



## APIs by Example: Office APIs — Search System Directory

[System iNetwork Programming Tips Newsletter](#)

System iNEWS Staff

Thu, 05/10/2001 (All day)

This installment of APIs by Example shows how to use one of the Office APIs, Search System Directory (QOKSCHD). The example consists of two objects, service program CBX005, and program CBX005I.

Procedure GetSmtpAddr in service program CBX005 accepts as input the user ID and address that constitute the system directory entry and returns the user's formatted SMTP address.

Here are the parameters for procedure GetSmtpAddr:

### Parameters:

PxUser            INPUT

This is the user ID portion of the system directory entry.

PxAddr            INPUT

This is the address portion of the system directory entry.

Return value      OUTPUT

This returns the formatted SMTP address for the specified user. If there is no SMTP address specified for the user or an error occurs during retrieval, the program returns blanks in this parameter.

There are four main steps performed by procedure GetSmtpAddr:

- a. The API input parameters are initialized.
- b. The API searches the system directory.
- c. The program returns a blank return value if there is no SMTP address specified for the user or an error.
- d. The procedure formats and returns the SMTP address.

The Office API used in service program CBX005 is

### \* Search System Directory (QOKSCHD)

This searches the system distribution directory and returns user information that matches the criteria input.

<http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/info/apis/QOKSCHD.htm>

Below is service program CBX005:

```
** Compile options:
**   CRTRPGMOD MODULE( CBX005 )
**   DBGVIEW( *LIST )
**
**   CRTSRVPGM SRVPGM( CBX005 )
**   MODULE( CBX005 )
**   ACTGRP( QSRVPGM )
**
***-- Header specifications: -----**
H NoMain Option( *SrcStmt )
***-- System Info Data Structure: -----**
D PgmSts          SDs
D PsJobUsr          10a   Overlay( PgmSts: 254 )
D PsCurUsr          10a   Overlay( PgmSts: 358 )
***-- Get user SMTP address: -----**
D GetSmtpAddr      Pr          321a
```

```

D PxUser          10a
D PxAddr          8a  Options( *NoPass )
***-- Get user SMTP address: -----**
P GetSmtpAddr     B          Export
D                Pi          321a
D PxUser          10a
D PxAddr          8a  Options( *NoPass )
***-- API error data structure: -----**
D ApiError        Ds
D AeBytPrv        10i 0 Inz( %Size( ApiError ) )
D AeBytAvl        10i 0 Inz
D AeMsgId         7a
D                 1a
D AeMsgDta        128a
***-- Search directory parameters: -----**
D Sreq0100        Ds
D SrCcsId         10i 0 Inz( 0 )
D SrChrSet        10i 0 Inz
D SrCodPag        10i 0 Inz
D SrWldCrd        4a  Inz
D SrCvtRcv        1a  Inz( '0' )
D SrSchDta        1a  Inz( '0' )
D SrRunVfy        1a  Inz( '1' )
D SrConHdl        1a  Inz( '0' )
D SrRscHdl        16a  Inz
D SrSrqFmt        8a  Inz( 'SREQ0101' )
D SrSrqOfs        10i 0 Inz( 110 )
D SrSrqNbrElm     10i 0 Inz
D SrRtnFmt        8a  Inz( 'SREQ0103' )
D SrRtnOfs        10i 0 Inz( 100 )
D SrRtnNbrElm     10i 0 Inz( 1 )
D SrRcvFmt        8a  Inz( 'SRCV0101' )
D SrRcvNbrElm     10i 0 Inz( 1 )
D SrUsrFmt        8a  Inz( 'SRCV0111' )
D SrOrdFmt        8a  Inz
D SrOrdRtnOpt     1a  Inz( '0' )
D                 3a
D Sr0103          Like( Sreq0103 )
D Sr0101          Like( Sreq0101 )
**
D Sreq0101        Ds          Inz
D S1Entry         Dim( 2 )
D S1EntLen        10i 0 Inz( %Size( S1Entry ) )
D                 Overlay( S1Entry: 1 )
D S1CmpVal        1a  Inz( '1' )
D                 Overlay( S1Entry: *Next )
D S1FldNam        10a  Overlay( S1Entry: *Next )
D S1PrdId         7a  Inz( '*IBM' )
D                 Overlay( S1Entry: *Next )
D S1DtaCas        1a  Overlay( S1Entry: *Next )
D                 1a  Overlay( S1Entry: *Next )
D S1ValLen        10i 0 Inz( %Size( S1ValMtc ) )
D                 Overlay( S1Entry: *Next )
D S1ValMtc        10a  Overlay( S1Entry: *Next )
**
D Sreq0103        Ds
D S3SpcRtn        10a  Inz( '*SMTP' )
**
D Srcv0100        Ds          32767
D R00BytRtn       10i 0
D R00OrdFldOfs    10i 0
D R00UsrEntOfs    10i 0
D R00DirEntNbr    10i 0
D R00ConHdl       1a
D R00RscHdl       16a
D R00UsrMtcAry    Like( Srcv0101 )
**
D Srcv0101        Ds          Based( pSrcv0101 )
D R01UsrDtaLen    10i 0
D R01RtnNbrFld    10i 0
D Srcv0111        Ds          Based( pSrcv0111 )
D R11FldNam       10a
D R11PrdId        7a
D                 3a
D R11CcsId        10i 0

```

```

D  R11CodPag          10i 0
D  R11RtnFldLen       10i 0
D  Srcv0111v         Ds          Based( pSrcv0111v )
D  R11RtnFld         256a
**-- Local constants & variables: -----**
D  SmtPdmn           s          256a  Varying
D  SmtPusrId         s          64a   Varying
**
D  At                c          '@'
**-- Search directory: -----**
D  SchDir            Pr          Extpgm( 'QOKSCHD' )
D  SdRcvVar          Like( Srcv0100)
D  SdRcvVarLen       10i 0 Const
D  SdFmtNam          8a   Const
D  SdFunction        10a   Const
D  SdKeepTmpRsc      1a   Const
D  SdRqsVar          Const Like( Sreq0100 )
D  SdRqsVarLen       10i 0 Const
D  SdRqsFmtNam       8a   Const
D  SdError           8a
**
**-- Get SMTP address: -----**
**
C          If          PxUser      = '*CURRENT'
C          Eval        PxUser      = PsCurUsr
C          EndIf
**
C          If          %Parms      = 1
C          Eval        SrSrqNbrElm = 1
C          Eval        S1ValMtc(1) = PxUser
C          Eval        S1FldNam(1) = 'USER  '
**
C          Else
C          Eval        SrSrqNbrElm = 2
C          Eval        S1ValMtc(1) = PxUser
C          Eval        S1ValMtc(2) = PxAddr
C          Eval        S1FldNam(1) = 'USRID  '
C          Eval        S1FldNam(2) = 'USRADDR'
C          EndIf
**
C          Eval        Sr0103      = Sreq0103
C          Eval        Sr0101      = Sreq0101
**
C          Callp       SchDir( Srcv0100
C                          : %size( Srcv0100 )
C                          : 'SRCV0100'
C                          : '*SEARCH'
C                          : '0'
C                          : Sreq0100
C                          : %Size( Sreq0100 )
C                          : 'SREQ0100'
C                          : ApiError
C                          )
**
C          If          AeBytAvl    > 0          Or
C          R00DirEntNbr = 0
**
C          Return      *Blanks
**
C          Else
C          Eval        pSrcv0101    = %Addr( Srcv0100 ) +
C                                     R00UsrEntOfs
C          Eval        pSrcv0111    = pSrcv0101      +
C                                     %Size( Srcv0101 )
**
C          Do          R01RtnNbrFld
**
C          Eval        pSrcv0111v   = pSrcv0111      +
C                                     %Size( Srcv0111 )
**
C          Select
C          When        R11FldNam     = 'SMTPUSRID'
C          Eval        SmtPusrId     = %Subst( R11RtnFld
C                                               : 1
C                                               : R11RtnFldLen )

```

```

**
C          When      R11FldNam   =  'SMTPDMN'
C          Eval      SmtprDmn    =  %Subst( R11RtnFld
C                                   : 1
C                                   : R11RtnFldLen )
C          EndSl
**
C          Eval      pSrcv0111   =  pSrcv0111      +
C                                   %Size( Srcv0111 ) +
C                                   R11RtnFldLen
C          EndDo
**
C          Return     SmtprUsrId + At + SmtprDmn
C
C          EndIf
**
P GetSmtprAddr      E

```

Program CBX00CI shows sample usage. The program is intended to be a break-handling program that receives messages sent to the user and forwards them to the SMTP address specified in the system directory.

```

/*                                                     */
/* Program      : CBX005I                               */
/* Program function: Break handling exit program         */
/*                                                     */
/* Program summary:                                     */
/*   Receives messages from a monitored message queue as they */
/*   arrive. The SMTP address of the current job user is then */
/*   retrieved from the system directory.                 */
/*                                                     */
/*   If an SMTP address is found, the incoming message will be */
/*   forwarded to that address and subsequently removed from */
/*   the message queue.                                   */
/*                                                     */
/*   To notify the user of the event, the message text is also */
/*   sent as a status message that appears at the bottom of */
/*   the current screen.                                  */
/*                                                     */
/* Parameters:                                          */
/*   MsgQ      INPUT      The name of the message queue */
/*                   receiving the message.              */
/*                                                     */
/*   MsgQlib   INPUT      The name of the library containing */
/*                   the message queue.                  */
/*                                                     */
/*   MsgKey    INPUT      The message reference key of the */
/*                   message received.                   */
/*                                                     */
/* Activation of break message handling:               */
/*   CHGMSGQ   MSGQ( message-queue-name )              */
/*           DLVRY( *BREAK )                            */
/*           PGM( CBX005I *ALWRPY )                    */
/*                                                     */
/* Compile options:                                    */
/*   CRTCLMOD  MODULE( CBX005CI )                      */
/*           DBGVIEW( *LIST )                          */
/*                                                     */
/*   CRTPGM    PGM( CBX005I )                          */
/*           MODULE( CBX005CI )                        */
/*           BNDSRVPGM( CBX005 )                      */
/*           ACTGRP( *CALLER )                        */
/*                                                     */
/*-----*/
Pgm      ( &MsgQ &MsgQlib &MsgKey )

/*-- Parameters:                                     ---*/
Dcl      &MsgQ      *Char    10
Dcl      &MsgQlib   *Char    10
Dcl      &MsgKey    *Char     4

```

```

/-- Global variables:                                ---/
Dcl      &Msg          *Char    512
Dcl      &MsgId        *Char     7
Dcl      &Sev          *Dec    ( 2 0 )
Dcl      &Sender       *Char    80
Dcl      &RtnType      *Char     2
Dcl      &SndUser      *Char    10
Dcl      &SmtprAddr    *Char    64

Dcl      &ToCallStkE   *Char    38
Dcl      &ErrorFlag   *Lgl     1  '0'

/-- Global error monitoring:  -----*/
MonMsg    CPF0000      *None      GoTo EndPgm
/-- Receive message and keep on queue:  ---/
RcvMsg    MsgQ( &MsgQlib/&MsgQ )      +
          MsgKey( &MsgKey )          +
          Rmv( *NO )                  +
          Msg( &Msg )                  +
          MsgId( &MsgId )              +
          Sev( &Sev )                  +
          Sender( &Sender )            +
          RtnType( &RtnType )          +

/-- Get SMTP address:                                ---/
CallPrc   GetSmtprAddr   Parm( '*CURRENT ' )      +
          RtnVal( &SmtprAddr )

If        ( &SmtprAddr > ' ' )      Do

/-- Retrieve sender user-id:                          ---/
ChgVar    &SndUser       %Sst( &Sender 11 10 )

/-- Send message to SMTP address and remove from queue:  ---/
SndDst    Type( *LMSG )              +
          ToIntNet( ( &SmtprAddr ) )  +
          DstD( &MsgQ              *Tcat ':' *Bcat +
              %Sst( &Msg 1 32 ) )      +
          LongMsg( ':/N'              *Bcat +
              'Sending user . . :'    *Bcat +
              &SndUser                *Bcat +
              ':/N'                  *Bcat +
              'Target queue . . :'    *Bcat +
              &MsgQ                    *Bcat +
              ':/P'                  *Bcat +
              'Message text . . :'    *Bcat +
              &Msg                      )
          RmvMsg    MsgQ( &MsgQlib/&MsgQ )      +
          MsgKey( &MsgKey )          +
          Clear( *BYKEY )

EndDo

/-- Send message to bottom of screen:  ---/
SndPgmMsg MsgId( CPF9897 )      +
          MsgF( QCPFMSG )        +
          MsgDta( &Msg )          +
          ToPgmQ( *EXT )          +
          MsgType( *STATUS )

EndPgm:
EndPgm

```

The above tip was written by Carsten Flensburg. For questions regarding this tip, contact Carsten at <mailto:flensburg@novasol.dk>.

**Source URL:** <http://iprodeveloper.com/rpg-programming/apis-example-office-apis-search-system-directory>