Micro Focus Operations Bridge Upgrade Version Check Tool

Overview of the utilities being made available to aid in identifying the installed components and configuration of Operations Bridge Manager

Date: 9th December 2020

Author: Andy Doran (andy.doran@microfocus.com)

Revision: 2020.12.09

Location: <https://marketplace.microfocus.com/itom/content/opsb-version-check-tools>

Support for: Operations Bridge Manager Classic 10.12, 10.60, 10.61, 10.62, 10.63, 10.70, 2018.05,

2018.11, 2019.02, 2019.05, 2019.11, 2020.05, 2020.10

This tool will provide you with valuable information for your upgrade planning to classic OBM 2020.10. OBM 2020.10 provides a Flash-independent user interface for Operational and administration tasks. Running a Flash-independent version of OBM by end of calendar year 2020 is very important if your company is following the Adobe Flash-Player removal initiative. Otherwise, please make sure that you can still run Adobe Flash-Player in your browsers beyond end of 2020.

In case you need more information about the Micro Focus Operations Bridge Evolution program, please contact your Micro Focus Support liaison, or send an email to:

[OpsBEvolution@MicroFocus.com](mailto:OpsBEvolution@MicroFocus.com)

Contents

[Background 3](#_Toc57816989)

[Overview of the Utility 4](#_Toc57816990)

[Current Checks 4](#_Toc57816991)

[Requirements 5](#_Toc57816992)

[Using the Utility 6](#_Toc57816993)

[Using opr-version-check.pl 7](#_Toc57816994)

[Examples 8](#_Toc57816995)

[All online checks 8](#_Toc57816996)

[OBM Version checks 8](#_Toc57816997)

[Management Pack Checks 9](#_Toc57816998)

[Indicator Mapping Rules 10](#_Toc57816999)

[UCMDB Content Packs Check 11](#_Toc57817000)

[Preparing Offline files 11](#_Toc57817001)

[Using “opr-checker” 11](#_Toc57817002)

[Using “opr-support-utils” 12](#_Toc57817003)

[Obtaining Management Pack Information 13](#_Toc57817004)

[APM Check Utility 14](#_Toc57817005)

[Examples 15](#_Toc57817006)

[Running on a Gateway Server on Windows 15](#_Toc57817007)

[Running on a Gateway Server on Windows 15](#_Toc57817008)

[Running on Windows system (not a Gateway server) 15](#_Toc57817009)

[Using the \_.netrc File 16](#_Toc57817010)

[OBR Check Utility 17](#_Toc57817011)

# Background

Micro Focus Operations Bridge (OpsBridge) is a suite of products that monitors IT environments and consolidates data from other monitoring tools, since Operation Bridge Manager (OBM/OMi) often used as single plane of glass in large IT organizations. Due to the nature of the OpsBridge suite and the potential huge list of integrated products, upgrade planning and execution could become tricky – e.g. understanding what components are installed, what versions and the overall configuration.

The Upgrade Version Check Tool (opr-version-check) is an evolving utility that aims to simplify the process of identifying what has been installed as an aid to the upgrade process, as well as highlighting where components may be out of date. The tool is provided by the ITOM Customer Success team with “community support” via ITOM Marketplace.

There are several Perl scripts provided that can be executed on either a Linux or Windows platform. Some of these need to run “live” (“online”), but others can also run “offline” if the information that they process has been gathered previously. A single driver script (opr-version-check.pl) is used to invoke each additional script as required.

# Overview of the Utility

## Current Checks

The Upgrade Version Check Tool currently collects/reports on the following information:

|  |  |  |
| --- | --- | --- |
| **Area/Category** | **Explanation** | **Offline?** |
| **OBM/OMi Version** | The installed OBM/OMi version and patches installed. If the version found is not the currently available version, this is highlighted. | yes |
| **O/S Information** | The O/S and version installed where the component (i.e. Gateway Server or Data Processing Server) | yes |
| **Server Role** | The role(s) of the server (i.e. Gateway server) | yes |
| **License Information** | The installed licenses – where a license is expired or fully consumed, this is highlighted. | yes |
| **Database Server** | The database server type (i.e. PostgreSQL, Oracle or MS SQL Server) and host along with connection information. | yes |
| **Management Packs** | The installed Management Packs along with their versions (this will include a check to see if they MP is the current version) | yes |
| **Database Server Detail** | The Database Server type and Version | no |
| **Count of Gateway Servers** | The number of Gateway Servers configured | no |
| **Data Processing Servers** | The number of Data Processing Servers configured | no |
| **Dual Role Servers** | Notification of the number of servers with both GW and DPS roles | no |
| **Connected Servers** | Information about the connected serves configured | no |
| **Users** | Count of OBM users – count of those which are “SuperAdmin” | no |
| **Connectors** | Information about installed connectors and the agents that they are running on | no |

Where “Offline” is supported, this means that the information can be gathered using the utilities available on the Gateway and Data Processing Servers (DPS). These can be used to generate files which can be processed by the Upgrade Version Check Tool. If a category is not marked as being available offline, then it can only be processed by running on a Gateway or DPS server.

**Note**: For the online database checks to be made, Java needs to be available on the system that is running the checker tool. If Java is not available, the database checks will not be made.

## Requirements

The scripts are all written in Perl. On Linux, this means that the Perl module must be installed. This is a pre-requisite for installing the OBM server components, and so should be present. To install the Perl module if required, refer to the documentation for the Linux platform being used. For example, on RedHat:

yum install perl -y

On Windows, Perl will need to be installed unless the OpsBridge Operations Agent is installed (the OA provides a Perl engine and if this is detected can be used by the scripts). If the scripts are to be executed from a Windows system without an OA present (for example when running to process offline files), then a Perl engine such as ActivePerl must be installed. This is available from:

<https://www.activestate.com/products/perl/downloads/>

If the OA Perl engine is to be used, a batch file (oaper.bat) is provided which will locate the Perl engine. Usage is described in the next section.

# Using the Utility

The Upgrade Version Check Tool is provided as a zip file which should be unzipped either to Windows or Linux. Once unzipped, the script that drives the process is called “opr-version-check.pl”. To execute this script on Linux, simply use:

./opr-version-check.pl <options>

Assuming it is in the current directory – specify the path to the file if it is not. To execute the script on Windows, either use:

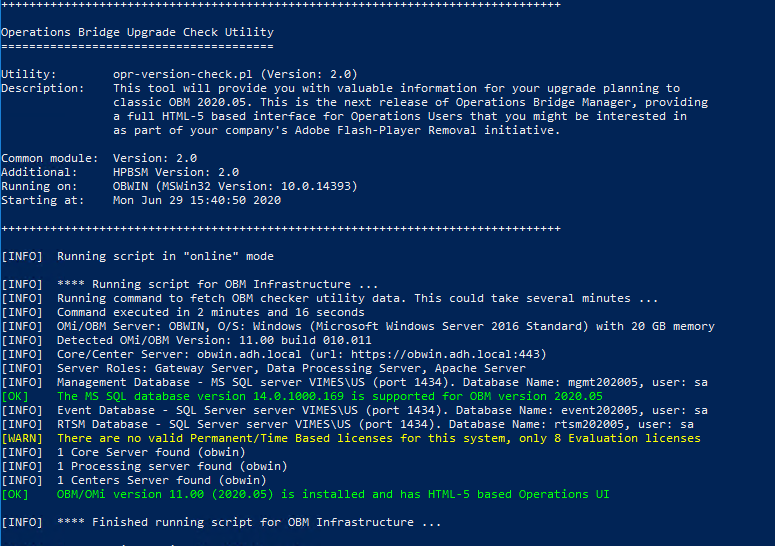
perl opr-version-check.pl <options>

Or

oaperl.bat opr-version-check.pl <options>

Again, this assumes the script is in the current directory – specify the path if it is not. The first example above assumes that ActivePerl (or another engine) has been installed. The second example assumes that the OA Perl engine will be used. The batch file “oaperl.bat” locates the OA Perl engine and invokes the script, passing the options to it.

Example:



## Using opr-version-check.pl

The Upgrade Version Check Tool supports two basic modes of execution – online or offline. By default, it will run in online mode which means that it expects at least the OBM admin account password. When running in offline mode, the script needs to be provided with the files to process which must be generated using the OBM utilities (described later).

When running in online mode, the Upgrade Version Check Tool must execute on either a Gateway or DPS server to access the OBM utilities that provide the information being processed.

The list of supported input parameters is as follows (and can be specified in any order):

|  |  |  |
| --- | --- | --- |
| **Option** | **Explanation** | **Example** |
| **-offline** | This is the default. If selected, then at least one input file is required | -offline |
| **-online** | If this switch is used, then the script must be executing on a Gateway or DPS server. Some checks require credentials with this switch specified | -online |
| **-obm-input <file>** | When using offline mode, this is the input file for the infrastructure checks (such as OBM version and patch level) | -obm-input /tmp/opr-server.txt |
| **-mp-input <file>** | When using offline mode, this is the input file that contains Management Pack information | -mp-input /tmp/mp-data.txt |
| **-user <user>** | When using online mode, this overrides the username assumed by default (admin) for logging on to OBM | -user operator |
| **-pwd <pass>** | The password for the OBM user | -pwd P@ssw0rd |
| **-dbuser <user>** | When using online mode, this overrides the login user for the backend database. This should not be required as the login user is determined during the OBM analysis | -dbuser sa |
| **-dbpwd <pass>** | The password for the OBM database user | -dbpwd P@ssw0rd |

If the db-user parameter is not supplied, then the database user specified in the OBM configuration will be used. The db-user parameter is used to override that setting. If the -db-pwd parameter is NOT supplied, then the password specified via the -pwd parameter will be used for the account that connects to the database server.

In addition, the following switches can be used:

|  |  |  |
| --- | --- | --- |
| **Option** | **Explanation** | **Example** |
| **-nocolor** | The output uses colors. For example, “OK” messages are in GREEN whilst failure messages are in RED. This switch disables the use of colors if required | -nocolor |
| **-log** | This switch enables logging. All output sent to the screen is also captured in a log file. A log is created for each utility, will be in the current directory and the name is given as each utility completes | -log |
| **-timeout <seconds>** | The default timeout for commands issued by the scripts is 5 minutes. This can be overridden using the -timeout switch to either lengthen or shorten that time period. This only applies when the scripts run on Linux | -timeout 120 |
| **-help** | Prints help text. If this is provided, then ONLY the help is shown – the script will immediately exit | -help |

In addition to the above parameters, there are some “advanced” parameters that can be used to override the default behaviour. Under normal circumstances these should not be used.

The first is the parameter “-dbport” – this can be used to override the port that is used to connect to the database server. Normally tis is obtained by the script when reading the configuration files, but in some cases, this may need to be changed dynamically.

The other two are switches that can be used with “offline” mode. The switch “-forcedb” will enable the database checks when the database server is found in the input file specified using the “-obm-input” parameter. If this switch is used, then the database password is also required. The switch “-forcetls” is similar – if an LDAP configuration is found then there will be a test for the LDAP server TLS version. This is nor,ally disabled for the “offline” mode, but this switch will enable it.

## Examples

### All online checks

Running all online checks (this requires the script being executed on a Gateway or DPS Server). This results in the information shown in the screenshot at the beginning of this section.: -

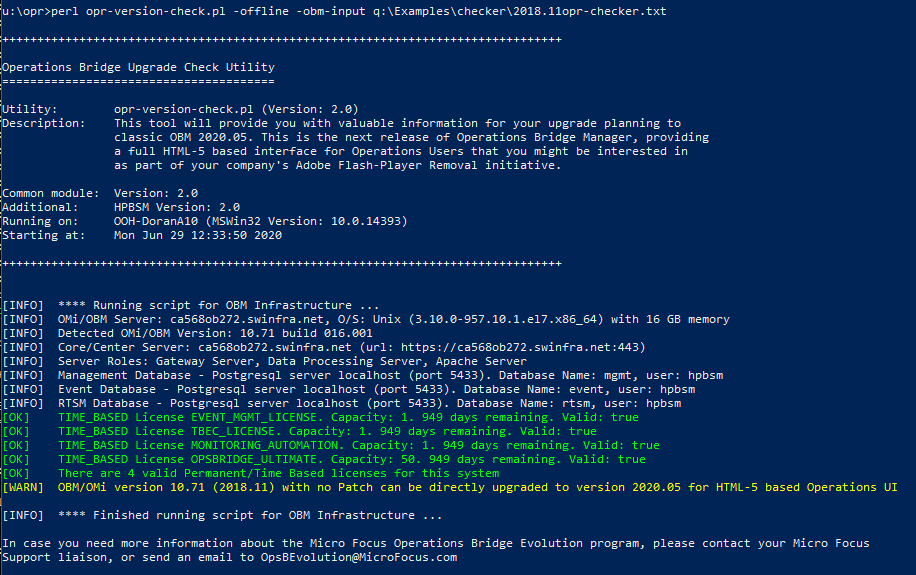
./opr-version-check.pl -online -pwd P@ssw0rd1 -db-pwd P@ssw0rd2

### OBM Version checks

Running offline to check the OBM version and upgrade recommendations:

./opr-version-check.pl -offline -obm-input /tmp/opr-server.txt

This results in information similar to:



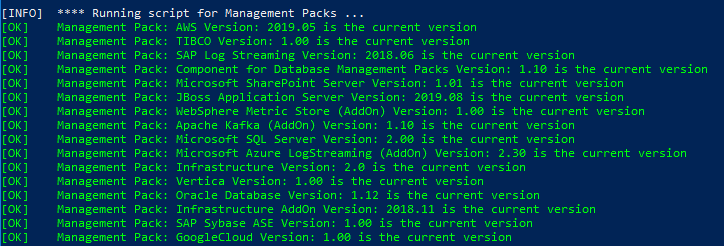
In this example, the input file has been analysed and it has been determined that the server which provided the file is running an older version of OBM which can be directly upgraded to the current release.

### Management Pack Checks

The online check will show the installed Management Packs and indicate whether they should be upgraded. The offline check will only include this if the -mp-input argument is used, specifying a file generated using the built-in tools (described later). To include this input file for the offline checks:

./opr-version-check.pl -offline -obm-input /tmp/opr-checker.txt -mp-input /tmp/mp.txt

This generate output like:



System Checks

The utility will read information form the Management Database (normally mgmt) to determine whether the DPS and Gateway servers meet the current recommendations for small/medium/large environments. Warnings will be given if this is not the case.

This check is only performed using the “online” mode.

### Indicator Mapping Rules

If Custom Indicator Mapping Rules are defined, it is still necessary to use a Flash based UI in order to make changes (this will be rectified in a future update of OBM). The utility will check whether any such rules have been defined and generate an informational “NOTE” message if any are found:



It is possible to list these Rules by name if that information is useful. To do that, the “Event” database information will be required. This is listed just before the message relating to making the checks:

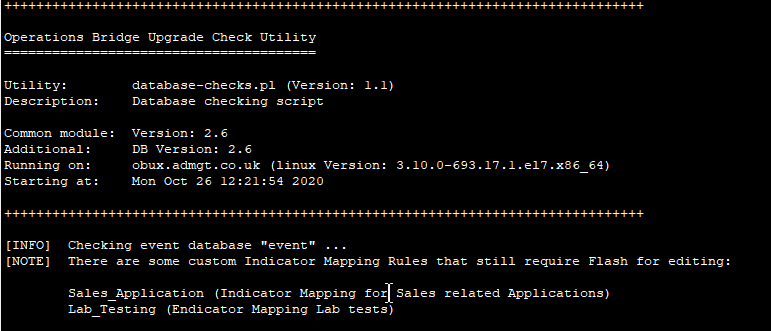


In this example, the event database is on the Postgres server on the localhost, available on port 5433 and accessed using the hpbsm user. In the case of an Oracle server, the SID will also be displayed.

In order to show the Indicator mapping rules by name, the “database-checks.pl” script can be called directly using the above information:-

./database-checks.pl -server localhost -dbevent event -user hpbsm -pwd P@ssw0rd -dbtype postgres -port 5433 -etidetail

The results would then be like:



It is important to specify the database type using the -dbtyoe switch. The valid types are

Postgres

Oracle

MSSQL

If the type is not specified, then you will be prompted to provide it. If the database type is Oracle, the SID must be provided using the -SID parameter (the SID will be shown with the other details when the opr-version-check.pl script runs).

This check is only made when the online checks are made.

### UCMDB Content Packs Check

An additional check is now included for the UCMDB database (normally named “rtsm”). When new Content Packs are installed, the old ones are not unregistered. This can consume quite a lot of space in the database – and on disk. The check will highlight if more tnan one UCMDB content pack is found.

Micro Focus has a blog entry that describes the steps that can be take to groom this information:

<https://community.microfocus.com/t5/CMS-Discovery-CMDB-User/Maintenance-tip-cleanup-of-the-old-Content-Packs/m-p/2776901#M20958>

## Preparing Offline files

When running in offline mode, the input files must have been generated previously. The “loggrabber” utility can be used to generate some of these files, but if a login is required – loggrabber cannot process the information. This section gives some examples in generating these files.

The examples are based on Linux but can be executed on Windows. The default location for the utilities on Windows is “C:\HPBSM\opr\support” and “C:\HPBSM\opr\bin”, although this can be changed if OBM is installed at a different location.

On Windows, the file extensions for all utilities is “.bat”

### Using “opr-checker”

The file used when the argument “-obm-input” is specified can be generated using the “opr-checker” tool with the switches “-sys -opr -rapid” or “-all -rapid”. For example:

/opt/HP/BSM/opr/support/opr-checker.pl -sys -opr -security -rapid > /tmp/opr-checker.txt

This file can then be copied to a location where the Upgrade Version Check Tool will be executed. Note that the switch “-all” can be used instead of “-sys -opr -security”.

In some cases, not all information is provided using the opr-checker tool. When this happens, the additional information can be obtained using the “opr-support-utils” tool and by including the file “TopazInfra.ini” (located in the “c:\HPBSM\conf” on Windows, “/opt/HP/BSM/conf” on Linux). These additional files should be concatenated (merged) with the output from the opr-checker tool (see later for examples).

Use if the opr-support-utils tool is described in the next section.

To determine whether the file generated by opr-checker has all the information required, use the following:

cat /tmp/opr-checker.txt | grep -I TopazInfra

cat /tmp/opr-checker.txt | grep -I opr.db.connection.dbname

cat /tmp/opr-checker.txt | grep -I odb.db.connection.dbname

or on Windows

c:\> type c:\temp\opr-checker.txt | findstr -I TopazInfra

c:\> type c:\temp\opr-checker.txt | findstr -I opr.db.connection.dbname

c:\> type c:\temp\opr-checker.txt | findstr -I odb.db.connection.dbname

using the file generated using the utility. If the check for “TopazInfra” returns no results, then the TopazInfra.ini file is required. If the checks for the “dbname” entries return no results, then the output from opr-support-utils is required (note that the check for odb.dn.connection.dbname is looking for information relating to the UCMDB – so if that has not been configured then it is expected to return no results).

### Using “opr-support-utils”

This tool can be used to retrieve information relating to the event and UCMDB databases if this information was not picked up by the opr-checker utility. If you do not require this information (or do not have the UCMDB database configured), then these steps can be skipped.

To generate a file that contains information for the event database:

/opt/HP/BSM/opr/support/opr-support-utils.sh -list\_settings -context opr > /tmp/opr.txt

To generate a file that contains information for the UCMDB (rtsm) database:

/opt/HP/BSM/opr/support/opr-support-utils.sh -list\_settings -context odb > /tmp/odb.txt

These files can be combined with the output of the opr-checker utility and TopazInfra.ini (if required) to generate a single input file for the version check util. See below for examples on doing this.

### Obtaining Management Pack Information

The Management Pack information is obtained by using the utility “ContentManager”. This will require user credentials:

/opt/HP/BSM/bin/ContentManager.sh -user admin -pw P@ssw0rd -l -verbose > /tmp/mp.txt

This file can then be provided as a separate input file using the -mp-input parameter for the upgrade checker tool to run in offline mode.

Combining multiple files into one

Running the upgrade checker too in offline mode expects a single file generated by the “opr-checker” utility to be an input file. In some circumstance as described earlier, the “opr-checker” tool will not obtain all information and so a second utility is used to provide that information (along with the TopazInfra.ini file”. These files need to be combined into a single file to use them with the upgrade check tool.

On Linux, this can be done as follows:

cat /tmp/opr-checker.txt /tmp/opr.txt /tmp/odb.txt /opt/HP/BSM/conf/TopazInfra.ini > /tmp/all-data.txt

This will result in a single file “all-data.txt” that can be used as the input file. On Windows, the equivalent would be:

C:> copy c:\temp\opr-checker.txt + c:\temp\opr.txt c:\temp\odb.txt + c:\HPBSM\conf\TopazInfra.ini > c:\temp\all-data.txt

# APM Check Utility

A separate utility is provided for use with APM. It uses the same support libraries that the other scripts use but is not driven using the “opr-version-check” script. Instead, the run-apm-checks.pl script must be used directly.

As with the other script utilities, run-apm-checks.pl can be executed from either a Windows system or a Linux system. There are some restrictions on the use of the script, depending on where it is executed from.

The main restriction is that the checks for information held in the APM management database can currently only run from a Windows platform. If the APM Gateway server is a Linux server, this means that the script may need to run on the Gateway server and then run again from a Windows system to connect to the database server and get all of the information.

Off-line mode is not supported by this script.

The list of supported input arguments and switches is in the table below, and can be passed in any order:

|  |  |  |
| --- | --- | --- |
| **Option** | **Explanation** | **Example** |
| **-host <host>** | The APM Gateway host (were the jmx server is running) | -host myserver |
| **-user <user>** | The APM username. If not provided, “admin” is assumed | -user operator |
| **-pwd <password>** | Password for the APM user | -pwd P@ssw0rd |
| **-netrc** | Use the \_.netrc file for credentials (see below) | -netrc |
| **-nojmx** | Do not perform any jmx checks | -nojemx |
| **-dbhost <host>** | The SQL or Oracle server hosting the APM management database (if using SQL then any instance should be specified – host\instance) | -host server1 |
| **-db <database>** | The APM Management database name | -db mgmt. |
| **-dbuser <user>** | The login ID with access to the APM database | -dbuser sa |
| **-dbpwd <password>** | The password for the login ID | -dbpwd P@ssw0rd |
| **-dbport <port>** | The port used to connect to the database server (if not specified, this defaults to 1433 for a SQL Server or 1521 for an Oracle Server) | -dbport 1432 |
| **-trusted** | If using SQL Server, specifying this switch means that the current Windows account will be used to connect | -trusted |
| **-dbsid <SID>** | For an Oracle connection ,the SID to use | -sid orcl |
| **-dbtype <type>** | APM databases are supported on MS SQL or Oracle. The options here are “MS SQL” or Oracle (“MS SQL” must be specified in quotes). The default is “MS SQL” | -dbtype “MS SQL” |

In addition, the following switches can be used:

|  |  |  |
| --- | --- | --- |
| **Option** | **Explanation** | **Example** |
| **-nocolor** | The output uses colors. For example “OK” messages are in GREEN whilst failure messages are in RED. This switch disables the use of colors if required | -nocolor |
| **-verbose** | Include additional informational messages | -verbose |
| **-log** | This switch enables logging. All output sent to the screen is also captured in a log file. A log is created for each utility, will be located in the current directory and the name is given as each utility completes | -log |
| **-help** | Prints help text. If this is provided then ONLY the help is shown – the script will immediately exit | -help |

## Examples

Below are some examples of running the script. On Windows, perl is required and either needs to be installed separately (for example ActivePerl) or if the Operations Agent has been installed – the oaperl.bat file can be used to execute the script. An OA agent is NOT installed by default on an APM server, so this would have to be done as a separate exercise.

### Running on a Gateway Server on Windows

C:\> perl c:\scripts\run-apm-checks.pl -pwd P@ssw0rd -dbpwd P@ssw0rd2

Because the script is running on the APM Gateway server, only the passwords will be required to access the JMX pages and backend database. The information about the connections to the JMX pages, and the database server will be picked up from the APM configuration

### Running on a Gateway Server on Windows

./run-apm-checks.pl -pwd P@ssw0rd

Because the script is running on the APM Gateway server, only the APM password is required. The database server cannot be checked from Linux, but the information on the server (type of Database, host name etc.) will be displayed.

### Running on Windows system (not a Gateway server)

C:\> perl c:\scripts\run-apm-checks.pl -host myhost -pwd P@ssw0rd -dbhost mydbserver -dbuser admin -dbpwd P@ssw0rd2 -sid orcl -dbtype Oracle

In this case, all information relating to the Gateway host (where the JMX pages are) and the database server are required. With this information, connections will then be made to those servers to retrieve the configuration data.

By default, access to the JMX pages is restricted to the localhost only. In order to access them remotely – configuration changes (changes to infrastructure) are required. See the APM documentation for information on how to do this, and the implications of doing so.

### Using the \_.netrc File

Using a \_.netrc file is a way to avoid passing credentials via the command line when accessing a url. This can be used with the script (but only for the JMX processing). For more information om configuring this file, see the information at:

<https://community.apigee.com/articles/39911/do-you-use-curl-stop-using-u-please-use-curl-n-and.html>

Note that on Windows, the file is expected to be in the %USERPROFILE% directory.

# OBR Check Utility

A separate utility is provided for use with OBR. It uses the same support libraries that the other scripts use but is not driven using the “opr-version-check” script. Instead, the run-obr-checks.pl script must be used directly.

As with the other script utilities, run-obr-checks.pl can be executed from either a Windows system or a Linux system. The script supports both “online” and “offline” mode – running “online” means that the script must be executed on an OBR server. To run in “offline” mode, the OBR capture tool must be executed manually.

The OBR capture tool is documented here:

<https://docs.microfocus.com/itom/Operations_Bridge_Reporter:10.40/Troubleshoot/Troubleshooting_SHR/Capture>

This must be installed and configured before the script can be used and must be executed manually to use the script in “offline” mode. When running manually, a zip file will be created (the location will be given as the capture tool runs). This zip file can then be used as the input to the script.

Supported parameters and switches:

|  |  |  |
| --- | --- | --- |
| **Option** | **Explanation** | **Example** |
| **-zipfile <filename>** | This argument is used to specify the filename of the zip file generated by manually running the OBR Capture tool. If this is not provided then the script will run the tool (meaning the script must run on the OBR server) | -zipfile /tmp/obr.zip |

Sample output (online mode):

