Apple Federal Smart Card Package Installation and Setup Guide

About the Federal Smart Card Package

The Apple Federal Smart Card Package (FSCP) is software you install on a Macintosh computer that lets users gain access to the computer using a Department of Defense Common Access Card.

Users can use a Common Access Card to verify their permission to

- Log in to the computer
- Access the computer when the screen saver is on
- Make changes to some System Preferences panes
- Install software

To use FSCP, you need the following:

- A Macintosh computer with Mac OS X v10.2.3 installed
- A Department of Defense Common Access Card issued since 2001
- An SCM Microsystems SCR331 USB High Speed EMV Reader

You can also use one of these smart card readers, but you must download and install driver software from the manufacturer's website:

- Gemplus GemPC430 USB Smart Card Reader
- OMNIKEY CardMan Desktop USB 2020
- Schlumberger Sema Reflex USB v.2 Reader or Reflex USB Lite Reader

If you are using your own directory service for user accounts, you need to be connected to your network.

Users should also be connected the first time they log in using the Common Access Card so that FSCP can access any Certificate Revocation Lists (CRL) needed to verify certificates.

Installing the Federal Smart Card Package

To install the Apple Federal Smart Card Package:

1. Log in as an administrator for your computer and insert the FSCP installation disc.

   The user you created when you set up Mac OS X is an administrator.

2. Double-click the "FederalSmartCardPackage.pkg" icon on the CD.

   A message asks you to enter your password and restart the computer.

3. Follow the onscreen instructions to install the software.

   FSCP installs the software necessary to use your smart card reader. It also installs the ReadCAC application in the Smartcard folder in Applications/Utilities.

Setting Up the Federal Smart Card Package

After you install the FSCP software, you need to set up your computer and the software. This includes setting up user accounts, setting login options, adding the EDI Identifier for each smart card to the FSCP software, and setting up the FSCP software.

Setting Up User Accounts

Each person using a smart card to log in to a computer needs a user account. You can use existing user accounts on the computer or create new accounts. You can also use user accounts in an existing directory service, such as LDAP or Active Directory. To learn more about doing this, see "Setting Up FSCP to Use Other Directories" later in this document.

To create a user account on the computer:

1. Open System Preferences and click Accounts, then click New User.

   To create a user, you need to log in as an administrator of the computer or click the lock icon in the Accounts preferences pane and enter an administrator name and password.

2. Type a name and short name for the user that is different from other user accounts.
3. Type a password for the user account. Users can change their password later using the My Account preferences pane.
4. If you want, select the "Allow user to administer this computer" checkbox.
5. Click Save.
6. If automatic login is turned on, a message asks if you want to turn it off. You should turn it off.

Setting Login Options

Mac OS X is set up to log in automatically as the user you create when you set up Mac OS X. Before using a Common Access Card to log in to your computer, you need to turn off automatic login. To reduce the possibility of someone circumventing the security of your computer, you can hide the Restart and Shut Down buttons that appear in the login window.

To change login options:

1. Open System Preferences and click Accounts.
2. Make sure you have deselected the "Log in automatically" checkbox.
3. To hide the Restart and Shut Down buttons in the login window, click Login Options and select the checkbox.

Getting the Identifier for a Common Access Card

For a user to authenticate using a Common Access Card, you must associate the user's account with the card by adding the card's Electronic Data Interchange (EDI) Identifier to the user account. The identifier is a ten-digit number stored on the card. You get this number using the ReadCAC application.

To get the EDI Identifier:

Open the ReadCAC application in the Smartcard folder (in Utilities), then insert the smart card in the reader and enter the PIN for the card.

The identifier appears with the label "DoD EDI Identifier" in the ReadCAC window.

FSCP saves the EDI Identifier for each card used with the computer in a file named CACRecords.txt, which is in your Documents folder.

Adding the EDI Identifier to a Mac OS X User Account

If you use Mac OS X user accounts for smart card authentication, you can use the `cac_addid` command in Terminal to add the EDI Identifier to the user account.

Note: If you are using a different directory service, you do not need to do this step.

Open Terminal (in Utilities) and execute this command (as root):

```
cac_addid username ID
```

The `username` is the short name for the user account and `ID` is the identifier.

If you're familiar with NetInfo Manager (in Utilities), you can use it to add the identifier to the user account. Open the application and in the columns at the top of the window, select "users" and then select the user account name. Choose New Property from the Directory menu. In the Property column, type "_DoD_EDI_Identifier". In the Value(s) column, type the identifier for the card. Choose Save Changes from the Domain menu.

Starting Authentication With the Common Access Card

To start using the Common Access Card to authenticate access to the computer, execute this command in Terminal (as root):

```
cac_setup
```

To stop authentication using the card and restore the standard Mac OS X authentication, execute this command:

```
cac_setup -off
```

Authenticating With the Common Access Card

You can now use the Common Access Card to gain access to the computer.

To log in to the computer the first time using the card:

1. Users should be connected to the network so that FSCP can access any Certificate Revocation Lists (CRLs) needed to verify certificates.

   To see where the CRLs are located, check the URI field in the Certificates pane of ReadCAC for a certificate.

2. Insert the card and type the PIN in the dialog.

   An "X" appears below the PIN text box for each time you type an incorrect PIN since the Common Access Card was inserted. If you exceed the maximum number of times you can enter an incorrect PIN (3), your card is locked. See your administrator if your card is locked.

   Note: If you need to log in using your password instead of a smart card, click the Other button in the login window and type your user name and password.

To authenticate access to the computer using the Common Access Card later:

When a message asks you to authenticate (for example, when you're changing a setting in System Preferences), insert your card and type the PIN for the card.

Because it takes place on the smart card, authentication takes longer than when you use a password.

Making Sure FSCP Is Running

For the smart card reader to work, an FSCP system daemon named “pcscd” must be running. To make sure the daemon is running and that it recognizes your reader, execute this command in Terminal:

```
pcsctest
```

You should see messages similar to these in the Terminal window:

```
MUSCLE PC/SC Lite Test Program

Testing SCardEstablishContext       : Command successful.
Testing SCardGetStatusChange
Please insert a working reader      : Command successful.
Testing SCardListReaders            : Command successful.
Reader 01: SCM SCR-331 CCID 0 0
Enter the reader number          :
```

Type the number 1.

If the Common Access Card is not inserted, you will see this message:

```
Waiting for card insertion
```

If so, insert the card. You will then see messages similar to these:

```
                                    : Command successful.
Testing SCardConnect                : Command successful.
Testing SCardStatus                 : Command successful.
Current Reader Name                 : SCM SCR-331 CCID 0 0
Current Reader State                : 34
Current Reader Protocol             : 0
Current Reader ATR Size             : 9
Current Reader ATR Value            : 3B 65 00 00 9C 02 02 07 02
Testing SCardDisconnect             : Command successful.
Testing SCardReleaseContext         : Command successful.
```

If you do not see messages similar to these, you may need to restart the “pcscd” daemon.

Restarting the FSCP Daemon

To restart the FSCP “pcscd” daemon, execute this in Terminal as root:

```
/System/Library/StartupItems/SmartCardServices/SmartCardServices
        restart
```

You can also use `stop` and `start` instead of `restart`.

Adding a Smart Card to SmartCardServices

The FSCP software contains the ATR values for the Common Access Cards currently available. If you are issued a new card that is not recognized by FSCP, you may be able to use the `pcsctool` command to add the card’s ATR value to FSCP.

To add a card to FSCP:

1 Execute this command in Terminal as root:

```
pcsctool
```

You should see several options.

2 Enter 1 for the Common Access Card bundle.

3 When a message asks, insert your smart card.

If the SmartCardServices daemon doesn’t recognize the ATR value of the card, it adds the value to the bundle.

Changing other configuration options

Several options are defined in a file named configuration.plist, which is in the SCLoginPlugin.bundle installed by FSCP. The SCLoginPlugin.bundle is located here:

/System/Library/CoreServices/SecurityAgentPlugins/

Here is the full pathname for the file:

/System/Library/CoreServices/SecurityAgentPlugins/SCLoginPlugin.bundle/Contents/Resources/configuration.plist

Note: To see the contents of SCLoginPlugin.bundle, hold down the Control key and click the SCLoginPlugin.bundle icon, then choose Show Package Contents from the menu. In the window that opens, double-click the Contents folder, then the Resources folder.

The configuration.plist file defines the options as XML key and value pairs. You can change the file using any XML editor. Because the file is located in the Mac OS X System folder, you need to log in as root to change it.

The contents of the file look like this:

```
<dict>
    <key>allowPasswordLogon</key>
    <string>1</string>
    <key>verboseMessages</key>
    <string>1</string>
    <key>crlVerificationOptional</key>
    <string>1</string>
</dict>
```

The value of the `allowPasswordLogon` key lets users log in using a user name and password. Set the value to `0` (zero) to prevent this.

The value of the `verboseMessages` key sets whether diagnostic information is on (`1`) or off (`0`). You probably do not need to change this setting or to use this information.

The value of the `crlVerificationOptional` key sets the Certificate Revocation List (CRL) verification. The default value (`1`) allows "lax" CRL verification. The value for strict CRL verification is zero (`0`).

Changing CRL verification

FSCP is set up to allow "lax" CRL verification so it doesn't prevent users from logging in or authenticating if the CRL for a certificate cannot be created or updated (for example, because the computer cannot connect to the CRL server).

To use strict CRL verification, change the value of the `crlVerificationOptional` key in the configuration.plist file to zero (`0`).

Setting up Mozilla to Work With Your Smart Card

You can use the Mozilla application for Mac OS X to send and receive signed and encrypted email messages using S/MIME with certificates stored on your Common Access Card.

Note: Mozilla is not included with Mac OS X and is not supported by Apple.

To download Mozilla for Mac OS X, visit the Mozilla website at www.mozilla.org and download version 1.2.1 or later.

To set up Mozilla to work with your Common Access Card, you need to:

- Add your smart card reader to Mozilla.
- Set up Mozilla to use your Common Access Card to sign or encrypt messages.
- Install root certificates on your computer.

Adding Your Smart Card Reader to Mozilla

You use Mozilla's Privacy & Security preferences to identify the smart card reader by specifying the location of a file installed by FSCP. The file is named "pkcs11.shlb" and it's located in the Library folder on your Mac OS X startup disk.

To add your smart card reader to Mozilla:

1 Be sure the reader is connected to your computer and insert the Common Access Card.

2 Open Mozilla and choose Preferences from the Mozilla menu.

3 Click the triangle next to Privacy & Security, then select Certificates.

4 Click Manage Security Devices, then click Load.

5 Enter a name in the Module Name box (for example, "CAC SmartCard"), then type the location of the file in the Module Filename box:

```
my_startup_disk:Library:Application Support:Mozilla:pkcs11.shlb
```

"My_startup_disk" should be the name of the Mac OS X startup disk for your computer.

6 Click OK to close the dialog.

Important Do not click the Enable FIPS button.

A message appears when Mozilla recognizes the card.

7 Click OK to close the Device Manager dialog and the Preferences dialog.

Important Do not click the Unload button to unload the "pkcs11.shlb" module.

Setting Up Mozilla to Sign and Encrypt Messages

To set up Mozilla to use your Common Access Card to authenticate signing and encrypting email messages:

1. Choose Mail & Newsgroups from the Window menu.

   You may need to set up your email account information before you continue.
2. Insert the Common Access Card in the reader, if necessary.
3. Choose Mail & Newsgroups Account Settings from the Edit menu, then click Security.
4. Click Select under Digital Signing.
5. If necessary, enter the PIN for your Common Access Card. (In the message, Mozilla calls the PIN the "master password.")
6. Click OK. Mozilla selects the correct certificate for signing your email.
7. If a message asks to use this certificate for encryption, click Cancel.
8. Click Select under Encryption, then click OK. Mozilla selects the correct certificate for encrypting your email.

Installing Root Certificates

Installing root certificates makes it easier to work with certificates for signing and encrypting email. You can install either test certificates or live certificates to use with the card.

To install test certificates:

1. Using the Mozilla web browser, open one of these webpages:

   For ORC Certificate Authority:

   - ID Certificates: https://ca-3.c3pki-OandM.orc.com
   - Email Certificates: https://email-ca-3.c3pki-OandM.orc.com

   For JITC Certificate Authority:

   - ID Certificates: https://idca.nit.disa.mil
   - Email Certificates: https://eca.nit.disa.mil
2. In the message that appears, make sure the checkbox is selected to accept this certificate temporarily, then click OK.
3. Messages might say that the webpage is encrypted or that the page contains unencrypted information. Click OK to close these messages.
4. In the page that appears, click the Retrieval tab, then click Import CA Certificate Chain in the list on the left.

5 Make sure the “Import the CA certificate chain into your browser” button is selected and click Submit.

A message appears asking you the purpose of trusting the new Certificate Authority.

6 If you are installing an email certificate, select the “Trust this CA to identify email users” checkbox. If you are installing an ID certificate, do not select any of the checkboxes.

Note: If you want to see the certificate you are accepting, click View.

7 When you are ready, click OK.

To install a live certificate, open this webpage:

dodpki.c3pki.chamb.disa.mil/rootca.html

Follow the onscreen instructions to install the live certificates. Make sure you install these certificates: dodroot.cac and dodrootmed.cac.

Using Mozilla to Send and Receive Email

You are now ready to send and receive signed or encrypted email messages. Before you open Mozilla, be sure your smart card reader is connected to your computer and insert your card. To receive encrypted email, be sure the person sending the message encrypts it using 3DES.

Setting Up FSCP to Use Other Directories

To authenticate a user when he or she logs in, FSCP gets information from the Common Access Card and sends it to Open Directory. Open Directory uses this information to look up the user record from the local NetInfo directory service and returns it to FSCP, which then passes the record to Mac OS X to finish the login process.

You can also use user accounts in an existing directory service, such as LDAP or Active Directory, to authenticate access to computers on your network. To do this, you make changes to the userLookupConfig.plist file installed by FSCP. This file specifies the search information necessary to look up the user record.

The userLookupConfig.plist file is in the SCLoginPlugin.bundle, which is located here:

/System/Library/CoreServices/SecurityAgentPlugins/

Here is the full pathname for the file:

/System/Library/CoreServices/SecurityAgentPlugins/SCLoginPlugin.bundle/Contents/
Resources/userLookupConfig.plist

Note: To see the contents of SCLoginPlugin.bundle, hold down the Control key and click the SCLoginPlugin.bundle icon, then choose Show Package Contents from the menu. In the window that opens, double-click the Contents folder, then the Resources folder.

The file specifies the search information as a series of XML key and value pairs. You change these pairs to specify the information to get from the Common Access Card to use for the search, the format of the search string, and the key for your directory service.

You can use any XML editor to change the userLookupConfig.plist file. Because the file is located in the Mac OS X System folder, you need to log in as root to change it.

Format of the userLookupConfig.plist file

When it’s installed, the userLookupConfig.plist file looks like this.

```
<array>
    <dict>
        <key>values</key>
        <array>
            <dict>
                <key>type</key>
                <string>DemographicData</string>
                <key>tag</key>
                <string>23</string>
                <key>value</key>
                <string>placeholder</string>
            </dict>
        </array>
        <key>formatString</key>
        <string>$1</string>
        <key>userLookupKey</key>
        <string>dsAttrTypeNative:_DoD_EDI_Identifier</string>
    </dict>
</array>
```

The definition of the user lookup is an array that consists of three keys and their values:

- The value of the `values` key is an array that specifies the search information to get from the card. Each element of the array specifies one piece of information that replaces the `placeholder` string of the `value` key of that element. The installed file specifies getting the EDI Identifier from the demographic data on the card. You can specify different information, including getting several items from the card.
- The value of the `formatString` key specifies the form of the search string sent to Open Directory. The `$1` in the installed file specifies sending the value of the `value` key from the first element of the `values` array to Open Directory.
- The `userLookupKey` key specifies the directory service data key to search. The first part of the value specifies the type of the key and the second part the key name.

You can change the lookup configuration based on your existing directory service. For example, if your directory service specifies user records by using the “NT Principal Name” from the signing certificate and the name of the key in the directory schema is “KeyName,” the userLookupConfig.plist file might look like this when you finish changing it:

```
</array>
    <dict>
        <key>values</key>
        <array>
            <dict>
                <key>type</key>
                <string>CertificateData</string>
                <key>certSelect</key>
                <string>1</string>
                <key>tag</key>
                <string>NT Principal Name</string>
                <key>value</key>
                <string>placeholder</string>
        </dict>
        </array>
        <key>formatString</key>
        <string>$1</string>
        <key>userLookupKey</key>
        <string>dsAttrTypeNative:KeyName</string>
    </dict>
</array>
```

The following sections provide additional information about the format of the file.

The `values` Array

The `values` array consists of a `type` key and two or three other keys depending on the type, which can have one of these values:

- `DemographicData`

  Includes first, middle, and last name, the EDI Identifier, and other information.
- `CertificateData`

  Information about the identity, signing, or encryption certificate on the card.

If the value of the `type` key is `DemographicData`, the array has two additional keys:

- The value of the `tag` key specifies the demographic data to get from the card. For example, specifying `23` for the `tag` value returns the EDI Identifier number.
- The value of the `value` key is a placeholder. It is replaced by the information returned by the card. For example, the EDI Identifier might be "1603987654".

You must specify the value of the `tag` as a decimal number. Demographic tags are defined in "Defense Manpower Data Center Common Access Card Application Programming Interface" using hexadecimal numbers. See "Demographic Tag Values" later in this document for a list of the tags and their hexadecimal and decimal values.

If the value of the `type` key is `CertificateData`, the array has three additional keys:

- The value of the `certSelect` key specifies which certificate to use. The value can be one of these (other values are not defined):

  ```
  0:  Identity Certificate
  1:  Signing Certificate
  2:  Encryption Certificate
  ```

- The value of the `tag` key specifies the information to get from the card. These values are allowed:

  ```
  // Fields in the subjectAltName
  "RFC 822 Name"               // e.g. smith@navy.mil
  "NT Principal Name"          // e.g. 0123456789@mil

  // Fields in the Subject Name
  "Common Name"            // e.g. SMITH.JOHN.Q.0123456789
  "Organizational Unit"    // e.g. USN
  "Organizational Unit"    // e.g. PKI
  "Organizational Unit"    // e.g. DoD
  "Organization"           // e.g. U.S. Government
  "Country"                // e.g. US
  ```

  To specify one of the organizational units, add `<:n>` to the end. For example,

  ```
  Organizational Unit:0
  ```

  returns "USN."

  Note: These values are the same as the labels shown for certificates in the ReadCAC application.

- The value of the `value` key is a placeholder. It is replaced by the information returned by the card.

The formatString Key

The value of the `formatString` key specifies the format of the search string. You can specify which items in the `values` array to use in the search string, their order, and any literal string elements needed to search your directory service for the user record.

To specify values in the `values` array to include in the search string, use `$<n>`, where `<n>` is an index to the item in the `values` array. The first item in the array is 1, the second 2, and so forth.

As seen in the examples above, the `formatString` value can be as simple as `$1`, which indicates using the first and only item in the `values` array for the search string.

You can form more complicated searches by combining several items from the `values` array with string literals. The following example includes the first, second, and third element from the `values` array separated by periods and followed by the string literal ".mil":

```
$1.$2.$3.mil
```

The userLookup Key

The `userLookupKey` key specifies the data key of the directory service to search. You specify the key by its type and name. There are two types of attributes: "standard" and "native."

- Open Directory defines the "standard" attribute keys. You identify a standard attribute key in the value of the `userLookupKey` key using this attribute type:

  ```
  dsAttrTypeStandard
  ```

- The directory service defines "native" attribute keys. You identify a native attribute in the value of the `userLookupKey` key using this attribute type:

  ```
  dsAttrTypeNative
  ```

  For example, if the name of the attribute in your directory service is "MyUniqueIDField," you would set the value of the `userLookupKey` key to

  ```
  dsAttrTypeNative:MyUniqueIDField
  ```

Note: For more information on the way Open Directory works, see *Inside Mac OS X*.

Open Directory defines both of the attribute type identifiers and the standard attribute keys in the file named DirServicesConst.h, located in

/System/Library/Frameworks/DirectoryService.framework/Versions/Current/Headers/

Examples of Searching User Records

This section provides additional examples for changing the userLookupConfig.plist file.

A Simple Change

This example gets the EDI Identifier from the Common Access Card.

```
<array>
    <dict>
        <key>values</key>
        <array>
            <dict>
                <key>type</key>
                <string>DemographicData</string>
                <key>tag</key>
                <string>23</string>
                <key>value</key>
                <string>placeholder</string>
            </dict>
        </array>
        <key>formatString</key>
        <string>$1@mil</string>
        <key>userLookupKey</key>
        <string>dsAttrTypeStandard:uniqueID</string>
    </dict>
</array>
```

The `formatString` specifies using the EDI Identifier returned by the card (for example, “1604678933”) followed by the literal string “@mil”. The resulting search string is “1604678933@mil”.

The `userLookupKey` specifies the standard Open Directory key, `uniqueID`.

FSCP passes the search string and key to Open Directory, which searches for a user record with the value of the search string as its unique ID.

Searching Using Certificate Data

This example searches for the user record using two search values from the certificate data on the Common Access Card.

```
<array>
    <dict>
        <key>values</key>
        <array>

        <dict>
            <key>type</key>
            <string>CertificateData</string>
            <key>certSelect</key>
            <string>1</string>
            <key>tag</key>
            <string>NT Principal Name</string>
            <key>value</key>
            <string>placeholder</string>
        </dict>
        <dict>
            <key>type</key>
            <string>CertificateData</string>
            <key>certSelect</key>
            <string>1</string>
            <key>tag</key>
            <string>Organizational Unit:0</string>
            <key>value</key>
            <string>placeholder</string>
        </dict>
        <key>formatString</key>
        <string>$1.$2.mil</string>
        <key>userLookupKey</key>
        <string>dsAttrTypeNative:CertInfo</string>
    </dict>
</array>
```

Both search items specify using the signing certification. The first search item looks up the NT Principal Name (for example, “The_Name”) while the second search item looks up an Organizational Unit, which is “USN”.

The `formatString` combines these two elements separated by a period. The resulting string is “The_Name.USN.mil”. The `userLookupKey` specifies searching for this value in a directory key named “CertInfo.”

Searching Using a Combination of Data

This example searches using both demographic data and certificate data.

```
<dict>
    <key>values</key>
    <array>
        <dict>
            <key>type</key>
            <string>DemographicData</string>
            <key>tag</key>
            <string>23</string>
            <key>value</key>
            <string>0123456789</string>
        </dict>
        <dict>
            <key>type</key>
            <string>CertificateData</string>
            <key>certSelect</key>
            <string>1</string>
            <key>tag</key>
            <string>Organizational Unit:0</string>
            <key>value</key>
            <string>placeholder</string>
        </dict>
    </array>
    <key>formatString</key>
    <string>$1@$2.mil</string>
    <key>userLookupKey</key>
    <string>dsAttrTypeNative:MyUserIdentifier</string>
</dict>
```

The first item in the `values` array specifies using the EDI Identifier demographic data from the card. The second item in the array specifies looking up data in the signing certificate for the Organizational Unit, which is "USN".

The `formatString` combines the two items separated by the "@" symbol and followed by the string literal ".mil". The resulting string might look similar to this:

```
1604678933@USN.mil
```

The `userLookupKey` specifies searching for this value in the directory service key named "MyUserIdentifier."

Demographic Tag Values

The “Defense Manpower Data Center Common Access Card Application Programming Interface” specifies demographic tags in hexadecimal. You specify the tag value in the userLookupConfig.plist file using the decimal equivalent. This table lists the demographic tags with their hexadecimal and decimal values.

Demographic tag	Decimal	Hexadecimal
Person First Name	1	1
Person Middle Name	2	2
Person Last Name	3	3
Person Cadency Name	4	4
Person Identifier	5	5
Date of Birth	6	6
Sex Category Code	7	7
Person Identifier Type Code	8	8
Blood Type Code	17	11
DoD EDI Person Identifier	23	17
Organ Donor	24	18
Identification Card Issue Date	98	62
Identification Card Expiration Date	99	63
Date Demographic Data was Loaded on Chip	101	65
Date Demographic Data on Chip Expires	102	66
Card Instance Identifier	103	67
Exchange Code	18	12
Commissary Code	19	13
MWR Code	20	14
Non-Medical Benefits Association End Date	27	1B
Direct Care End Date	28	1C

Demographic tag	Decimal	Hexadecimal
Civilian Health Care Entitlement Type Code	208	D0
Direct Care Benefit Type Code	209	D1
Civilian Health Care End Date	210	D2
Meal Plan Type Code	26	1A
DoD Contractor Function Code	25	19
US Government Agency/Subagency Code	32	20
Branch of Service Code	36	24
Pay Grade Code	37	25
Rank Code	38	26
Personnel Category Code	52	34
Non-US Government Agency/Subagency Code	53	35
Pay Plan Code	54	36
Personnel Entitlement Condition Code	211	D3

022-0584