

CAST Report Generator - CAST Report Generator for Dashboards

- [Introduction](#)
- [Where can I obtain the CAST Report Generator for Dashboards?](#)
 - [1.10.0](#)
 - [1.9.0](#)
- [Limitations](#)
- [Installation prerequisites](#)
 - [.NET Core](#)
 - [Supported Operating Systems](#)
 - [CAST RestAPI](#)
 - [Folder access permissions](#)
- [Installation and configuration process](#)
 - [Step 1 - Deploy files](#)
 - [Step 2 - Configure for Standard Compliance Report Generation in CAST Dashboards](#)
 - [Edit .properties file](#)
 - [Restart the application](#)
- [Using CAST Report Generator for Dashboards in CLI mode](#)
 - [Using an API Key instead of a password](#)
 - [Log files](#)
 - [Return codes](#)



Summary: this page describes the **CAST Report Generator for Dashboards** release.

Introduction

CAST Report Generator is now provided as an **cross-platform ZIP file** alongside the traditional Microsoft Windows GUI "installable" release. In other words this release can run on both **Microsoft Windows** and **Linux** operating systems and can be simply unzipped to the install location (no installer is provided). The tool has two main uses:

- the **primary goal of this tool** is to provide reporting capabilities to the **CAST RestAPI** - this integration is first and foremost used by the CAST dashboards to generate **Standard Compliance reports** from within the dashboard interface (see [Engineering Dashboard - Report Generation](#) and [Security Dashboard - Report Generation](#)).
- The tool can also be used in **CLI mode** to generate reports, if required.

Where can I obtain the CAST Report Generator for Dashboards?

1.10.0

You can obtain the Report Generator for Dashboards from [CAST Extend](#):



ReportGenerator for Dashboards

Cast Product 1.10.0-funcrel

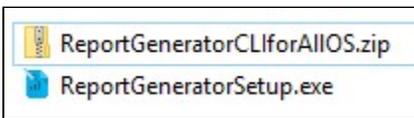
The CAST Report Generator for Dashboard is a standalone service, that will be used by CAST Dashboards, for automatic generation of reports based on data generated by CAST AIP. The solution provides the opportunity (for example) to

Last published 24/06/2019 5 Downloads

A ZIP file will be downloaded. Extract the ZIP and deploy the files as described below in [Installation process](#).

1.9.0

You can obtain the CAST Report Generator for Dashboards from the [CAST Product Center](#) - this requires a CAST login. Download the latest release, unpack the ZIP, and you will find another ZIP called **ReportGeneratorCLIforAllIOS.zip** alongside the traditional GUI based ReportGeneratorSetup.exe:



Extract the ZIP called **ReportGeneratorCLIforAllIOS.zip** and deploy the files as described below in [Installation process](#).

Limitations

- The tool has no GUI - it is designed to be used **primarily** when generating reports from within the CAST Dashboard interface (see [Engineering Dashboard - Report Generation](#) and [Security Dashboard - Report Generation](#)) and also in **CLI mode**.
- No PDF export capabilities in CLI mode.

Installation prerequisites

To install and use the CAST Report Generator for Dashboards, the following is required:

.NET Core

CAST Report Generator for Dashboards requires the **.NET Core runtime/SDK** as follows:

Report Generator release	Minimum .NET Core	Download	Notes
1.14.0	3.1.x SDK	https://dotnet.microsoft.com/download/dotnet-core/3.1	A change to the SDK rather than the runtime to match the requirements for Report Generator GUI.
1.13.x	3.0.x runtime	https://dotnet.microsoft.com/download/dotnet-core/3.0	Future releases of CAST Report Generator will require .NET Core 3.1.x due to the end-of-life of .NET Core 3.0.x scheduled for 3rd March 2020 , therefore CAST recommends installing .NET Core 3.1.x to avoid the need to re-install .NET Core in the future.
1.9.0 - 1.12.0	2.2.x runtime	https://dotnet.microsoft.com/download/dotnet-core/2.2	-

This runtime/SDK is available for both Microsoft Windows and for Linux operating systems depending on the Operating System used to host Apache Tomcat:

Microsoft Windows

.NET Core can be downloaded as an installer - choose the **.NET Core Installer runtime/SDK** that matches your Windows environment:

Release information	Build apps - SDK 	
v3.1.2 Release notes Released 2020-02-18	SDK 3.1.102 Visual Studio support Visual Studio 2019 (v16.4) Included runtimes .NET Core Runtime 3.1.2 ASP.NET Core Runtime 3.1.2 Desktop Runtime 3.1.2 Language support C# 8.0 F# 4.7	
OS	Installers	Binaries
Linux	Package manager instructions	ARM32 ARM64 x64 Alpine x64 RHEL 6 x64
macOS	x64	x64
Windows	x64 x86	ARM32 x64 x86
All	dotnet-install scripts	

[Localized IntelliSense](#)

Linux

Package manager instructions are provided for many different Linux distributions - choose the **Package Manager Instructions** option:

Release information	Build apps - SDK															
v3.1.2 Release notes Released 2020-02-18	SDK 3.1.102 Visual Studio support Visual Studio 2019 (v16.4) Included runtimes .NET Core Runtime 3.1.2 ASP.NET Core Runtime 3.1.2 Desktop Runtime 3.1.2 Language support C# 8.0 F# 4.7															
	<table border="1"><thead><tr><th>OS</th><th>Installers</th><th>Binaries</th></tr></thead><tbody><tr><td>Linux</td><td>Package manager instructions</td><td>ARM32 ARM64 x64 Alpine x64 RHEL 6 x64</td></tr><tr><td>macOS</td><td>x64</td><td>x64</td></tr><tr><td>Windows</td><td>x64 x86</td><td>ARM32 x64 x86</td></tr><tr><td>All</td><td>dotnet-install scripts</td><td></td></tr></tbody></table>	OS	Installers	Binaries	Linux	Package manager instructions	ARM32 ARM64 x64 Alpine x64 RHEL 6 x64	macOS	x64	x64	Windows	x64 x86	ARM32 x64 x86	All	dotnet-install scripts	
OS	Installers	Binaries														
Linux	Package manager instructions	ARM32 ARM64 x64 Alpine x64 RHEL 6 x64														
macOS	x64	x64														
Windows	x64 x86	ARM32 x64 x86														
All	dotnet-install scripts															
	Localized IntelliSense															

Then select the relevant Linux distribution to view the installation instructions:

Ubuntu 19.04 Package

04/12/2019 • 3 minutes to read • 🔍

- Ubuntu 19.04 - x64 ^
- Ubuntu 19.10 - x64
- Ubuntu 19.04 - x64
- Ubuntu 18.04 - x64
- Ubuntu 16.04 - x64
- CentOS 7 - x64
- Debian 10 - x64
- Debian 9 - x64



Additional tips can be found here: <https://www.ostechnix.com/how-to-install-microsoft-net-core-sdk-on-linux/>.

Supported Operating Systems

CAST Report Generator for Dashboards only requires that .NET Core runtime is present on the operating system. Therefore any operating system that supports the installation of **.NET Core** can be used for the CAST Report Generator for Dashboards. CAST has tested Microsoft Windows and CentOS (Linux).

CAST RestAPI

1.9.x



Folder access permissions

The user that Apache Tomcat is running as must have **read/write access** to the following locations:

- folder into which the Report Generator for Dashboards is unzipped
- "Logs" sub-folder.



In Linux environments, typically the "rw-" permission is sufficient.

Installation and configuration process

Step 1 - Deploy files

Installing CAST Report Generator for Dashboards consists only of unzipping the downloaded ZIP file to a location on the target server, for example:

Microsoft Windows	D:\data\CAST\ReportGeneratorforDashboards
Linux	/opt/reportgeneratorfordashboards

If you:

- need to use the tool to **generate Standard Compliance reports in the CAST Engineering or Security Dashboards**, you should deploy the tool on the **Apache Tomcat server which hosts your dashboards** and then complete [Step 2 below](#).
- simply want to use the tool via **CLI to generate reports** (see [below](#)) you can deploy the tool **on any server providing it has network access to a CAST RestAPI interface** (i.e. a dashboard). There is nothing further to do in this case.

Step 2 - Configure for Standard Compliance Report Generation in CAST Dashboards

If you need to use the tool to generate Standard Compliance reports in the CAST Engineering or Security Dashboards, a further configuration process is required:

Edit .properties file

Edit the following file with a text editor:

```
WAR 1.x
CATALINA_HOME\webapps\<>deployed_war>\WEB-INF\report.properties

WAR 2.x
CATALINA_HOME\webapps\<>deployed_war>\WEB-INF\classes\application.properties

ZIP 2.x
<unpacked_zip>\application.properties
```

Find the following options and modify as explained below:

```
# Set the Report Generator path
# If this variable is not set then the document generation is considered as disabled.
# The path is probably something such as (Linux/Windows):
#report.reportGenerator=dotnet /opt/report-generator/CastReporting.Console.Core.dll
#report.reportGenerator=dotnet D:\\data\\CAST\\ReportGeneratorforDashboards\\CastReporting.Console.Core.dll

# Set the directory of reports
#report.directory=/tmp/reports
#report.directory=D:\\temp\\reports

# Set the current Web Service URL. The current REST API called back by the Report Generator.
report.webServiceURL=http://localhost:8888/CAST-RESTAPI/rest
```

report. reportGenerator=dotnet	<p>Add a new line pointing to the location of the <code>CastReporting.Console.Core.dll</code> (delivered with CAST Report Generator - CAST Report Generator for Dashboards) on the server hosting Apache Tomcat. You can also uncomment and modify an example line. For example:</p> <pre>report.reportGenerator=dotnet c:\\ReportGeneratorforDashboards\\CastReporting.Console.Core.dll</pre> <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;">  Note that the path to <code>CastReporting.Console.Core.dll</code> when using Microsoft Windows must always use double back slashes (\\) or single forward slashes (/) - the single back slash (\) is not valid. </div>
report.directory=	<p>Add a new line pointing to the temporary folder where the reports will be generated on the server hosting Apache Tomcat. You can also uncomment and modify an example line. For example:</p> <pre>report.directory=c:\\temp\\reports</pre> <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;">  Note that: <ul style="list-style-type: none"> the path to the temporary folder when using Microsoft Windows, must always use double back slashes (\\) or single forward slashes (/) - the single back slash (\) is not valid. The user that Apache Tomcat is running as must have read/write access to this location. In Linux environments, typically the "rw-" permission is sufficient. </div>
report. webServiceURL=	<p>Modify the existing line to point to the RestAPI in your CAST Engineering Dashboard or CAST-RESTAPI-integrated. war deployment. This is used by CAST Report Generator for Dashboards. For example:</p> <pre>WAR 1.x / 2.x report.webServiceURL=http://<server>:<port>/<dashboard>/rest ZIP 2.x report.webServiceURL=http://<server>:<port>/rest</pre>

For example:

```
# Set the Report Generator path
# If this variable is not set then the document generation is considered as disabled.
# The path is probably something such as (Linux/Windows):
#report.reportGenerator=dotnet /opt/report-generator/CastReporting.Console.Core.dll
#report.reportGenerator=dotnet c:\\ReportGenerator\\CastReporting.Console.Core.dll

report.reportGenerator=dotnet c:\\ReportGeneratorforDashboards\\CastReporting.Console.Core.dll

# Set the directory of reports
#report.directory=/tmp/reports
#report.directory=c:\\temp\\reports

report.directory=c:\\temp\\reports

# Set the current Web Service URL. The current REST API called back by the Report Generator.
report.webServiceURL=http://localhost:8888/CAST-RESTAPI/rest
```

Restart the application

Restart the web application to ensure the changes are taken into account.

Using CAST Report Generator for Dashboards in CLI mode

As mentioned previously, the **primary goal** of this release is to provide reporting capabilities for the CAST Dashboards (Engineering and Security) - However, it is also possible to use CAST Report Generator for Dashboards yourself via its CLI. The CLI instructions are **identical** to those provided for the **standard CAST Report Generator GUI** - see [here](#) for more instructions.

You must run the CAST Report Generator for Dashboards in a specific way when used in CLI mode. The `CastReporting.Console.Core.dll` is provided with the tool:

```
dotnet %path_to_CastReporting.Console.Core.dll% -webservice %webservice% -username %username% -password %password% -application %application% -domain %domain% -template %template% -file %file%
```

For example:

```
dotnet "D:\data\CAST\ReportGeneratorCore\CastReporting.Console.Core.dll" -webservice "http://<server>:<port>/CAST-RESTAPI/rest/" -username cio -password cast -application "Jurassic Park" -domain "AED" -template "Executive-summary.docx" -file "D:\temp\Core_Report.docx"
```

Using an API Key instead of a password

 Available in v. **1.11.0**.

If you have enabled **SAML authentication mode** for your CAST Dashboard/RestAPI deployment, you will need to use an **API Key** instead of a password. To do, so you will need to modify the command line as follows:

- add the **-apikey true** option
- enter the API Key in the **-password** option

For example:

```
dotnet "D:\data\CAST\ReportGeneratorCore\CastReporting.Console.Core.dll" -webservice "http://<server>:<port>/CAST-RESTAPI/rest/" -username cio -password myapikey -apikey true -application "Jurassic Park" -domain "AED" -template "Executive-summary.docx" -file "D:\temp\Core_Report.docx"
```

Log files

The CAST Report Generator for Dashboards saves a log of all interactions in the folder into the **Logs folder** (located in the folder in to which you have unzipped the CAST Report Generator for Dashboards). One log file per day is generated:

```
ReportGenerator_<timestamp>.log
```

For example **ReportGenerator_20190328.log**.

Return codes

One of the following codes will be returned on completion of the CLI:

0	Report successfully generated
1	Bad arguments given to the Report Generator
2	Web service cannot be accessed or is badly configured
3	Report failed