

Validate and Accept the Delivery

On this page:

- [Introduction](#)
- [Reviewing the Delivery Log](#)
 - [Via the CAST Management Studio](#)
 - [Review file catalog and alerts](#)
- [Generate reports](#)
 - [Generate report on found files](#)
 - [Delivery report](#)
- [Accepting/Rejecting the delivery](#)
 - [Rejection](#)
 - [Acceptance](#)

Introduction

The CAST AI Admin is responsible for validating the source code delivered by the Delivery Manager(s) before accepting the delivery and proceeding with the configuration and execution of the analysis.



Why is it important?

- avoid time consuming rework and delay due to incomplete delivery
- improve accuracy of analysis
- ensure consistency with analysis scope (application boundary)

Typically issues at this stage are due to missing files and other deficiencies that result in an incomplete delivery. Resolution includes either a request for the missing component or the decision (wherever possible) to proceed with a redefined analysis boundary that excludes the undelivered components.

There are significant differences in the delivery validation depending on whether the validation pertains to a first time delivery during an application on-boarding or to delivery of a new version of the source code of an application previously on-boarded.

Where you are undertaking an application reanalysis, the delivery validation can often be limited to inspecting the CAST Delivery Manager Tool delivery log (see **Reviewing the Delivery Log** below). Examining the delta between two deliveries can help to assess if the changes are expected or are "reasonable" and therefore do not require a partial or full re-qualification and a new on-boarding of the application.

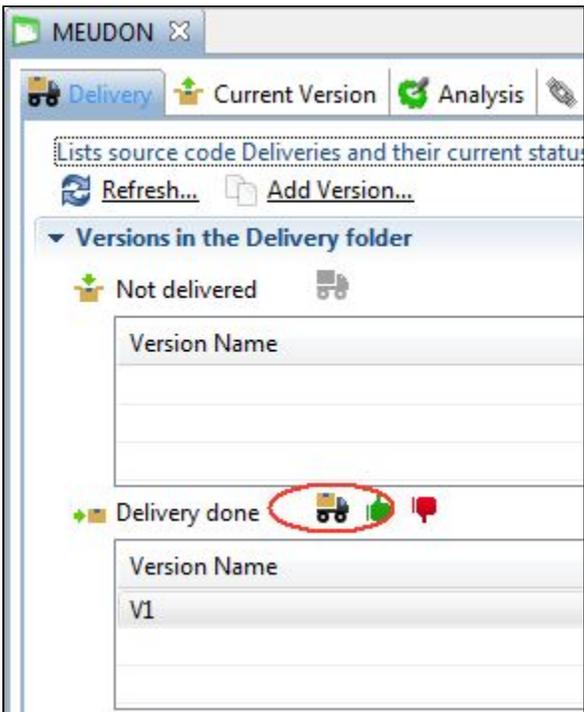
During the an initial application on-boarding, the review and acceptance of the source code delivery requires a more in depth assessment against what is discovered during the [Qualification](#) step.

Reviewing the Delivery Log

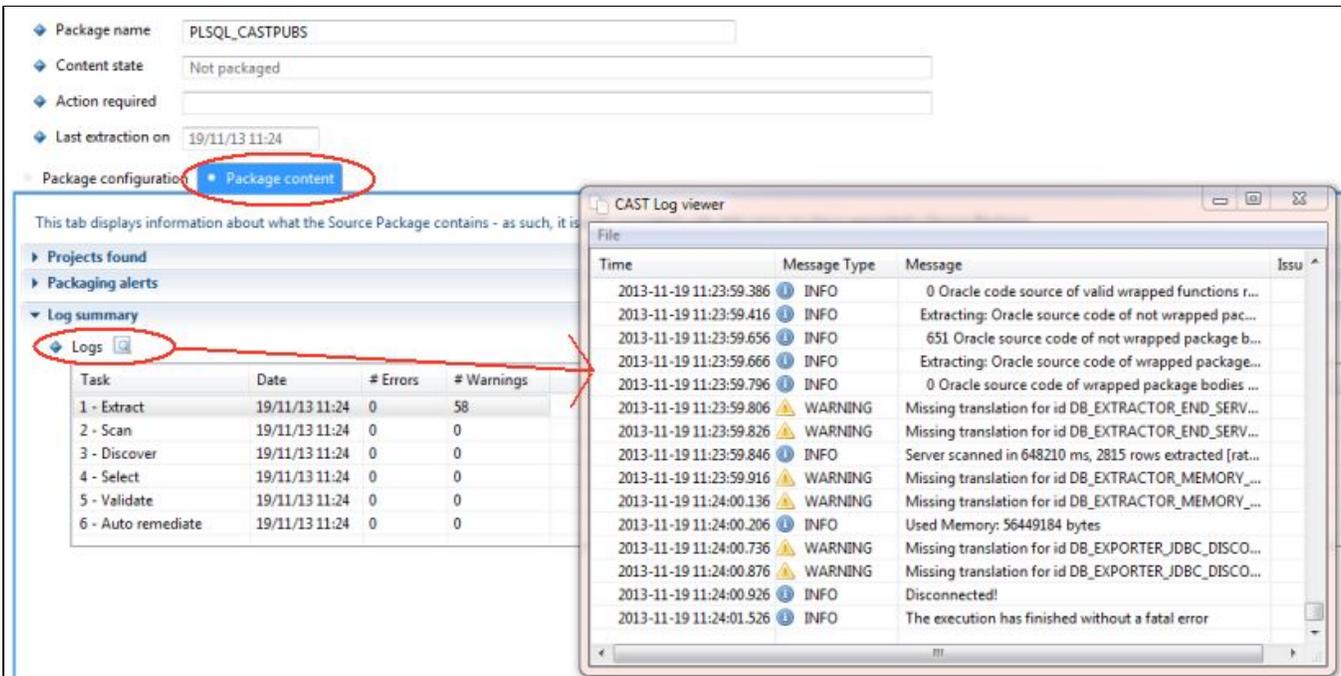
This step involves reviewing the package log and any warnings that may be displayed. You can access the log in one of two ways:

Via the CAST Management Studio

Launching the **CAST Delivery Manager Tool** from the CAST Management Studio (Application editor) and inspecting the content of EACH package that has been delivered:



Launch the DMT using the above icon



Accessing the Log Viewer from the Package Content tab in the DMT

i Note that when running the CAST Delivery Manager Tool direct from the CAST Management Studio, the Java Heap Space memory threshold is set to 640 MB. If you are experiencing out of memory issues, then CAST highly recommends launching the CAST Delivery Manager Tool from the CAST AIC Portal ensuring that a 64bit JRE is being used.

Review file catalog and alerts

You can also use the DMT (Package Content tab) to view information about the **projects** that have been identified, the **alerts** that have been generated and the **files** that have been found:

Package configuration • **Package content**

This tab displays information about what the Source Package contains - as such, it is only populated with data once you have generated a Source Package.

Projects found

Number of Added/Removed items is based on the cloned version of this package

Project summary by type

Project type	Technology	# Selected	# Ignored	# Added	# Removed
Eclipse Java project	Java EE	1	0	0	0
Web JSP	Java EE	0	1	0	0

List of projects for the selected type

T	Name	Path	Selected/Ignored	Compared with last version
	CastPubs	./project	Selected	N/A

Packaging alerts

Alert summary by type

Alert type	Project type	Number of alerts

Files found

Number of Added/Removed items is based on the cloned version of this package

File summary by type

File extension	Total files	Added files	Removed files	Folder
*.bak	2	0	0	<Package root>
*.class	21	0	0	<Package root>
*.classpath	1	0	0	<Package root>
*.css	1	0	0	<Package root>
*.db	1	0	0	<Package root>

Please see the CAST Delivery Manager Tool help (tap F1 in any location in the DMT to open the help) for more information about the information provided in the Package Content tab.

i The Delivery Manager Tool displays "delta" information when (Added/Removed columns) when you are delivering a new Version of the same Application.

During an application on-boarding, the validation steps include cross validation of the Package Content report with information gathered during the pre-analysis step via the [Qualification](#) process and then reviewing any warning, error or alert raised.

i It is important to note here that the DMT, in the current version, only provides a simple list of what has been extracted and included in the package. The inspection of this list may trigger the need for further analysis of the delivered source code and in some cases the need for pre-processing it before analysis (see [Qualification](#)).

Generate reports

The CAST Delivery Manager Tool offers two reporting functions: **discovered files** and **a delivery report**. You can use these functions once the packaging action is complete:

Generate report on found files

V1

◆ Version name

◆ Delivery State

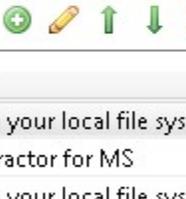
◆ Copied from version

◆ Version Date

◆ Content State

◆ Action required

The source code is delivered to the AI Center in packages. A package

◆ Packages 

Location	Name
Folder on your local file system	JEE
CAST extractor for MS	TSQL
Folder on your local file system	NET
Folder on your local file system	VB
Folder on your local file system	NET2
Folder on your local file system	MAVEN

◆ Detailed report on files found

Delivery report

V1

◆ Version name

◆ Delivery State

◆ Copied from version

◆ Version Date

◆ Content State

This option will generate an XML based report listing all items in the delivery, for example:

- Projects found
- Project removed
- Alerts
- etc.

The report will open with the default application assigned to open XML files. The report will only contain information once the **Package action** has been run. Full details of what is provided in the Delivery Report are shown below:

- **folderPath**: The path of the source folder
- **generationDate**: The date the report was generated

- **totalDuration**: Time taken to generate the package
- **releaseInfo**: The version of Delivery Report
- **Application**: details about the Application
 - **applicationName**: The name provided for the Application
 - **applicationGuid**: ID assigned to the Application
 - **Version**: details about the Version
 - **versionName**: The name of the Version
 - **applicationGuid**: ID assigned to the Version
 - **FileLanguages**: details about the files that will be considered
 - **FileLanguage**: The files in the selected source path are identified according to their language and represented in chronological order with respect to language Id
 - **languageId**: The ID used to identify the language
 - **extensions**: The file extensions considered
 - **total**: Total number of files for the language
 - **added**: No. of files added during the current packaging compared with previous
 - **removed**: No. of files removed during current packaging compared with previous
 - **modified**: No. of files modified during next packaging compared with previous
 - **totalSize**: Total size of the files
 - **addedSize**: Size of the added files compared with previous
 - **removedSize**: Size of the removed files compared with previous
 - **modifiedSize**: Size of the modified files compared with previous
 - **loc**: Lines of count of the projects
 - **notInProjects**: The files not included in the project
 - **Extension**: Information about the file extensions discovered.
 - **extensionId**: The file extensions discovered
 - **total**: Total number of files of the particular extension that have been discovered
 - **added**: No. of files added during the current packaging compared with previous
 - **removed**: No. of files removed during the current packaging compared with previous
 - **modified**: No. of files modified during next packaging compared with previous
 - **totalSize**: Total size of the discovered files
 - **addedSize**: Size of the added discovered files compared with previous
 - **removedSize**: Size of the removed discovered files compared with previous
 - **modifiedSize**: Size of the modified discovered files compared with previous
 - **loc**: Lines of count of the discovered files
 - **notInProjects**: The files that are not included in the project
 - **Framework**: Information about the discovered frameworks
 - **frameworkId**: The frameworks used in the source code
 - **version**: Framework version.
 - **Projects**: The projects identified during packaging
 - **Project**: Information about the project
 - **projectType**: The type of project discovered
 - **discovererName**: The name of the DMT discoverer that identified the project
 - **total**: Total number of projects that have been discovered
 - **added**: No. of projects added during the current packaging compared with previous
 - **removed**: No. of projects removed during the current packaging compared with previous
 - **modified**: No. of projects modified during next packaging compared with previous
 - **Alerts**: Alerts generated during the packaging.
 - **Alert**: information about the alert
 - **id**: alert ID
 - **total**: total number of alerts of the particular ID
 - **Log Messages**: Log messages generated during the packaging
 - **id**: log message ID
 - **total**: total number of log messages of the particular ID
 - **parameter**: the name of the parameter
 - **id**: parameter ID
 - **value**: value of the particular parameter



Log messages can be as per the following types:

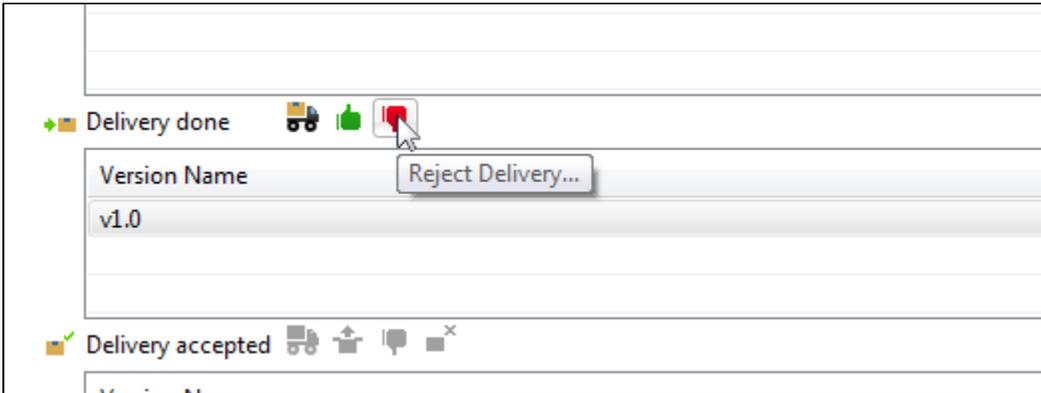
1. Extraction Log messages
2. Scan Log messages
3. Discovery Log messages
4. Selection Log messages
5. Validation Log messages
6. Remediation Log messages

The level of message can be Fatal, Error or Warning. If there are multiple parameters then the parameter id is mentioned as 1, 2, 3...]

Accepting/Rejecting the delivery

Rejection

When any issue is detected and/or unresolved questions are raised, the analysis process should be halted as the delivery cannot be accepted until these issues are fully resolved. The CAST AI Admin should therefore reject the delivery in the CAST Management Studio:



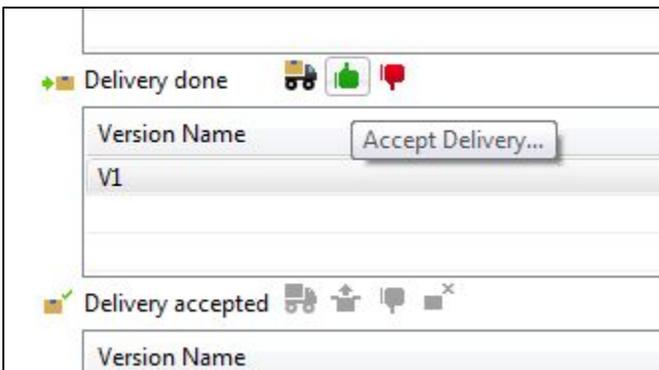
When the delivery is rejected, the CAST Management Studio does not automatically notify the Delivery Manager(s) of the rejection. Moreover there is no justification why this has occurred. Thus while the Delivery Manager can discover that a prior delivery has been rejected by reopening the DMT and noting the change in status of the prior delivery, they will not know why this has been rejected by the CAST AI Admin.

CAST recommends that when a delivery is rejected, the CAST AI Admin notifies (via email) the Delivery Manager of the rejection, providing a reason and possible remediation actions that may be required before a new delivery.

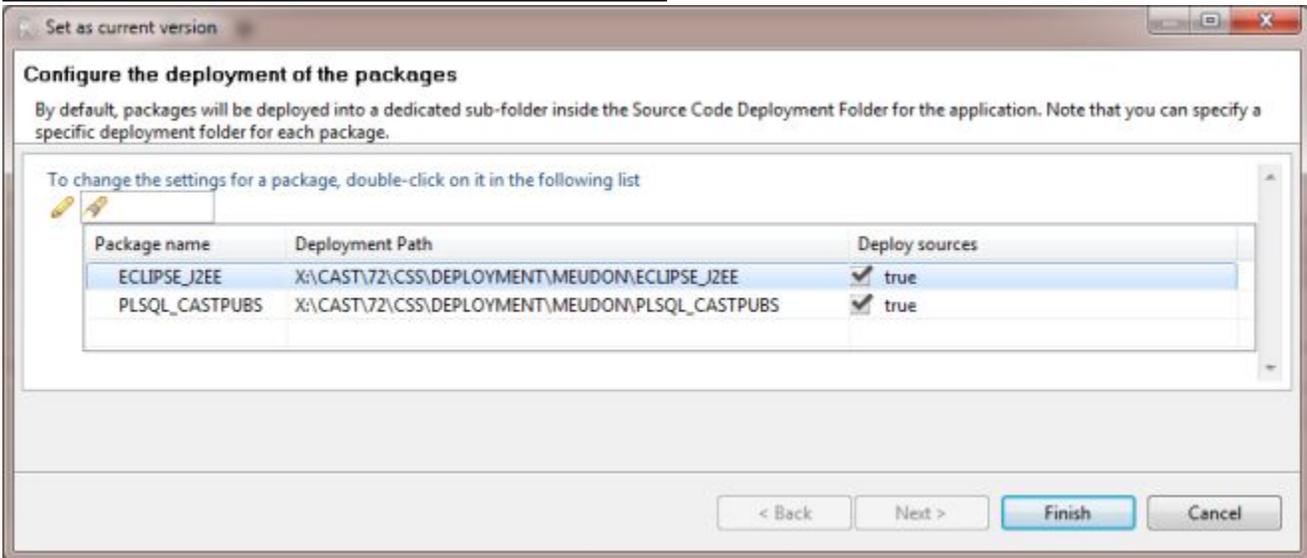
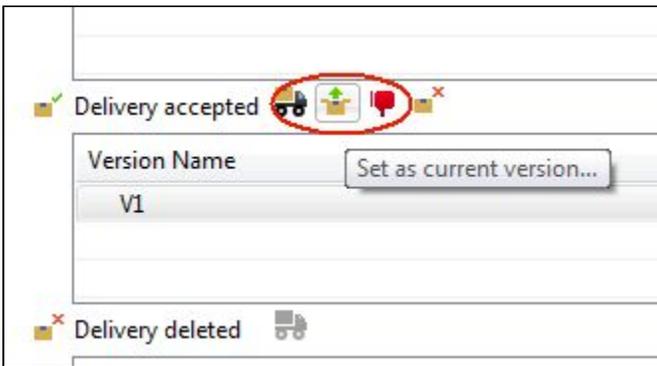
Acceptance

Accepting the delivery is a two step process that results in the transfer and de-archiving of the delivered package(s) into the **Deployment folder** (see Window > Preferences > Platform Settings in the CAST Management Studio):

- Accept the delivery



- Set the delivered version as the **current version**



- You will be able to directly inspect the code only after the deployment steps are completed.
- You should now proceed to [Run and validate the analysis](#).

i By default the **Deployment Path** points to the **Deployment folder** configured in the CAST Management Studio (Window > Preferences > Platform Settings), however you are free to choose any location you require. CAST recommends using the default suggestion to ease future source code deployment. Please also refer to [Set-up a file repository for your analysis data](#) for more information.

Note that if, for any reason, the delivery is rejected **following** the deployment of the delivered package, the folders created as a result of the package deployment will not be removed from the **Deployment folder** until a new delivery of the application is completed.