



**Professional Services**

**Nuance Integration Web Service**

**Kaiser Permanente**

**16th November, 2018**

# Copyright Information

©2018. SecureAuth® is a registered trademark of SecureAuth Corporation. SecureAuth’s IdP software, appliances, and other products and solutions are copyrighted products of SecureAuth Corporation.

Version 1.0

October 2017

# Table of Contents

[Copyright Information 2](#_Toc529443881)

[Table of Contents 3](#_Toc529443882)

[1. Introduction 4](#_Toc529443883)

[2. Document Usage 4](#_Toc529443884)

[3. Classic Platform Integration Overview 4](#_Toc529443885)

[4. Provisioning Platform Integration Overview 4](#_Toc529443886)

[5. Nuance Caller Processing Sequence 5](#_Toc529443887)

[Registration 5](#_Toc529443888)

[Password Reset 6](#_Toc529443889)

[6. User Registration Status 6](#_Toc529443890)

[Is User Registered Request 6](#_Toc529443891)

[Is User Registered Response 7](#_Toc529443892)

[7. PIN Validation 7](#_Toc529443893)

[Validate PIN Request 7](#_Toc529443894)

[Validate PIN Response 8](#_Toc529443895)

[8. Set Secret Date 8](#_Toc529443896)

[Set Secret Date Request 8](#_Toc529443897)

[Set Secret Date Response 9](#_Toc529443898)

[9. Authenticate User 9](#_Toc529443899)

[Authenticate User Request 9](#_Toc529443900)

[Authenticate User Response 10](#_Toc529443901)

[10. Password Reset 10](#_Toc529443902)

[Reset Password Request 10](#_Toc529443903)

[Reset Password Response 11](#_Toc529443904)

[11. Current v7.x ASP Interfaces 11](#_Toc529443905)

[12. Sample v7.x Dialog Trace 13](#_Toc529443906)

## 

# Introduction

The QuickFactor SOAP web service interface allows external applications to submit requests to the Access Assurance Suite (AAS) for processing.

# Document Usage

This document is intended to be used by SecureAuth and Nuance software developers for interfacing to the AAS system

# Classic Platform Integration Overview

This provides an overview of the current production solution on AAS v7.x



# Provisioning Platform Integration Overview

This provides an overview of the planned production solution on AAS v9.1.2



# Nuance Caller Processing Sequence

## Registration

1. **Is User Registered**
   1. Caller calls designated number
   2. Caller enters NUID
   3. Nuance checks ServiceNow for Caller status
      1. Caller must be ‘active’ to proceed
   4. Nuance checks registration status with AAS
      1. Registered: PIN=’disabled’
      2. Not Registered: PIN≠’disabled’
   5. If not registered, Caller is to record their voice print
2. **Validate PIN**
   1. Caller enters PIN
   2. Nuance requests AAS to verify the PIN
3. **Set Secret Date**
   1. Caller enters Secret Date
   2. Nuance requests AAS to store Secret Date
      1. AAS stores Secret Date (encrypted)
      2. AAS sets PIN=’disabled’
   3. Nuance records a hint for the Secret Date
   4. Nuance stores the voice recording directly in the database table ‘IVR\_Answers’
4. **Registration is complete.**

## Password Reset

1. **Is User Registered**
   1. Caller calls designated number
   2. Caller enters NUID
   3. Nuance checks ServiceNow for Caller status
      1. Caller must be ‘active’ to proceed
   4. Nuance checks registration status with AAS
      1. Registered: PIN=’disabled’
      2. Not Registered: PIN≠’disabled’
   5. If registered, Caller can proceed to reset their password
2. **Authenticate User**
   1. Caller enters their Secret Date
   2. Nuance requests AAS to authenticate Caller by Secret Date
      1. AAS returns set of password reset groups
3. **Password Reset**
   1. Caller selects password group to reset
   2. Nuance generates a random password
   3. Nuance requests AAS to reset the target group with the random password
   4. AAS checks profile Status=‘active’
      1. Return failure if not active
4. **Password reset is complete.**

# User Registration Status

This process will:

1. Verify the user profile status is ‘Active’
2. Check the PIN status (encrypted)
   1. Registered: PIN=’disabled’
   2. Not Registered: PIN≠’disabled’

## Is User Registered Request

POST /QuickFactorServer2.0.2/Service.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://quickfactor.com/SPC\_CheckRegistration"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<SPC\_CheckRegistration xmlns="http://quickfactor.com/">

<sessionID>string</sessionID>

<NUID>string</NUID>

</SPC\_CheckRegistration>

</soap:Body>

</soap:Envelope></soap:Envelope>

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

## Is User Registered Response

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<SPC\_CheckRegistrationResponse xmlns="http://quickfactor.com/">

<sessionID>string</sessionID>

<SPC\_CheckRegistrationStatus>string</SPC\_CheckRegistrationStatus> -- String = 0 (fail),1 (success)

</SPC\_CheckRegistrationResponse>

</soap:Body>

</soap:Envelope>

# PIN Validation

This process will:

1. Authenticate the user using the PIN for initial Nuance registration
   1. The PIN is encrypted in the SecureAuth database

## Validate PIN Request

POST /QuickFactorServer2.0.2/Service.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://quickfactor.com/SPC\_ValidatePIN"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<SPC\_ValidatePIN xmlns="http://quickfactor.com/">

<sessionID>string</sessionID>

<NUID>string</NUID>

<UserPIN>string</UserPIN>

</SPC\_ValidatePIN>

</soap:Body>

</soap:Envelope>

## Validate PIN Response

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<SPC\_ValidatePINResponse xmlns="http://quickfactor.com/">

<sessionID>string</sessionID>

<SPC\_ValidatePINStatus>string</SPC\_ValidatePINStatus> -- String = 0 (fail), 1 (success)

</SPC\_ValidatePINResponse>

</soap:Body>

</soap:Envelope>HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

# Set Secret Date

This process will:

1. Set the Secret Date that the user registered via Nuance
   1. The Secret Date will be stored encrypted in the SecureAuth database
2. Set the PIN=’disabled’ (encrypted)

## Set Secret Date Request

POST /QuickFactorServer2.0.2/Service.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://quickfactor.com/SPC\_SetSecretDate"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<SPC\_SetSecretDate xmlns="http://quickfactor.com/">

<sessionID>string</sessionID>

<NUID>string</NUID>

<SecretDate>string</SecretDate> -- String = yyyymmdd

</SPC\_SetSecretDate>

</soap:Body>

</soap:Envelope>

## Set Secret Date Response

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<SPC\_SetSecretDateResponse xmlns="http://quickfactor.com/">

<sessionID>string</sessionID>

<SPC\_SetSecretDateRStatus>string</SPC\_SetSecretDateRStatus> -- String = 0 (fail), 1 (success)

</SPC\_SetSecretDateResponse>

</soap:Body>

</soap:Envelope>

# Authenticate User

This process will:

1. Authenticate the user using the Secret Date for registered users in Nuance
2. Return a list of reset groups and targets

## Authenticate User Request

POST /QuickFactorServer2.0.2/Service.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://quickfactor.com/SPC\_AuthenticateUser"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<SPC\_AuthenticateUser xmlns="http://quickfactor.com/">

<sessionID>string</sessionID>

<NUID>string</NUID>

<SecretDate>string</SecretDate> -- String = yyyymmdd

</SPC\_AuthenticateUser>

</soap:Body>

</soap:Envelope>

## Authenticate User Response

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<SPC\_AuthenticateUserResponse xmlns="http://quickfactor.com/">

<sessionID>string</sessionID>

<SPC\_AuthenticateUserStatus>string</SPC\_AuthenticateUserStatus> -- String = 0 (fail), 1 (success)

<SPC\_TargetGroups>string</SPC\_ TargetGroups> -- Colon delimited list

</SPC\_AuthenticateUserResponse>

</soap:Body>

</soap:Envelope>

# Password Reset

This process will:

1. Reset the account password to the Nuance generated password

## Reset Password Request

POST /QuickFactorServer2.0.2/Service.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://quickfactor.com/SPC\_ResetPassword"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<SPC\_ResetPassword xmlns="http://quickfactor.com/">

<sessionID>string</sessionID>

<NUID>string</NUID>

<System>string</System> --- TargetGroup to be reset

<NewPassword>string</NewPassword>

</SPC\_ResetPassword>

</soap:Body>

</soap:Envelope>

## Reset Password Response

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<SPC\_ResetPasswordResponse xmlns="http://quickfactor.com/">

<sessionID>string</sessionID>

<SPC\_ResetPasswordStatus>string</SPC\_ResetPasswordStatus> -- String = 0 (fail), 1 (success)

</SPC\_ResetPasswordResponse>

</soap:Body>

</soap:Envelope>

# Current v7.x ASP Interfaces

The following are provided for reference purposes only.

**IVR\_ValidatePIN.asp**

* Variable Assignment:
  + Status = -1
  + 'this is the right product and currently supports ValidateUser + Profile + TargetList

ProductID = "PRO\_BLDR"

* + AccountDisable = "false"
  + NUID = Trim(Request("NUID"))
  + UserPIN = Trim(Request("PIN"))
  + Client\_IP = Request.ServerVariables("REMOTE\_HOST")
* Functions invoked with the variables
  + SessionID = PB\_ctrl.Init( 0, ProductID, Client\_IP )
  + SessionID = PWC\_ctrl.Init( SessionID, ProductID, Client\_IP )
  + Result = PWC\_ctrl.PBAuthenticateUser( SessionID, NUID + "," + UserPIN, AccountDisable )
* Here’s what gets rendered in the page.
  + If combination of NUID and UserPIN is authenticated and account is not disabled, Result returned is empty. Otherwise, an error message is returned. Status returned is 0 if authenticated, otherwise, Status is set to 1. A status of -1 can also be returned if SessionID returned is either 0 or -1.
  + <body>

       <center>

              <h1>IVR Validate PIN Page</h1>

              <h2>Returning status:  <%=Status%></h2>

              <h3>Returning session ID:  <%=SessionID%></h3>

              <status><%=Status%></status>

              <result><%=Result%></result>

              <sessionID><%=SessionID%></sessionID>

       </center>

</body>

**IVR\_SetSecretDate.asp**

* Variable Assignment:

*Note: NUID is being pulled from the request but I didn’t see any reference to it in any of the function calls.*

* + ProductID = "PRO\_BLDR"
  + Client\_IP = Request.ServerVariables("REMOTE\_HOST")
  + status = -1
  + NUID = Trim(Request("NUID"))
  + secretDate = Trim(Request("SecretDate"))
  + sessionID = Trim(Request("SessionID"))
  + 'this works after trial+error, in the end it's just a string

'the first value ties the new value to a field in database

sFieldValues = "0,"+secretDate

* Functions invoked with the variables
  + sessionID = PWC\_ctrl.Init(sessionID, ProductID, Client\_IP )
  + Result = PWC\_ctrl.UpdateProfile( sessionID, sFieldValues, 0, "SUCCESS" )
* Here’s what gets rendered in the page.
  + Status is set to 0 if Successful, otherwise set to 1. Result will contain the word “successful” if the update succeeded.
  + <body>

       <center>

              <h1>IVR Secret Date Page</h1>

              <h2>Status:  <%=status%></h2>

              <h3>Result:  <%=Result%></h3>

              <status><%=status%></status>

              <result><%=Result%></result>

              <sessionID><%=sessionID%></sessionID>

       </center>

</body>

**IVR\_AuthenticateUser.asp**

* Variable Assignment:
  + Status = -1
  + ProductID = "PWD\_SS\_COUR"
  + AccountDisable = "false"
  + NUID = Trim(Request("NUID"))
  + SecretDate = Trim(Request("SecretDate"))
  + Client\_IP = Request.ServerVariables("REMOTE\_HOST")
* Functions invoked with the variables:
  + SessionID = PB\_ctrl.Init( 0, ProductID, Client\_IP )
  + SessionID = PWC\_ctrl.Init( SessionID, ProductID, Client\_IP )
  + PWC\_ctrl.ValidateUser( SessionID, NUID, AccountDisable )
  + Target\_ctrl.AuthenticateUser(SessionID, SecretDate, AccountDisable)
* Here’s what gets rendered in the page.
  + Group names and target names are colon-delimited
  + Status returned is 2 if user was NOT successfully validated through NUID + AccountDisable. If validated successfully, Status returned is 0 if user was authenticated through SecretDate, otherwise, status returned is 1.
  + Result returned is empty if successful.
  + <body>

       <center>

              <h1>IVR Authentication Web Page</h1>

              <h2>Status:  <%=Status%></h2>

              <h3>Number of Groups:  <%=nGroups%></h3>

              <h4>Group Names:  <%=GroupNames%></h4>

              <status><%=Status%></status>

              <result><%=Result%></result>

              <sessionID><%=SessionID%></sessionID>

              <NumberOfGroups><%=nGroups%></NumberOfGroups>

              <GroupName><%=groupNames%></GroupName>

              <TargetName><%=targetNames%></TargetName>

       </center>

</body>

**IVR\_ResetPassword.asp**

* Variable Assignment:

*Note: NUID is being pulled from the request but I didn’t see any reference to it in any of the function calls.*

* + status = -1
  + ProductID = "PWD\_SS\_COUR"
  + Client\_IP = Request.ServerVariables("REMOTE\_HOST")
  + NUID = Trim(Request("NUID"))
  + system = Trim(Request("System"))
  + sessionID = Trim(Request("SessionID"))
  + password = Trim(Request("NewPassword"))
  + targetName = Trim(Request("Target"))
  + comment = "Reset Password"
  + system = Replace(system, "\_\_", " ")
  + targetName = Replace(targetName, "\_\_", " ")
* Functions invoked with the variables
  + sessionID = PWC\_ctrl.Init(sessionID, ProductID, Client\_IP)
  + Result = PWC\_ctrl.StartAction(sessionID, system, targetName, result)
  + Result1 = PWA\_ctrl.ResetPassword(sessionID, password, comment)
* Here’s what gets rendered in the page.
  + If *StartAction* call is successful, Status returned is 0 for Success and 2 for NonSuccess. Result1 throws “Success” on successful reset.
  + <body>

       <center>

              <h1>Reset Password Page</h1>

              <h2>Result1 value: <%=Result1%></h2>

              <status><%=status%></status>

              <result><%=Result1%></result>

       </center>

</body>

# Sample v7.x Dialog Trace

The following are provided for reference purposes only.

**Authenticate User**

2018-11-06 21:35:17,185 | D6864DB7A96DABEFA34C3A3D91450ECC | [http-nio-8080-exec-7] | ERROR | AbstractKaiserBase | AuthenticateUser.execute: request [http://cscrdvapp011.crdc.kp.org/IVR/IVR\_AuthenticateUser.asp?NUID=K249877&SecretDate=19930315]

2018-11-06 21:35:18,498 | D6864DB7A96DABEFA34C3A3D91450ECC | [http-nio-8080-exec-7] | ERROR | AbstractKaiserBase | AuthenticateUser.execute: response [

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Password Express - IVR User Authentication</title>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

</head>

<body>

<center>

<h1>IVR Authentication Web Page</h1>

<h2>Status: 0</h2>

<h3>Number of Groups: 4</h3>

<h4>Group Names: All Mainframe:Sametime:Windows Logon:WSSO</h4>

<status>0</status>

<result></result>

<sessionID>940353763</sessionID>

<NumberOfGroups>4</NumberOfGroups>

<GroupName>All Mainframe:Sametime:Windows Logon:WSSO</GroupName>

<TargetName>All Mainframe:KP-PRO02/PSVR/KAIPERM - Brett M Christenson/CA/KAIPERM:CS - K249877:ssoi.kp.org - uid=K249877,ou=kppeople,ou=people,dc=kp,dc=org</TargetName>

</center>

</body>

</html>

**Password Reset**

]

2018-11-06 21:35:36,029 | D6864DB7A96DABEFA34C3A3D91450ECC | [http-nio-8080-exec-1] | ERROR | AbstractKaiserBase | ResetPassword.execute: request [http://cscrdvapp011.crdc.kp.org/IVR/IVR\_ResetPassword.asp?NUID=K249877&System=Sametime&SessionID=940353763&NewPassword=BqyFZkj1&Target=KP-PRO02/PSVR/KAIPERM\_\_-\_\_Brett\_\_M\_\_Christenson/CA/KAIPERM]

2018-11-06 21:35:39,905 | D6864DB7A96DABEFA34C3A3D91450ECC | [http-nio-8080-exec-1] | ERROR | AbstractKaiserBase | ResetPassword.execute: response [

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Password Express - IVR Reset Password</title>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

</head>

<body>

<center>

<h1>Reset Password Page</h1>

<h2>Result1 value: Success</h2>

<status>0</status>

<result>Success</result>

</center>

</body>

</html>

]