

Automating CAST Transaction Configuration Center tasks

In this section:

- [Prerequisites](#)
- [How does it work?](#)
 - [Miscellaneous information](#)
- [Command arguments](#)
 - [-ShowHelp](#)
 - [-HideGUI](#)
 - [-ConnectProfile <profile-name>](#)
 - [-ImportTemplatesSetup <file-path>](#)
 - [-UpdateTemplatesSetup <file-path>](#)
 - [-ExportTemplatesSetup <file-path>](#)
 - [-ImportAppliSetup <file-path> <system-name>/<application-name>](#)
 - [-ExportAppliSetup <file-path> <system-name>/<application-name>](#)
 - [-SaveCalibrationChanges \(YES|NO|CANCEL\)](#)
 - [-RunSnapshotPreparation \(YES|NO|CANCEL\)](#)
 - [-ComputeAppliFP <system-name>/<application-name>](#)
 - [-ExportAppliEntryPoints <file-path> <system-name>/<application-name>](#)
 - [-CreateAppliStatsForTR <file-path> BRIEF <system-name>/<application-name>](#)
 - [-GenerateSets](#)
 - [-SaveAfterAutomatedOperations](#)
 - [-ExitAfterAutomatedOperations](#)

Target audience:

- CAST administrators



Summary: This section of documentation provides an explanation of how to use the CAST Transaction Configuration Center in an automated and repeatable way.

Prerequisites

Before you can starting using the CAST Transaction Configuration CLI, you must ensure that you meet the following requirements:



An **installation of CAST AIP** is required, which must be located on the machine you wish to run the CLI from.

How does it work?

The CAST Transaction Configuration CLI includes multiple command arguments to perform various different actions including:

- Import Templates Setup
- Update Templates Setup
- Export Templates Setup
- Import Application Setup
- Export Application Setup
- Save Calibration Changes
- Run Snapshot Preparation
- Compute Application FP
- Export Application Entry Points
- Create Application Stats For TR
- Generate sets
- Save After Automated Operations
- Exit After Automated Operations

To execute an action via the command line, you must specify the **CAST-TransactionConfig-CLI.exe** executable (rather than the CAST-TransactionConfig.exe executable which is used for the GUI). Command arguments are expressed as verbs introduced by a dash (-), followed by various mandatory arguments that must be enclosed with double quotes (") if containing blank spaces. Some command arguments should appear only once (such as **-ConnectProfile** for example), whereas others can appear as many times as necessary (such as **-ImportAppliSetup** or **-ExportAppliSetup** for example).

For options which require a **file path**, if the given path is not absolute, the file will be created into, or read from, the same CAST temporary directory as used by the CAST Management Studio.

Miscellaneous information

- Boolean format: [YES|NO|CANCEL]

- String format: enclose with double quotes if containing blank spaces
- The examples listed below should be placed on one line if they are being used in a batch file.

Command arguments

-ShowHelp

This option makes the CAST Transaction Configuration Center display the syntax of its command line on the standard output, then exit. Thus this option should only be used when invoking the CAST Transaction Configuration Center from a command shell session such as CMD.EXE.

Example

```
CAST-TransactionConfig-CLI.exe -ShowHelp
```

-HideGUI

This option prevents the CAST Transaction Configuration Center from displaying its splash screen and graphical user interface. Thus this option should only be used when invoking the CAST Transaction Configuration Center from a command shell session such as CMD.EXE. In addition, all messages, errors, exceptions, etc. will be logged on the standard output instead of inside the log window, and no dialog box will be displayed if the application is requested to terminate when some pending changes have not been saved.

Example

```
CAST-TransactionConfig-CLI.exe -HideGUI
```

-ConnectProfile <profile-name>

This option connects the CAST Transaction Configuration Center to the CAST Management Service (MNGT), using the Connection Profile whose name is provided in argument. If the CAST Management Service is not provided on the command line, the CAST Management Studio Database connection manager dialog will be displayed before executing the commands which depend upon a connection (namely **-ImportTemplatesSetup**, **-ExportTemplatesSetup**, **-ImportAppliSetup** and **-ExportAppliSetup**).

Example

```
CAST-TransactionConfig-CLI.exe -ConnectProfile CAST_MNGT
```

-ImportTemplatesSetup <file-path>

This option imports a setup from file into the root [Templates] node - the path and file name are required (the file extension .TCCSetup can be omitted). The currently existing configuration, if any changes occurred, is discarded before being replaced by the newly imported one.

Required arguments

- [-ConnectProfile <profile-name>](#)

Example

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ImportTemplatesSetup "C:\TCC\MyTemplatesSetup.TCCSetup"
```

-UpdateTemplatesSetup <file-path>

This option updates the root [Templates] node configuration with the setup from file - the path and file name are required (the file extension .TCCSetup can be omitted), merging the current [Templates] node configuration with the one of the file according to rules explained in the "Update Configuration" section of the CAST Transaction Configuration Center help. Thus, it is particularly useful for configuring the root [Templates] node with Setups related to multiple technologies. This option can appear multiple times in a command if multiple setup files need to be imported.

Required arguments

- [-ConnectProfile <profile-name>](#)

Example

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -UpdateTemplatesSetup "C:\TCC\MyTemplatesSetupForJava.TCCSetup" -UpdateTemplatesSetup "C:\TCC\MyTemplatesSetupForPHP.TCCSetup" -UpdateTemplatesSetup "C:\TCC\MyTemplatesSetupForSHELL.TCCSetup"
```

-ExportTemplatesSetup <file-path>

This option exports the Setup of the root [Templates] node to file - the path and file name are required (the file extension .TCCSetup can be omitted).

Required arguments

- [-ConnectProfile <profile-name>](#)

Example

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ExportTemplatesSetup "C:\TCC\MyTemplatesSetup.TCCSetup"
```

-ImportAppliSetup <file-path> <system-name>/<application-name>

This option imports a setup from file - the path and file name are required (the file extension .TCCSetup can be omitted) into the specified System > Application node. This option can appear multiple times in a command if the configuration of several applications must be imported. If <system-name> contains a slash (/), it must be enclosed with single quotes (').

Required arguments

- [-ConnectProfile <profile-name>](#)

Examples

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ImportAppliSetup "C:\TCC\MyApplicationSetup.TCCSetup" "My System/My Application"
```

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ImportAppliSetup "C:\TCC\CustomersBillingSetup.TCCSetup" "My System/Customers Billing" -ImportAppliSetup "C:\TCC\AcctMngtSetup.TCCSetup" "My System/Account Management"
```

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ImportAppliSetup "C:\TCC\MyApplicationSetup.TCCSetup" "'My Systems/John'/My Application"
```

-ExportAppliSetup <file-path> <system-name>/<application-name>

This option exports the Setup of the System > Application node to file - the path and file name are required (the file extension .TCCSetup can be omitted). This option can appear multiple times in a command if the configuration of several applications must be exported. If <system-name> contains a slash (/), it must be enclosed with single quotes (').

Required arguments

- [-ConnectProfile <profile-name>](#)

Examples

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ExportAppliSetup "C:\TCC\MyApplicationSetup.TCCSetup" "My System/My Application"
```

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ExportAppliSetup "C:\TCC\CustomersBillingSetup.TCCSetup" "My System/Customers Billing" -ExportAppliSetup "C:\TCC\AccntMngtSetup.TCCSetup" "My System/Account Management"
```

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ExportAppliSetup "C:\TCC\MyApplicationSetup.TCCSetup" "'My Systems/John'/My Application"
```

-SaveCalibrationChanges (YES|NO|CANCEL)

This option configures the behavior of the CAST Transaction Configuration Center when there exists some unsaved changes in the calibration of data functions and/or transactions of an application, just before starting an action which needs these changes to be saved, such as for example the computation of the function points of this application. The following settings are available:

- **YES:** the calibration changes for the application are saved, then the action is performed, taking the saved changes into account
- **NO:** the calibration changes for the application are reverted, then the action is performed, taking the initial (or most recently saved) changes into account
- **CANCEL:** the action is cancelled, and the calibration changes for the application are kept (unsaved)

The default setting is CANCEL. Once a setting has been changed via the CLI, it remains in place until it is overridden or all CLI arguments have been processed.



Warning: In the list of CLI arguments, this option must appear before the CLI arguments related to the actions whose results depend upon the chosen setting, because this option not only affects the results of these actions, but also even controls whether their execution may be cancelled.

Example

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -SaveCalibrationChanges YES -ComputeAppliFP "My System/Customers Billing" -SaveCalibrationChanges NO -ComputeAppliFP "My System/Account Management"
```

-RunSnapshotPreparation (YES|NO|CANCEL)

This option configures the behavior of the CAST Transaction Configuration Center when the snapshot data of an application doesn't exist or is obsolete, whereas an action needing up-to-date snapshot data for this application is about to start, such as the computation of its function points for example. The following settings are available:

- **YES:** the Snapshot Preparation action is run, then the action is performed
- **NO:** the Snapshot Preparation action is not run, and the action is performed using the current (and maybe obsolete) snapshot data
- **CANCEL:** the action is cancelled, and the snapshot data are left unchanged

The default setting is CANCEL. Once a setting has been changed via the CLI, it remains in place until it is overridden or all CLI arguments have been processed.



Warning: In the list of CLI arguments, this option must appear before the CLI arguments related to the actions whose results depend upon the chosen setting, because this option not only affects the results of these actions, but also even controls whether their execution may be cancelled.

Example

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -RunSnapshotPreparation YES -ComputeAppliFP "My System/Customers Billing" -RunSnapshotPreparation CANCEL -ComputeAppliFP "My System/Account Management"
```

-ComputeAppliFP <system-name>/<application-name>

This option computes the function points of the specified application. This option can appear multiple times if the function points of several applications need to be computed. If the <system-name> contains a slash (/), it must be enclosed with single quotes (').

If the application has unsaved changes in the calibration of its data functions and/or of its transactions, or if the Snapshot Preparation action should be run prior to the computation of the function points of the application, the computation will continue, or abort, according to the current settings of, respectively, the **-SaveCalibrationChanges** and the **-RunSnapshotPreparation** options.

Required arguments

- [-ConnectProfile <profile-name>](#)

Examples

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ComputeAppliFP "My System/My Application"

CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -RunSnapshotPreparation YES -
SaveCalibrationChanges YES -ComputeAppliFP "My System/Customers Billing" -ComputeAppliFP "My System/Account
Management"

CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ComputeAppliFP "'My Systems/John'/My
Application"
```

-ExportAppliEntryPoints <file-path> <system-name>/<application-name>

This option exports the Transaction Entry Points of the System > Application > Setup > Transaction entry points node, ordered by Full Name, to file in XML format - the path and file name are required (the file extension .xml can be omitted). If the <system-name> contains a slash (/), it must be enclosed with single quotes (').

Required arguments

- [-ConnectProfile <profile-name>](#)

Example

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ExportAppliEntryPoints "C:
\TCC\MyAppEntryPoints.xml" "My System/My Application"
```

If Transaction Entry Points have to be exported for an application for which function points have never been computed, or are now out of date because the application and/or its TCC Setup have changed since the last time function points have been computed, you should add a **-ComputeAppliFP** option about the application of interest in the list of commands provided to the CLI before the **-ExportAppliEntryPoints**, to ensure the application's Transaction Entry Points will be accurate.

Example

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ComputeAppliFP "'My System/John'/My
Application" -ExportAppliEntryPoints "C:\TCC\JohnApp.xml" "'My System/John'/My Application"
```

-CreateAppliStatsForTR <file-path> BRIEF <system-name>/<application-name>

This option asks the CAST Transaction Configuration Center to compute statistics about transactions of a specified application (via the system and name). Statistics can be brief or detailed, according to the desired quantity of information to be produced. If the <system-name> contains a slash (/), it must be enclosed with single quotes (').

After the statistics have been computed, statistics will be written into the file specified by its full path/name (the file extension can be omitted, and if so will be **.txt** by default for brief statistics, or **.csv** by default for detailed statistics; if the given path is not absolute, the file will be created into the CAST temporary directory).

Only the following setting is available at the moment:

- **BRIEF:** the brief statistics will contain, in text (.txt) file format, only numeric values (such as the count of transactions, function points, % of artifacts involved in transactions, etc.).

Required arguments

- [-ConnectProfile <profile-name>](#)

Example

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -CreateAppliStatsForTR "C:\TCC\MyApp_BriefStats.txt" BRIEF "My System/My Application"
```

If statistics have to be computed for an application whose function points have never been computed, or are now out of date because the application and/or its TCC Setup have changed since the last time function points have been computed, you should add a **-ComputeAppliFP** option for the specified application in the list of commands provided to the CLI, before the **-CreateAppliStatsForTR** is run, to ensure that the application's statistics are accurate.

Example

```
CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ComputeAppliFP "My System/My Application" -CreateAppliStatsForTR "C:\TCC\MyApp_BriefStats.txt" BRIEF "My System/My Application"
```

-GenerateSets

This option asks the CAST Transaction Configuration Center to generate sets for:

- a group of rules, identified by their **type**:
 - **ENTRY** (i.e. all Entry Point rules)
 - **DATA** (i.e. all Data Entities rules)
 - **END** (i.e. all End Point rules)
 - **EXCLUDED** (i.e. all Excluded Items rules)
- a specific rule, identified by its **ID**



Note that:

- All rules are taken into account: i.e. they can be created under "Free Definition", "By Naming", "By Inheritance" and "By Type".
- a rule's ID is stored in the **cal_objsetdef** table in the CAST Management Service schema. Therefore, you can query this table to find the ID of the rule you need. For example, a rule called "TABLES" has been created in the GUI. Running the following query will return the ID of the rule:

```
select set_id from mngt.cal_objsetdef
where setname = 'TABLES'
```

Required arguments

- **-ConnectProfile** <profile-name>

Example

```
By rule type: CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile <profile_name> -GenerateSets "SYSTEM /APPLICATION" <ENTRY | DATA | END | EXCLUDED>
```

```
By rule ID: CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile <profile_name> -GenerateSets "SYSTEM /APPLICATION" <rule_ID>
```

Results

Assuming a rule returns objects, then the following will be displayed:

```

Connected to CSS -> //CSS:2280/postgres, cast_mngt
Generate sets
Connection...
Computing sets...
Computing exclusions...
Aggregating sets...
SET_OBJECT:516;Orders;Microsoft table;default.CASTPUBS.dbo.Orders
SET_OBJECT:492;salesdetail;Microsoft table;default.CASTPUBS.dbo.salesdetail
SET_OBJECT:6423;SalesCreated.html;HTML5 source code;C:\CAST\DEPLOY\MEUDON\JEE\SalesCreated.html
/CAST_HTML5_SourceCode
SET_OBJECT:504;roysched;Microsoft table;default.CASTPUBS.dbo.roysched
SET_OBJECT:530;Invoice_Line;Microsoft table;default.CASTPUBS.dbo.Invoice_Line
SET_OBJECT:464;stores;Microsoft table;default.CASTPUBS.dbo.stores
SET_OBJECT:523;Order_Line;Microsoft table;default.CASTPUBS.dbo.Order_Line
SET_OBJECT:457;titleauthor;Microsoft table;default.CASTPUBS.dbo.titleauthor
SET_OBJECT:537;Invoice;Microsoft table;default.CASTPUBS.dbo.Invoice
SET_OBJECT:510;publishers;Microsoft table;default.CASTPUBS.dbo.publishers
SET_OBJECT:499;sales;Microsoft table;default.CASTPUBS.dbo.sales
SET_OBJECT:2853;ListTitles.jsp;eFile;[C:\CAST\DEPLOY\MEUDON\JEE>ListTitles.jsp]
SET_OBJECT:484;Stock;Microsoft table;default.CASTPUBS.dbo.Stock
SET_OBJECT:2810;ErrorPage.jsp;eFile;[C:\CAST\DEPLOY\MEUDON\JEE>ErrorPage.jsp]
SET_OBJECT:6425;Index.html;HTML5 source code;C:\CAST\DEPLOY\MEUDON\JEE\Index.html/CAST_HTML5_SourceCode
SET_OBJECT:553;authors;Microsoft table;default.CASTPUBS.dbo.authors
SET_OBJECT:2863;Connect.jsp;eFile;[C:\CAST\DEPLOY\MEUDON\JEE\Connect.jsp]
SET_OBJECT:545;discounts;Microsoft table;default.CASTPUBS.dbo.discounts
SET_OBJECT:474;Stock_Line;Microsoft table;default.CASTPUBS.dbo.Stock_Line
SET_OBJECT:2779;NewSales.jsp;eFile;[C:\CAST\DEPLOY\MEUDON\JEE\NewSales.jsp]
SET_OBJECT:2340;BookDemoServletMaster;Java Class;com.castsoftware.bookdemo.servlet.BookDemoServletMaster
SET_OBJECT:2342;NewSale;Java Class;com.castsoftware.bookdemo.servlet.NewSale
SET_OBJECT:2811;ListAuthors.jsp;eFile;[C:\CAST\DEPLOY\MEUDON\JEE>ListAuthors.jsp]
SET_OBJECT:444;titles;Microsoft table;default.CASTPUBS.dbo.titles
Number of generated objects: 24
Number of generated objects excluded: 0

```

Notes

When using rule types

- The activation status of the rules will be **respected**, i.e. the CLI will only generate the set for rules which are active
- If the rule type is ENTRY | DATA | END:
 - the CLI will also generate all the (active) EXCLUDED rules to take them into consideration
 - results will show all objects generated by any of the rules (with the prefix "SET_OBJECT:"); plus all the objects generated by any of the rules, but which were ALSO generated by any excluded item rules (with the prefix "SET_OBJECT_EXCLUDED:"). For example:

```

...
SET_OBJECT:objectId1;objectName1;objectType1;objectFullName1
SET_OBJECT:objectId2;objectName2;objectType2;objectFullName2
SET_OBJECT:objectId3;objectName3;objectType3;objectFullName3
...
SET_OBJECT_EXCLUDED:objectId4;objectName4;objectType4;objectFullName4
SET_OBJECT_EXCLUDED:objectId5;objectName5;objectType5;objectFullName5
...

```

- If the rule type is EXCLUDED:
 - the CLI will generate ONLY the excluded item rules
 - the results will show all the objects generated by the rules (with the prefix "SET_OBJECT:")

```

...
SET_OBJECT:objectId1;objectName1;objectType1;objectFullName1
SET_OBJECT:objectId2;objectName2;objectType2;objectFullName2
SET_OBJECT:objectId3;objectName3;objectType3;objectFullName3
...

```

When using rule IDs

- The activation status of the rule will be **ignored**, i.e. the CLI will generate the set no matter whether the rule is active or inactive
- If the selected rule ID is an ENTRY | DATA | END type:
 - the CLI will also generate all the (active) EXCLUDED rules to take them into consideration
 - results will show all objects generated by any of the rules (with the prefix "SET_OBJECT:"); plus all the objects generated by any of the rules, but which were ALSO generated by any excluded item rules (with the prefix "SET_OBJECT_EXCLUDED:"). For example:

```

...
SET_OBJECT:objectId1;objectName1;objectType1;objectFullName1
SET_OBJECT:objectId2;objectName2;objectType2;objectFullName2
SET_OBJECT:objectId3;objectName3;objectType3;objectFullName3
...
SET_OBJECT_EXCLUDED:objectId4;objectName4;objectType4;objectFullName4
SET_OBJECT_EXCLUDED:objectId5;objectName5;objectType5;objectFullName5
...

```

- If the type is EXCLUDED or a Generic Set
 - the CLI will generate the set
 - the results will show all the objects generated by the rules (with the prefix "SET_OBJECT:")

```

...
SET_OBJECT:objectId1;objectName1;objectType1;objectFullName1
SET_OBJECT:objectId2;objectName2;objectType2;objectFullName2
SET_OBJECT:objectId3;objectName3;objectType3;objectFullName3
...

```

-SaveAfterAutomatedOperations

This option asks the CAST Transaction Configuration Center to save all pending changes such as, for example, those resulting from a Setup import at the level of either the root Templates node or the level of one/some System > Application nodes. Saving is mandatory for ensuring that the results of any operation that has modified the configuration will not be lost after the CAST Transaction Configuration Center is exited.

Note: The pending changes will only be saved if they do not contain any errors.

Example

```

CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ImportAppliSetup "C:\TCC\MyApplicationSetup.
TCCSetup" "My System/My Application" -SaveAfterAutomatedOperations

```

-ExitAfterAutomatedOperations

This option asks the CAST Transaction Configuration Center to exit after all CLI commands have been executed. If any pending changes should be saved before exit, use this option along with **-SaveAfterAutomatedOperations** as shown in the example below.

Example

```

CAST-TransactionConfig-CLI.exe -HideGUI -ConnectProfile CAST_MNGT -ImportAppliSetup "C:\TCC\MyApplicationSetup.
TCCSetup" "My System/My Application" -SaveAfterAutomatedOperations -ExitAfterAutomatedOperations

```