



Rocky Enterprise Linux 9.2 Manual Pages on command 'rct.8'

\$ man rct.8

RCT(8) RHSM Certificate Tool RCT(8)

NAME

rct - Displays information (headers) about or size and statistics of a entitlement, product, or identity certificate used by Red Hat Subscription Manager.

SYNOPSIS

```
rct cat-cert [--no-content] [--no-products] /path/to/certificate.pem
rct stat-cert /path/to/certificate.pem rct cat-manifest [--no-content]
/path/to/consumer_export.zip rct dump-manifest [--destination /path]
[--force] /path/to/consumer_export.zip
```

DESCRIPTION

Red Hat Subscription Manager uses X.509 certificates to identify a registered system (identity certificate), the products installed on that system (product certificates), and the subscriptions attached to the system (entitlement certificates), including available content repositories, products, and support levels. All of the information that Subscription Manager requires is contained in the body of the certificate.

COMMANDS

stat-cert

Prints the size of the certificate and other details about the certificate. The precise details depend on the type of certificate being checked.

cat-cert

Prints the information that is contained in the certificate itself, such as the certificate headers, serial numbers, products, and content sets. Two options, --no-content and --no-products, can be used to shorten the output to include only header and descriptive information.

cat-manifest

Prints the information that is contained in the subscription service manifest. The manifest is an archive of JSON files which contain all of the subscription information for subscriptions allocated to the on-premise service. The --no-content option can be used to reduce the detail shown in the output.

dump-manifest

Extracts the contents of the manifest archive.

THE STAT-CERT COMMAND

The `rct` tool is used to gather information about the already-issued certificates being used by Subscription Manager. The main reason for that is that certificate sizes, for a number of reasons, impact content delivery service performance.

For large accounts and organizations, there can be a very large number of products and content sets available. Older versions of entitlement certificates (version 1.0) used different (less efficient) DER encoding, so that large amounts of information results in very large certificates. (This is what caused timeouts or crashes when dealing with some content services.) Newer entitlement certificate versions (version 3.0) use more efficient encoding on large content sets, resulting in smaller certificate content sizes and better service performance.

If there are problems with the content service timing out or returning errors, then the `rct stat-cert` command can be used to check the size

and version of a given entitlement certificate quickly.

A large number of content sets is anything over 185 total sets. Both the total number of content sets and the size of the DER encoding in the certificate could affect performance.

OPTIONS

`/path/to/cert.pem`

Gives the full path and filename to the PEM certificate for the given subscription, product, or system. This is required.

EXAMPLES

The statistics for an entitlement certificate show both the DER size and the number of content sets, among other information:

- * Type (entitlement certificate)
- * Version (of the certificate style); newer versions will be 3.x, with better performance for handling large content sets
- * DER size, which gives the size of the certificate contents (not the size of the certificate file itself)
- * Key size, for the associated key file, in bytes
- * The total number of available content sets in the subscription

For example:

```
[root@server ~]# rct stat-cert /etc/pki/entitlement/2027912482659389239.pem
```

Type: Entitlement Certificate

Version: 1.0

DER size: 47555b

Subject Key ID size: 553b

Content sets: 100

While the size of the certificate is less of an issue for identity and product certificates (which are quite small), the stat-cert command can still be used to view the size and statistics of the certificates.

For a product certificate, the stat-cert command shows:

- * Type (product certificate)
- * Version (of the certificate style)
- * DER size, which gives the size of the certificate contents (not the size of the certificate file itself)

For example:

```
[root@server ~]# rct stat-cert /etc/pki/product/69.pem
```

Type: Product Certificate

Version: 1.0

DER size: 1558b

For an identity certificate:

- * Type (identity certificate)

- * Version (of the certificate style)

- * DER size, which gives the size of the certificate contents

(not the size of the certificate file itself)

- * Key size, for the associated key file, in bytes

For example:

```
[root@server ~]# rct stat-cert /etc/pki/consumer/cert.pem
```

Type: Identity Certificate

Version: 1.0

DER size: 1488b

Subject Key ID size: 20b

THE CAT-CERT COMMAND

Each certificate contains a complete set of information with all of the details for whatever element is being identified. That information can be displayed, in pretty-print form, using the cat-cert command.

OPTIONS

/path/to/cert.pem

Gives the full path and filename to the PEM certificate for the given subscription, product, or system. This is required.

--no-content

Returns all of the certification information, order information, and product information, but excludes all of the Content sections, which significantly reduced the information printed to stdout. This is for an entitlement certificate only.

--no-products

Returns all of the certification information, order information, and content (repository) information, but excludes all of the

Product sections, which significantly reduced the information printed to stdout. This is for an entitlement certificate only.

/path/to/cert.pem

Gives the full path and filename to the PEM certificate for the given subscription, product, or system.

OUTPUT

The command returns the most basic information about the certificate -- such as its directory path, its serial number and subject name, and its validity period (start and end dates) -- in the Certificate section:

- * Path -- the filesystem location where the certificate is in?
stalled
- * Version -- the certificate format version -- P * Serial -- the serial number for the certificate
- * Start/End Date -- the validity period for the certificate
- * Alt Name -- the subject alternative name, which uses the host? name of the system rather than the UUID (for identity certificates only)

The Subject DN of the certificate is in the Subject section.

For example, for the identity certificate:

```
[root@server ~]# rct cat-cert /etc/pki/consumer/cert.pem
```

```
+-----+
```

Identity Certificate

```
+-----+
```

Certificate:

Path: /etc/pki/consumer/cert.pem

Version: 1.0

Serial: 824613308750035399

Start Date: 2012-11-09 16:20:22+00:00

End Date: 2013-11-09 16:20:22+00:00

Alt Name: server.example.com

Subject:

CN: e94bc90e-44a1-4f8c-b6fc-0a3e9d6fac2b

A product certificate contains additional information in a Product sec?

tion, which defines the information for the specific installed product, such as its name, product version, and any yum tags used for that prod?

uct. For example:

```
[root@server ~]# rct cat-cert /etc/pki/product/69.pem

+-----+
      Product Certificate
+-----+

Certificate:

    Path: /etc/pki/product/69.pem
    Version: 1.0
    Serial: 12750047592154746449
    Start Date: 2012-10-04 18:45:02+00:00
    End Date: 2032-09-29 18:45:02+00:00

Subject:

    CN: Red Hat Product ID [b4f7ac9e-b7ed-45fa-9dcc-323beb20e916]

Product:

    ID: 69
    Name: Red Hat Enterprise Linux Server
    Version: 6.4
    Arch: x86_64
    Tags: rhel-6,rhel-6-server
```

The most information is contained in the entitlement certificate. Along with the Certificate and Subject, it also has a Product section that defines the product group that is covered by the subscription.

Then, it contains an Order section that details everything related to the purchase of the subscription (such as the contract number, service level, total quantity, quantities assigned to the system, and other details on the subscription).

A subscription for a product covers the version purchased and every previous version of the product. For example, when a subscription is purchased for Red Hat Enterprise Linux 6.4, the subscription provides full access to all RHEL 6 repositories, plus access to all RHEL 5 repositories and then other included product content repositories, like

Subscription Asset Manager. Every available content repository is listed in a Content section that contains the repository name, associated tags, its URL, and a notice on whether the yum repository is enabled by default. For example:

```
[root@server ~]# rct cat-cert /etc/pki/entitlement/2027912482659389239.pem
```

```
+-----+
```

Entitlement Certificate

```
+-----+
```

Certificate:

Path: /etc/pki/entitlement/2027912482659389239.pem

Version: 1.0

Serial: 2027912482659389239

Start Date: 2011-12-31 05:00:00+00:00

End Date: 2012-12-31 04:59:59+00:00

Subject:

CN: 8a99f9843adc8b8f013ae5f9de022b73

Product:

ID: 69

Name: Red Hat Enterprise Linux Server

Version:

Arch: x86_64,ia64,x86

Tags:

Order:

Name: Red Hat Enterprise Linux Server, Premium (8 sockets) (Up to 4 guests)

Number: 2673502

SKU: RH0103708

Contract: 10011052

Account: 5206751

Service Level: Premium

Service Type: L1-L3

Quantity: 100

Quantity Used: 1

Socket Limit: 8

Virt Limit:

Virt Only: False

Subscription:

Stacking ID:

Warning Period: 0

Provides Management: 0

Content:

Type: yum

Name: Red Hat Enterprise Linux 6 Server (RPMs)

Label: rhel-6-server-rpms

Vendor: Red Hat

URL: /content/dist/rhel/server/6/\$releasever/\$basearch/os

GPG: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

Enabled: True

Expires: 86400

Required Tags: rhel-6-server

THE CAT-MANIFEST COMMAND

A subscription management service is allocated a specific block of subscriptions that are available to an account. This list of subscriptions is the manifest for the service. The `cat-manifest` command reads and prints the details of the manifest, such as the creation date, the system UUID and name, available products, and subscription details. There are multiple JSON files in the archive, identifying different aspects of the subscription service and subscription configuration, such as the general manifest properties, subscription information, content and repository information, and product information.

OPTIONS

`--no-content`

Excludes all of the Content Sets sections, which significantly reduces the information printed to stdout.

`/path/to/consumer_export.zip`

Gives the path and filename (by default, `consumer_export.zip`) for the manifest file on the local system. This is required.

EXAMPLES

The command pretty-prints all of the details about the manifest itself and the allocated subscriptions, products, and content.

```
[root@server ~]# rct cat-manifest /tmp/consumer_export.zip
```

```
+-----+
```

Manifest

```
+-----+
```

General:

Server: candlepin

Server Version: 1.3

Date Created: 13 April 2013

Creator: admin

Consumer:

Name: server.example.com

UUID:

Content Access Mode: entitlement

Type: system

Subscriptions:

Name: Red Hat Enterprise Linux

Quantity: 249237

Created: 12/01/2011

Start Date: 01/01/2012

End Date: 01/01/2022

Service Level: Premium

Service Type: Physical

Architectures: x86,x86_64

SKU: SYS0395

Contract: 12345678

Order: 09876543

Account: abcd1234

Entitlement File: /etc/pki/entitlement/2027912482659389239.pem

Certificate File: /etc/pki/product/69.pem

Certificate Version: 3

THE DUMP-MANIFEST COMMAND

A subscription management service is allocated a specific bloc of subscriptions that are available to an account. This list of subscriptions is the manifest for the service. The cat-manifest command prints the contents of the manifest.

OPTIONS

/path/to/consumer_export.zip

Gives the path and filename (by default, consumer_export.zip) for the manifest file on the local system. This is required.

--destination=PATH

Specifies an export directory to which to extract and save the contents of the manifest archive. If no destination is given, then the archive is extracted to the local directory.

--force, -f

Overwrites any existing archive files. If a manifest archive already exists in the specified location (for example, if the manifest has already been dumped once), then attempting to dump the manifest to the same location will fail. Using the --force option forces the dump operation to complete and overwrites the previous file.

EXAMPLES

This command simply extracts the manifest files to a given location (the working directory by default). The manifest itself contains multiple JSON files, with separate JSON files providing details on the manifest itself, each individual product, each individual subscription, and details for the specific, on-premise subscription management service.

For example:

```
[root@server ~]# rct dump-manifest --destination /export/archives/sam/manifest /tmp/consumer_export.zip
```

The manifest has been dumped to the /export/archives/sam/manifest directory.

FILES

* Product certificates: /etc/pki/product/*.pem

* Subscription certificates: etc/pki/entitlement/<serial#>.pem

* System identity certificates: /etc/pki/consumer/cert.pem

* The manifest: consumer_export.zip

BUGS

This tool is part of Red Hat Subscription Manager. To file bugs against this command-line tool, go to <<https://bugzilla.redhat.com>>, and select Red Hat > Red Hat Enterprise Linux > subscription-manager.

AUTHORS

Deon Lackey <dlackey@redhat.com>, Michael Stead <mstead@redhat.com>, and James Bowes <jbowes@redhat.com>. The rct tool was written by James Bowes.

COPYRIGHT

Copyright (c) 2012 Red Hat, Inc. This is licensed under the GNU General Public License, version 2 (GPLv2). A copy of this license is available at <http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt>.

RCT(8)