

HTML5 and JavaScript - Packaging, delivering and analyzing your source code

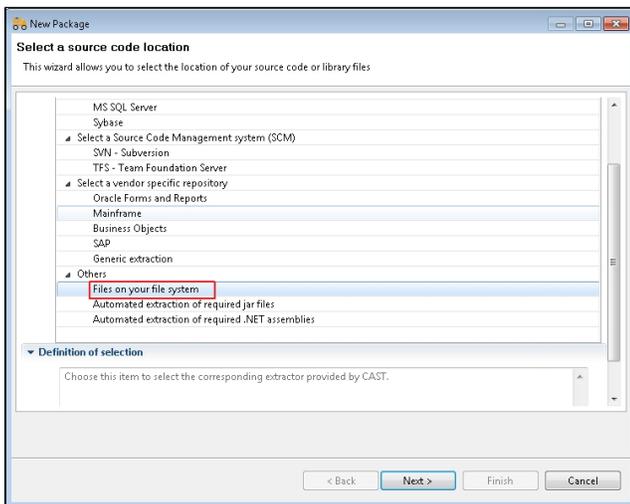
Packaging and delivery

i Note that the **HTML5/JavaScript** extension does not contain any CAST Delivery Manager Tool **discoverers or extractors**, therefore, no "HTML5/JavaScript" projects will be detected by the CAST Delivery Manager Tool. You therefore have two choices:

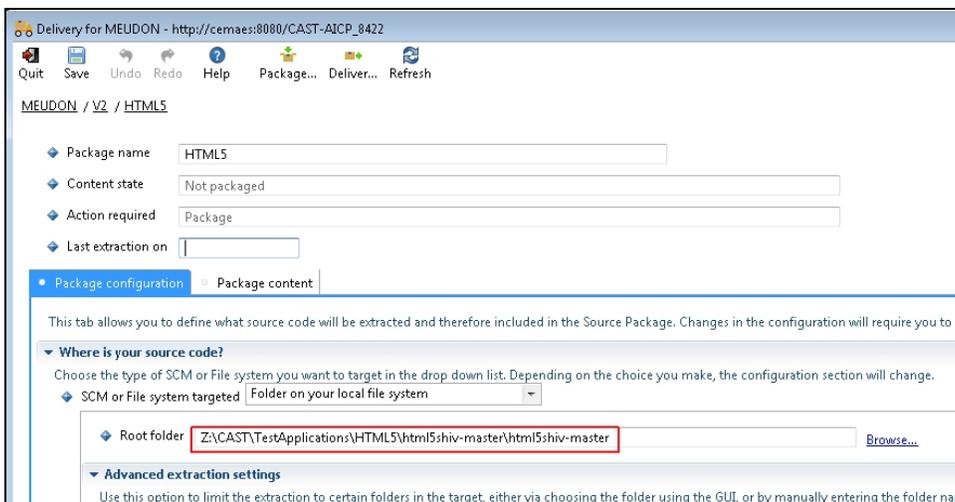
- manually create an Analysis Unit in the CAST Management Studio.
- or download and install the [Web Files Discoverer](#) to automatically detect projects in the CAST Delivery Manager and therefore Analysis Units in the CAST Management Studio.

Using the CAST Delivery Manager Tool:

- create a new **Version**
- create a new **Package** for your HTML5/JavaScript source code using the **Files on your file system** option:



- Define the **root folder** of your Application source code:

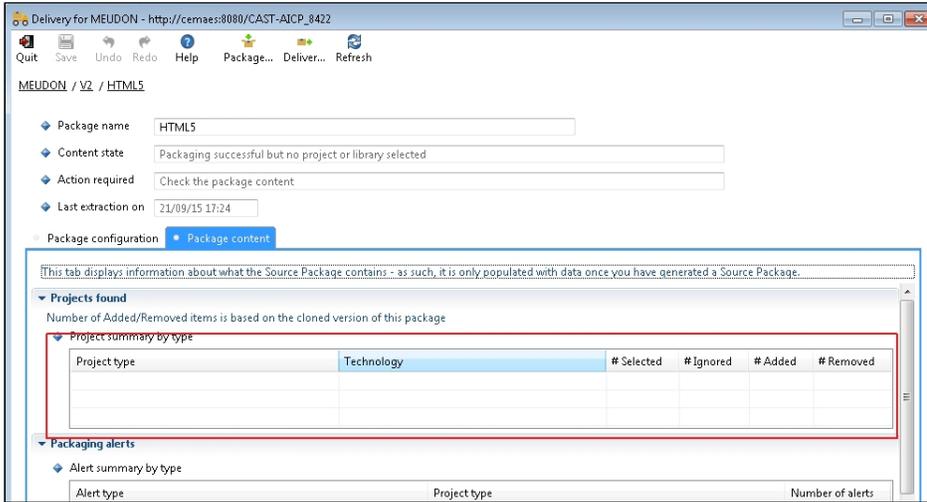


- Run the **Package action**
- Before delivering the source code, check the **packaging results**:

Without the Web Files Discoverer

If you are not using the [Web Files Discoverer](#), the following will occur:

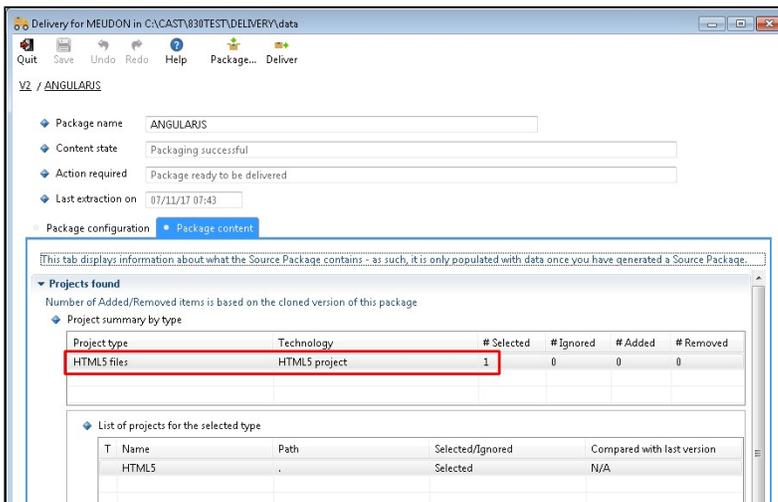
- the CAST Delivery Manager Tool will **not** find any "projects" related to the HTML5/JavaScript application source code - this is the **expected behaviour**. However, if your HTML5/JavaScript related source code is part of a larger application (for example a JEE application), then other projects may be found during the package action (click to enlarge):



With the Web Files Discoverer

If you are using the [Web Files Discoverer](#), the following will occur:

- the CAST Delivery Manager Tool will **automatically detect "HTML5 file projects"** (see [Web Files Discoverer](#) for more technical information about how the discoverer works) related to the HTML5/JavaScript application source code. In addition, if your HTML5/JavaScript related source code is part of a larger application (for example a JEE application), then other projects may also be found during the package action (click to enlarge):



- Deliver the **Version**

Analyzing

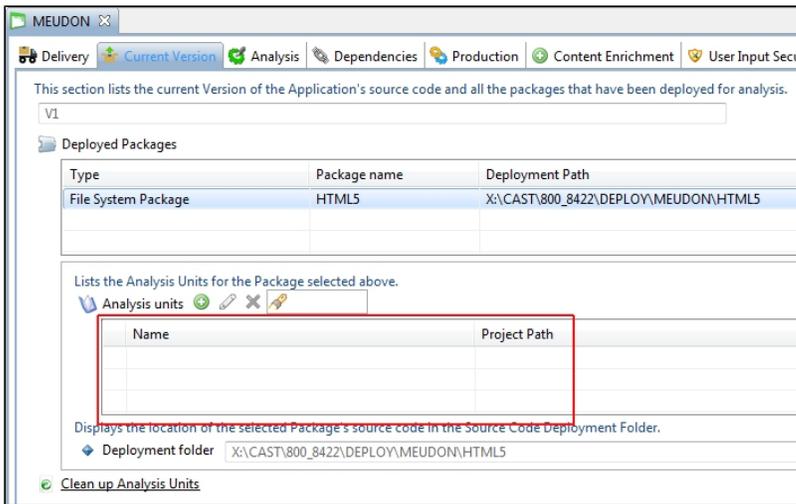
Using the CAST Management Studio:

- Accept and deploy the **Version** in the CAST Management Studio.

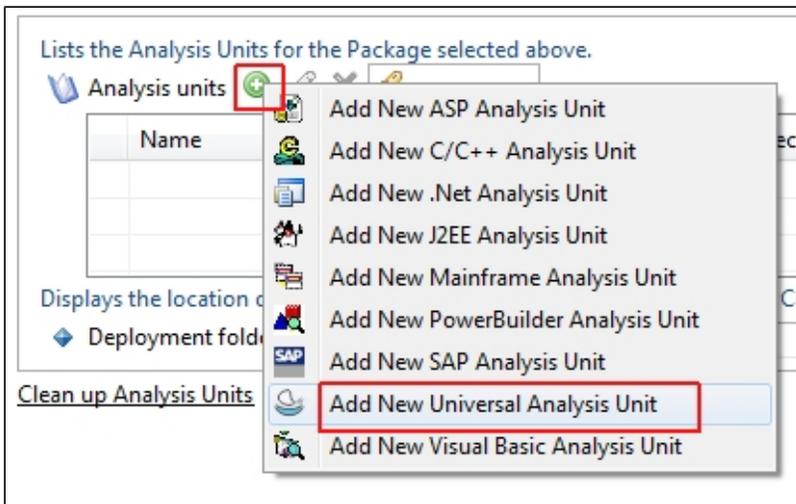
Without the Web Files Discoverer

If you are not using the [Web Files Discoverer](#), the following will occur:

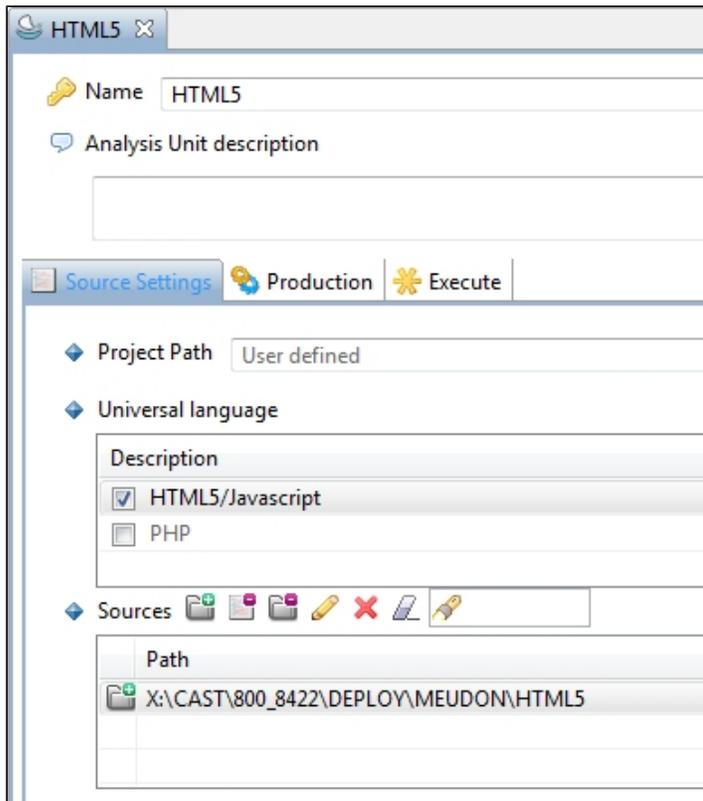
- No **Analysis Units** will be created automatically relating to the HTML5/JavaScript source code - this is the **expected behaviour**. However, if your HTML5/JavaScript related source code is part of a larger application (for example a JEE application), then other Analysis Units may be created automatically:



- In the **Current Version** tab, add a new Analysis Unit specifically for your HTML5/JavaScript source code, selecting the **Add new Universal Analysis Unit** option:



- Edit the new Analysis Unit and configure in the **Source Settings** tab:
 - a **name** for the Analysis Unit
 - ensure you tick the **HTML5/JavaScript** option
 - define the **location** of the deployed HTML5/JavaScript source code (the CAST Management Studio will locate this automatically in the **Deployment** folder):



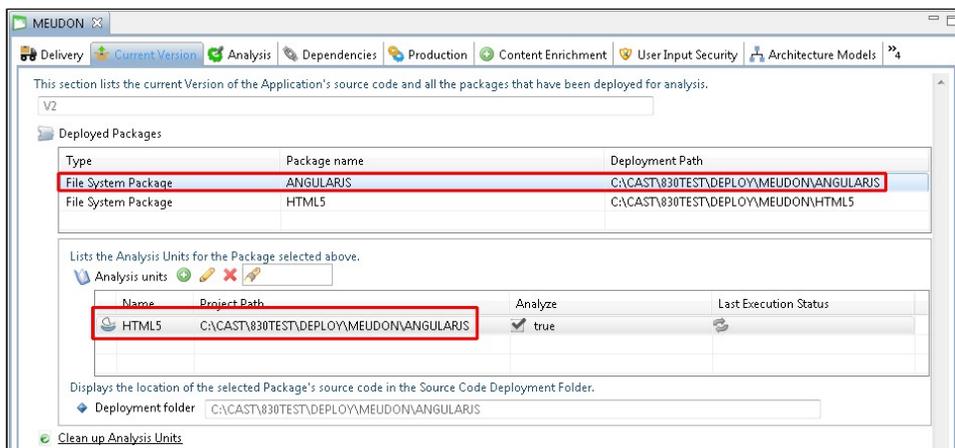
⚠ Do not add a dependency from this Analysis Unit to another client Analysis Unit.

- Run a **test analysis** on the Analysis Unit before you generate a **new snapshot**.

With the Web Files Discoverer

If you are using the [Web Files Discoverer](#), the following will occur:

- "HTML5" Analysis Units will be created **automatically** (see [Web Files Discoverer](#) for more technical information about how the discoverer works) related to the HTML5/JavaScript application source code. In addition, if your HTML5/JavaScript related source code is part of a larger application (for example a JEE application), then other Analysis Units may also be created:



⚠ Do not add a dependency from this Analysis Unit to another client Analysis Unit.

- There is nothing further to do, you can now run a **test analysis** on the Analysis Unit before you generate a new snapshot.

Using the filters.json file

The HTML5 and JavaScript extension is provided with a file called **filters.json** at the root of the extension folder. This file lists all items that will be ignored during an analysis. The file is pre-populated with items, but it is also possible to modify the file manually and add your own custom filters. See [HTML5 and JavaScript - using the filters.json file](#).

Analysis warning and error messages

The following warnings and error messages may be displayed in the analysis log:

Message ID	Message Type	Logged during	Impact	Remediation	Action
HTML5-001	Warning	Analysis	A jsp file has not been entirely analyzed because an internal issue.		if the jsp file is important for you, contact CAST Technical Support
HTML5-002	Warning	Analysis	An asp/asp/htc/cshtml file has not been entirely analyzed because an internal issue.		if the file is important for you, contact CAST Technical Support
HTML5-003	Warning	Analysis	A file has not been opened/analyzed.	See if you have access rights to this file.	
HTML5-004	Warning	Analysis	An HTTP request has not been created, a transaction link could be missing between a client and a server.		if the file is important for you, contact CAST Technical Support
HTML5-005	Warning	Analysis	One statement of a file has not been correctly parsed.		if the file is important for you and/or many objects/links have not been created about this file, contact CAST Technical Support
HTML5-006	Warning	Post analyses	An important sql query has failed when removing files skipped by analysis. Impact is that lines of code (LOC) will be higher than expected.		contact CAST Technical Support
HTML5-007	Warning	Post analyses	An important sql query has failed when removing bad links created by UA through grep (internal). Impact is that transactions will be badly altered by these bad links.		contact CAST Technical Support
HTML5-008	Warning	Post analyses	Metrics of a file have not been reported (code lines, comment lines). LOC could be wrong for this file.		contact CAST Technical Support
HTML5-009	Warning	Post analyses	A link useful for transactions has not been created between a file created by JEE analyzer and an object representing the same file created by HTML5/Javascript analyzer.		contact CAST Technical Support
HTML5-010	Warning	Post analyses	Objects corresponding to jsp files could be counted twice because they were not set external in JEE project.		contact CAST Technical Support
HTML5-011	Warning	Post analyses	Objects corresponding to asp files could be counted twice because they were not set external in DOTNET project.		contact CAST Technical Support
HTML5-012	Warning	Post analyses	Objects corresponding to aspx files could be counted twice because they were not set external in DOTNET project.		contact CAST Technical Support
HTML5-013	Warning	Post analyses	Objects corresponding to htc files could be counted twice because they were not set external in DOTNET project.		contact CAST Technical Support

Message ID	Message Type	Logged during	Impact	Remediation	Action
EXTDOTNET-001	Warning	Analysis	An ASP DOTNET operation has not been created correctly.		Contact CAST Technical Support
EXTDOTNET-002	Warning	Analysis	A file has not been opened/analyzed.	See if you have access rights to this file.	

End of comment and File skipped messages for .js files

You may find that the analysis log contains the following messages for **.js** files:

```
Warning MODULMSG ; Job execution end of comment '\*\/' not found 0 ; 0 0 0 [Module name] 0 0
Warning MODULMSG ; Job execution File skipped : K:\CAST\Ref\auto-refi2-staticresources\testing\inflow.js
```

These messages occur when the underlying Universal Analyzer raises a warning whenever it encounters what it considers a syntax error. The "**End of comment**" message is logged and then a following message is logged stating that the file has been "**skipped**". These warnings should be **ignored** as they have **no impact**: the HTML5/JavaScript extension will analyze the file and raise the following message in the analysis log:

```
Information MODULMSG ; Job execution [com.castsoftware.html5] Light parsing of file K:  
\CAST\CASTMS\Deploy\Test_source\Ref\auto-refi2-staticresources\testing\inflow.js
```