SAP Password Reset Tool

Terry W. Earls
Microsoft Corporation
December 2003

Note: This document is for reference purposes only. No guarantee or warrantee is made in this document.

PROBLEM OVERVIEW 3

SOLUTION OVERVIEW..3

SOLUTION DETAILS...3

Process Flow...3
User Perspective 3
Client Computer, Web Page, Web Services, and IIS Perspectives 3
SAP Perspective...5

ADDITIONAL INFO..6

SAMPLE CODE 8

Generate Random Password Function Module...8

Password Reset Main Function Module 10

Password Reset Wrapper..16

Get RFC Destinations..18

CONTACT INFORMATION...20

Problem Overview

Resetting users' passwords typically consumes a significant amount of support resources. In addition, other than verifying the authenticity of these requests, support personnel spend time manually performing password resets, without adding value to the process.

Solution Overview

If we can verify the credentials of the user making the Password Reset request, we are willing to assume that it is okay to reset that user's password on any of the SAP systems. A user cannot request a Password Reset on behalf of another user.

A solution was developed to verify the user's credentials against the network and to immediately perform an RFC call into the appropriate SAP system to reset the password. Four custom RFCs were created in SAP to accomplish this solution.

Since users operate in a homogeneous NT-domain environment, and web servers are Internet Information Services (IIS), verifying user authenticity is simple. Likewise, the ABAP required for the function modules is straight-forward.

Solution Details

Process Flow

User Perspective

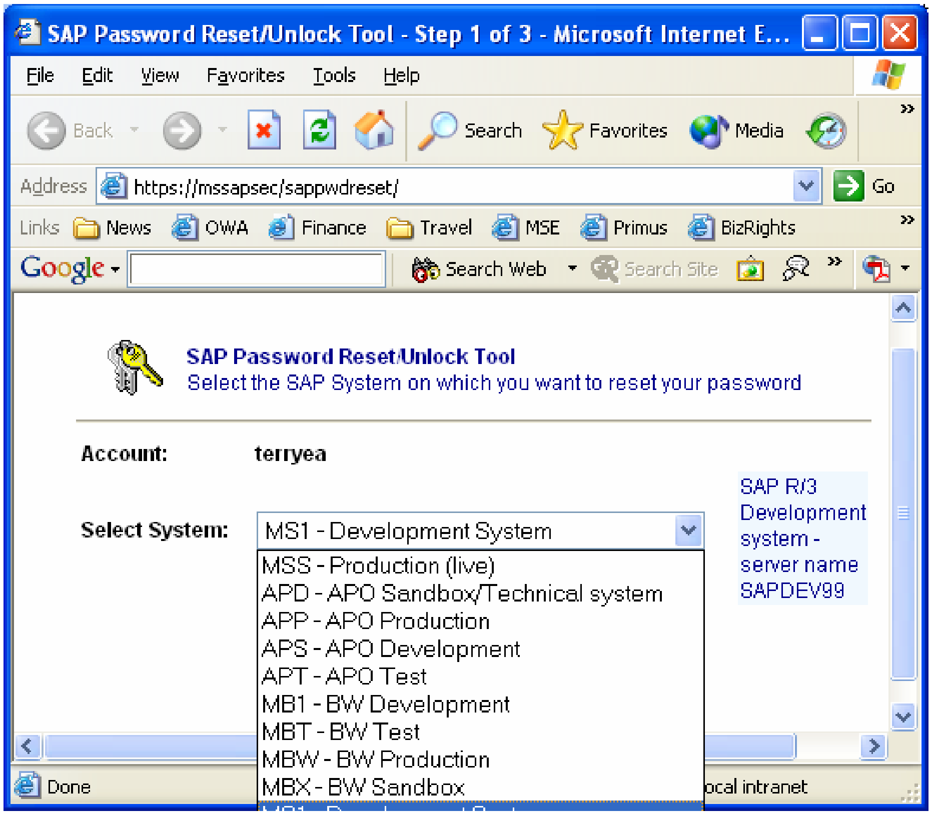
1. User logs into Windows computer (in the morning, for example).
2. User visits an intranet https:// website to reset an SAP password.
3. Website presents a dropdown list of all SAP systems, and user selects the appropriate system.
4. Once the appropriate SAP system is selected, user submits the password reset request.
5. After a few moments, the new password is displayed on the website page. If an error was encountered, the appropriate error message is displayed.
6. Website screen clears after 120 seconds (2 minutes). This gives the user enough time to copy the new password to the clipboard, or otherwise remember the password. The password is displayed as text, so that copy & paste functionality operates as normal in a Windows environment.

Client Computer, Web Page, Web Services, and IIS Perspectives

1. When user logs into the computer on the NT domain, the user's computer receives an "NT credential" which can be queried to verify who the user is. (This is automatic and no programming or special configuration is required.)
2. When the user visits the secure intranet website (using SSL), hosted on an IIS box, several events happen:

    - The webpage (ASP.net) queries the host header to determine the visiting user's identity. (The IIS website will have "anonymous access" disabled.) The user's NT identity is returned (such as "terryea" for a user name). This can be queried in .asp by requesting **Request.ServerVariables("LOGON_USER")**
    - The webpage queries SAP Production through Web Services (or DCOM) to determine which SAP systems are available for password reset.
    - SAP passes back through the Web Services the appropriate list of SAP systems and short descriptions of each.

3. The webpage dynamically builds a drop down box, containing each SAP system available for password resets. The user name is displayed (hard-coded) on the webpage, and as dropdown values are selected, the short description is displayed next to the drop down.



https://mssapsec/sappwdreset/

SAP Password Reset/Unlock Tool

Select the SAP System on which you want to reset your password

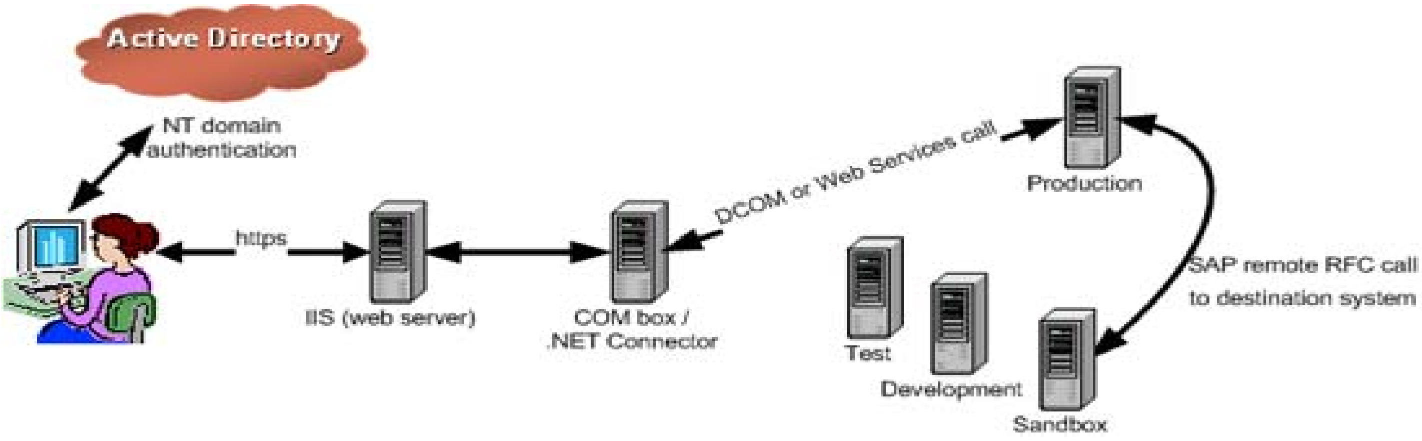
Account: terryea

Select System: MS1 - Development System

MSS - Production (live)
APD - APO Sandbox/Technical system
APP - APO Production
APS - APO Development
APT - APO Test
MB1 - BW Development
MBT - BW Test
MBW - BW Production
MBX - BW Sandbox

SAP R/3 Development system - server name SAPDEV99

4. The user selects the desired system and presses the submit button.
5. The webpage makes a call to a custom RFC in SAP Production, passing the user name and the desired system for password reset.
6. The SAP Production RFC receives this information, and determines if the desired system is Production. If the destination system is not Production, the RFC makes a remote call to the target SAP system, passing along the user name and destination system.
7. When the ultimate SAP system receives the RFC call (either Production or other destination system), it generates a unique, strong password and builds a "Call Transaction" into transaction "SU01_NAV" to reset the user's password. Upon success, the password is passed back along to Production, where the ultimate status notification is returned to the calling web object, returning the success code and new password.
8. The web page changes to inform the user of success or failure, and displays the new password if reset was successful. A refresh timer begins to count down from 120 seconds, and after expiration, the webpage is redirected to a main portal page.



SAP Perspective

All calls from the web for password resets make the request to the Production system. This is done to support friendly error handling if the destination system is unavailable or otherwise experiences errors. For example, if the user calls for a password reset on a sandbox machine, the box may be unavailable. If the web process called the sandbox directly, the error returned to the user would be the equivalent of a timeout error. However, by calling into Production (which has virtually guaranteed availability), the function module which calls to the destination system can more nicely handle this error, and provide the user with a friendly and informative message.

In order to support this approach, SM59 Destination Strings exist in Production for each “destination” system.

There are four custom function modules developed for the entire password reset solution. All four function modules exist in all SAP systems:

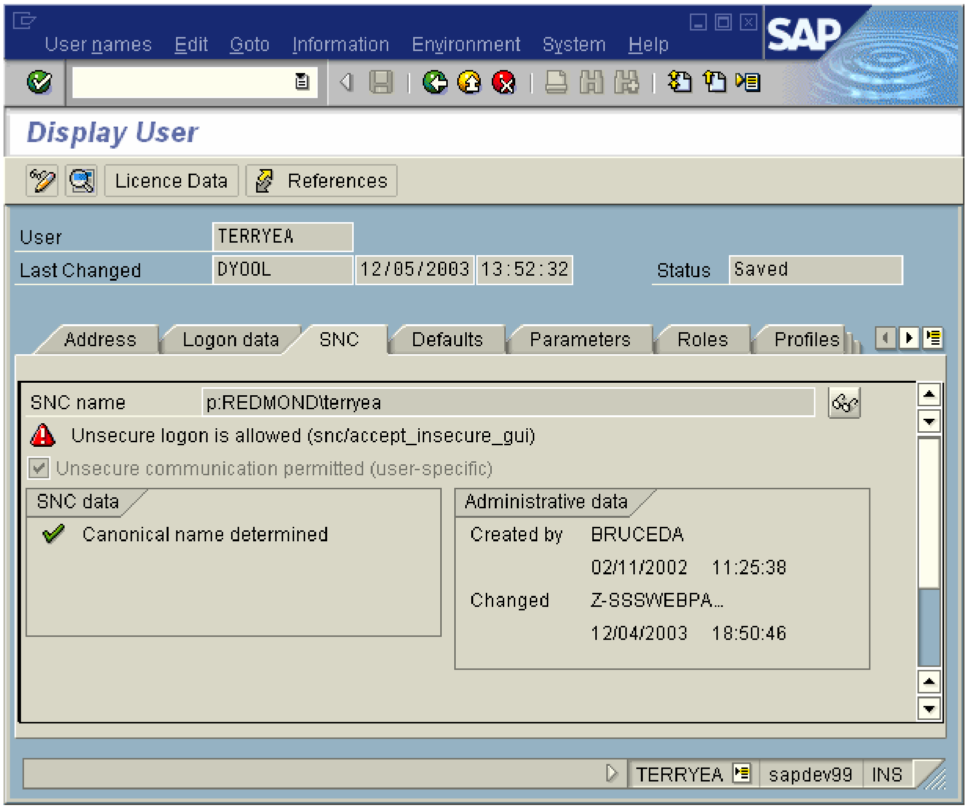
- Password Reset Enabled Systems: This function module is called by the web process to determine which SAP systems are enabled for password resets. Its input parameters: none. Its output parameters: (1) table containing (a) SAP system ID; (b) Short system description. This function module determines which SAP systems are available for reset by looking for a specific naming convention in SM59 destinations. For example, SM59 R/3 connection strings might all start with “PWD[sys][cli]”, such as “PWDMS1122” to indicate this connection is for the password reset tool to system MS1, client 122.
- Password Reset Wrapper: This function module “wraps” the Core Password Reset module. Since all calls for resets go into the Production system, a “wrapper” is needed to hand off the request to the target system if it is not Production. This wrapper will call the destination system via an SM59 connection. Its input parameters are (1) user name to reset; (2) destination system. Its output parameters are (1) new password; (2) success or failure return code.
    - If the wrapper does not need to call a destination system, it calls the Core Password Reset function module.
    - If the wrapper does need to call a destination system, it calls the wrapper on the destination system. There, the wrapper determines no further calls are needed, and it calls the Core Password Reset module on itself.

- Core Password Reset: This function module contains the bulk of the tool, and causes the password to actually be reset. Its input parameter is (1) user name to be reset. Its output parameters are (1) new password; (2) success or failure return code. Once this function module is called, it verifies that the user name exists on the current system, calls the "Strong Password Generator" and builds & executes a Call Transaction for SU01_NAV to actually reset the password.
- Strong Password Generator: This function module is called by the Core Password Reset function module. When called, it generates a unique and strong password. Its input parameter is: (1) user name. Its output parameter is: (1) password. The user name is used only to ensure that the generated password does not contain any pattern of the username. Random passwords are generated by leveraging SAP's standard function module RSEC_GENERATE_PASSWORD.

Additional Info

When making the initial call into SAP to determine which SAP systems support Password Reset, you could pass in the user's alias – and only return SAP systems on which the user has an active account. On our implementation, we opted not to take this approach for several reasons. First, it would slow down the user experience substantially, especially if any of the systems is offline. We decided to always present the complete list of SAP systems. If the user submits a request for a reset on a system where he does not have an account, we inform him of such, and provide links to information on how to set up his account.

There are other variations which can support particular situations. For example, we actually have single sign on into SAP for the majority of our users. This Password Reset tool for them is more like a "User Account Reset." When users submit a Password Reset request, we also populate their user account with their domain credentials. This allows them to self-update their single sign on information if they change domains, or change aliases. When a new account is set up for the first time, users are required to use the Password Reset tool to generate their initial password & to initially populate their Single Sign On information (available on the SU01 screen for single sign on enabled systems):



Display User

User TERRYEA
Last Changed DYOOL 12/05/2003 13:52:32 Status Saved

SNC name p:REDMOND\terryea

Unsecure logon is allowed (snc/accept_insecure_gui)

☑ Unsecure communication permitted (user-specific)

SNC data

✓ Canonical name determined

Administrative data

Created by BRUCEDA
02/11/2002 11:25:38
Changed Z-SSSWEBPA...
12/04/2003 18:50:46

TERRYEA sapdev99 INS

When NT user credentials match the SAP user IDs, the user mapping is easy. In our implementation, the NT user “REDMOND\terryea” maps to the SAP user “TERRYEA”. However, if your mapping is not that straight-forward, this process will need a step where you perform the mapping between the NT credentials and the SAP user ID.

Sample Code

The concepts and sample code provided are done so for reference only. Use at your own risk. There is NO SUPPORT on this code. It is strongly encouraged to use this sample code as a reference document only in consultation with your own developers and program managers. Each SAP implementation is unique and code may need to be adjusted as appropriate.

Generate Random Password Function Module

The concepts and sample code provided are done so for reference only. Use at your own risk. There is NO SUPPORT on this code. It is strongly encouraged to use this sample code as a reference document only in consultation with your own developers and program managers. Each SAP implementation is unique and code may need to be adjusted as appropriate.

```
FUNCTION Z_S_EAS_GEN_RANDOM_PASSWORD.
*"----------------------------------------------------------------------
*"*"Local interface:
*"       IMPORTING
*"             VALUE(UNAME) TYPE  CHAR12 OPTIONAL
*"       EXPORTING
*"             VALUE(PWD) TYPE  CHAR8
*"       EXCEPTIONS
*"              UNKNOWN_ERROR
*"----------------------------------------------------------------------
DATA: LEAVE(1),
      ERROR(1),
      COUNTER TYPE I.

DATA: OFF TYPE I VALUE 0,
      LEN TYPE I VALUE 3,
      SLEN TYPE I,
      STRING(3) TYPE C.

  CONSTANTS: C_ALPHABET(46) VALUE
          'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()'.

  CLEAR: LEAVE, ERROR, COUNTER.

  DO.
    COUNTER = COUNTER + 1.
    CALL FUNCTION 'RSEC_GENERATE_PASSWORD'
         EXPORTING
              ALPHABET      = C_ALPHABET
              OUTPUT_LENGTH = 8
         IMPORTING
              OUTPUT        = PWD
         EXCEPTIONS
              SOME_ERROR    = 1
              OTHERS        = 2.

    IF SY-SUBRC <> 0.
      RAISE UNKNOWN_ERROR.
    ENDIF.

    LEAVE = 'X'.
```

```
* Check for at least one number in the password
    IF NOT PWD CA '0123456789'.
      CLEAR LEAVE.
    ENDIF.

* Check for at least one special character in positions 2-7
    IF NOT PWD+1(5) CA '!@#$%^&*()'.
      CLEAR LEAVE.
    ENDIF.

* Check for at least one Alpha character in the password
    IF NOT PWD CA 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'.
      CLEAR LEAVE.
    ENDIF.
```

Password Reset Main Function Module

The concepts and sample code provided are done so for reference only. Use at your own risk. There is NO SUPPORT on this code. It is strongly encouraged to use this sample code as a reference document only in consultation with your own developers and program managers. Each SAP implementation is unique and code may need to be adjusted as appropriate.

```
FUNCTION Z_S_EAS_R_PASSWORD_RESET.
*"----------------------------------------------------------------------
*"*"Local interface:
*"  IMPORTING
*"     VALUE(USER_ACCOUNT_IN) LIKE  USR02-BNAME
*"     VALUE(SNC_IN) LIKE  USRACL-PNAME OPTIONAL
*"  EXPORTING
*"     VALUE(RESULT_OUT) TYPE  CHAR2
*"     VALUE(DESCR_OUT) TYPE  CHAR80
*"     VALUE(PASSWORD_OUT) TYPE  CHAR8
*"  EXCEPTIONS
*"      NO_ERROR
*"      ACCOUNT_NOT_FOUND
*"      ACCOUNT_INACTIVE
*"      SYSTEM_LOCKED
*"      UNEXPECTED_ERROR_IN_UNLOCK
*"      UNEXPECTED_ERROR_IN_RESET
*"      USER_ACCOUNT_LOCKED
*"----------------------------------------------------------------------
  DATA: BEGIN OF BDC_TABLE OCCURS 0.
          INCLUDE STRUCTURE BDCDATA.
  DATA: END OF BDC_TABLE.

  RESULT_OUT = '00'.

* Verify that the User Exists.
  SELECT SINGLE * FROM USR02 WHERE BNAME = USER_ACCOUNT_IN.
  IF SY-SUBRC <> 0.
    RESULT_OUT = '01'.                 "Account Not Found
    DESCR_OUT = 'Account Not Found.'.
    EXIT.
  ENDIF.

  AUTHORITY-CHECK OBJECT 'S_USER_GRP'
           ID 'CLASS' FIELD USR02-CLASS
           ID 'ACTVT' FIELD '05'.
  IF SY-SUBRC NE 0.
    RESULT_OUT = '05'.
    DESCR_OUT = 'No Authority to Reset Passwords.'.
    EXIT.
  ENDIF.

* Verify that the account is valid
  IF ( USR02-GLTGV > SY-DATUM ) OR ( USR02-GLTGB < SY-DATUM ).
    RESULT_OUT = '02'.                 "Account is inactive
    DESCR_OUT = 'Account is inactive.'.
    EXIT.
  ENDIF.

* Verify that the user has at least one role assigned.
  SELECT * FROM UST04 WHERE BNAME = USER_ACCOUNT_IN.
  ENDSELECT.
  IF SY-SUBRC <> 0.
    RESULT_OUT = '02'.                 "Account is inactive
    DESCR_OUT = 'Account is inactive.'.
    EXIT.
  ENDIF.

* Account has an administrative lock, do not unlock and reset.
  IF USR02-UFLAG = '64'.
    RESULT_OUT = '03'.
    DESCR_OUT = 'Account Status is "system locked"'.
    EXIT.
  ENDIF.

* If account is locked, unlock the account
  IF USR02-UFLAG = '128'.
    PERFORM UNLOCK_ACCOUNT TABLES BDC_TABLE
                            USING USER_ACCOUNT_IN
                         CHANGING RESULT_OUT
                                  DESCR_OUT.
  ENDIF.
* Check once and retry
  SELECT SINGLE * FROM USR02 WHERE BNAME = USER_ACCOUNT_IN.
  IF USR02-UFLAG <> 0.
    PERFORM UNLOCK_ACCOUNT TABLES BDC_TABLE
                            USING USER_ACCOUNT_IN
                          CHANGING RESULT_OUT
                                   DESCR_OUT.
* If not, exit with error.
    SELECT SINGLE * FROM USR02 WHERE BNAME = USER_ACCOUNT_IN.
    IF USR02-UFLAG <> 0.
      RESULT_OUT = '04'.               "Unexpected Error with Unlock
      DESCR_OUT = 'Unexpected Error with Unlock.'.
      EXIT.
    ENDIF.
  ENDIF.

* Reset Password
  CLEAR BDC_TABLE.
  REFRESH BDC_TABLE.
  PERFORM PASSWORD_RESET TABLES BDC_TABLE
                          USING USER_ACCOUNT_IN
                       CHANGING RESULT_OUT
                                DESCR_OUT
                                PASSWORD_OUT.

  IF NOT SNC_IN IS INITIAL.

    REFRESH BDC_TABLE.
    PERFORM ADD_BDC_VALUES TABLES BDC_TABLE  USING:
* main su01 screen
       'X'      'SAPLSUU5'          '0050',
       ' '      'USR02-BNAME'        USER_ACCOUNT_IN,
       ' '      'BDC_OKCODE'        'CHAN',
* change screen
       'X'      'SAPLSUU5'          '0100',
       ' '      'BDC_OKCODE'        'SNC',
* snc screen
       'X'      'SAPLSUU5'          '0100',
       ' '      'USRACL-PNAME'      SNC_IN,
       ' '      'USRACL-GUIFLAG'    'X',
       ' '      'BDC_OKCODE'        'UPD',
       'X'      'SAPLSUU5'          '0050',
       ' '      'BDC_OKCODE'        'BACK'.

    PERFORM SUBMIT_BDC1 TABLES BDC_TABLE
                      CHANGING DESCR_OUT.

    IF SY-SUBRC = 0.
      IF DESCR_OUT IS INITIAL.
        RESULT_OUT = '00'.
        DESCR_OUT = 'Success'.
      ELSE.
        RESULT_OUT = 04.
        DESCR_OUT = 'User is locked!'.
        CLEAR PASSWORD_OUT.
      ENDIF.
    ELSE.
      RESULT_OUT = '05'.
      DESCR_OUT = 'Unexpected Error with Reset.'.
      CLEAR PASSWORD_OUT.
    ENDIF.

  ENDIF.
  EXIT.

ENDFUNCTION.

************************************************************************
* form unlock_account
************************************************************************
FORM UNLOCK_ACCOUNT TABLES BDC_TABLE STRUCTURE BDCDATA
                    USING  USER
                 CHANGING  RESULT_OUT
                           DESCR_OUT.

  PERFORM ADD_BDC_VALUES TABLES BDC_TABLE USING:
* main SU01 screen
     'X'      'SAPLSUU5'          '0050',
     ' '      'USR02-BNAME'       USER,
     ' '      'BDC_OKCODE'        'LOCK',
* Lock/Unlock User Screen
     'X'      'SAPLSUU5'          '0500',
     ' '      'BDC_OKCODE'        'UNLO'.

  PERFORM SUBMIT_BDC1 TABLES BDC_TABLE
                     CHANGING DESCR_OUT.


  SELECT SINGLE * FROM USR02 WHERE BNAME = USER.
  IF USR02-UFLAG NE '0'.
    RESULT_OUT = '04'.
    EXIT.
  ENDIF.

ENDFORM.                               " unlock_account

************************************************************************
* form password_reset
************************************************************************
FORM PASSWORD_RESET TABLES BDC_TABLE STRUCTURE BDCDATA
                     USING USER
                  CHANGING RESULT_OUT
```

```
                           DESCR_OUT
                           PASSWORD_OUT.

* Call a function to generate a strong password
  CALL FUNCTION 'Z_S_EAS_GEN_RANDOM_PASSWORD'
       EXPORTING
            UNAME         = USER
       IMPORTING
            PWD           = PASSWORD_OUT
       EXCEPTIONS
            UNKNOWN_ERROR = 1
            OTHERS        = 2.
  IF SY-SUBRC <> 0.
    RESULT_OUT = '06'.
    DESCR_OUT = 'UNEXPECTED_RESET_ERROR'.
    EXIT.
  ENDIF.


  PERFORM ADD_BDC_VALUES TABLES BDC_TABLE  USING:
* main SU01 screen
     'X'      'SAPLSUU5'          '0050',
     ' '      'USR02-BNAME'        USER,
     ' '      'BDC_OKCODE'        'PASS',
* Lock/Unlock User Screen
     'X'      'SAPLSUU5'          '0400',
     ' '      'G_PASSWORD1'       PASSWORD_OUT,
     ' '      'G_PASSWORD2'       PASSWORD_OUT,
     ' '      'BDC_OKCODE'        'PASS',
     'X'      'SAPLSUU5'          '0050',
     ' '      'BDC_OKCODE'        'BACK'.

  PERFORM SUBMIT_BDC1 TABLES BDC_TABLE
                  CHANGING DESCR_OUT.

  IF SY-SUBRC = 0.
    RESULT_OUT = '00'.
    DESCR_OUT = 'Success'.
  ELSE.
    RESULT_OUT = '05'.
    DESCR_OUT = 'Unexpected Error with Reset.'.
    CLEAR PASSWORD_OUT.
  ENDIF.

ENDFORM.                               "password_reset.

*============================================================
FORM SUBMIT_BDC1 TABLES BDC_TABLE STRUCTURE BDCDATA
               CHANGING DESCR_OUT.
*============================================================

  DATA: MESSTAB LIKE BDCMSGCOLL OCCURS 0 WITH HEADER LINE.
  DATA: BDC_MODE(1) VALUE 'N',
        BACKSUBRC LIKE SY-SUBRC.
  CLEAR DESCR_OUT.

  data: Option type CTU_PARAMS.

  option-DISMODE  =  BDC_MODE.
  option-UPDMODE  = 'S'.
  option-CATTMODE = 'N'.
  option-DEFSIZE  = 'X'.
  option-RACOMMIT = 'X'.      "to not exit after a commit work

  REFRESH MESSTAB.  CLEAR MESSTAB.
  CALL TRANSACTION 'SU01_NAV' USING BDC_TABLE
                          OPTIONS FROM  option
                          MESSAGES INTO MESSTAB.
BACKSUBRC = SY-SUBRC.

  LOOP AT MESSTAB WHERE MSGID = '01' AND
                        MSGNR = '410'.
    DESCR_OUT = 'LOCK'.
  ENDLOOP.
SY-SUBRC = BACKSUBRC.

ENDFORM.                               " submit_bdc

*============================================================
FORM ADD_BDC_VALUES TABLES BDC_TABLE STRUCTURE BDCDATA
*============================================================
                   USING VALUE(PARAM1)
                         VALUE(PARAM2)
                         VALUE(PARAM3).
  CLEAR BDC_TABLE.
  IF PARAM1 = 'X'.
    BDC_TABLE-DYNBEGIN = 'X'.
    BDC_TABLE-PROGRAM = PARAM2.
    BDC_TABLE-DYNPRO  = PARAM3.
  ELSE.
    BDC_TABLE-FNAM = PARAM2.
    BDC_TABLE-FVAL = PARAM3.
  ENDIF.
  APPEND BDC_TABLE.
ENDFORM.                               " add_bdc_value

* Validate that the first character is not "?" and is not "!"
    IF PWD(1) = '?' OR
       PWD(1) = '!'.
       CLEAR LEAVE.
    ENDIF.

* Validate that the first three characters are not the same
    IF PWD(1) = PWD+1(1) AND
       PWD+1(1) = PWD+2(1).
       CLEAR LEAVE.
   ENDIF.

IF NOT UNAME IS INITIAL.
   SLEN = STRLEN( PWD ).
   SLEN = SLEN - LEN.
   SLEN = SLEN + 1 .           " to ensure we check until the last char
   OFF = 0.

  DO SLEN TIMES.
    CLEAR STRING.
    STRING = PWD+OFF(LEN).
    OFF = OFF + 1.
    IF UNAME CS STRING.
       CLEAR LEAVE.
       OFF = 0.
       EXIT.
    ENDIF.

  ENDDO.

ENDIF.

    IF LEAVE EQ 'X'.
      EXIT.
    ELSE.
      IF COUNTER > 100.
        MESSAGE E777(S3) WITH
               'Error: Password generation looped more than 100 times.'.
      ENDIF.
    ENDIF.

  ENDDO.

ENDFUNCTION.
```

Password Reset Wrapper

The concepts and sample code provided are done so for reference only. Use at your own risk. There is NO SUPPORT on this code. It is strongly encouraged to use this sample code as a reference document only in consultation with your own developers and program managers. Each SAP implementation is unique and code may need to be adjusted as appropriate.

```
FUNCTION Z_S_EAS_R_PWD_RESET_WRAPPER.
*"----------------------------------------------------------------------
*"*"Local interface:
*"       IMPORTING
*"             VALUE(SAP_SYSTEM_IN) LIKE  RFCDES-RFCDEST
*"             VALUE(USER_ACCOUNT_IN) LIKE  USR02-BNAME
*"             VALUE(SNC_IN) LIKE  USRACL-PNAME OPTIONAL
*"       EXPORTING
*"             VALUE(RESULT_OUT) TYPE  CHAR2
*"             VALUE(DESCR_OUT) TYPE  CHAR80
*"             VALUE(PASSWORD_OUT) TYPE  CHAR8
*"       EXCEPTIONS
*"              NO_ERROR
*"              ACCOUNT_NOT_FOUND
*"              ACCOUNT_INACTIVE
*"              SYSTEM_LOCKED
*"              UNEXPECTED_ERROR_IN_UNLOCK
*"              UNEXPECTED_ERROR_IN_RESET
*"              USER_ACCOUNT_LOCKED
*"              UNKNOWN_SAP_SYSTEM
*"              SAP_SYSTEM_UNAVAILABLE
*"----------------------------------------------------------------------

  IF SAP_SYSTEM_IN = 'LOCAL'.
    CALL FUNCTION 'Z_S_EAS_R_PASSWORD_RESET'
         EXPORTING
              USER_ACCOUNT_IN            = USER_ACCOUNT_IN
              SNC_IN                     = SNC_IN
         IMPORTING
              RESULT_OUT                 = RESULT_OUT
              DESCR_OUT                  = DESCR_OUT
              PASSWORD_OUT               = PASSWORD_OUT
         EXCEPTIONS
              NO_ERROR                   = 1
              ACCOUNT_NOT_FOUND          = 2
              ACCOUNT_INACTIVE           = 3
              SYSTEM_LOCKED              = 4
              UNEXPECTED_ERROR_IN_UNLOCK = 5
              UNEXPECTED_ERROR_IN_RESET  = 6
              OTHERS                     = 7.
    IF SY-SUBRC = 7.
      RESULT_OUT = '05'.
      DESCR_OUT = 'Unknown Error returned from Function Call.'.
    ENDIF.

  ELSE.
    CALL FUNCTION 'Z_S_EAS_R_PASSWORD_RESET' DESTINATION SAP_SYSTEM_IN
        EXPORTING
             USER_ACCOUNT_IN            =  USER_ACCOUNT_IN
             SNC_IN                     =  SNC_IN
        IMPORTING
             RESULT_OUT                 =  RESULT_OUT
             DESCR_OUT                  =  DESCR_OUT
             PASSWORD_OUT               =  PASSWORD_OUT
        EXCEPTIONS
             NO_ERROR                   = 1
             ACCOUNT_NOT_FOUND          = 2
             ACCOUNT_INACTIVE           = 3
             SYSTEM_LOCKED              = 4
             UNEXPECTED_ERROR_IN_UNLOCK = 5
             UNEXPECTED_ERROR_IN_RESET  = 6
             OTHERS                     = 7.
    IF SY-SUBRC = 7.
      RESULT_OUT = '05'.
      DESCR_OUT = 'Unknown Error returned from Function Call.'.
    ENDIF.

  ENDIF.                               "Local/Remote System

ENDFUNCTION.
```

Get RFC Destinations

The concepts and sample code provided are done so for reference only. Use at your own risk. There is NO SUPPORT on this code. It is strongly encouraged to use this sample code as a reference document only in consultation with your own developers and program managers. Each SAP implementation is unique and code may need to be adjusted as appropriate.

```
FUNCTION Z_S_EAS_R_GET_DESTINATION.
*"----------------------------------------------------------------------
*"*"Local interface:
*"       IMPORTING
*"             VALUE(PREFIX_IN) LIKE  RFCDES-RFCDEST
*"       EXPORTING
*"             VALUE(RC) TYPE  CHAR2
*"             VALUE(DESCR_OUT) TYPE  CHAR200
*"       TABLES
*"              RFCDEST_OUT STRUCTURE  ZRFCDOC_PASSRESET
*"       EXCEPTIONS
*"              NO_ERROR
*"              NO_MATCHES_FOUND
*"              UNDEFINED_ERROR
*"----------------------------------------------------------------------

  DATA: L_RFCDEST LIKE RFCDES-RFCDEST,
        T_LINES LIKE SY-TABIX.

  DATA: BEGIN OF I_RFCDOC OCCURS 0.
          INCLUDE STRUCTURE ZRFCDOC_PASSRESET.
  DATA: END OF I_RFCDOC.

  TABLES: RFCDES, RFCDOC.

  RC = '00'.

* Select RFC Destinations into internal table.
  SELECT RFCDEST INTO L_RFCDEST FROM RFCDES
                      WHERE RFCTYPE = '3'.    "R3 Connections.
    IF L_RFCDEST(3) = PREFIX_IN.
      SELECT SINGLE * FROM RFCDOC WHERE RFCDEST = L_RFCDEST
                                    AND RFCLANG = 'EN'.
      IF SY-SUBRC = 0.
        MOVE RFCDOC-RFCDEST TO I_RFCDOC-RFCDEST.
        MOVE RFCDOC-RFCDOC1 TO I_RFCDOC-RFCDOC1.
        CONCATENATE RFCDOC-RFCDOC2 RFCDOC-RFCDOC3 INTO
                    I_RFCDOC-RFCDOC2.
        APPEND I_RFCDOC.
        CLEAR  I_RFCDOC.
      ENDIF.
    ENDIF.
  ENDSELECT.

  DESCRIBE TABLE I_RFCDOC LINES T_LINES.
  IF T_LINES = 0.
    RC = '01'.
    DESCR_OUT = 'No RFC Destination Matches Found.'.
    EXIT.
  ELSE.
    RFCDEST_OUT[] = I_RFCDOC[].
    RC = '00'.
    EXIT.
  ENDIF.

ENDFUNCTION.


** Note:  ZRFCDOC_PASSRESET structure **
```

Component	Type	Data	Lngth	Description
RFCDEST	RFCDEST	CHAR	32	Logical Destination (Specified in Function Call)
RFCDOC1	RFCDOC_D	CHAR	72	Description of RFC connection
RFCDOC2	CHAR200	CHAR	200	Text field length 200

Contact Information

This whitepaper is provided solely as a sample reference document, and no formal support is provided. However, the author will provide best effort to answer questions. You can contact Terry Earls at terryea@microsoft.com

The concepts and sample code provided are done so for reference only. Use at your own risk. There is NO SUPPORT on this code. It is strongly encouraged to use this sample code as a reference document only in consultation with your own developers and program managers. Each SAP implementation is unique and code may need to be adjusted as appropriate.