

Data Collection Guideline for vMRF

Virtual Multimedia Resource Function

Operating Instructions

Copyright

© Ericsson AB 2016, 2017. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.



Contents

1	Introduction	1
2	Collect Data for vMRF Troubleshooting	2
3	Overview of Data Collection in vMRF	3





1 Introduction

The purpose of this document is to instruct how to collect troubleshooting data that is to be enclosed in a Customer Service Request (CSR) in case a problem is experienced with the Virtual Multimedia Resource Function (vMRF).



2 Collect Data for vMRF Troubleshooting

This section describes how to collect all troubleshooting data with the `collectData.py` script. Other possible options for the script are described in [Overview of Data Collection in vMRF](#) on page 3.

Prerequisites:

- An incident has happened that requires opening a CSR.
- There is enough free disk space available in the `/tmp` and `cluster/storage` directories, that is, more than 500 MB free disk space, or more than 50% of total disk space capacity of the disk partition.

Steps

To collect troubleshooting data, do the following:

1. Open an SSH connection to the Operation And Maintenance (O&M) IP address of the Virtual Multimedia Resource Function (vMRF) Virtualized Network Function (VNF) instance using the following command:

```
ssh <user ID>@<O&M IP address>
```

Note: This command opens an SSH connection to the active SC VM, where data collection is performed. It is not recommended to log into a PL VM to collect troubleshooting data.

2. Run the following command:

```
collectData.py -a
```

Note: It is recommended to run `collectData.py` with the `-a` option to collect all available troubleshooting data. The reason is that the script runs for a reasonable amount of time even with all options enabled, and collecting all information maximizes the help given for troubleshooters.

`dcgm` and `getinfo.sh` are aliases to `collectData.py -a` and running them provides the same result.

3. Attach the resulting `.tar.gz` file to the CSR. The `.tar.gz` file is located in `/cluster/storage/collectdata/<date_time>/.tar.gz`.



3 Overview of Data Collection in vMRF

To collect troubleshooting data, the `collectData.py` script is used. The script collects troubleshooting information as specified by the options used when running the script.

Each command execution, except for options `-h` and `-v`, generates a `.tar.gz` file containing the requested information from the VNF in the following output format:

```
<hostname>_<machine_timestamp>_collectdata.tar.gz
```

The collected data can be analyzed after unpacking the `.tar.gz` file located in `/cluster/storage/collectdata/<date_time>/` by performing the following steps:

Steps

1. `gunzip`
`<hostname>_<machine_timestamp>_collectdata.tar.gz`
2. `tar xvf <hostname>_<machine_timestamp>_collectdata.tar`

The script accepts any of the following options, and it is also possible to combine multiple options (for example, `collectData.py -bcdl`):

- | | |
|-----------------------|--|
| -h, --help | This option shows the help message without collecting any data. It also lists the commands run by other options. For further information on Command Line Interface (CLI) tool commands, refer to <i>CLI Commands</i> . |
| -a, --all | This option collects all of the troubleshooting information, that is, binaries, Performance Management (PM) data, system logs, and the result of CLI tool and system commands. A detailed description of data collected is provided in the description of options below. |
| -b, --binaries | This option collects all the binaries under the <code>opt/mrf_agent</code> , <code>opt/mrf_director</code> , and <code>opt/ip_pipeline</code> directories needed, for example, for debugging crash dumps. |
| -c, --commands | This option automatically runs a set of CLI tool and system commands and collects data in the <code>commands.log</code> file in each VM. The list of commands run is printed out when using this option. |



Option `-c` collects the following data:

- Information related to counters for traffic performance
- Information related to error, discard, and debug counters
- Information on active contexts
- A summary of the vMRF signaling state, H.248 interface association state and SCTP operational state
- IP configuration
- Internal configuration of and load on IPP
- Node status check information

-d, --data

This option collects the following information:

- Output of measurement jobs, that is, data stored in `.rop` files
- Alarm and alert logs
- Network configuration data
- Dump files

-l, --logs

This option collects system logs from the `journalctl` command.

-n, --node

This option collects troubleshooting information only in a vMRF VNF specified by its IP address. The option must be combined with at least one data-collecting option as shown below:

```
collectData.py -n <VNF_IP address> -  
<option(s)>.
```

For example, `collectData.py -n <VNF_IP address> -lb` collects system logs and binaries on the specified vMRF VNF.

-o, --output

This option specifies the output directory.

-s, --show-results

This option prints the summary of data collection to the screen. The option must be combined with other data collection options.

**-v, --verbose**

This option prints the output of several CLI tool and system commands, that is, no `.tar.gz` file is generated. If combined with other options, `-v` ignores them, and the script runs as if `-v` was the only option.

The option collects the following data:

- Information related to counters
- Information on active contexts
- A summary of the vMRF signaling state, H.248 interface association state and SCTP operational state
- Internal configuration of and load on IPP
- Node status check information