Copyright

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.
Contents

About This Document .................................................................................................................. v
Document Conventions ............................................................................................................. vi

Chapter 1

System Overview ....................................................................................................................... 1

Overview ................................................................................................................................... 1
Introduction .............................................................................................................................. 1
Managing Public/Private Key Pairs ............................................................................................ 2
Decrypting Files ....................................................................................................................... 9

NCC Glossary of Terms ............................................................................................................. 17

Index .......................................................................................................................................... 19
About This Document

Scope

This document describes how Printshop:

- generates and distributes the security key to the operator
- decrypts the operator-provided voucher batch files

It also explains the format of the Voucher Batch file.

It does not include detailed design of the service.

Audience

This guide is intended for use by personnel of the print shop who will be responsible for the end-to-end voucher printing process.

Related documents

The following documents are related to this document:

- CCS Technical Guide
- CCS Voucher Management User's Guide
Document Conventions

Typographical Conventions

The following terms and typographical conventions are used in the Oracle Communications Network Charging and Control (NCC) documentation.

<table>
<thead>
<tr>
<th>Formatting convention</th>
<th>Type of information</th>
</tr>
</thead>
</table>
| **Special Bold**      | Items you must select, such as names of tabs.  
Names of database tables and fields. |
| **Italics**           | Name of a document, chapter, topic or other publication.  
Emphasis within text. |
| **Button**            | The name of a button to click or a key to press.  
**Example:** To close the window, either click **Close**, or press **Esc**. |
| **Key+Key**           | Key combinations for which the user must press and hold down one key and then press another.  
**Example:** Ctrl+P, or Alt+F4. |
| **Monospace**         | Examples of code or standard output. |
| **Monospace Bold**    | Text that you must enter. |
| **variable**          | Used to indicate variables or text that should be replaced. |
| **menu option > menu option >** | Used to indicate the cascading menu option to be selected, or the location path of a file.  
**Example:** Operator Functions > Report Functions  
**Example:** /IN/html/SMS/Helptext/ |
| **hypertext link**    | Used to indicate a hypertext link on an HTML page. |

Specialized terms and acronyms are defined in the *Glossary* at the end of this guide.

**Terminology**

This topic explains any terminology specific to this manual.

**Operator**

An operator is the telecommunications service provider which generates the vouchers or calling cards which need printing.
Chapter 1

System Overview

Overview

Introduction

This chapter provides an overview of the software and formats used in preparing a voucher batch file for
printing.

In this chapter

This chapter contains the following topics.

Introduction ........................................... 1
Managing Public/Private Key Pairs .......... 2
Decrypting Files ...................................... 9

Introduction

Charging Control Services files and encryption

Charging Control Services produces encrypted voucher and account batch files for printing. The
encryption is used to provide security for the vouchers or subscriber accounts the files hold. Before the
files are printed, the encrypted files must be decrypted using the same public private key pair that was
used for the encryption.

For more information about how CCS generates vouchers and accounts, see CCS User's Guide and
CCS Technical Guide.

Public and private key encryption

Public and private key encryption (also known as asymmetric encryption) involves a pair of keys:

1. a public key which is used encrypt the file, and
2. a private key which is used to decrypt the file.

Both keys are generated by the holder of the private key. The public key is made available to others
who want to send encrypted files to the private key holder. In this case, the print shop will generate the
public and private keys and provide the public key to the operator.

For more information about:

- generating keys, see Managing Public/Private Key Pairs (on page 2).
- decrypting files, see Decrypting Files (on page 9).

More information about public and private key encryption is widely available in publications and on the
Internet.
Recommended software

Oracle uses GnuPG to encrypt batch files. These files can be decrypted using any software which supports gnupg public private keys. This guide covers the GnuPG command line tool, and the GPG4Win WindowsXP-compatible software.

Note: Other software such as PGP can also be used successfully for generating and exporting keys and decrypting files. Please use the software which is most suitable for your platform.

For more information about GnuPG (including downloadable software), see http://www.gnupg.org.
For more information about GPG4Win (including downloadable software), see http://www.gpg4win.org.
For more information about PGP (including purchasable software), see http://www.pgp.com.

Managing Public/Private Key Pairs

Generating GPG keys

A public and private GPG key can be generated from a pass-phrase. The private key is held only by the print shop and used only to decode the encrypted batch file. The public key is used to encrypt the file and must therefore be supplied to the operator who will be responsible for generating the voucher batch file.

For more information about using GPG keys with exported files, see Print Shop Operations Guide.

Generating keys using gpg

Follow these steps to generate a key using GnuPG.

Important: Additional documentation is available at http://www.gnupg.org. Always consult the recent documentation for your version of GnuPG if you are unsure of any steps in the procedure.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Log into the machine which has the GnuPG tool installed.</td>
</tr>
</tbody>
</table>
| 2    | Run the gpg binary.  
**Example command:** ./gpg --gen-key  
**Result:** Text similar to the following will appear:  
gpg (GnuPG) 1.4.6; Copyright (C) 2006 Free Software Foundation, Inc.  
This program comes with ABSOLUTELY NO WARRANTY.  
This is free software, and you are welcome to redistribute it under certain conditions. See the file COPYING for details.  
gpg: directory `/home/users/cmorris/.gnupg' created  
gpg: can't open `/gnupg/options.skel': No such file or directory  
gpg: keyring `/home/users/cmorris/.gnupg/secring.gpg' created  
gpg: keyring `/home/users/cmorris/.gnupg/pubring.gpg' created  
Please select what kind of key you want:  
(1) DSA and Elgamal (default)  
(2) DSA (sign only)  
(5) RSA (sign only)  
Your selection? |
| 3    | Enter the kind of algorithm you have agreed with the operator you are printing for.  
**Result:** Text similar to the following will appear:  
DSA keypair will have 2048 bits.  
ELG-E keys may be between 2048 and 4096 bits long.  
What keysize do you want? (2048) |
### Chapter 1

#### System Overview

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 4 | **Step** 4: Enter the keysize you have agreed with the operator.  
**Result:** Text similar to the following will appear.  
Requested keysize is 2048 bits  
Please specify how long the key should be valid.  
- 0 = key does not expire  
- \(<n>\) = key expires in \(n\) days  
- \(<n>w\) = key expires in \(n\) weeks  
- \(<n>m\) = key expires in \(n\) months  
- \(<n>y\) = key expires in \(n\) years  
Key is valid for? (0) |
| 5 | **Step** 5: Enter the expiry period you have agreed with the operator.  
**Result:** Text similar to the following will appear.  
Key does not expire at all  
Is this correct? (y/N) y |
| 6 | **Step** 6: If all the details entered so far are correct, type y and press Enter.  
**Result:** Text similar to the following will appear:  
You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form:  
"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"  
Real name:  
Type your real name and press Enter.  
**Result:** Text similar to the following will appear.  
Email address:  
Type your email address and press Enter.  
**Result:** Text similar to the following will appear.  
Comment:  
Type a comment and press Enter. The comment should identify who the printshop is, and may also identify the operator.  
**Result:** Text similar to the following will appear.  
You selected this USER-ID:  
"ExampleUser (TelcoEurope-Printshop) <example.user@example.com>"  
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? |
| 7 | **Step** 7: If the details which have been displayed are correct, type O and press Enter.  
**Result:** Text similar to the following will appear.  
You need a Passphrase to protect your secret key.  
Type a passphrase and press Enter.  
**Important:**  
- This passphrase must be entered when the files are decrypted. If the passphrase is not available, the files will not be able to be decrypted and a new pair of keys and batch file will have to be generated.  
- The passphrase is an important contributor to the overall security of the encryption. Ensure you follow any guidelines set by the operator, and that you pick a secure password.  
**Result:** Text similar to the following will appear:  
Confirm passphrase:  
Type the passphrase from step 11 again and press Enter.  
**Result:** Text similar to the following will appear as gpg generates the keys.  
We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number |
### Exporting keys using gpg

Follow these steps to export keys which have been generated by gpg.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>On the machine where the gpg generated the keys, instruct gpg to export the keys.</td>
</tr>
</tbody>
</table>

#### Example command:

```
gpg --armor --export <email_addr_from_key_creation> > <file>.key
```

#### Result: gpg will export the public key to the specified file.

If you cat the file, it will look similar to the example key shown below:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.7 (GNU/Linux)
mQGiBEfsGcYRBADQn11NupDz4MFMPFxcCiBRKMBz1k1scAixqhZ2TKK8Eo66tM6cbXYy2a09yKbUJykJ
x1bYbDEwBQQT5d5+4wADQg8FCkTB6uUC9y91I0ms5MqIPhGtIQLJcJWCG0o2J3q5kMr6/2zhFL7D1JU
Mk0V5aGZMmRSrdIS+XHXp4fLJ3jazLykmKfKQQ2Z842Zb1h1j+g7jyJWyn85r6gDFYDTy/eOyY2KMeL6q5
9jL890dDua8eo2z5k1idG8sc4XEPfjF3A1rFvXhC1j1uU51/SimuTFH9bQp3e3bFJ12ZpIM6OnouMVI
aSmuNv6T1YAQTgEUIyGh485SUayCYjC8YF3Zme7kc0RqgIAnn9Q36nt0e77/c0FL8wIQLTBCMK
7Y71t7YNOcLCQgAmAEFEQQAI94A9vCnkm6NJcmataJezBtI7E5rikVQXyMKn1nNnDKX9b01JSHeduBFp
wq0US5l1tpzqJ31tUEVqKurJ/3DAIvS2rgDsdwMVXhwbtwRXdYFaPHz42ur+w7/lqk6eodUuxYyZ
Gd9/lpZl3e1j3jyYs8f3aDvY3BecvXoC4BIVfqug4zGiQx2QWqAgMBA4B8eA7Jp7JFMR/kOvBD3ZEV
DExn/mUGjO9H1yPxTmx7q0eB0vHyk4aXmg4d4GwqducOW23kbRyJ5m59g/aC8ckkPAcn7H5yXyYi
vxO1UJeH3zHTIT02Hwxk21L9V/Lp8PckY2ze2i3QP1mQOLBzJdij6KPOZypzfAWKT5H5vogne0u0L4
F8AKVXc6QyZpQ/SIP71R7qjCQq0MCN6GbKIdxAbIBGA6aY4ISLh2zPnwk9GQvZXXJ22AIG17RbaA7d7v
ISBisn/0T1a6j1lUQW4AYB2m9U091LXDRM2JwXWb81I0UgCxnX2Cv8v7mMkwbhx7CKDQRH7BnGIEaMcgw
NyyXybabQDrzIt5zFzjgok1BAU4C+WzgIbA9zR1cSFLhbNj9GaW0aWuZACoMxh4haCkgPFBhWBDc
0R53aNjwuIU3ObfcdofdmwvyoU3DOL12ZdWyMWFpCwqCgkA02ikjmc1y5M1Sjg4a1s1q0AEALHRCgQw
DPMCNvL9w7QFq0QDP72UY59y8sFB83MnU1214tjG8dmyQDy7a21S75er6cu5tjncPcky70c
jPTKYJ0YPMWNKRPH1a9WyP7w9C7RIU/ix19Qqhph9q3w+zh0/2zge/ZTtx/CIjdOq7yqAJ
6mPngu/TGq/4rgu5oMoSO3TQrPo3s66ckB2M+ONCj8wUDGVk/V4RQzFDC7hU8LLnyI1yGpZRXITlh5v
Hrte60W6Swpxh7z29Y8EaHi9UU77FCizadoaVLPVLpcc89pw2e3Cqkgpo1Q6eeNvJc9gobZTHNgg/cL
EIQP+yccNio2aHcR+CY7DqfjZBFSUR0jQpFBE8LXLJH87B8igRagABJqBH7nBAhAAoJEFJ/mqR0H
UNzxmYBms65TherAMMMtjQ0Xj/XySSAAjQg4BWP2c9ujw8rG05e2Q==
=Q4J
-----END PGP PUBLIC KEY BLOCK-----
```
Step | Action
--- | ---
2 | Send the public key to the operator. The operator will import it into Charging Control Services and will use it when generating voucher and subscriber account (calling card) batches.

**Generating keys using Gpg4Win**

Follow these steps to generate a new key using Gpg4Win.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1 | Start GNU Privacy Assistant - Keyring Editor.  
Result: The GNU Privacy Assistant - Keyring Editor screen will open. |
| 2 | Select Keys, New Key.  
Result: The Generate key screen opens. |
3 Select values from the **Algorithm:** and **Key size (bits):** fields. The values you pick should have been agreed with the operator you are printing for.

4 In the **User ID:** field, enter your email address.

5 In the **Email:** field, enter your email address.

6 In the **Comment:** field enter a description of this key. The comment should identify who the printshop is, and may also identify the operator.

7 In the **Passphrase:** and **Repeat passphrase:** fields, type the passphrase to use with this key.

**Important:**

- This passphrase must be entered when the files are decrypted. If the passphrase is not available, the files will not be able to be decrypted and a new pair of keys and batch file will have to be generated.

- The passphrase is an important contributor to the overall security of the encryption. Ensure you follow any guidelines set by the operator, and that you pick a secure password.

8 In the **Expiration** area, select an expiry date as agreed with the operator.

9 Click **OK**.

**Results:**

- The gpa.exe screen will appear. It will have a moving bar which indicates it is...
Chapter 1

System Overview

Step | Action
--- | ---
| | generating the new key.

- When the new key has been generated, it will appear in the GNU Privacy Assistant - Keyring Editor screen.

---

Exporting keys using Gpg4Win

Follow these steps to export keys which have been generated by gpg.

Step | Action
--- | ---
1 | On the machine where Gpg4Win generated the keys, start GNU Privacy Assistant - Keyring Editor.
Result: The GNU Privacy Assistant - Keyring Editor screen will open.
Chapter 1

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Select the key you want to export and click <strong>Export</strong>. Result: The Export public keys to file screen appears.</td>
</tr>
<tr>
<td>3</td>
<td>Browse to the directory you want the exported keys to be stored in.</td>
</tr>
<tr>
<td>4</td>
<td>Type the name of the file you want the exported keys to be stored in and click <strong>OK</strong>.</td>
</tr>
</tbody>
</table>
Step | Action
--- | ---
5 | Click **Close**.
6 | Send the public key to the operator. The operator will import it into Charging Control Services and will use it when generating voucher and subscriber account (calling card) batches.

**Decrypting Files**

**Introduction**

The batch file provided by the operator for printing will have been encrypted using the public key provided by the printshop. This file will need to be decrypted using the matching private key. There are two methods for decrypting the files.

The voucher batch file should be placed on the Printshop target PC. The PGP software on the PC should be used to decrypt the voucher batch file.

**Decrypting files using gpg**

Follow these steps to decrypt a file using gpg.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Copy the batch file to the machine where the key was exported from.</td>
</tr>
</tbody>
</table>
| 2 | Use gpg to decrypt the batch file.  
**Example command:** gpg -o batchFile.txt --decrypt batchFile.gpg  
**Result:** Text similar to the following will appear.  
You need a passphrase to unlock the secret key for  
user: "ExampleUser (TelcoEurope-Printshop) <example.user@example.com>"  
2048-bit ELG-E key, ID 69372FCB, created 2009-01-16 (main key ID 2E72F865)  

Enter passphrase: |
| 3 | Type the passphrase used when the key was generated and press **Enter**.  
**Result:** gpg will use the passphrase to decrypt the file. Text similar to the following will appear.  
user: "ExampleUser (TelcoEurope-Printshop) <example.user@example.com>"  
2048-bit ELG-E key, ID 69372FCB, created 2009-01-16 (main key ID 2E72F865)  

The decryption should now be complete. |
Decryption of files using Gpg4Win

Follow these steps to decrypt an encrypted file using Gpg4Win.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Using Windows Explorer, browse to the file you want to decrypt.</td>
</tr>
</tbody>
</table>
| 2    | Select the file and right-mouse-click on the document.  
**Result:** The right-mouse-click menu appears. |
| 3    | Select GPGee, Verify/Decrypt.  
**Result:** The Verify/Decrypt Files and Enter passphrase screens open. |
| 4    | Type the passphrase for the specified key in the **Enter passphrase** field. |
| 5    | Click **Ok**.  
**Result:** The results of the file decryption will be displayed in the Verify/Decrypt Files screen and the decrypted file will be saved to the same directory as the source file. |
Exported voucher batch files

Voucher batch file format is controlled by the security library, and the voucher writer plugin used to generate the batch. Which libraries and plugins are used is defined by the Authentication Module (PAM) and the Authentication Rule specified in the New Voucher Batch screen.

Header fields are in the format "<Key field name>=<value>". Key field names always start with an alphabetic character. This makes it easy to distinguish them from voucher records (which always start with a number).

The following header fields are used in the voucher batch file header, (although downstream processors should detect any "<Key field name>=<value>" lines).

<table>
<thead>
<tr>
<th>Header field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BilingEngineName=&lt;str&gt;</td>
<td>The name of the Voucher and Wallet Server where the voucher resides.</td>
</tr>
<tr>
<td>VoucherTypeName=&lt;str&gt;</td>
<td>The name of the voucher type as created on the NCC platform. The voucher type contains the following information:</td>
</tr>
<tr>
<td></td>
<td>1 Pre-use expiry period (number of days and hours that this voucher is valid in a pre-use state)</td>
</tr>
<tr>
<td></td>
<td>2 Wallet expiry period (change the current wallet expiry date by this many days and hours)</td>
</tr>
<tr>
<td></td>
<td>3 Voucher number length</td>
</tr>
<tr>
<td></td>
<td>4 Voucher PIN length</td>
</tr>
<tr>
<td></td>
<td>5 A list of all the balance types, associated values and balance expiry date modifications which will be changed/updated when this voucher is redeemed</td>
</tr>
</tbody>
</table>
Chapter 1

Note: It will be up to the operator to provide the details of the voucher type described here to the print shop so that any specific voucher details can be printed on the final vouchers.

AuthRuleName=<str> The name of the authentication rule which was used for creating the voucher number and PIN.
AuthModName=<str> The name of the pluggable authentication module (PAM) (NCC specific) used for creating the voucher PIN.
VoucherBatchBatch=<str> A two character identifier (non unique) for this voucher batch.
VoucherBatchID=<int> The system generated ID for this voucher batch.
OriginalCount=<int> The number of vouchers created in this batch.
StartOfRange=<int> Beginning of the range of voucher numbers.
EndOfRange=<int> End of the range of voucher numbers.

A line consisting of a single equal sign (=) terminates the header lines. All subsequent lines are voucher detail records.

CCS3 DES voucher batch example

This text shows an example export voucher batch file generated by ccsVoucher_CCS3 using the DES encryption library (and a bespoke voucher file writer plugin to format the non-header details), but no GnuPG key.

# # Voucher file for batch 83 # Generated by ccsVoucher at Tue Nov 11 12:55:27 2008 # (key=value or voucherserialnumber, vouchernumber, vouchersecret, vouchercontext, voucherprivate_secret ) # BillingEngineName=PCDEV VoucherTypeName=DES AuthRuleName=DES (VL=10 VP=4) AuthModName=DES VoucherBatchBatch= VoucherBatchID=83 OriginalCount=2 StartOfRange=1000000001 EndOfRange=1000000002 = # # Voucher records start # 1000000001,8986 1000000002,4887 # # End of voucher records #

CCS3 CB10 voucher batch example

This text shows an example export voucher batch file generated by ccsVoucher_CCS3 using the 'CB10 HRN' encryption library using the 'HRNGEN' encryption algorithm, but no GnuPG key.

# # Voucher file for batch 85 # Generated by ccsVoucher at Tue Nov 11 12:55:27 2008 # (key=value or voucherbatch, preuseexpiry, hrn, serialnumber) # BillingEngineName=PCDEV VoucherTypeName=CB10
Chapter 1

AuthRuleName=CB10  (S=14 R1=2 R2=2 R3=0)
AuthModName=CB10 HRN
VoucherBatchBatch=
VoucherBatchID=85
OriginalCount=2
StartOfRange=00000000000001
EndOfRange=00000000000002
#
# Voucher records start
#
85,20090101000000,631599527570333589,1000000138
85,20090101000000,855619036698319621,1000000139
#
# End of voucher records
#

CCS3 CB10 GPG voucher batch example
This text shows an example export voucher batch file generated by ccsVoucher_CCS3 using the 'CB10 HRN' encryption library using the 'HRNGEN' encryption algorithm, and GnuPG encryption.

Note: This file has been decrypted using the gpg key.
#
# Voucher file for batch 86
# Generated by ccsVoucher at Tue Nov 11 12:55:27 2008
# (key=value or voucherserialnumber,hrnserialnumberseed,hrn,nnrnumber,hrnc)
#
BillingEngineName=PCDEV
VoucherTypeName=CB10 HRN
AuthRuleName= CB10 (S=14 R1=2 R2=2 R3=0)
AuthModu
leName=CB10 HRN
VoucherBatchBatch=
VoucherBatchID=86
OriginalCount=2
StartOfRange=00000000000003
EndOfRange=00000000000004
#
# Voucher records start
#
86,20090101000000,057195727842702414,1000000138
86,20090101000000,363323157948027866,1000000139
#
# End of voucher records
#

Exported card/account batch files
Subscriber account/calling card batch file format is controlled by the account writer plug-in used to generate the batch. Which libraries are used is defined by the authentication name specified in the New Subscriber Batch screen.

Header fields are in the format "Key_field_name=value". Key field names always start with an alphabetic character. This makes it easy to distinguish them from voucher records (which always start with a number).

The following header fields are used in the voucher batch file header, (although downstream processors should detect any "Key_field_name=value" lines).

<table>
<thead>
<tr>
<th>Header field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountBatchID=int</td>
<td>The ID of the subscriber account batch.</td>
</tr>
<tr>
<td>Header field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ServiceProviderID=int</td>
<td>The ID number of the service provider the subscriber batch belongs to. When ccsAccount is started by the screens the value of this field is populated by the id of the service provider which is selected in the Service Provider field of the Subscriber Management screen when the New button is clicked.</td>
</tr>
<tr>
<td>AccountTypeID=int</td>
<td>The product type the subscriber batch has. When ccsAccount is started by the screens the value of this field is populated by the Product Type field on the New Subscriber Batch screen.</td>
</tr>
<tr>
<td>maxConcurrent=int</td>
<td>The maximum number of concurrent connections wallets generated with this subscriber batch can have. When ccsAccount is started by the screens the value of this field is populated by the Maximum Concurrent Accesses field on the New Subscriber Batch screen.</td>
</tr>
<tr>
<td>BatchSize=int</td>
<td>The number of subscriber accounts in this batch. When ccsAccount is started by the screens the value of this field is populated by the Batch Size field on the New Subscriber Batch screen.</td>
</tr>
<tr>
<td>RangeStart=int</td>
<td>Beginning of the range of subscriber account numbers. When ccsAccount is started by the screens the value of this field is populated by the Card Number Start Range field on the New Subscriber Batch screen.</td>
</tr>
<tr>
<td>RangeEnd=int</td>
<td>End of the range of subscriber account numbers. When ccsAccount is started by the screens the value of this field is populated by the Card Number End Range field on the New Subscriber Batch screen.</td>
</tr>
<tr>
<td>AuthenticationModuleID=int</td>
<td>The ID of the authentication module used for: • Encryption and/or random generation of PINs for this batch • (optionally) sends the output file for encryption by gpg. When ccsAccount is started by the screens the value of this field is populated by the PAM Name field on the New Subscriber Batch screen.</td>
</tr>
<tr>
<td>BillingEngineID=int</td>
<td>The ID number of the Voucher and Wallet Servers.</td>
</tr>
<tr>
<td>CurrencyID=int</td>
<td>The ID of the currency the wallets generated with this subscriber batch will use. When ccsAccount is started by the screens the value of this field is populated by the Wallet Currency field on the New Subscriber Batch screen.</td>
</tr>
<tr>
<td>LimitType=str</td>
<td>The type of limit the wallets generated with this subscriber batch will use.</td>
</tr>
<tr>
<td>BalanceType=int</td>
<td>The balance type ID of the balance type this wallet will have any initial value stored in.</td>
</tr>
</tbody>
</table>

A line consisting of a single equal sign (=) terminates the header lines. All subsequent lines are voucher detail records.

**Card/account output file**
This text shows an example export subscriber account/calling card output file.

# Account Batch Output File
# Generated Wed Dec 31 01:24:29 2008
# AccountBatchID=59
ServiceProviderID=1
AccountTypeID=7
maxConcurrent=1
BatchSize=10
RangeStart=8815000000
RangeEnd=8819990000
AuthenticationModuleID=4
BillingEngineID=2
CurrencyID=2
LimitType=DEBT
BalanceType=1

Dec 31 01:24:29.861203 ccsAccount(15179) NOTICE: Beginning account generation.
16309877,3415992,7,GS.H3zcJxKzby,8800127
19052821,0363266,7,GSfRbQyO15unk,8800128
18627603,5447142,7,GSzefn9Gh2gSY,8800129
16635167,9003194,7,GSnkF67MozS9g,8800130
19498256,8441931,7,GSfZtbQvbO1g,8800131
18758105,8744644,7,GSgU9UULMLZttw,8800132
17349265,3517347,7,GSgH/ML4NHZs,8800133
16223817,0064708,7,GSbgIe4gPO.U,8800134
16089674,7771756,7,GS1Xy7ySSzszVw,8800135
16405822,1207166,7,GSJugOSguxj7qg,8800136
Dec 31 01:24:35.514685 ccsAccount(15179) NOTICE: Progress 10/10 (100.0%) Complete
Dec 31 01:24:35.515578 ccsAccount(15179) NOTICE: Account generation complete.
NCC Glossary of Terms

CCS
1) Charging Control Services (or Prepaid Charging) component.
2) Common Channel Signalling. A signalling system used in telephone networks that separates signalling information from user data.

HRN
Hidden Reload Number

HTML
HyperText Markup Language, a small application of SGML used on the World Wide Web. It defines a very simple class of report-style documents, with section headings, paragraphs, lists, tables, and illustrations, with a few informational and presentational items, and some hypertext and multimedia.

Oracle
Oracle Corporation

PC
Point Code. The Point Code is the address of a switching point.

PIN
Personal Identification Number

Service Provider
See Telco.

SGML

Telco
Telecommunications Provider. This is the company that provides the telephone service to customers.

Telecommunications Provider
See Telco.
<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
</tr>
<tr>
<td>About This Document • v</td>
</tr>
<tr>
<td>Audience • v</td>
</tr>
<tr>
<td><strong>C</strong></td>
</tr>
<tr>
<td>Card/account output file • 16</td>
</tr>
<tr>
<td>CCS • 17</td>
</tr>
<tr>
<td>CCS3 CB10 GPG voucher batch example • 14</td>
</tr>
<tr>
<td>CCS3 CB10 voucher batch example • 14</td>
</tr>
<tr>
<td>CCS3 DES voucher batch example • 13</td>
</tr>
<tr>
<td>Charging Control Services files and encryption • 1</td>
</tr>
<tr>
<td>Copyright • ii</td>
</tr>
<tr>
<td><strong>D</strong></td>
</tr>
<tr>
<td>Decrypting Files • 1, 10</td>
</tr>
<tr>
<td>Decrypting files using gpg • 10</td>
</tr>
<tr>
<td>Decrypting files using Gpg4Win • 11</td>
</tr>
<tr>
<td>Document Conventions • vi</td>
</tr>
<tr>
<td><strong>E</strong></td>
</tr>
<tr>
<td>Exported card/account batch files • 14</td>
</tr>
<tr>
<td>Exported voucher batch files • 12</td>
</tr>
<tr>
<td>Exporting keys using gpg • 5</td>
</tr>
<tr>
<td>Exporting keys using Gpg4Win • 8</td>
</tr>
<tr>
<td><strong>G</strong></td>
</tr>
<tr>
<td>Generating GPG keys • 2</td>
</tr>
<tr>
<td>Generating keys using gpg • 2</td>
</tr>
<tr>
<td>Generating keys using Gpg4Win • 5</td>
</tr>
<tr>
<td><strong>H</strong></td>
</tr>
<tr>
<td>HRN • 17</td>
</tr>
<tr>
<td>HTML • 17</td>
</tr>
<tr>
<td><strong>I</strong></td>
</tr>
<tr>
<td>Introduction • 1, 10</td>
</tr>
<tr>
<td><strong>M</strong></td>
</tr>
<tr>
<td>Managing Public/Private Key Pairs • 1, 2</td>
</tr>
<tr>
<td><strong>O</strong></td>
</tr>
<tr>
<td>Operator • vi</td>
</tr>
<tr>
<td>Oracle • 17</td>
</tr>
<tr>
<td>Overview • 1</td>
</tr>
<tr>
<td><strong>P</strong></td>
</tr>
<tr>
<td>PC • 17</td>
</tr>
<tr>
<td>PIN • 17</td>
</tr>
<tr>
<td>Public and private key encryption • 1</td>
</tr>
</tbody>
</table>

**R**
- Recommended software • 2
- Related documents • v

**S**
- Scope • v
- Service Provider • 17
- SGML • 17
- System Overview • 1

**T**
- Telco • 17
- Telecommunications Provider • 17
- Terminology • vi
- Typographical Conventions • vi