remediation button.
Object Type	JavaScript function
Included In	Remediate.thtml
Usage	Required
Example	The following code calls the HideRemed() function: <pre><% IF showRemedOption %> <input name="btnHideRemed" type="button" value="<% textRemedOption %>" onClick="HideRemed(); <% disabled %>"> <% HC_REMED_SHOW_ADV_PREFS %> <% END %></pre>
Related Documentation	Remediate.thtml / Remediate-mobile-webkit.thtml / Remediate-ipad.thtml

Home

	References the top-level directory in the zip file.
Object Type	Template variable
Included In	All templates
Usage	You can use this variable or the double-dot (..) convention to reference the top-level directory.
Example	Both of the following references are valid:

```
<% Home %>/images/logo.gif  
../images/logo.gif
```

Related Documentation [Using the Template Toolkit](#)

I18N_UAC_DOWNLOAD_AND_INSTALL_64_BIT

	Contains the link to download and install Odyssey Access Client on 64-bit Windows manually.
Object Type	Template variable
Included In	AgentInstall.thtml
Usage	Optional (Required for Pulse Policy Secure).
Example	Both of the following references are valid:

```
<%IF I18N_UAC_DOWNLOAD_AND_INSTALL_64_BIT %>  
<span><% I18N_UAC_DOWNLOAD_AND_INSTALL_64_BIT FILTER verbatim %></span>  
<%END%>
```

Related Documentation

ie

	Boolean variable that indicates whether or not the browser is the Microsoft Internet Explorer.
Object Type	Template variable
Included In	LoginMeeting.thtml
Usage	Required
Example	The following example checks to see if the browser is Internet Explorer, and if it is, enables ActiveX.

```
<% IF ie %>  
<OBJECT  
  <% activex_ini FILTER verbatim%>  
</OBJECT>  
<% END %>
```

Related Documentation [ie_winxp](#)

IEonWindows

	Boolean indicating if the browser is Internet Explorer and the OS is Windows.
Object Type	Template variable
Included In	AgentInstall.html, AgentlessInstall.html
Usage	Optional. The variable value is 1 if the browser is Internet Explorer and the OS is Windows.
Example	The following example checks to see if the browser is Internet Explorer, and if it is, enables ActiveX.

```
<% IF IEonWindows %>
```

Related Documentation

ie_winxp

	Boolean indicating if the browser is Internet Explorer and the OS is Windows XP.
Object Type	Template variable
Included In	PleaseWait.html
Usage	Optional. The variable value is 1 if the browser is Internet Explorer and the OS is Windows XP.
Example	

```
<% IF ie_winxp %>
```

Related Documentation [PleaseWait.html](#) / [PleaseWait-mobile-webkit.html](#) / [PleaseWait-ipad.html](#)

install

	Boolean indicating whether or not the meeting applet should be installed.
Object Type	Template variable
Included In	MeetingAppletInstaller.html, MeetingRun.html
Usage	Required
Example	

```
<% IF install %>
```

Related Documentation

- [MeetingAppletInstaller.html](#)
- [MeetingRun.html](#)

instanceld

	Contains the ID of the current meeting instance.
Object Type	Template variable
Included In	MeetingAppletInstaller.shtml
Usage	Required
Example	<pre><PARAM NAME="Parameter0" VALUE="meeting_id=<% instanceld %>; user_name=<%username%>;cert_md5=<%cert_md5%>; ncp_read_timeout=<%ncp_read_timeout%>"></pre>
Related Documentation	MeetingAppletInstaller.shtml

instructions

	Instructional text displayed to the user on the login page.
Object Type	Template variable
Default Value	"Please sign in to begin your secure session"
Included In	LoginPage.shtml
Usage	Optional
Example	<pre><% instructions FILTER verbatim %></pre>
Related Documentation	None

isLinux

	Boolean indicating whether or not the end-user's operating system is a Linux system.
Object Type	Template variable
Included In	AgentInstall.shtml, PleaseWait.shtml, Logout.shtml
Usage	Optional
Example	<pre><% isLinux %></pre>
Related Documentation	PleaseWait.shtml / PleaseWait-mobile-webkit.shtml / PleaseWait-ipad.shtml

isMac

	Boolean indicating whether or not the end-user's operating system is a Macintosh system.
Object Type	Template variable
Included In	PleaseWait.shtml
Usage	Required

Example

```
<% isMAC %>
```

Related Documentation [PleaseWait.shtml](#) / [PleaseWait-mobile-webkit.shtml](#) / [PleaseWait-ipad.shtml](#)

isSAMEnabled

Boolean indicating whether or not JSAM application is enabled and running.

Object Type Template variable

Included In [Logout.shtml](#)

Usage Optional. Allows the end-user to close the JSAM window.

Example The following code checks to see if JSAM is enabled and running. If it is, then the system uses the JavaScript below to close the JSAM window:

```
<% IF isSAMEnabled %>
<!-- terminate JSAM -->
<SCRIPT LANGUAGE="JavaScript">
var win = "<%jsamwindowid %>";
win = win.replace(/[^0-9a-zA-Z]/g,"a");
w = window.open("", win, "height=1,width=1");
if (w) {
  w.bSessionClose = true;
  w.close();
}
</SCRIPT>
<% END %>
```

Related Documentation [Logout.shtml](#) / [Logout-mobile-webkit.shtml](#) / [Logout-ipad.shtml](#)

isUAC

Boolean indicating whether the server is for Pulse Policy Secure.

Object Type Template variable

Included In [PleaseWait.shtml](#), [Remediate.shtml](#)

Usage Required.

Example

```
<% isUAC %>
```

Related Documentation [Logout.shtml](#) / [Logout-mobile-webkit.shtml](#) / [Logout-ipad.shtml](#)

keyboard.js

Enables keyboard types of functions for a mouseless environment.

Object Type JavaScript file

Included In	LoginPage.thtml (Kiosk)
Usage	Required in Kiosk zip . This template displays an error if the user signs out, if the user's session times out, or if Pulse Connect Secure uninstalls Host Checker or Cache Cleaner from the user's system. For more detailed information, see " Logout.thtml / Logout-mobile-webkit.thtml / Logout-ipad.thtml ". You must always include this template in your Kiosk.zip file.
Example	None
Related Documentation	LoginPage.thtml / LoginPage-mobile-webkit.thtml / LoginPage-ipad.thtml

labelInstructions

	Instructions tag, indicating a location for instructions to appear on the page.
Object Type	Template variable
Included In	Remediate.thtml
Usage	Optional
Example	The following code loops through a list of failed policies, displays each policy name and remediation instructions:

```
<% FOREACH failedPolicy = listFailedPolicies %>
  <% loop.count %>
  <% failedPolicy.name %>
  <% labelInstructions %>
  <% failedPolicy.remediation FILTER verbatim%>
<% END %>
```

Related Documentation	Remediate.thtml / Remediate-mobile-webkit.thtml / Remediate-ipad.thtml
------------------------------	----------------------------------------------------------------------------------------

listFailedPolicies

	Template Toolkit hash value of failed policies (keys: name and remediation).
Object Type	Template variable
Included In	Remediate.thtml , Remediate-ppc.thtml , Remediate-ipad.thtml , Remediate-mobile-webkit.thtml
Usage	Required
Example	The following example checks to see if the page should show failed policies and, if so, loops through the list of failed policies (the hash list listFailedPolicies) and displays the name of each policy along with label instructions and remediation instructions:

```
<% IF showPolicies %>
<% FOREACH failedPolicy = listFailedPolicies %>
  <% loop.count %>
  <% failedPolicy.name %>
  <% labelInstructions %>:
  <% failedPolicy.remediation FILTER verbatim%>
```

```
<% END %>
```

Related Documentation [Remediate.shtml](#) / [Remediate-mobile-webkit.shtml](#) / [Remediate-ipad.shtml](#)

loading

Contains default text to be printed when ActiveX/applet is loading.

Object Type Template variable

Included In **Remediate.shtml**

Usage Optional

Example The following code tests to see if the page is still loading and displays messages to the end-user:

```
<% IF showLoading %>  
<% loading %>  
<% pleasewait %>
```

Related Documentation [Remediate.shtml](#) / [Remediate-mobile-webkit.shtml](#) / [Remediate-ipad.shtml](#)

locale

Returns a locale that applies to an applet the page is attempting to upload.

Object Type Template variable

Included In **Logout.shtml, MeetingAppletInstaller.shtml, MeetingTrouble.shtml**

Logout.shtmlUsage Required

Example The following code sets the HTML form locale based on the template variable:

```
<PARAM NAME="locale" VALUE="<% locale %>">
```

Related Documentation [safari](#)

loginmode

Passes the user's mode to the server using a hidden value.

Object Type HTML form field

Included In **GeneratePin.shtml, NewPin.shtml, NextToken.shtml, Defender.shtml, SM- NewPinSelect.shtml, SM-NewPinSystem.shtml, SM- SM-NewUserPin.shtml, SM- NextToken.shtml**

Usage Required. Typically names a template directive that returns the user's login mode from a server.

Example The following code assigns the secure ID loginmode to the form field:

```
<INPUT type=hidden name="loginmode" value="<% secid_loginmode %>">
```

Related Documentation [secid_loginmode](#)

Login()

Verifies that realms exist, remembers currently selected authentication realms, and establishes the user's time zone.

Object Type JavaScript function

Included In **LoginPage.html**

Usage Required. This function includes two actions:

- **Remember the selected authentication realm**—Use this action within the function if you want to allow users to select from multiple realms in the sign-in page.
- **Establish the user's time zone**—Required. The function uses the **tz_offset** (time zone offset) variable and **getTimezoneOffset()** function to determine the user's time zone.

Example

```
function Login() {  
  // Remember currently selected auth realm  
  if (document.frmLogin.realm != null &&  
      document.frmLogin.realm.type == "select-one") {  
    SetLastRealm(  
      document.frmLogin.realm.options  
      [document.frmLogin.realm.selectedIndex].text);  
  }  
  if (document.frmLogin.tz_offset != null) {  
    var wdate = new Date (95, 12, 1);  
    var sdate = new Date (95, 6, 1);  
    var winter = (-1) * wdate.getTimezoneOffset();  
    var summer = (-1) * sdate.getTimezoneOffset();  
    document.frmLogin.tz_offset.value = winter < summer ? winter : summer;  
  }  
  return true;  
}
```

Related Documentation [LoginPage.html](#) / [LoginPage-mobile-webkit.html](#) / [LoginPage-ipad.html](#)

LoginMeeting.html

Displays the user credentials form to enable a user to sign in to a meeting.

Object Type Template file

Usage Required in the uploaded **Meeting.zip**. This template provides a page on which the end-user can enter credentials to gain access to a meeting.

The form action must specify **login_meeting.cgi**. The template includes the JavaScript function **grab()**, which checks to see which type of browser the user is running, and if ActiveX, Java, and cookies are enabled, depending on the browser type.

You can modify the alert contained in the required JavaScript. The alert is:

```
alert("You must enable cookies through your Web browser in order to sign into the meeting.");
```

Required Variables

- "activex_ini"
- "cboxuser"
- "focus"
- "Home"
- "ie"
- "mid"
- "mid_param"
- "occurrence"
- "username"
- "Remember my name" (not required, but recommended)
- "Security token" (not required, but recommended)

Example None

Related Documentation [grab\(\)](#)

LoginPage.shtml / LoginPage-mobile-webkit.shtml / LoginPage-ipad.shtml

Calls the RSA Soft ID client, enabling the user to sign into Pulse Connect Secure using Soft ID authentication.

Object Type Template

Usage Required. You must always include this template in your zip file. You must always include **LoginPage.shtml** in your zip file, even if you are using an authentication server that does not require a username or password. This is the standard Pulse Connect Secure sign-in page that collects the user's name, password, and authentication realm, and displays an error if authentication fails. For information about hiding this page from users authenticated against a(n):

- Anonymous server, see "[AnonymousAuthentication](#)"
- Certificate server, see "[CertificateAuthentication](#)"
- eTrust SiteMinder server using client-side certificate authentication, see "[Login\(\)](#)"

LoginPage.shtml contains the following suggested form definition. In this form definition, most elements are required:

```
<form name="frmLogin" action="login.cgi" method="post" autocomplete="off"
```

`onsubmit="return Login()">`

Required Variables ["LoginPageErrorMessage"](#)

Example None

Related Documentation

- [Login\(\)](#)

LoginPageErrorArgs

Contains the arguments for an error or message.

Object Type Template variable

Included In **GraceLoginUsed.thtml, PasswordExpiration.thtm**

Usage Optional

Example In the following example, **GraceLoginsRemaining** is the argument to error code 2016. The directive that contains **LoginPageErrorArgs.GraceLoginsRemaining** returns the number of remaining login attempts Pulse Connect Secure allows a user.

```
<% IF LoginPageErrorCode == 2016 %>
  Your password has expired. A grace login was used.
  You have <% LoginPageErrorArgs.GraceLoginsRemaining %> grace logins
  remaining.
<% ELSE %>
  <% LoginPageErrorMessage %>
<% END %>
```

Related Documentation [LoginPageErrorCode](#)

LoginPageErrorCode

Codes indicating errors that you can display to users.

Object Type Template variable

Included In **ExceededConcurrent.thtml, GraceLoginUsed.thtml, LoginPage.thtml,**
PasswordExpiration.thtml, SSL.thtml

Usage Optional. Although this template variable is optional, you should leave intact the template variables that are already included in the various templates. You can modify many of the error messages, but you must leave the codes as defined. Pulse Connect Secure returns these codes on various conditions. Some may be no more than confirmation messages. Others are conditions to which the user needs to respond.

Example The following example shows how you can use the Template Toolkit conditional statements to define two alternate error conditions. In this example, the different error codes display appropriate HTML pages. If Pulse Connect Secure does not return either of the error codes, the user never sees the two error pages.

```

<head>
<% IF LoginPageErrorCode == 1021%>
<meta http-equiv=" refresh" content=" 0;url=AcctDisabledError.html" >

```

Table 6 lists possible errors that the user may see on this page as well as the corresponding text returned by the **LoginPageErrorMessage** variable:

Table6: Login Error Messages, Codes, and Related Notes

Error message	Error Code	Comments
Invalid Username or Password	1002	Wrong user credentials.
This login requires a digital certificate.	1003	Client certificate not passed for authentication.
The digital certificate is invalid.	1004	
Only admins are permitted to login.	1006	
Logins are not permitted from this IP address.	1007	The source IP failed the IP Restriction check.
Logins are not permitted using this browser	1008	The User Agent header failed the User Agent Restriction check.
There are no auth servers defined for this user.	1009	Displayed if a realm is not sent to Pulse Connect Secure. This error is only displayed if a coding error exists in LoginPage.thtml.
Exceeded the number of concurrent users.	1010	Exceeded the number of concurrent users based on the user license plus a 10% allowance.
The IP has been blocked due too many concurrent sign-in attempts.	1011	
The password is too short.	1012	The password failed the Password Restriction check. When using eTrust auto-signon, the end-user should never see this event.
Not using SSLv3	1015	This site requires Secure Sockets Layer (SSL) protocol version 3.
This site requires Strong ciphers.	1016	

Too many users have logged in	1017	The number of concurrent users signed in to the system has exceeded the system limit but is still within the 10% allowance.
Sign on for administrators is disabled. You can try again in a few minutes.	1018	Displayed when a super administrator is signed in and Pulse Connect Secure is in recovery mode.
Your New PIN has been saved. Please make sure to remember it.	1019	Only used with ACE authentication.
Password Change Success	1020	Used with the password management feature.
Account Disabled	1021	Used with the password management feature.
Account Locked Out	1022	Used with the password management feature.
Account Expired	1023	Used with the password management feature.
Pulse Connect Secure session has been terminated.	1024	End-user explicitly logged off.
You do not have permission to login.	1025	This event can be triggered when the client machine does not pass any realm restrictions (hence, no realm), or does not satisfy any end-point security policies (HC/CC).
This realm has reached the max session limit.	1027	Exceeded maximum number of users for this authentication realm.
Unlicensed Feature	1028	License required to use this access method is missing.
Access denied. Please try signing in again using a host name (for example, https://department.yourcompany) instead of an IP address (such as http://10.11.12.13)	1029	Only used with eTrust SiteMinder authentication.
You do not have the correct privileges to access the system. Please contact your administrator for more information.	1030	The end-user is not allowed to sign in because the role mapping rules of the authentication realm do not map the end-user to any role.
Certificate has been revoked.	1032	The client certificate has been revoked.

Cannot determine the status of your certificate.	1033	Pulse Connect Secure encountered a failure while performing an OCSP check.
You must use your source site's Intersite Transfer Service to access this resource.	1034	Pulse Connect Secure cannot access any SAML assertion while performing SAML SSO. The user is prompted to access the source site's Intersite Transfer Service to continue.
The number of concurrent Advanced Endpoint Defense (Malware Protection) users signed into the system has exceeded the system limit.	1035	This event will be triggered only when the feature is turned ON. The system limit is built-in 25 user license plus any purchased Advanced Endpoint Defense user license.

Related Documentation [LoginPageErrorMessage](#)

LoginPageErrorMessage

Pulse Connect Secure error message text that you can display to users.

Object Type

Template variable

Included In

ExceededConcurrent-ppc.html, ExceededConcurrent-mobile-webkit.html, ExceededConcurrent-ipad.html, ExceededConcurrent.html, GraceLoginUsed-ppc.html, GraceLoginUsed-mobile-webkit.html, GraceLoginUsed-ipad.html, GraceLoginUsed.html, LoginPage-ppc.html, LoginPage-ipad.html, LoginPage-mobile-webkit.html, LoginPage.html, Logout-ppc.html, Logout-ipad.html, Logout-mobile-webkit.html, Logout.html, PleaseWait.html, PasswordExpiration-ppc.html, PasswordExpiration-ipad.html, PasswordExpiration-mobile-webkit.html, PasswordExpiration.html, SecondaryLoginPage-ipad.html, SecondaryLoginPage-mobile-webkit.html, SecondaryLoginPage-ppc.html, SecondaryLoginPage.html, SSL-ppc.html, SSL-ipad.html, SSL-mobile-webkit.html, SSL.html

Example

The following code prompts the user to re-enter credentials or to sign in at another time. Note that the messages can be whatever you want them to be:

```
<% IF LoginPageErrorCode == 1002 %>
  <b> Your username or password is incorrect. Please reenter your credentials.
</b>
<%ELSIF LoginPageErrorCode == 1006 %>
  <b> The system is undergoing maintenance. Only administrators are allowed
to sign in at this time.</b>
<% END %>
```

The following code tests the pleasewait condition and, if true, displays the Login page error message that occurs as a result:

```
if (document.getElementById('pleasewait1')) {
  document.getElementById('pleasewait1').innerHTML =
    "<% LoginPageErrorMessage FILTER verbatim%>";
}
```

Related Documentation [LoginPageErrorCode](#)

LoginPage-new-ux.thtml

	Provides the user with user credentials form for logging in.
Object Type	Template
Usage	Required. Used for displaying Login page.
Required Variables	mobile_browser_login hcRunning ccRunning ccInAcTimeout hcInAcTimeout setcookies signinId AnonymousAuthentication CertificateAuthentication SAMLAuthentication prompts RealmList signin isSdpController

login_required

	Boolean indicating whether or not login to the meeting session is required.
Object Type	Template variable
Included In	MeetingRunJava.thtml
Usage	Required
Example	The following example checks to see if login is required. If yes, then the page redirects the user back to the meeting sign-in page:

```
<% IF login_required %>
  redirectParent();
<% END %>
```

Related Documentation [MeetingRunJava.shtml](#)

[Logout.shtml](#) / [Logout-mobile-webkit.shtml](#) / [Logout-ipad.shtml](#)

Displays an error if the user signs out, if the user's session times out, or if Pulse Connect Secure uninstalls Host Checker or Cache Cleaner from the user's system.

Object Type Template

Usage Required. You must always include **Logout.shtml** in your zip file.

This template contains JavaScript for detecting, stopping, installing, and uninstalling the Host Checker and Cache Cleaner components, images and text to display to users while Pulse Connect Secure performs these actions, and JavaScript that closes and opens the J-SAM window.

This template contains JavaScript for detecting, stopping, installing, and uninstalling Host Checker components, as well as images and text to display to users while the performs these actions.

This template also includes variables telling users to wait while components install, a link that users can use to sign back into Pulse Connect Secure, and error message variables.

When configuring the **LoginPageErrorMessage** variable included in this template, note that the text that Pulse Connect Secure displays to users corresponds with a code returned by **LoginPageErrorCode**. You cannot change this text on Pulse Connect Secure, but you can include code in your template to display different text based on the error code returned by the **LoginPageErrorCode** variable. For example:

```
<% IF LoginPageErrorCode == 1001 %>
  <b> Your secure gateway session has ended due to inactivity.</b>
<% END %>
```

Described below are the possible errors that the **LoginPageErrorCode** may return as well as the corresponding text returned by the **LoginPageErrorMessage** variable:

Table7: Additional Login Error Messages and Codes

Error Message	Error Code
Maximum Session Timeout Reached.	1000
Idle Time Out.	1001

For detailed information about the JavaScript and variables described here, see comments in the template.

Required Variables

- "delivery_mode"
- "geckoBrowser"

- "locale"
- "LoginPageErrorMessage"
- "nc_logging"
- "Netscape4xx"
- "pleasewaitLogoutJSCode"
- "PleaseWaitObjectCC"
- "pleasewaitWin32"
- "safari"
- "Setupappversion"
- "UninstallCC"
- delivery_mode
- LoginPageErrorMessage
- signinAgain
- signinAgainUrl
- pleasewaitMac
- setup_ini
- fallback_setup_applet_vitals

Example None

meeting_ini

Contains an Active-X control.

Object Type Template variable

Usage Required if running Internet Explorer on a Windows system. This variable contains the name of the Active-X control that must be loaded to run Pulse Secure Collaboration in the browser.

Required Variables

- activex_ini
- FILTER verbatim

Example <%meeting_ini FILTER verbatim%>

Related Documentation

- [MeetingRun.shtml](#)
- [MeetingSelect.shtml](#)

MeetingAppletInstaller.shtml

Enables Pulse Connect Secure to display list of and install meeting-related applets.

Object Type Template

Usage Required. The page displays troubleshooting messages if the applets do not install. You cannot change the text, as Pulse Connect Secure generates the troubleshooting messages. You can, however, add links if you want to provide additional meeting-related applets to your users. These applets are not covered by the certificate validation Pulse Connect Secure performs on Pulse Secure-supplied applets.

Required Variables

- ["applet_ini"](#)
- ["cert_md5"](#)
- ["install"](#)
- ["instanceld"](#)
- ["locale"](#)
- ["ncp_read_timeout"](#)
- ["username"](#)
- ["startSessionReDir"](#)
- [text](#)
- ["troubleshooting"](#)
- [uninstallReDir](#)

Example None

Related Documentation

- [MeetingRun.shtml](#)
- [MeetingSelect.shtml](#)

MeetingRun.shtml

Tests Sun JVM compatibility.

Object Type Template

Usage Required

Required Variables

- [errors](#)
- ["expired"](#)
- ["Home"](#)
- ["message"](#)
- ["mid"](#)
- ["occurrence"](#)
- [title](#)
- ["type"](#)
- ["warnings"](#)

Example None
Related Documentation [LoginMeeting.shtml](#)

MeetingRunJava.shtml

Launches the Meeting Java client.

Object Type Template
Usage Required
Required Variables

- "appstr"
- href
- "login_required"
- url

Example None
Related Documentation [LoginMeeting.shtml](#)

meetings

Returns an array of meetings that the **MeetingSelect.shtml** template uses as a way to cycle through logins to active meetings.

Object Type Template variable
Included In **MeetingSelect.shtml**
Usage Required. You should not modify or delete this variable.
Example The following code returns a list of instantiated meetings:

```
<% IF rows %>
<% FOREACH meeting = meetings %>
<input type="button" value=" &gt; " onclick=
"self.location='/dana-na/meeting/<%signin%>login_meeting.cgi?
mid=<%mid%>&occurrence=<%meeting.3%><%params%>'">
<%started%>
<%meeting.0%>
<%meeting.1%>
<%meeting.2%>
<% END %>
<%END%>
```

Related Documentation [rows](#)

MeetingSelect.shtml

Enables user to select recurring meeting occurrences.

Object Type	Template
Usage	Required
Required Variables	<ul style="list-style-type: none">• “meetings”• “mid”• “params”• “rows”• “started”• text• title
Example	None
Related Documentation	LoginMeeting.shtml

MeetingTestJava.shtml

	Tests Sun JVM compatibility.
Object Type	Template
Usage	Required
Required Variables	<ul style="list-style-type: none">• “mid”• “mid_param”• “testjava_ini”
Example	None
Related Documentation	LoginMeeting.shtml

MeetingTestJS.shtml

	This is the first page in the platform compatibility checker sequence. Tests JavaScript compatibility.
Object Type	Template
Usage	Required. If the end-user’s browser does not support JavaScript, or if they need to enable JavaScript functionality in their browser, this template displays appropriate messages depending on the browser type and the problem, and offers hints to address the problem.
Required Variables	<ul style="list-style-type: none">• “activex_ini”• “mid”• “mid_param”

-
- "win32"

Example None

Related Documentation [LoginMeeting.shtml](#)

MeetingTestMSJava.shtml

Tests Microsoft JVM compatibility.

Object Type Template

Usage Required

Required Variables

- "applet_ini"
- "mid"
- "mid_param"

Example None

Related Documentation [LoginMeeting.shtml](#)

MeetingTestResult.shtml

Displays Meeting test results.

Object Type Template

Usage Required

Required Variables

- "check"
- "message"
- "remedy1"
- "remedy2"

Example None

Related Documentation

- [MeetingTestMSJava.shtml](#)
- [MeetingTestJava.shtml](#)
- [MeetingTestJS.shtml](#)
- [MeetingTestResult.shtml](#)

MeetingTrouble.shtml

Template Page for the Meeting Test Result page.

Object Type Template

Usage Required

Required Variables

- "Home"
- "locale"
- "mid"
- "midStr"
- "occurrence"
- "startSessionReDir"
- "type"
- "uninstall_action"
- "uninstallReDir"

Example

None

Related Documentation

[MeetingTestResult.shtml](#)

message

Returns a message based on a server response.

Object Type

Template variable

Included In

MeetingRun.shtml, MeetingTestResult.shtml, SecondaryLoginPage.shtml

Usage

Optional

Example

```
<% IF message %>
  <% message %>
<% END %>
```

Related Documentation

- [MeetingRun.shtml](#)
- [MeetingTestResult.shtml](#)
- [SecondaryLoginPage.shtml / SecondaryLoginPage-mobile-webkit.shtml / SecondaryLoginPage-ipad.shtml](#)

mid

Contains the meeting ID.

Object Type

Template variable

Included In

LoginMeeting.shtml, MeetingRun.shtml, MeetingTestJava.shtml, MeetingTestJS.shtml, MeetingTestMSJava.shtml, MeetingTrouble.shtml, PasswordChange.shtml

Usage

Required

Example

The following code assigns the meeting ID to a form input object:

```
<input type="hidden" name="mid" value="<% mid %>">
```

Related Documentation

- [LoginMeeting.shtml](#)
- [MeetingRun.shtml](#)
- [MeetingTestJava.shtml](#)
- [MeetingTestJS.shtml](#)
- [MeetingTestMSJava.shtml](#)
- [MeetingTrouble.shtml](#)
- [PasswordChange.shtml / PasswordChange-mobile-webkit.shtml / PasswordChange-ipad.shtml](#)

mid_param

	Contains the URL parameters for the platform compatibility checker page.
Object Type	Template variable
Included In	LoginMeeting.shtml, MeetingTestJava.shtml, MeetingTestJS.shtml, MeetingTestMSJava.shtml
Usage	Required
Example	The following code passes the meeting ID URL to the JavaScript compatibility CGI:

```
<a href="/dana-na/meeting/<% signin %>meeting_testjs.cgi<% mid_param %>">  
Check Meeting Compatibility
```

Related Documentation [mid](#)

midStr

	Contains the URL parameter for the meeting.
Object Type	Template variable
Included In	MeetingTrouble.shtml
Usage	Required
Example	The following line shows how the midStr variable is appended to a CGI URL, to send the user to the correct compatibility checker for the current meeting.

```
“Use the <a href="/dana-na/meeting/meeting_testjs.cgi<% midStr %>">  
compatibility checker</a> to see if you need to upgrade any of your  
components.”
```

Related Documentation [MeetingTrouble.shtml](#)

MinimumPasswordLength

	Argument for error code 2006. The argument is the minimum length of password required.
Object Type	Template variable
Included In	PasswordChange.thtml
Usage	Optional
Example	None
Related Documentation	<ul style="list-style-type: none"> LoginPageErrorCode PasswordChange.thtml / PasswordChange-mobile-webkit.thtml / PasswordChange-ipad.thtml

msg

	Contains any extra remediation related default instructions.
Object Type	Template variable
Included In	Remediate.thtml
Usage	Optional
Example	<p>The following code checks to see if a heading is to be shown. If yes, then the page displays the heading, followed by any remediation message that might be available.</p> <pre><% IF showHeading %> <% heading %> <% msg FILTER verbatim %> <% END %></pre>
Related Documentation	Remediate.thtml / Remediate-mobile-webkit.thtml / Remediate-ipad.thtml

nc_logging

	Initiates Network Connect logging.
Object Type	Template variable
Included In	Logout.thtml
Usage	Optional. Available when the user signs out of a session from a Macintosh Safari browser. Network Connect logging is initiated and the log captured.
Example	None
Related Documentation	safari

ncp_read_timeout

Determines the read timeout value over the Network Control Protocol.

Object Type	Template variable
Included In	MeetingAppletInstaller.thtml
Usage	Required. Passed to MeetingAppletInstaller.thtml when it installs a meeting applet.
Example	

```
<% IF install %>
<APPLET
  <% applet_ini FILTER verbatim %>
  <PARAM NAME="Parameter0"
VALUE="meeting_id=<%instanceId%>;user_name=<%username%>;
cert_md5=<%cert_md5%>;ncp_read_timeout=<%ncp_read_timeout%>">
  <PARAM NAME="StartSessionReDir" VALUE="<%startSessionReDir%>">
  <PARAM NAME="UninstallReDir" VALUE="<%uninstallReDir%>">
  <PARAM NAME="locale" VALUE="<%locale%>">
</APPLET>
```

Related Documentation [MeetingAppletInstaller.thtml](#)

netacekey

Passes the user's eTrust ACE key to the server using a hidden value.

Object Type	HTML form field
Included In	SM-NewUserPin.thtml, SM-NewPinSelect.thtml, SM-NextToken.thtml
Usage	Required. The form assigns the netesecid_key value to netacekey .
Example	The following example shows the use of netacekey in the INPUT statement:

```
<INPUT type=hidden name="netacekey" value="<% netesecid_key %>">
```

Related Documentation [netesecid_key](#)

netesecid_key

Contains the user's eTrust ACE key.

Object Type	Template variable
Included In	SM-NewPinSelect.thtml, SM-NextToken.thtml, SM-NewPinSystem.thtml, SM-NewUserPin.thtml
Usage	Required. The value is passed by the page to the server.
Example	The following example shows the use of netesecid_key in the INPUT statement:

```
<INPUT type=hidden name="neteacekey" value="<% netesecid_key %>">
```

Related Documentation [neteacekey](#)

netesecid_loginmode

Returns a value of Next Token mode or New PIN mode.

Object Type	Template variable
Included In	SM-NewPinSelect.thtml, SM- NextToken.thtml, SM-NewPinSystem.thtml, SM- SM-NewUserPin.thtml
Usage	Required. You must send this value to the server, but if you do not want to expose it to the user, you may send it as a hidden form field.
Example	The following hidden INPUT statement sets the login mode based on the netesecid_loginmode variable.

```
<INPUT type=hidden name="loginmode" value="<% netesecid_loginmode %>">
```

Related Documentation [loginmode](#)

netesecid_pinerr

Error code and corresponding message that informs the user that he mis-entered his PIN.

Object Type	Template variable
Included In	SM-NewUserPin.thtml
Usage	Optional
Example	The following code checks the value of netesecid_pinerr , a boolean, to see if the PIN the user enters matches the PIN on the authentication server:

```
<% IF netesecid_pinerr == 0 %>
<IMG border='0' src='imgs/custom-logininfo.gif' alt='Info' width='21' height='20'>
New PIN Assignment
<% ELSIF netesecid_pinerr == 1 %>
The Two PINs Entered Do Not Match
<% END %>
```

Related Documentation [SM-NewUserPin.thtml](#)

netesecid_realm

The user's realm.

Object Type	Template variable
--------------------	-------------------

Included In	SM-NewPinSelect.shtml, SM- NextToken.shtml, SM-NewPinSystem.shtml, SM-NewUserPin.shtml
Usage	Optional
Example	The following code shows the configuration of the page based on various server-provided values:

```
<FORM name="frmNewPin" action="login.cgi" method="POST"
autocomplete=off>
<INPUT type=hidden name="loginmode" value="<% netesecid_loginmode %>">
<INPUT type=hidden name="realm" value="<% netesecid_realm %>">
<INPUT type=hidden name="neteacekey" value="<% netesecid_key %>">
```

Related Documentation

- [SM-NewUserPin.shtml](#)
- [SM-NextToken.shtml / SM-NextToken-mobile-webkit.shtml / SM-NextToken-ipad.shtml](#)

netesecid_username

	The user's username.
Object Type	Template variable
Included In	SM-NewPinSelect.shtml, SM- NextToken.shtml, SM-NewPinSystem.shtml, SM-NewUserPin.shtml
Usage	Optional
Example	The following INPUT statement assigns the netesecid_username to the form username field:

```
<INPUT type="text" name="username" value="<% netesecid_username %>">
```

Related Documentation [netesecid_realm](#)

netesecidactionCancel

	Cancels a eTrust secure ID action currently in progress.
Object Type	HTML form field
Included In	SM-NewPinSelect.shtml, SM- NextToken.shtml, SM-NewPinSystem.shtml, SM-NewUserPin.shtml
Usage	Required. The value of either this field or of netesecidactionEnter must be sent to the server:

```
<INPUT type="submit" value="Cancel" name="netesecidactionCancel">
```

Related Documentation [netesecidactionEnter](#)

netesecidactionEnter

	Initiates the submission of the current eTrust secure ID action.
Object Type	HTML form field
Included In	SM-NewPinSelect.shtml , SM-NextToken.shtml , SM-NewPinSystem.shtml , SM-NewUserPin.shtml
Usage	Required. The value of either this field or of netesecidactionCancel must be sent to the server.
Example	<pre><INPUT type="submit" value="Enter" name="netesecidactionEnter"></pre>
Related Documentation	netesecidactionCancel

Netscape4xx

	Checks for Netscape version 4.x browser.
Object Type	Template variable
Included In	Logout.shtml
Usage	Required. Checks to see if the user's browser is Netscape version 4.x. If so, then, on logout, error messages are passed to the page in a compatible way.
Example	None
Related Documentation	Logout.shtml / Logout-mobile-webkit.shtml / Logout-ipad.shtml

NextToken.shtml / NextToken-mobile-webkit.shtml / NextToken-ipad.shtml

	Prompts the user to verify his credentials by entering his SecureID token code.
Object Type	Template
Usage	Optional. Although this template is optional, you should always include it in the zip file that you upload to Pulse Connect Secure.
Required Variables	<ul style="list-style-type: none">• "secid_contextid"• "secid_loginmode"
Example	None

NextToken-new-ux.shtml

	Displays the Template page for Next Auth token.
Object Type	Template
Usage	Optional. Displays page for providing next Auth token for authentication. Required in case SecureID Auth server is configured and requires customization.
Required Variables	secid_key

NewPin.thtml / NewPin-mobile-webkit.thtml / NewPin-ipad.thtml

	Prompts ACE users to enter a new PIN or create a system-generated PIN before signing into the system.
Object Type	Template
Usage	Required. When configuring the secid_pinselectmode variable (which sets the user's PIN select mode), note that possible values include:

Table8: PIN Error Messages

Mode	Description
0	User must use the system-generated PIN. He cannot enter I
1	User may enter his own PIN or use the system-generated P
2	User must enter his own PIN. He cannot use the system-ge

When configuring the **secid_pinserr** variable (which stores the error code and message that informs the user if he mis-entered his PIN), note that possible values include:

Table9: Additional PIN Error Messages

Code	Value
0	New PIN is required.
1	The two PINs entered do not match.
2	PIN Format is invalid.
3	PIN length is invalid.

Required Variables

- "secid_contextid"
- "secid_loginmode"

Example None

NewPin-new-ux.thtml

	Displays the Template page for authenticating with New PIN.
Object Type	Template
Usage	Optional. Displays page capturing new PIN for authentication. Required in case ACE/Radius Auth server is configured and requires customization.
Required Variables	secid_key secid_pinselectmode

occurrence

	Contains the recurring meeting occurrence ID.
Object Type	Template variable
Included In	LoginMeeting.thtml, MeetingRun.thtml, MeetingSelect.thtml, MeetingTrouble.thtml
Usage	Required
Example	The following code shows how the occurrence variable is used in an INPUT statement:

```
<input type="hidden" name="occurrence" value="<% occurrence %>">
```

Related Documentation [mid](#)

onsubmit

	Event trigger in a form.
Object Type	HTML form element
Included In	GeneratePin.thtml, LoginPage.thtml, NewPin.thtml, NextToken.thtml, SM-NewUserPin.thtml
Usage	Required if you are using Pulse Secure Collaboration. Returns the time zone offset value set by the Login() function.
Example	The following code shows a typical use of onsubmit: onsubmit="return Login()"
Related Documentation	None

p

	Specifies the page name as CASQUEAPP.
Object Type	HTML form field
Included In	Defender.thtml
Usage	Required
Example	The following code assigns the value CASQUEAPP to the form INPUT field p:

```
<INPUT type=hidden name="p" value="casqueapp">
```

Related Documentation [secid_challenge](#)

PageErrorArgs

	Contains the arguments for an error or message.
Object Type	Template variable

Included In PasswordChange.shtml

Usage Optional

Example The following code checks to see if a password length error occurs, then displays an informational message:

```
<% IF PageErrorCode == 2015 %>  
New password must not be longer than  
<% PageErrorArgs.MaximumPasswordLength %> characters.
```

Related Documentation

- [PageErrorCode](#)
- [PasswordChange.shtml / PasswordChange-mobile-webkit.shtml / PasswordChange-ipad.shtml](#)

PageErrorCode

Contains the error code in case of an error or message.

Object Type Template variable

Included In LoginPage.shtml, PasswordChange.shtml

Usage Optional

Table10: Page Error Codes

Error Code	Error message
1020	Password Change Success
1021	Account Disabled
1022	Account Locked out
1023	Account Expired
2001	Password Mismatch
2002	Invalid Password
2003	Password Change not allowed
2004	Password too new
2006	Password too short
2007	Password too trivial
2008	Password exists in password history
2009	Password Expired Message
2010	Password must change at next logon message
2011	Password does not meet complexity requirements

2012	Could not connect to auth server to change password
2013	Password change failed
2014	Password successfully changed
2015	Password will expire in some amount of time
2016	You have some number of grace logins.

For more errors and messages, see “[LoginPageErrorCode](#)” and “[LoginPageErrorMessage](#)”.

Example

You cannot change the messages that appear in [Table 10](#), but you can create your own messages and include them in your customized pages. For example:

```
<% IF PageErrorCode == 2009 %>
Your password has expired. Please contact your system administrator.
<% END %>
```

Related Documentation

- [LoginPageErrorCode](#)
- [LoginPageErrorMessage](#)

params

Contains parameters required to sign in to a meeting.

Object Type

Template variable

Included In

MeetingSelect.shtml

Usage

Required.

Example

The following code fills the form with the names of available meeting instances:

```
<% IF rows %>
<% FOREACH meeting = meetings %>
<input type="button" value=" &gt; "
onclick="self.location='/dana-na/meeting/<%signin%>login_meeting.cgi?
mid=<%mid%>&occurrence=<% meeting.3 %><% params %>'>
<%started%> <%meeting.0%>
<%meeting.1%>
<%meeting.2%>
<% END %>
<%END%>
```

Related Documentation

[MeetingSelect.shtml](#)

password

Indicates the password for an optional JavaScript function in the **Defender.shtml** template.

Object Type

JavaScript variable

Included In	Defender.thtml
Usage	Optional
Example	The header of Defender.thtml contains the following optional function to automatically place the cursor in the password field of the Defender page. If you choose to include this function in the header, you must also call it from the body of Defender.thtml . Also note that you should not change the JavaScript function name, form name, or form field name and that the form and form fields should have the same name.

```
function FinishLoad() {
    document.frmDefender.>password.focus();
}
```

Related Documentation [FinishLoad\(\)](#)

password

	Sets an HTML form field as a password field.
Object Type	Input form field
Included In	LoginPage.thtml, NewPin.thtml, NextToken.thtml, LoginMeeting.thtml, Defender.thtml, PasswordChange.thtml, SM-SM- NewPinSelect.thtml, NewPinSystem.thtml, SM- SM- NewUserPin.thtml, SM- NextToken.thtml, GeneratePin.thtml
Usage	This field is required for all types of authentication servers except anonymous servers and certificate servers.
Example	The following code assigns the password field:

```
<input type="password" name="password">
```

Related Documentation

- [LoginPage.thtml / LoginPage-mobile-webkit.thtml / LoginPage-ipad.thtml](#)
- [password2](#)

password

	Contains the <i>security token</i> , which is equivalent to the meeting password. If meeting password is configured to be sent in the invitation emails, a security token is added to the meeting URL. This allows the user to join the meeting with one-click, without the requirement to enter the meeting password manually.
Object Type	Template variable
Included In	LoginMeeting.thtml
Related Documentation	<ul style="list-style-type: none"> • LoginPage.thtml / LoginPage-mobile-webkit.thtml / LoginPage-ipad.thtml

password2

	Verified password.
Object Type	HTML form field
Included In	NewPin.html, SM-NewPinSelect.html, SM-NewUserPin.html, SM-NextToken.html, GeneratePin.html
Usage	Required. The password2 and password fields pass the password and verified password that the user enters to the server for verification.
Example	The value of the following fields must match:

```
<INPUT type="password" name="password">
<INPUT type="password" name="password2">
```

“password”

PasswordChange.html / PasswordChange-mobile-webkit.html / PasswordChange-ipad.html

	Notifies the user that his password is due to expire and prompts him to change it.
Object Type	Template
Usage	<p>Optional. Note that the changePasswordTitle variable in this template specifies whether the changed password is for the user’s primary or secondary authentication server. When you include this variable in your template, Pulse Connect Secure adds a title to the standard old/new/confirm password text boxes specifying whether the user must change his primary or secondary password.</p> <p>Optional. When you include the adds a title to the standard old/new/confirm password text boxes specifying whether the user must change his password. The showChangePasswordTitle variable indicates whether or not to show the change password text. The variable value is true if primary and secondary authentication both exist in the login process. The changePasswordTitle variable is a regular text (title) message that indicates whether the password change page is for a primary or a secondary password. Without this variable, the password change page only displays the old/new/confirm password text boxes and a message above, suggesting that the user change password. With the changePasswordTitle variable, the password change page displays the old/new/confirm password text boxes, a message above suggesting that the user change password, and another heading at the top of that page, stating Change Primary (or Secondary) Password.</p>
Required Variables	None
Example	None
Related Documentation	<ul style="list-style-type: none"> changePasswordTitle showChangePasswordTitle

PasswordComplexityMessage

	The password complexity message.
Object Type	Template variable
Included In	PasswordChange.thtml
Usage	Optional. Argument for error code 2011 .
Example	The following code notifies the user that the password does not meet complexity requirements:

```
<% IF PageErrorCode == 2011 %>  
  New password not complex enough.  
  Must consist of <% PageErrorArgs.PasswordComplexityMessage %>  
  characters.  
<% END %>
```

Related Documentation

- [PageErrorArgs](#)
- [PageErrorCode](#)

PasswordExpiration.thtml / PasswordExpiration-mobile-webkit.thtml / PasswordExpiration-ipad.thtml

	Notifies the user that his password has expired and prompts him to change it.
Object Type	Template
Usage	Optional
Required Variables	“startPageLink”
Example	None

PasswordExpirationFriendlyTime

	Message displaying how soon password expires.
Object Type	Template variable
Included In	PasswordExpiration.thtml
Usage	Argument for error code 2015 . Shows time in days, hours, and minutes.
Example	The following code checks for error code 2015 , and displays a message containing the time remaining before the password expires:

```
<% IF LoginPageErrorCode == 2015 %>  
  Password expires in  
  <% LoginPageErrorArgs.PasswordExpirationFriendlyTime %>.  
<% END %>
```

Related Documentation

- [LoginPageErrorArgs](#)
- [LoginPageErrorCode](#)
- [PasswordExpirationSeconds](#)

PasswordExpirationSeconds

The number of seconds in which the current user's password will expire.

Object Type

Template variable

Included In

PasswordExpiration.thtml

Usage

Optional. Argument for error code 2015.

Example

The following code indicates that the user's password will expire in a certain number of seconds:

```
<% IF LoginPageErrorCode == 2015 %>  
  Password will expire in  
  <% LoginPageErrorArgs.PasswordExpirationSeconds %> seconds.  
<% END %>
```

Related Documentation

- [LoginPageErrorArgs](#)
- [LoginPageErrorCode](#)
- [PasswordExpirationFriendlyTime](#)

PasswordHistoryLength

The length of the list of passwords the system has saved for a given end-user.

Object Type

Template variable

Included In

PasswordChange.thtml

Usage

Optional. Argument for error code **2008**. Some authentication systems save passwords and do not allow an end-user to assign the same password twice in succession, within a certain amount of elapsed time, or as long as the password still exists in the password history list.

Example

The following code shows the use of **PasswordHistoryLength**:

```
<% IF PageErrorCode == 2008 %>  
  New password must not repeat previous  
  <% PageErrorArgs.PasswordHistoryLength %> passwords.
```

Related Documentation

[PasswordComplexityMessage](#)

pleasewait

	Default text printed to the page when either ActiveX or an applet is loading.
Object Type	Template variable
Included In	Remediate.shtml
Usage	Required
Example	The following code shows a table containing two directives. The first, the loading directive, tests whether or not ActiveX or an applet is loading. If yes, then the pleasewait directive displays a message, prompting the user to wait.

```
<% loading %>
<% pleasewait %>
```

Related Documentation

- [loading](#)
- [Remediate.shtml / Remediate-mobile-webkit.shtml / Remediate-ipad.shtml](#)

PleaseWait-mtg.shtml

	Meeting client launch intermediate page.
Object Type	Template
Usage	Required
Required Variables	<ul style="list-style-type: none">• home• meeting_ini• startSessionReDir

PleaseWait.shtml / PleaseWait-mobile-webkit.shtml / PleaseWait-ipad.shtml

	Displays a page asking the user to wait while application components install or uninstall.
Object Type	Template
Usage	Optional. This template differs from the standard PleaseWait.shtml template in the types and degree of validation and status checking performed while you wait. You may choose to customize this template if you have configured Host Checker or Cache Cleaner at the realm level. Use this page to control Host Checker and Cache Cleaner starts during pre-authentication and post-authentication. This template contains JavaScript for detecting, stopping, installing, and uninstalling the Host Checker and Cache Cleaner components as well as images and text to display to users while Pulse Connect Secure performs these actions. This template also contains JavaScript that periodically checks the user's status to determine if Host Checker or Cache Cleaner is loaded on his system and redirects him to the sign-in page (welcome.cgi) if necessary.

For detailed information about the JavaScript and variables described here, see comments in the template.

Required Variables

- `"LoginPageErrorMessage"`
- `"PleaseWaitObjectCC"`
- `isLinux`
- `isMAC`
- `userAgentTypeApple` (PleaseWait-mobile-webkit.thtml)
- `userAgentTypeAndroid` (PleaseWait-mobile-webkit.thtml)
- `serverdetails` (PleaseWait-mobile-webkit.thtml)

Example

None

Related Documentation

- [Logout.thtml / Logout-mobile-webkit.thtml / Logout-ipad.thtml](#)

pleasewait-ipad.thtml

Displays a page asking the user to wait while application components install or uninstall. This is the iPad version of **PleaseWait.thtml**.

Object Type

Template

Usage

Optional. This template differs from the standard **PleaseWait.thtml** template in the types and degree of validation and status checking performed while you wait. You may choose to customize this template if you have configured Host Checker or Cache Cleaner at the realm level. Use this page to control Host Checker start during pre-authentication and post-authentication.

Required Variables

- `showHeading`
- `delivery_mode`
- `pleasewaitObjectHC`
- `setup_classid`
- `setup_codebase`

Example

None

Related Documentation

-

pleasewaitLogoutJSCode

JavaScript code for **pleasewait** logout.

Object Type

Template variable

Included In

Logout.thtml

Usage

Required

Example

The following code shows the variable passed to the page verbatim, prior to the script section of the HTML page:

```
<html>
<head>
<meta http-equiv="Content-Language">
<meta http-equiv="Content-Type" content="text/html">
<meta name=robots content="none">
<title>Instant Virtual Extranet</title>

<% pleasewaitLogoutJSCode FILTER verbatim %>
<%IF pleasewaitObjectHC OR pleasewaitObjectCC OR pleasewaitWin32%>
<%IF pleasewaitObjectHC OR pleasewaitWin32%><SCRIPT LANGUAGE="JavaScript">
```

Related Documentation

- [Logout.shtml / Logout-mobile-webkit.shtml / Logout-ipad.shtml](#)
- [PleaseWaitObjectCC](#)
- [pleasewaitWin32](#)

PleaseWait-new-ux.shtml

Prompts the user to wait while application components install or uninstall.

Object Type

Template

Usage

Required. Displays waiting page until the client components are installed and launched.

Required Variables

mobile_browser_login
delivery_mode
pleasewaitObjectEP
pleasewaitObjectJAM
pleasewaitJSCode
pleasewaitObjectCC
pleasewaitObjectHC
pleasewaitTO_msec
pleasewaitObjectHC_js
geckoBrowser
jpaAction
jpaInstall
isVista

PleaseWaitObjectCC

	If a Cache Cleaner object is enabled, this variable contains the Cache Cleaner object tag.
Object Type	Template variable
Included In	Logout.thtml, PleaseWait.thtml
Usage	Required
Example	The following code checks to see if the Cache Cleaner object is enabled. If so, the code displays a message:

```
<%IF pleasewaitObjectCC %>
  <span><div id="cc">Cache Cleaner Component</div></span>
<%END%>
```

Related Documentation

- [Logout.thtml / Logout-mobile-webkit.thtml / Logout-ipad.thtml](#)
- [PleaseWait.thtml / PleaseWait-mobile-webkit.thtml / PleaseWait-ipad.thtml](#)
- [PleaseWaitObjectHC](#)
- [pleasewaitWin32](#)

pleaseWaitObjectEP

	If Odyssey Access Client is enabled, this variable contains the Odyssey Access Client object tag.
Object Type	Template variable
Included In	AgentInstall.thtml
Usage	Optional (Required for Pulse Policy Secure).
Example	The following code checks to see if the Pulse Secure client is enabled. If so, the code displays a message:

```
<%IF pleaseWaitObjectEP %>
  <span><div id="ep">OAC Object</div></span>
<%END%>
```

PleaseWaitObjectHC

	If a Host Checker object is enabled, this variable contains the Host Checker object tag.
Object Type	Template variable
Included In	AgentInstall.thtml, Logout-ipad.thtml, Logout-mobile-webkit.thtml, Logout-ppc.thtml, Logout.thtml, PleaseWait-ppc.thtml, PleaseWait.thtml

Usage Required

Example The following code checks to see if the Host Checker object is enabled. If so, the code displays a message:

```
<%IF pleasewaitObjectHC %>
<span><div id="cc">Host Checker Component</div></span>
<%END%>
```

Related Documentation

- [PleaseWait.shtml / PleaseWait-mobile-webkit.shtml / PleaseWait-ipad.shtml](#)
- [PleaseWaitObjectCC](#)
- [pleasewaitWin32](#)

pleasewaitObjectJAM

If Pulse Secure client is enabled, this variable contains the Pulse Secure client object tag.

Object Type Template variable

Included In **AgentInstall.shtml, PleaseWait.shtml**

Usage Required

Example The following code checks to see if the Pulse Secure client is enabled. If so, the code displays a message.

```
<%IF pleasewaitObjectJAM %>
<span><div id="jam"> Pulse Object</div></span>
<%END%>
```

Related Documentation

- [PleaseWait.shtml / PleaseWait-mobile-webkit.shtml / PleaseWait-ipad.shtml](#)
- [pleasewaitWin32](#)

pleasewaitTO_msec

Timeout value for HC to be completed on an end-user system.

Object Type Template variable

Included In **PleaseWait-ppc.shtml**

Usage Optional (Required for mobile HC clients)

Example

```
<% pleasewaitTO_msec %>
```

pleasewaitWin32

Boolean indicating that you must stop one or more Win32 components from the **logout** page.

Object Type	Template variable
Included In	Logout.thtml
Usage	Required. Set to 1 if any components need to be stopped.
Example	The following code checks to see if any of the components are causing Pulse Connect Secure to display a Please Wait message and if the components are java-based. If yes, the code calls the stopComponents() JavaScript function to stop the component.

```
<% IF (pleasewaitObjectCC || pleasewaitObjectHC || pleasewaitWin32) &&
delivery_mode == 'java'%>
onload="javascript:stopComponents()"
<% END %>
```

Related Documentation

- [PleaseWait.thtml / PleaseWait-mobile-webkit.thtml / PleaseWait-ipad.thtml](#)
- [PleaseWaitObjectCC](#)
- [PleaseWaitObjectHC](#)
- [stopComponents\(\)](#)

plugintype

Provides the information about type of defender server application.

Object Type	Template variable
Included In	Defender.thtml
Usage	Optional. Valid value is CASQUEAPP. Pulse Connect Secure sets the plugintype variable to CASQUEAPP when finding a Casque RADIUS server.
Example	The following code loads the frmCasque HTML form if the plugintype contains the value CASQUEAPP :

```
<% IF plugintype == 'CASQUE' %>
<form name="frmCasque" method="POST" autocomplete=off>
<input type=hidden name="key" value="<% secid_key %>">
<INPUT type=hidden name="p" value="casqueapp">
<a href="#" onClick="submitFrmCasque(); return false;">
LAUNCH CASQUE PLAYER</a><BR>
</form>
```

Related Documentation

- [Defender.thtml / Defender-mobile-webkit.thtml / Defender-ipad.thtml](#)
- [frmCasque](#)

portal

	User interface string that displays the sign-in page's portal name.
Object Type	Template variable
Default Value	Secure SSL VPN
Included In	LoginPage.html, LogoutPage.html, SSL.html, SecondaryLoginPage.html, Logout.html, Defender.html, ExceededConcurrent.html, GeneratePin.html, NewPin.html, NextToken.html, ShowSystemPin.html
Usage	Optional
Example	<% portal FILTER verbatim %>
Related Documentation	None

postAuthSNDecline

	Text for "Decline" button on sign-in notifications page.
Object Type	Template variable
Default value	None
Included In	SigninNotifPostAuth.html
Usage	Optional

postAuthSNProceed

	Text for "Proceed" button on sign-in notifications page.
Object Type	Template variable
Default value	None
Included In	SigninNotifPostAuth.html
Usage	Optional

postAuthSNText

	Content of sign-in notification on sign-in notifications page.
Object Type	Template variable
Default value	None
Included In	SigninNotifPostAuth.html
Usage	Required

postAuthSNTitle

Title of sign-in notification on sign-in notifications page.

Object Type	Template variable
Default value	None
Included In	SigninNotifPostAuth.thtml
Usage	Optional

preAuthSNDecline

Text for "Decline" button on sign-in notifications page.

Object Type	Template variable
Default value	None
Included In	SigninNotifPreAuth.thtml
Usage	Optional

preAuthSNProceed

Text for "Proceed" button on sign-in notifications page.

Object Type	Template variable
Default value	None
Included In	SigninNotifPreAuth.thtml
Usage	Optional

preAuthSNText

Content of sign-in notification on sign-in notifications page.

Object Type	Template variable
Default value	None
Included In	SigninNotifPreAuth.thtml
Usage	Required

preAuthSNTitle

Title of sign-in notification on sign-in notifications page.

Object Type	Template variable
Default value	None
Included In	SigninNotifPreAuth.thtml
Usage	Optional

prompts

	Contains display information for username/password prompts.
Object Type	Template variable
Included In	LoginPage.html, NewPin.html, NextToken.html, Defender.html, SecondaryLoginPage.html, SM-NewPinSelect.html, SM-NewUserPin.html, SM-NextToken.html
Usage	<p>Required when including SecondaryLoginPage.html template in your zip file to collect secondary authentication credentials. The prompt variable is a hash list used to display primary and/or secondary username and password prompts in the sign-in page. You can use the prompts variable instead of hardcoding username and password labels.</p> <p>If you want to prompt the user to enter secondary sign-in credentials, and you do not include the prompts variable in your template, Pulse Connect Secure brings up a secondary sign-in page to collect the user's secondary credentials. If you include the prompts variable, Pulse Connect Secure collects the secondary credentials along with the primary credentials in the main sign-in page.</p>
Example	<p>The following example is a required piece of code in the SecondaryLoginPage.html template:</p> <pre><% FOREACH prompt = prompts %> <% NEXT IF !prompt.required %> <% prompt.promptText %> <input type="<% prompt.type %>" name="<% prompt.name %>" size="20"> <% END %></pre>
Related Documentation	None SecondaryLoginPage.html / SecondaryLoginPage-mobile-webkit.html / SecondaryLoginPage-ipad.html

psaldownloadmacapp.html

	Displays the Template page for Pulse Application Launcher download for MAC clients.
Object Type	Template
Usage	Optional. Used for downloading PSAL on MAC machines in Cloud Secure use case.
Required Variables	go_back_url install_url psal_install_url_mac_app i18n_fb_download i18n_psal_continue_message i18n_do_not_want_to_proceed main_window_name psalText

psaldownloadmacapp-new-ux.thtml

	Displays the Template page for Pulse Application Launcher download for MAC clients.
Object Type	Template
Usage	Optional. Used for downloading PSAL on MAC machines in Cloud Secure use case.
Required Variables	go_back_url install_url psal_install_url_mac_app i18n_fb_download i18n_psal_continue_message i18n_do_not_want_to_proceed main_window_name psalText

psaldownloadwindowsapp-new-ux.thtml

	Displays the Template page for Pulse Application Launcher download for Windows clients.
Object Type	Template
Usage	Optional. Used for downloading PSAL on Windows machines in Cloud Secure use case.
Required Variables	go_back_url psal_install_url_windows_app i18n_fb_download i18n_psal_continue_message i18n_do_not_want_to_proceed main_window_name psalText

psalinstall-new-ux.thtml

	Displays the Template Page for installing Pulse Application Launcher.
Object Type	Template
Usage	Required. Displays install page for Pulse Application Launcher.
Required Variables	go_back_url win_store_app mac_store_app i18n_psal_installer_download i18n_psal_download_auto_close mac_application

i18n_psal_manual_download
i18n_psal_installer_continue
i18n_psal_installer_remember
app
i18n_psal_installer_systray
i18n_do_not_want_to_proceed
i18n_psal_installer_steps_img
psal_download_url
psal_mac_bin
psal_win_bin
psalInstallText

psallaunch.thtml

	Displays the Template page for launching Pulse Application Launcher.
Object Type	Template
Usage	Required. Displays launch page for Pulse Application Launcher.
Required Variables	isMobileSafari psalText psalurlparams

psallaunch-new-ux.thtml

	Displays the Template page for launching Pulse Application Launcher.
Object Type	Template
Usage	Required. Displays launch page for Pulse Application Launcher.
Required Variables	isMobileSafari psalText psalurlparams I18N_PSAL_LAUNCH_AUTO_CLOSE win1

psalwait-new-ux.thtml

	Displays the Template page for waiting to check Pulse Application Launcher.
Object Type	Template
Usage	Required. Displays wait page for launching Pulse Application Launcher.
Required Variables	apptoken app

back_url
 continue_url
 I18N_PSAI_WAIT_INSTALL
 install_url
 I18N_FB_DOWNLOAD
 I18N_DO_NOT_WANT_TO_PROCEED
 main_window_name
 psalWaitText

realm

	Returns name of the current user's realm.
Object Type	HTML form field
Included In	LoginPage.thtml, SM- NewPinSelect.thtml, SM-NewPinSystem.thtml, SM-NewUserPin.thtml, SM- NextToken.thtml, GeneratePin.thtml, NewPin.thtml
Usage	Required. Login.cgi passes the realm value posted here to the appropriate authentication server. See example below.
Example	The following example appears in several template files:

```
<% IF RealmList.size == 0 %>
  <td>LoginRealm</td><td> </td><td>
  <input type="text" name="realm" size="20">
  </td>
  <% ELSIF RealmList.size == 1 %>
  <input type="hidden" name="realm" value="<% RealmList.0 %>">
  <% ELSE %>
  <td>LoginRealm</td><td> </td><td>
  <select size="1" name="realm">
  <% FOREACH r = RealmList %>
  <option value="<% r %>" ><% r %></option>
  <% END %>
  </select>
```

If you only associate one realm with a URL, and you want to eliminate the realm text box and realm list box altogether, you can eliminate this code with the exception of the following line:

```
<input type="hidden" name="realm" value="<% RealmList.0 %>">
```

You must include this line. Pulse Connect Secure populates this variable with the correct value. You can use this approach if you are sure that only one realm is associated with the sign-in URL.

Related Documentation [RealmList](#)

realm

	User interface label for the string "realm" .
Object Type	Template variable
Default value	Realm
Included In	LoginPage.thtml
Usage	Optional
Example	<p>In the following example, if no list of realms is available, display a textbox for the user to enter their realm. The realm variable is a label for the textbox.</p> <pre><% IF RealmList.size == 0 %> <% realm %> <input type="text" name="realm" value="" size="20"></pre>
Related Documentation	RealmList

RealmList

	List of realms available to the user.
Object Type	Template variable
Included In	LoginPage.thtml, GeneratePin.thtml, NewPin.thtml
Usage	<p>Optional. If you use the RealmList variable in the LoginPage.thtml template, the following rules apply:</p> <ul style="list-style-type: none">• If RealmList.size = 0, then allow the user to type the realm name into a text box.• If RealmList.size => 1, then allow the user to choose the realm name from a list box.
Example	<p>The following example from LoginPage.thtml shows RealmList being used as an index in a foreach loop that returns the available realms.</p> <pre><% FOREACH r = RealmList %> <option value="<% r %>" ><% r %></option> <% END %></pre>
Related Documentation	<ul style="list-style-type: none">• CertificateAuthentication• realm

recallLastRealmUsed()

	Use this function if you want to allow users to select from multiple realms in the sign-in page.
Object Type	JavaScript function
Included In	LoginPage.thtml
Usage	Optional

Example

The function checks that a realm exists and then gets the last selected authentication realm.

```
function recallLastRealmUsed() {
  if (document.frmLogin.realm != null &&
      document.frmLogin.realm.type == "select-one") {
    // try to remember which auth realm was last used
    var sLastRealm = GetCookieValue("lastRealm");
    if (sLastRealm.length > 0) {
      var cmb = document.frmLogin.realm;
      var nNumRealms = cmb.options.length;
      for (var n=0; n < nNumRealms; n++) {
        if (cmb.options[n].text == sLastRealm) {
          cmb.selectedIndex = n;
        }
      }
    }
  }
}
```

Related Documentation [FinishLoad\(\)](#)

redirect()

Redirects the page to a compatibility checking page.

Object Type

JavaScript function

Included In

MeetingRunJava.shtml, MeetingTestMSJava.shtml, MeetingTestJava.shtml

Usage

Optional

Example

The following two code samples show how the **redirect()** function redirects to a meeting page generated by a given CGI file:

From **MeetingTestMSJava.shtml**:

```
function redirect () {
  document.location.replace("/dana-na/meeting/
  <%signin%>meeting_testjava.cgi?<%mid_param%>")
}
```

From **MeetingTestJava.shtml**:

```
function redirect () {
  document.location.replace("/dana-na/meeting/
  <%signin%>meeting_testresult.cgi?<%mid_param%>")
}
```

Related Documentation

- [MeetingTestMSJava.shtml](#)

-
- [MeetingRunJava.thtml](#)
 - [MeetingTestJava.thtml](#)
 - [MeetingTestJS.thtml](#)

Remediate.thtml / Remediate-mobile-webkit.thtml / Remediate-ipad.thtml

Provides the user with the ability to fix the Host Checker starts during pre-authentication and post-authentication, before the Welcome page displays.

Object Type

Template

Usage

Optional. If you have enabled Host Checker remediation, this page appears before/after the sign-in page and displays messages to the user when Host Checker policies with custom instructions have failed.

Required Variables

- "listFailedPolicies"
- "showButtons"
- "showClose"
- "showContinue"
- "showHeading"
- "showLoading"
- "showPolicies"
- "showRemedOption"
- "showTryAgain"
- showHeading (Remediate-mobile-webkit.thtml and Remediate-ipad.thtml)
- showPolicies (Remediate-mobile-webkit.thtml and Remediate-ipad.thtml)
- showButtons (Remediate-mobile-webkit.thtml and Remediate-ipad.thtml)
- showContinue (Remediate-mobile-webkit.thtml and Remediate-ipad.thtml)
- listFailedPolicies (Remediate-mobile-webkit.thtml and Remediate-ipad.thtml)

Example

None

Remediate-new-ux.thtml

Displays the Template page for Host Checker Remediation.

Object Type

Template

Usage

Required. Displays page capturing remediation instructions and failure policy reasons.

Required Variables

isUAC
showHeading
showPolicies

listFailedPolicies
 showButtons
 showTryAgain
 showRemedOption
 textRemedOption
 showReinstall
 textReinstall
 showContinue
 textContinue
 showClose
 textClose
 showLoading
 loading
 pleasewait
 HC_REMED_POLICIES_CHECK

remedy1

Contains the first possible remedy if your system is not fully compatible.

Object Type	Template variable
Included In	MeetingTestResult.thtml
Usage	Required
Example	The following code displays the first remedy:

```
<% IF remedy1 %>
  <% remedy1 FILTER verbatim %>
<% END %>
```

Related Documentation [remedy2](#)

remedy2

Contains the second possible remedy if your system is not fully compatible.

Object Type	Template variable
Included In	MeetingTestResult.thtml
Usage	Required
Example	The following code displays the second remedy:

```
<% IF remedy2 %>
```

```
<% remedy2 FILTER verbatim %>
<% END %>
```

Related Documentation [remedy1](#)

RoleList

List of roles available to the user.

Object Type Template variable

Included In **SelectRole.shtml**

Usage Optional

Example The following code uses the **RoleList** variable to process a loop of roles (in this case, two roles):

```
<%FOREACH role = RoleList %>
<a href="login.cgi?loginmode=mode_selectedrole&role=<%role.0%>
  &selectrolekey=<%SelectRoleKey%>&passwordExpiration=
  <%PasswordExpiration%>"><%role.1%></a>
<%END%>
```

Related Documentation [SelectRole.shtml](#) / [SelectRole-mobile-webkit.shtml](#) / [SelectRole-ipad.shtml](#)

ROWS

Contains the number of occurrences that match the search criteria.

Object Type Template variable

Included In **MeetingSelect.shtml**

Usage Required

Example The following code checks to see if there are any matching occurrences. If yes, the code then generates a list of meeting instances:

```
<% IF rows %>
<%FOREACH meeting = meetings%>
<input type="button" value=" &gt; "
  onclick="self.location=/dana-na/meeting/
  <%signin%>login_meeting.cgi?mid=<%mid%>&
  occurrence=<%meeting.3%><%params%>"">
  <%started%>
  <%meeting.0%>
  <%meeting.1%>
  <%meeting.2%>
<% END %>
<%END%>
```

Related Documentation [MeetingSelect.shtml](#)

runOnLoad

	Called when page loads the <body> .
Object Type	JavaScript function
Included In	Remediate.thtml
Usage	Optional. Useful when you need to load ActiveX or an applet as needed.
Example	None
Related Documentation	Remediate.thtml / Remediate-mobile-webkit.thtml / Remediate-ipad.thtml

safari

	Boolean that indicates whether or not the user's browser is Macintosh Safari.
Object Type	Template variable
Included In	Logout.thtml, MeetingTestJS.thtml
Logout.thtmlUsage	Required
Example	The following code fragment tests the variable to discover if the browser is the Safari browser:

```
<% IF safari %>
```

Related Documentation [geckoBrowser](#)

secid_challenge

	Challenge string obtained from a server.
Object Type	Template variable
Included In	Defender.thtml
Usage	Required. Used to return a challenge string from a RADIUS server in Defender.thtml .
Example	The following code displays the challenge to the user.

```
<% secid_challenge FILTER verbatim %>
```

Related Documentation

- [Defender.thtml](#) / [Defender-mobile-webkit.thtml](#) / [Defender-ipad.thtml](#)
- [p](#)

secidcontextid

	Form field for the value represented by secid_contextid .
Object Type	HTML form field

Included In [GeneratePin.shtml](#), [NewPin.shtml](#), [NextToken.shtml](#), [Defender.shtml](#)

Usage Required

Example The following code assigns the handle of the current transaction to a form field:

```
<INPUT type=hidden name="secidcontextid" value="<% secid_contextid %>">
```

Related Documentation [secid_contextid](#)

secid_contextid

Current transaction handle.

Object Type Template variable

Included In [GeneratePin.shtml](#), [NewPin.shtml](#), [NextToken.shtml](#)

Usage Required. You must send this value to the server, but if you do not want to expose it to the user, send it as a hidden form field.

Example The following code assigns the current transaction handle from the server to the HTML form.

```
<INPUT type=hidden name="secidcontextid" value="<% secid_contextid %>">
```

Related Documentation [secidcontextid](#)

secid_loginmode

Returns a value of Next Token mode or New PIN mode.

Object Type Template variable

Included In [GeneratePin.shtml](#), [NewPin.shtml](#), [NextToken.shtml](#), [Defender.shtml](#)

Usage Required. You must send this value to the server, but if you do not want to expose it to the user, you may send it as a hidden form field.

Example The following code assigns the mode to the HTML form:

```
<INPUT type=hidden name="loginmode" value="<% secid_loginmode %>">
```

Related Documentation

- [secidcontextid](#)
- [secid_contextid](#)

secid_pinerr

Error code and corresponding message that informs the user if he has entered his PIN incorrectly.

Object Type	Template variable
Included In	GeneratePin.thtml, NewPin.thtml
Usage	Optional
Example	The following code shows four different conditions using secid_pinerr :

```
<% IF secid_pinerr == 0 %>  
  New PIN required  
<% ELSIF secid_pinerr == 1 %>  
  The Two PINs Entered Do Not Match  
<% ELSIF secid_pinerr == 2 %>  
  Invalid PIN Format  
<% ELSIF secid_pinerr == 3 %>  
  Invalid PIN Length  
<% END %>
```

Related Documentation

- [secid_pinformat](#)
- [secid_pinselectmode](#)

secid_pinformat

Error message informing the end-user of the restrictions on the required PIN.

Object Type	Template variable
Included In	GeneratePin.thtml, NewPin.thtml
Usage	Optional
Example	The following fragment shows how to use this variable to display a particular restriction:

```
You must create a new Personal Identification Number (PIN) before you can sign  
in. Your PIN should be <% secid_pinformat %> long.
```

Related Documentation [secidpinformat](#)

secid_pinselectmode

Sets the user's PIN select mode.

Object Type	Template variable
Included In	GeneratePin.thtml, NewPin.thtml
Usage	Optional
Example	The following code creates a link and sets the login mode:

```
<% IF secid_pinselectmode == 1 %>  
  If you prefer, the system can
```

```
<a href="secid_genpin.cgi?loginmode=<% secid_loginmode %>"
onclick="gCancelNewPinMode=false;">
generate a PIN</a> for you.
<% END %>
```

Related Documentation [secidpinselectmode](#)

secid_systempin

	System-generated PIN.
Object Type	Template variable
Included In	ShowSystemPin.thtml , GenerateNewPin.thtml , NewPin.thtml
Usage	Optional
Example	The following code sets the user's PIN.

```
Your new PIN is <% secid_systempin %>
Be sure to remember it, because you need your PIN each time you sign in.
```

Related Documentation [secidactionSavePin](#)

secid_username

	System-supplied username.
Object Type	Template variable
Included In	GeneratePin.thtml , NewPin.thtml , NextToken.thtml , Defender.thtml , LoginPage.thtml
Usage	Required
Example	The following code posts the username passed into the form back to the server:

```
<INPUT type=hidden name="username" value="<% secid_username %>">
```

Related Documentation [username](#)

secidactionSavePin

	Specifies an action to be taken when a new PIN is created.
Object Type	HTML form field
Included In	NewPin.thtml , GeneratePin.thtml
Usage	Required
Example	The value of one of the following fields must be sent to the Pulse Connect Secure server:

```
<INPUT type="submit" name="secidactionSavePin" value="Save PIN">
<INPUT type="submit" name="secidactionCancel" value="Cancel">
```

Related Documentation

- [GeneratePin.shtml](#) / [GeneratePin-mobile-webkit.shtml](#) / [GeneratePin-ipad.shtml](#)
- [NewPin.shtml](#) / [NewPin-mobile-webkit.shtml](#) / [NewPin-ipad.shtml](#)

secidpinformat

Error message informing the end-user of the restrictions on the required PIN.

Object Type	HTML form field
Included In	NewPin.shtml , GeneratePin.shtml
Usage	Required
Example	The following code assigns the secure ID PIN format:

```
<INPUT type=hidden name="secidpinformat" value="<% secid_pinformat %>">
```

Related Documentation [secid_pinformat](#)

secidpinselectmode

Passes the user's PIN select mode to the server.

Object Type	HTML form field
Included In	NewPin.shtml , GeneratePin.shtml
Usage	Required
Example	The following code assigns the PIN select mode:

```
<INPUT type=hidden name="secidpinselectmode"
value="<% secid_pinselectmode %>">
```

Related Documentation [secid_pinselectmode](#)

[SecondaryLoginPage.shtml](#) / [SecondaryLoginPage-mobile-webkit.shtml](#) / [SecondaryLoginPage-ipad.shtml](#)

Collects the user's name and password for a secondary authentication server.

Object Type	Template
Usage	Optional. You must include the prompts variable in LoginPage.shtml in order to call this template.

The **prompts** variable provides the proper form input names and indicates whether or not the inputs are required. However, if you prefer, you can hardcode the form input names into the page and it still works. Further, the sign-in process accepts form input names **user#2** and **password#2** as the secondary credentials for any authentication server. The names in the prompt variables include the authentication server name in this format: **user@ldap**, **password@ldap**.

Required Variables None

Example None

SelectRole.shtml / SelectRole-mobile-webkit.shtml / SelectRole-ipad.shtml

Appears after the sign-in page and displays a list of roles from which the user can choose.

Object Type Template

Usage Optional. You may choose to customize this template if you have assigned your users to multiple roles, but have not permissively merged those roles.

Required Variables None

Example None

serverdetails

Contains some server details that are used on the end-user system.

Object Type Template variable

Included In **PleaseWait-ipad.shtml, PleaseWait-mobile-webkit.shtml**

Usage Optional (Required for Pulse HC on Android or iOS end-user systems)

Required Variables None

Example

```
<% serverdetails %>
```

setFailed()

Called if the component is stopped intentionally.

Object Type JavaScript function

Included In **Logout.shtml, PleaseWait.shtml**

Usage component signifies the component that is being checked to see if it has been stopped intentionally.

Example The following code stops Host Checker:

```
<%IF pleasewaitObjectHC %> setSucceeded('hc');  
<%END%>
```

```

}catch (e) {
  <%IF pleasewaitObjectHC %> setFailed('hc'); <%END%>
}

```

Related Documentation [setStarted\(\)](#)

setFinished

Called if the component has finished starting up.

Object Type JavaScript function

Included In **Logout.thtml, PleaseWait.thtml**

Usage Optional

component—Valid values for component are hc for Host Checker and cc for Cache Cleaner.

componentExample The following code shows how **setFinished** is used in the **Logout.thtml** template:

```

function setFinished(component)
{
  document.getElementById(component).style.fontWeight = "normal";
}

```

Related Documentation

- [Logout.thtml / Logout-mobile-webkit.thtml / Logout-ipad.thtml](#)
- [PleaseWait.thtml / PleaseWait-mobile-webkit.thtml / PleaseWait-ipad.thtml](#)

SetLastRealm()

Enables users to select from multiple realms on the sign-in page.

Object Type JavaScript function

Included In **LoginPage.thtml**

Usage When the user signs in, this function retrieves the authentication realm selected by the user and sets an expiration date of 30 days.

sValue—The name of the realm the user selects.

Example The following code shows the function:

```

function SetLastRealm(sValue) {
  var dtExpire = new Date();
  dtExpire.setDate(dtExpire.getDate() + 30);
  document.cookie = "lastRealm=" +
    escape(sValue) + "; expires=" + dtExpire.toGMTString();
}

```

Related Documentation [LoginPage.thtml / LoginPage-mobile-webkit.thtml / LoginPage-ipad.thtml](#)

setStarted()

Called if the component is started in order to stop it.

Object Type	JavaScript function
Included In	Logout.thtml, PleaseWait.thtml
Usage	Required.

component—Name of the component.

Example The following code shows the function:

```
function setStarted(component)
{
  document.getElementById(component).style.fontWeight = "bold";
}
```

Related Documentation [setFailed\(\)](#)

setSucceeded()

Called if the component is successfully stopped.

Object Type	JavaScript function
Included In	Logout.thtml, PleaseWait.thtml
Usage	Required

Example The following code calls the function if Host Checker has been successfully stopped:

```
<%IF pleasewaitObjectHC %> setSucceeded('hc'); <%END%>
```

Related Documentation [setStarted\(\)](#)

setthankyou()

Sets the timeout value and calls the **thankyou()** function, which displays the login page after the user has signed out.

Object Type	JavaScript function
Included In	Logout.thtml
Usage	Required

Example The following code shows the function:

```
function setthankyou()
{
  setTimeout("thankyou()", 2000);
}
```

}

Related Documentation [thankyou\(\)](#)

setup_classid

	The CLASSID of the ActiveX object.
Object Type	Template variable
Included In	Logout.thtml, Logout-ipad.thtml, Logout-mobile-webkit.thtml, Logout-ppc.thtml, PleaseWait-ppc.thtml, PleaseWait.thtml, AgentInstall.thtml, AgentlessInstall.thtml
Usage	Required
Example	The following example assigns the CLASSID if the delivery mechanism is ActiveX.

```
<%IF delivery_mode == 'activex'%>
<OBJECT <%setup_classid FILTER verbatim%> id=NeoterisSetup
```

Related Documentation

- [setup_codebase](#)

setup_codebase

	The path and version information (CODEBASE) of the ActiveX object.
Object Type	Template variable
Included In	Logout.thtml, Logout-ipad.thtml, Logout-mobile-webkit.thtml, Logout-ppc.thtml, PleaseWait-ppc.thtml, PleaseWait.thtml, AgentInstall.thtml, AgentlessInstall.thtml
Usage	Required
Example	The following example assigns the CODEBASE if the delivery mechanism is ActiveX.

```
<%IF delivery_mode == 'activex'%>
<OBJECT <%setup_classid FILTER verbatim%> id=NeoterisSetup
<%setup_codebase FILTER verbatim%>
width=0 height=0 >
</OBJECT>
```

Related Documentation

- [setup_classid](#)

setup_ini

	Contents of the ActiveX object.
Object Type	Template variable
Included In	Logout.thtml, Logout-ppc.thtml
Usage	Required
Example	The following example assigns the CODEBASE if the delivery mechanism is ActiveX.

```
<%IF setup_ini %>
<OBJECT <%setup_ini FILTER verbatim%>
```

Related Documentation

-

Setupappversion

The version of the Java installer setup application.

Object Type Template variable

Included In Logout.thtml

Usage Required

Example The following example assigns the `setupappversion` after determining the delivery mode:

```
<% IF delivery_mode == 'java' %>
<APPLET code=dsSetupApplet.class
code=dsSetupApplet
id=NeoterisSetup
archive="NeoterisSetupApplet.jar"
MAYSCRIPT
width=0 height=0
codebase="/dana-cached/setup/" >
<PARAM NAME="SetupAppVersion" VALUE="<% setupappversion %>">
```

Related Documentation [Logout.thtml](#) / [Logout-mobile-webkit.thtml](#) / [Logout-ipad.thtml](#)

setup_codebase

Codebase of Pulse Secure Setup ActiveX control.

Object Type Template variable

Included In Logout.thtml, PleaseWait.thtml

Usage Optional

Example The following code declares the `codebase`:

```
<OBJECT classid="clsid:4CC35DAD-40EA-4640-ACC2-A1A3B6FB3E06"
id=NeoterisSetup
<% setup_codebase FILTER verbatim %>
width=0 height=0 >
</OBJECT>
```

Related Documentation

- [Logout.thtml](#) / [Logout-mobile-webkit.thtml](#) / [Logout-ipad.thtml](#)
- [PleaseWait.thtml](#) / [PleaseWait-mobile-webkit.thtml](#) / [PleaseWait-ipad.thtml](#)

showButtons

Boolean indicating whether or not to show any of the action buttons.

Object Type	Template variable
Included In	Remediate.html , Remediate-ppc.html , Remediate-ipad.html , Remediate-mobile-webkit.html
Usage	Required
Example	The following code tests showButtons . If true, then the code continues to test other directives to determine which action buttons to display:

```
<% IF showButtons %>
<% IF showTryAgain %>
  <input name="btnTryAgain" type="button" value="<% textTryAgain %>"
  onClick="TryAgain()" <% disabled %>>
<% END %>
<% IF showRemedOption %>
  <input name="btnHideRemed" type="button"
  value="<% textRemedOption %>" onClick="HideRemed();
  <% disabled %>">
  <% HC_REMED_SHOW_ADV_PREFS %>
<% END %>
<% IF showContinue %>
  <input name="btnContinue" type="button" value="<% textContinue %>"
  onClick="Continue()">
<% END %>
<% IF showClose %>
  <input name="btnClose" type="button" value=" <% textClose %> "
  onClick="javascript:self.close()">
<% END %>
<% END %>
```

Related Documentation

- [showContinue](#)
- [showClose](#)
- [showRemedOption](#)
- [showTryAgain](#)

showChangePasswordTitle

Boolean indicating whether or not to show primary/secondary password change title.

Object Type	Template variable
Included In	PasswordChange.html
Usage	Optional.
Example	The following code tests the variable. If true, the code then displays the value contained in changePasswordTitle :

```
<%IF showChangePasswordTitle %>
  <% changePasswordTitle FILTER verbatim %>
<%END%>
```

Related Documentation

- [changePasswordTitle](#)
- [PasswordChange.html / PasswordChange-mobile-webkit.html / PasswordChange-ipad.html](#)

showClose

Boolean indicating whether or not to show the **Close** button.

Object Type Template variable

Included In **Remediate.html**

Usage Required

Example The following code tests **showClose**. If true, the code constructs a button that closes the page when the user clicks it:

```
<% IF showClose %>
  <input name="btnClose" type="button" value=" <% textClose %> "
  onClick="javascript:self.close()">
<% END %>
```

Related Documentation [showContinue](#)

showContinue

Boolean indicates whether or not to show the **Continue** button.

Object Type Template variable

Included In **Remediate.html** , **Remediate-ppc.html** , **Remediate-ipad.html** , **Remediate-mobile-webkit.html**

Usage Required

Example The following code shows a **Continue** button if the **showContinue** variable is true:

```
<% IF showContinue %>
  <input name="btnContinue" type="button" value="<% textContinue %>"
  onClick="Continue()"></td>
<% END %>
```

Related Documentation [Remediate.html / Remediate-mobile-webkit.html / Remediate-ipad.html](#)

showHeading

	flag to determine if we should show heading
Object Type	Template variable
Included In	Remediate.thtml , Remediate-ppc.thtml , Remediate-ipad.thtml , Remediate-mobile-webkit.thtml
Usage	Required
Example	The following code displays message text if the showHeading variable is true:

```
<% IF showHeading %>  
<% msg FILTER verbatim %>  
<% END %>
```

Related Documentation [Remediate.thtml](#) / [Remediate-mobile-webkit.thtml](#) / [Remediate-ipad.thtml](#)

showLoading

	Boolean indicates whether or not to show text during the loading of ActiveX or an applet.
Object Type	Template variable
Included In	Remediate.thtml
Usage	Required
Example	The following code displays message text if the showLoading variable is true:

```
<% IF showLoading %>  
<% pleasewait %>  
<% HC_REMED_POLICIES_CHECK %>
```

Related Documentation [Remediate.thtml](#) / [Remediate-mobile-webkit.thtml](#) / [Remediate-ipad.thtml](#)

showReinstall

	Boolean indicates whether or not to show the Reinstall button.
Object Type	Template variable
Included In	Remediate.thtml , Remediate-ppc.thtml
Usage	Required
Example	The following code displays message text if the showReinstall variable is true:

```
<% IF showReinstall %>  
<% pleasewait %>  
<% HC_REMED_POLICIES_CHECK %>
```

Related Documentation

showRemedOption

Boolean indicates whether or not to show the button that displays the message: Do not show remediation for this session.

Object Type	Template variable
Included In	Remediate.shtml , Remediate-ppc.shtml , Remediate-ipad.shtml , Remediate-mobile-webkit.shtml
Usage	Required
Example	The following code tests the value of showRemedOption . If true, the code hides the remediation button:

```
<% IF showRemedOption %>
<input name="btnHideRemed" type="button"
value="<% textRemedOption %>" onClick="HideRemed(); <% disabled %>">
<% HC_REMED_SHOW_ADV_PREFS %>
<% END %>
```

Related Documentation [Remediate.shtml](#) / [Remediate-mobile-webkit.shtml](#) / [Remediate-ipad.shtml](#)

showPolicies

Boolean indicating whether or not to show remediation policies.

Object Type	Template variable
Included In	Remediate.shtml , Remediate-ppc.shtml , Remediate-ipad.shtml , Remediate-mobile-webkit.shtml
Usage	Required
Example	The following code tests the values of several template variables:

```
<% IF showHeading || showPolicies || showButtons %>
```

Related Documentation [Remediate.shtml](#) / [Remediate-mobile-webkit.shtml](#) / [Remediate-ipad.shtml](#)

ShowSystemPin.shtml / ShowSystemPin-mobile-webkit.shtml / ShowSystemPin-ipad.shtml

Displays the system-generated PIN to the user.

Object Type	Template
Usage	Optional
Required Variables	None
Example	None

showTryAgain

	Boolean that indicates whether or not to show the Try Again button.
Object Type	Template variable
Included In	Remediate.thtml
Usage	Required
Example	The following code checks the variable and, if true, displays the Try Again button. The code also disables the button once the end-user has clicked it:

```
<% IF showTryAgain %>
<input name="btnTryAgain" type="button" value="<% textTryAgain %>"
onClick="TryAgain()" <% disabled %>>
<% END %>
```

Related Documentation [TryAgain](#)

signinAgain

	Boolean that indicates whether or not to display the login page after users log out.
Object Type	Template variable
Included In	Logout-ipad.thtml, Logout-mobile-webkit.thtml, Logout-ppc.thtml, Logout.thtml
Usage	Optional
Example	The following code checks the variable and, if true, redirects the user to the sign-in page (welcome.cgi).

```
<% IF signinAgain %>
<a href="welcome.cgi" ><div id=signindiv>
<% signinAgain FILTER verbatim %>
```

Related Documentation None

signinAgain

	Boolean that indicates whether or not to display the login page after users log out.
Object Type	Template variable
Included In	Logout.thtml
Usage	Optional
Example	The following code checks the variable and, if true, redirects the user to the sign-in page (welcome.cgi).

```
<% IF signinAgain %>
<a href="welcome.cgi" ><div id=signindiv>
<% signinAgain FILTER verbatim %>
```

Related Documentation None

signinAgainUrl

Contains the signin url to be used after the user logs out.

Object Type Template variable

Included In **Logout-ipad.thtml, Logout-mobile-webkit.thtml, Logout.thtml**

Usage Optional.

Example `<% <% signinAgainUrl %> %>`

Related Documentation

[SigninNotifPostAuth.thtml / SigninNotifPreAuth-mobile-webkit.thtml / SigninNotifPostAuth-ipad.thtml](#)

Displays Sign-in Notification to user and asks for agreement to proceed.

Object Type Template

Usage You can implement pre-auth sign-in notifications, post-auth sign-in notifications, or both.

Required Variables postAuthSNText

Example None

[SigninNotifPostAuth-new-ux.thtml](#)

Displays the Template page for Sign-in Notification(Post Auth).

Object Type Template

Usage Optional. Displays page capturing post auth sign-in notifications. Required in case Post Auth sign-in notifications are configured and needs customization.

Required Variables postAuthSNText
postAuthSNProceed
postAuthSNDecline

[SigninNotifPreAuth.thtml / SigninNotifPreAuth-ipad.thtml](#)

Displays Sign-in Notification to user and asks for agreement to proceed.

Object Type Template

Usage You can implement pre-auth sign-in notifications, post-auth sign-in notifications, or both.

Required Variables preAuthSNText

Example None

SigninNotifPreAuth-new-ux.shtml

	Displays the Template page for Sign-in Notification(Pre Auth).
Object Type	Template
Usage	Optional. Displays page capturing pre auth sign-in notifications. Required in case Pre Auth sign-in notifications are configured and needs customization.
Required Variables	preAuthSNText preAuthSNProceed preAuthSNDecline

SM-NewPinSelect.shtml / SM-NewPinSelect-mobile-webkit.shtml / SM-NewPinSelect-ipad.shtml

	Prompts the user to enter a new PIN or create a system-generated PIN before signing into Pulse Connect Secure.
Object Type	Template
Usage	Required for SiteMinder authentication.
Required Variables	"secid_loginmode"
Example	None

SM-NewPinSystem.shtml / SM-NewPinSystem-mobile-webkit.shtml / SM-NewPinSystem-ipad.shtml

	Enables the user to create a system-generated PIN if the user selects the System PIN option in the SM-NewPinSelect.shtml page.
Object Type	Template
Usage	Optional
Required Variables	"secid_loginmode"
Example	None
Related Documentation	SM-NewPinSelect.shtml / SM-NewPinSelect-mobile-webkit.shtml / SM-NewPinSelect-ipad.shtml

SM-NewUserPin.shtml

	Prompts the user to create a new PIN if he selects the Enter PIN option in the SM-NewPinSelect.shtml page.
Object Type	Template
Usage	Optional. Also determines if the two PINs entered by the user match and alerts the user if necessary. When configuring the secid_pinserr variable (which stores the error code and message that informs the user if he mis-entered his PIN), note that possible values include:

Table11: New PIN Assignment Messages

Code	Value
0	New PIN Assignment
1	The Two PINs Entered Do Not Match

Required Variables	"secid_loginmode"
Example	None
Related Documentation	SM-NewPinSelect.shtml / SM-NewPinSelect-mobile-webkit.shtml / SM-NewPinSelect-ipad.shtml

[SM-NextToken.shtml](#) / [SM-NextToken-mobile-webkit.shtml](#) / [SM-NextToken-ipad.shtml](#)

Prompts the user to verify his credentials by entering his SecureID token code.

Object Type	Template
Usage	Optional. For information the JavaScript, form definitions, form fields, and variables contained in these templates, see comments in the templates.



NOTE: If you prohibit the use of System-generated PINS, you do not need to include **SM-NewPinSelect.shtml** and **SM-NewPinSystem.shtml** in the zip file that you upload to Pulse Connect Secure.

Required Variables	"secid_loginmode"
Example	None
Related Documentation	<ul style="list-style-type: none"> SM-NewPinSelect.shtml / SM-NewPinSelect-mobile-webkit.shtml / SM-NewPinSelect-ipad.shtml SM-NewPinSystem.shtml / SM-NewPinSystem-mobile-webkit.shtml / SM-NewPinSystem-ipad.shtml

[softid_error](#)

Set if the ACE/SoftID authentication server returns errors.

Object Type	Template variable
Included In	GeneratePin.shtml (ACE/softID), LoginPage.shtml (ACE/softID), NewPin.shtml (ACE/softID), NextToken.shtml (ACE/softID)
Usage	Required
Example	The following code assigns any SoftID errors to a JavaScript variable:

```
var error= "<% softid_error %>";
```

Related Documentation	softid_time
------------------------------	-----------------------------

softid_time

	Contains the ACE/SoftID authentication server time.
Object Type	Template variable
Included In	LoginPage.thtml (ACE/softID)
Usage	Required by authenticate() function.
Example	The following code assigns the ACE/SoftID time to a time variable:

```
var time = <% softid_time %> ;
```

Related Documentation [LoginPage.thtml](#) / [LoginPage-mobile-webkit.thtml](#) / [LoginPage-ipad.thtml](#)

SSL.thtml / SSL-mobile-webkit.thtml / SSL-ipad.thtml

	Displays an error if authentication fails and the user is not allowed to sign into Pulse Connect Secure.
Object Type	Template
Usage	You must always include SSL.thtml in your zip file. This is a standard Pulse Connect Secure page that displays an error if authentication fails or realm-level checks fail and the user is not allowed to sign into Pulse Connect Secure. This template does not contain any JavaScript or forms. However, it does contain error message variables. For more information, see comments in the template.
Required Variables	None
Example	None

SSL-new-ux.thtml

	Displays the Template page that displays an error if authentication fails.
Object Type	Template
Usage	Required
Required Variables	LoginPageErrorMessage

Start_onclick()

	Initiates the proper applets when beginning a meeting session.
Object Type	JavaScript function
Included In	MeetingRun.thtml
Usage	Required
Example	The following code shows the Start_onclick() function:

```
function Start_onclick() {
  if ( typeof(NeoterisSetup) != "undefined") {
    try {
      NeoterisSetup.startSession();
      return 0;
    } catch (e) {
      return Start_onclick_applet();
    }
  } else {
    return Start_onclick_applet();
  }
}
```

Related Documentation [MeetingRun.shtml](#)

Start_onclick_java()

Configures the session manager.

Object Type JavaScript function

Included In **MeetingRun.shtml**

Usage Required. Called by various pages from which the user starts and stops meeting sessions.

Example The following code illustrates how the function begins to configure the session manager:

```
function Start_onclick_java() {
  var sOptions = "resizable=no,scrollbars=no,status,width=250,height=150";
  var x,y;
  if (screen) {
    if ((screen.availWidth) && (screen.availHeight)) {
      x = screen.availWidth - 290; // 40 pixel slop
      y = screen.availHeight - 190; // 40 pixel slop
      sOptions += ",left=" + x + ",screenX=" + x + ",top=" + y + ",screenY=" + y;
    }
  }
}
```

Related Documentation [MeetingRun.shtml](#)

start_status

Status of the started Host Checker or Cache Cleaner instance.

Object Type JavaScript variable

Included In **PleaseWait.shtml**

Usage Required. The return status equals 1 if a test of this variable fails.

Example The following code shows how the `getStatus` function uses the `start_status` variable:

```
function getStatus()
```

```

    {
      return start_status;
    }

```

Related Documentation [PleaseWait.html](#) / [PleaseWait-mobile-webkit.html](#) / [PleaseWait-ipad.html](#)

StartCC()

Enables the current user to start Cache Cleaner.

Object Type JavaScript function

Included In **PleaseWait.html**

Usage Optional

Example None

Related Documentation

- [PleaseWait.html](#) / [PleaseWait-mobile-webkit.html](#) / [PleaseWait-ipad.html](#)

StartHC()

Enables the current user to start Host Checker.

Object Type JavaScript function

Included In **PleaseWait.html**

Usage Optional

Example None

Related Documentation

- [PleaseWait.html](#) / [PleaseWait-mobile-webkit.html](#) / [PleaseWait-ipad.html](#)
- [StartCC\(\)](#)

started

Boolean indicating that a meeting instance has started.

Object Type Template variable

Included In **MeetingSelect.html**

Usage Required

Example The following code checks to see if meetings have started, and if so, populates the page with sign-in components corresponding to each meeting.

```

<% IF rows %>
<%FOREACH meeting = meetings%>
<input type="button" value=" &gt; "onclick=
"self.location=/dana-na/meeting/<%signin%>login_meeting.cgi?mid=<%mid%>
&occurrence=<%meeting.3%><%params%>">

```

```
<% started %>
<%meeting.0%>
<%meeting.1%>
<%meeting.2%>
<% END %>
<%END%>
```

Related Documentation [MeetingSelect.shtml](#)

startPageLink

Page Link for the Welcome Page.

Object Type Template variable

Included In **GraceLoginUsed.shtml, PasswordExpiration.shtml**

Usage Required. Must be added as an **href** link. Provides a link in the page, allowing user to navigate to the Welcome page.

Example The following code shows this variable used as an **href** link:

```
<a href="<% startPageLink %>">Click here to go to Welcome Page</a>
```

Related Documentation

- [GraceLoginUsed.shtml / GraceLoginUsed-mobile-webkit.shtml / GraceLoginUsed-ipad.shtml](#)
- [PasswordExpiration.shtml / PasswordExpiration-mobile-webkit.shtml / PasswordExpiration-ipad.shtml](#)

startSessionReDir

Initiates a redirect on finding that a meeting applet needs to be installed.

Object Type Template variable

Included In **MeetingAppletInstaller.shtml, MeetingTrouble.shtml, PleaseWait-mtg.shtml**

Usage Required

Example

```
<PARAM NAME="StartSessionReDir" VALUE="<% startSessionReDir %>">
```

Related Documentation

- [MeetingAppletInstaller.shtml](#)
- [MeetingTrouble.shtml](#)

stopComponents()

Performs a variety of shutdown checks and operations, shutting down Host Checker, Cache Cleaner, and Win32 applets, if necessary, when the end-user signs out.

Object Type	JavaScript function
Included In	Logout.thtml
Usage	Required
Example	<pre><% IF (pleasewaitObjectCC pleasewaitObjectHC pleasewaitWin32) && delivery_mode == 'java'%> onload="javascript:stopComponents()" <%END%></pre>

Related Documentation [delivery_mode](#)

SubmitClicked()

Saves or cancels a PIN per the user's request.

Object Type	JavaScript function
Included In	GeneratePin.thtml, NewPin.thtml, NextToken.thtml, SM-NewUserPin.thtml
GeneratePin.thtml, NewPin.thtml, NextToken.thtml, SM-NewUserPin.thtml Usage	Required. You must include this function in the header of GeneratePin.thtml .

Example From **GeneratePin.thtml**:

```
function SubmitClicked() {
    gCancelNewPinMode = false;
    return true;
}
```

From **SM-NewUserPin.thtml**:

```
function SubmitClicked() {
    if (document.frmNewPin.password2.value !=
document.frmNewPin.password3.value) {
        alert("The Two PINs Entered Do Not Match");
        document.frmNewPin.password2.focus();
        return false;
    }
}
```

Related Documentation

- [GeneratePin.thtml / GeneratePin-mobile-webkit.thtml / GeneratePin-ipad.thtml](#)
- [NewPin.thtml / NewPin-mobile-webkit.thtml / NewPin-ipad.thtml](#)
- [NextToken.thtml / NextToken-mobile-webkit.thtml / NextToken-ipad.thtml](#)
- [SM-NewUserPin.thtml](#)

submitFrmCasque()

Initiates the download of the CASQUE RADIUS server.

Object Type JavaScript function

Included In **Defender.thtml**

Usage Required if CASQUE RADIUS server is needed.

Example The following code shows the action to initiate the CASQUE server:

```
function submitFrmCasque() {  
    document.frmCasque.action = "/dana-na/download/x.casque?  
    url=/dana-na/auth/welcome.cgi";  
    document.frmCasque.submit();  
}
```

Related Documentation

- [Defender.thtml / Defender-mobile-webkit.thtml / Defender-ipad.thtml](#)
- [frmCasque](#)

testjava_ini

Boolean indicating whether or not Pulse Connect Secure needs to perform a compatibility check for Java.

Object Type Template variable

Included In **MeetingTestJava.thtml**

Usage Required

Example

```
<APPLET  
<% testjava_ini FILTER verbatim %>  
<PARAM NAME="TestJavaApplet" VALUE="true">  
<PARAM NAME="RedirectUri"  
VALUE="/danana/meeting/<%signin%>meeting_testresult.cgi  
?java=1&<%mid_param%>">  
</APPLET>
```

Related Documentation [MeetingTestJava.thtml](#)

textClose

Default text for the **Close** button.

Object Type Template variable

Included In **Remediate.thtml**

Usage Required. Used by Pulse Connect Secure to dynamically update the button name, depending upon the user's activity.

Example The following code changes the button to a **Close** button:

```
<% IF showClose %>
<input name="btnClose" type="button" value=" <% textClose %> "
onClick="javascript:self.close()">
<% END %>
```

Related Documentation

- [textContinue](#)
- [textRemedOption](#)
- [textTryAgain](#)

textContinue

Default text for the **Continue** button.

Object Type Template variable

Included In **Remediate.thtml**

Usage Required. Used by Pulse Connect Secure to dynamically update the button name, depending upon the user's activity.

Example

```
<% IF showContinue %>
<input name="btnContinue" type="button" value="<% textContinue %>"
onClick="Continue()">
<% END %>
```

Related Documentation

- [textClose](#)
- [textRemedOption](#)
- [textTryAgain](#)

textRemedOption

Default text for the **Do not show remediation for this session** button.

Object Type Template variable

Included In **Remediate.thtml**

Usage Required. Used by Pulse Connect Secure to dynamically update the button name, depending upon the user's activity.

Example

```
<% IF showRemedOption %>
<input name="btnHideRemed" type="button"
```

```
value="<% textRemedOption %>"
onClick="HideRemed(); <% disabled %>"
<% HC_REMED_SHOW_ADV_PREFS %>
<% END %>
```

Related Documentation

- [textClose](#)
- [textContinue](#)
- [textTryAgain](#)

textTryAgain

Default text for the **Try Again** button.

Object Type Template variable

Included In **Remediate.thtml**

Usage Required. Used by Pulse Connect Secure to dynamically update the button name, depending upon the user's activity.

Example

```
<% IF showTryAgain %>
<input name="btnTryAgain" type="button" value="<% textTryAgain %>"
onClick="TryAgain()" <% disabled %>>
<% END %>
```

Related Documentation

- [textClose](#)
- [textContinue](#)
- [textRemedOption](#)

thankyou()

Removes components and displays the login page after an end-user signs out.

Object Type JavaScript function

Included In **Logout.thtml**

Usage Optional

Example The function checks to see which application is being stopped and removed, then displays either an error or the sign-in page.

```
function thankyou()
{
<%IF pleasewaitObjectHC %>
removeBulb('hc');
removeComponent('hc');
<%END%>
```

```
<%IF pleasewaitObjectCC %>
removeBulb('cc');
removeComponent('cc');
<%END%>
document.getElementById('pleasewait1').innerHTML =
"<% LoginPageErrorMessage FILTER verbatim%>";
document.getElementById('signindiv').innerHTML =
"Click here to sign in again";
}
```

Related Documentation [setthankyou\(\)](#)

TotpAuthRegister-new-ux.shtml

Displays the Template page for registering with TOTP Auth server.

Object Type	Template
Usage	Optional. Displays page capturing TOTP Auth registration. Required in case TOTP Auth server is configured and requires customization.
Required Variables	totp_key totp_user_secret_key totp_user_backup_codes_string totp_user_name totp_only_username totp_issuer

TotpAuthTokenEntry-new-ux.shtml

Displays the Template page for TOTP Auth token.

Object Type	Template
Usage	Optional. Displays page for providing TOTP Auth token for authentication. Required in case TOTP Auth server is configured and requires customization.
Required Variables	totp_key

troubleshooting

Contains the troubleshooting text string.

Object Type	Template variable
Included In	MeetingAppletInstaller.shtml, MeetingRunJava.shtml
Usage	Required
Example	The following code checks to see if there is a troubleshooting string. If yes, then it displays the text string, as is:

```
<% IF troubleshooting %>
<% troubleshooting FILTER verbatim %>
<% END %>
```

Related Documentation

- [MeetingAppletInstaller.shtml](#)
- [MeetingRunJava.shtml](#)

TryAgain

Submits the hidden form when the user clicks the **Try Again** button.

Object Type JavaScript function

Included In **Remediate.shtml**

Usage Required

Example

```
<% IF showTryAgain %>
<input name="btnTryAgain" type="button" value="<% textTryAgain %>"
onClick="TryAgain()" <% disabled %>>
<% END %>
```

Related Documentation [showTryAgain](#)

type

Contains the type of client to use when running Meeting.

Object Type Template variable

Included In **MeetingTrouble.shtml**

Usage Required. Allowable values are:

- activex
- java
- applet

Example

The following code checks to see if the type of client is ActiveX:

```
<% IF type == 'activex' %>
<OBJECT
<% ini FILTER verbatim %>
<PARAM NAME="StartSessionReDir" VALUE="<%startSessionReDir%>">
<PARAM NAME="UninstallReDir" VALUE="<%uninstallReDir%>">
<PARAM NAME="locale" VALUE="<%locale%>">
</OBJECT>
<% END %>
```

Related Documentation [MeetingTrouble.shtml](#)

tz_offset

	Contains the time zone offset.
Object Type	HTML form field
Included In	LoginPage.shtml
Usage	Required. The Login() function uses the tz_offset (time zone offset) value to help determine the user's time zone.
Example	The following code shows how the tz_offset field is used in an HTML form.

```
<input type="hidden" name="tz_offset">
```

Related Documentation [LoginPage.shtml](#) / [LoginPage-mobile-webkit.shtml](#) / [LoginPage-ipad.shtml](#)

uninstall_action

	Indicates the type of action to take when having trouble launching a meeting instance.
Object Type	Template variable
Included In	MeetingTrouble.shtml
Usage	Required
Example	The following code illustrates the use of the uninstall_action variable:

```
<% IF type == 'activex' || type == 'applet' %>  
If you encounter problems launching Pulse Secure Collaboration, uninstall the version  
that is currently on your system.  
<input type="button" value="Uninstall" id=Uninstall name=button2  
LANGUAGE=javascript onclick="<% uninstall_action %>">  
After uninstalling, return to the "Join Meeting" page and click "Start Meeting"  
or "Join Meeting" to reinstall the latest Pulse Secure Collaboration application.<br>  
<% END %>
```

Related Documentation [Uninstall_applet\(\)](#)

Uninstall_applet()

	Initiates the uninstallation of an applet.
Object Type	JavaScript function
Included In	MeetingTrouble.shtml
Usage	Required. Note that the first character is upper cased.
Example	The following code is the uninstall_applet() function:

```
Uninstall_applet() {
  window.location = "/dana-na/meeting/meeting_appletuninstall.cgi?
  mid=<%mid%>&occurrence=<%occurrence%>&type=<%type%>";
  return true;
}
```

Related Documentation [MeetingTrouble.shtml](#)

Uninstall_onclick()

Initiates the uninstall script.

Object Type JavaScript function

Included In **MeetingTrouble.shtml**

Usage Required. Note that the first character is upper cased.

Example The following code is the **uninstall_onclick()** function:

```
Uninstall_onclick() {
  if ( typeof(NeoterisSetup) != "undefined" ) {
    NeoterisSetup.uninstall();
    return 1;
  }
  else {
    return 0;
  }
}
```

Related Documentation [MeetingTrouble.shtml](#)

uninstallReDir

Contains the location of the page to go to on an uninstall.

Object Type Template variable

Included In **MeetingAppletInstaller.shtml, MeetingTrouble.shtml**

Usage Required

Example The following code assigns the uninstall redirect page to a form parameter:

```
<PARAM NAME="UninstallReDir" VALUE="<% uninstallReDir %>">
```

Related Documentation

- [MeetingAppletInstaller.shtml](#)
- [MeetingTrouble.shtml](#)

UninstallCC

	Indicates whether or not Cache Cleaner uninstall is enabled.
Object Type	Template variable
Included In	Logout.thtml
Usage	Required. Not to be confused with JavaScript variable <code>uninstallcc</code> .
Example	The following code checks to see that the Cache Cleaner uninstall is enabled. If yes, then the code sets the JavaScript <code>uninstallcc</code> variable to 1, which indicates that the system should uninstall if the JavaScript function <code>stopCC()</code> is running:

```
<%IF uninstallCC %>  
  uninstallcc = 1;  
<%END%>
```

Related Documentation [StartCC\(\)](#)

upAndRunning()

	Indicates whether or not Host Checker or Cache Cleaner are up and running after they are started.
Object Type	JavaScript function
Included In	PleaseWait.thtml
Usage	Optional
Example	None
Related Documentation	PleaseWait.thtml / PleaseWait-mobile-webkit.thtml / PleaseWait-ipad.thtml

userAgentTypeAndroid

	Boolean indicating whether or not the end-user's operating system is an Android or iOS system.
Object Type	Template variable
Included In	PleaseWait-ipad.thtml , PleaseWait-mobile-webkit.thtml
Usage	Optional (Required for Pulse HC on Android or iOS end systems)
Example	<pre><% userAgentTypeAndroid %></pre>

Related Documentation

username

This field is required for all types of authentication servers except anonymous servers and certificate servers.

Object Type	Template variable
Included In	LoginMeeting.thtml, MeetingAppletInstaller.thtml
Usage	Required. As an example of username's use, Login.cgi (a Perl script on Pulse Connect Secure) passes the username value to the appropriate authentication server.
Example	The following example shows the use of the username variable in a meeting sign-in sequence:

```
<PARAM NAME="Parameter0"  
VALUE="meeting_id=<%instanceId%>;user_name=<% username %>;  
cert_md5=<%cert_md5%>;ncp_read_timeout=<%ncp_read_timeout%>">
```

Related Documentation [password](#)

warnings

Contains meeting warnings.

Object Type	Template variable
Included In	MeetingRun.thtml
Usage	Required
Example	The following code is included in MeetingRun.thtml . You should not modify the code:

```
<% warnings FILTER verbatim %>
```

Related Documentation [MeetingRun.thtml](#)

welcome

Contains the welcome message string.

Object Type	Template variable
Default Value	"Welcome to the"
Included In	LoginPage.thtml, LogoutPage.thtml, SSL.thtml, SecondaryLoginPage.thtml, Logout.thtml, Defender.thtml, ExceededConcurrent.thtml, GeneratePin.thtml, NewPin.thtml, NextToken.thtml, ShowSystemPin.thtml
Usage	Optional
Example	<pre><form name="frmLogin" action=login.cgi method="POST" autocomplete=off onsubmit="return Login(<% setcookies %>)"> <input type="hidden" name="tz_offset"> <% welcome FILTER verbatim %></pre>

Related Documentation None

win32

	Indicates whether or not the client operating system is Windows.
Object Type	Template variable
Included In	Logout.shtml, MeetingTestJS.shtml
Logout.shtmlUsage	Required
Example	The following code checks to see if the operating system is Windows. If yes, the code loads and redirects the page to a page to test the compatibility of Microsoft Java:

```
function checkActiveX () {
var win = <% win32 %>;
try {
  NeoterisSetup.isValid();
  document.form1.submit();
  return true;
} catch (e) {
var url = win ? "/dana-na/meeting/
<%signin%>meeting_testmsjava.cgi?
<%mid_param%>" : "/dana-na/meeting/<%signin%>
meeting_testjava.cgi?<%mid_param%>";
document.location.replace(url);
}
}
```

Related Documentation [MeetingTestMSJava.shtml](#)

Index

- [Index](#)

A

- Accept-Language.....
- Accept-Language header abbreviations
- ActivePerl.....
- AnonymousAuthentication variable, discussed
- AppConfig.....
- arithmetic operators, defined
- authentication servers
 - SiteMinder
 - custom sign in pages ...

B

- block directives, defined.....

C

- Cache Cleaner
 - custom pages.....
- CALL directive, restriction
- Cancel.html, discussed.....
- CASE directive, discussed...
- CertificateAuthentication variable, discussed
- conditional operators, discussed
- conditional test, creating
- custom help files
- custom sign-in pages
- localizing
- customer support
- contacting PSGSC
- Cygwin.....

D

- Defender.html, discussed....
- directive, defined.....

E

- ELSIF directives, discussed .
- END directive, discussed
- errors
 - modifying.....
- ExceededConcurrent.html, discussed

F

- FILTER directive, restriction .
- FinishLoad() function, discussed
- FOREACH loop, discussed..

G

- GeneratePin.html, discussed
- GET directive, discussed.....
- getTimezoneOffset() function, discussed
- GraceLoginUsed.html, discussed

H

- Home variable, discussed....
- Host Checker
 - custom pages.....

I

- IF directive, discussed
- INCLUDE.....
- INSERT
- internationalization.....
- INTERPOLATE directive, restriction

J

- J-SAM See Secure Application Manager, J-SAM

K

- keyboard.js.....
- Kiosk.zip.....
- Kiosk.zip, discussed

L

- languages.....
- Linux.....
- localization.....
- Login() function, discussed..
- Login.cgi, discussed
- LoginPage.html, customizing
- LoginPage.html, discussed .
- LoginPageErrorCode.....
- LoginPageErrorCode variable, discussed
- LoginPageErrorMessage variable, discussed
- Logout.html, discussed.....

M

- Mac OS X.....
- mathematical operators, defined
- Meeting.zip

N

NewPin.html, discussed
 NextToken.html, discussed .

P

PasswordChange.html, discussed
 PasswordExpiration.html, discussed
 Perl AppConfig.....
 PERL directive, restriction....
 custom sign in pages.....
 PROCESS

R

RAWPERL directive, restriction
 RealmList variable, discussed
 recallLastRealmUsed() function, discussed
 RSA Soft ID client

RSA SoftID client, custom pages

S

sample templates
 downloading

Sample.zip

samples.zip, discussed

secid_pinselectmode variable, discussed
 secid_pinserr variable, discussed

Secure Application Manager
 J-SAM
 custom pages

SET directive, discussed.....

SetLastRealm(sValue) function, discussed

ShowSystemPin.html, discussed

sign-in page, customizing.....

SigninNotifPostAuth.html, discussed
 SigninNotifPreAuth.html, discussed
 SM-NewPinSelect.html, discussed
 SM-NewPinSystem.html, discussed
 SM-NewUserPin.html, discussed
 SM-NextToken.html, discussed

SoftID.zip

SoftID.zip, discussed

SSL.html, discussed

support, technical See technical support

SWITCH directive, discussed

T

TAGS directive, restriction ...
 technical support

 contacting PSGSC

template comments

Template Toolkit.....

Template Toolkit documentation

template toolkit language, discussed

template variable
 LoginPageErrorCode

html files

tpage

ttree

tz_offset variable, discussed
 tz_offset variable, discussed

U

USE directive, restriction

W

WHILE loop, discussed

Windows.....