Import an Application managed with CAST Management Studio into AIP Console

⚠ This information is valid only for **1.18.x** releases of AIP Console. If you are not yet using **1.18.x**, please **upgrade** before importing your Applications.

ⓘ **Summary:** this page describes how to import existing Applications (i.e Applications that are being managed using the **CAST Management Studio**) into AIP Console. All the existing data for the Applications (configuration, versions, analysis results, snapshots) will be preserved.

Prerequisites and best practices

✓	**One Application per combined schema installation**	You can import multiple applications into AIP Console, however, there must be only **one Application** (the one to be imported) declared **in any existing Management schema**. In other words, the Applications to be imported must have dedicated combined schemas (Management, Analysis and Dashboard schemas).
✓	**Application /schema version**	The application you want to import into AIP Console must have the same AIP schema version number as the AIP Node you are importing into. ⓘ Note that in **AIP Console 1.25**, this limitation has been removed: you can import an Application with an AIP schema version number that is older than the AIP Node you are importing into. You can then upgrade this application to match the AIP Node AIP version number.
✓	**Delivery folder**	**Format** AIP Console requires a different format of **Delivery folder** (since version 1.13) than the format that is used in legacy deployments of CAST AIP. Therefore, before an application can be imported into AIP Console, the files stored in the existing /source Delivery folder relative to the Applications to be imported must be **transformed** into the format that the AIP Console can understand. This process is explained below in **Step 1**. CAST's recommended **best practice** if you want to transform multiple applications in **Step 1** is to only specify in **config.json** the applications that will then be imported in **Step 3** into the **same AIP Node**.
		Location Each AIP Node managed by AIP Console requires a Delivery folder in which to store Application data. CAST's recommended best practice is to use a **common/shared Delivery folder for all AIP Nodes that are managed in AIP Console**. This Delivery folder can be located on a network share. See **Configure AIP Node storage folder locations - optional - v. 1.x**.
✓	**Deployment folder**	The Deployment folder configured for an application you would like to import does not need to be identical in CAST Management Studio and on the AIP Node. It can be identical if required, but it is not mandatory.

✓	**CAST Storage Service / PostgreSQL instance**	In an AIP Console environment, each AIP Node (a server where CAST AIP core is installed and which is managed through AIP Console) can only be associated with **one single CAST Storage Service/PostgreSQL instance**, therefore the CAST AIP schemas (Management, Analysis and Dashboard schemas) associated with the applications you want to transform and import into AIP Console **must already be present on the CAST Storage Service/PostgreSQL instance associated to the AIP Node in to which the applications will be imported**.
		
		You can check which CAST Storage Service/PostgreSQL instance is associated to your target AIP Node using the `<installation>\AipNode\data\aip-node-app.properties` file and checking the `database.server.name` option:
		
		`# ============================`
		`# CSS Server parameters`
		`# ----------------------------`
		`database.server.name=192.168.200.104:2282`
		
		This requirement **may not** already be the case. For example:
		
		• You would like to import 5 Applications into **AIP Node "1"**
		• **AIP Node "1"** is associated with **CAST Storage Service/PostgreSQL instance "A"**
		• Of the Applications that will be imported to **AIP Node "1"**, three applications use CAST AIP schemas on CAST Storage Service/PostgreSQL **instance "A"** and two applications use CAST AIP schemas on CAST Storage Service/PostgreSQL **instance "B"**.
		
		In this scenario, you will need to perform the following:
		
		• **backup** the Management, Analysis and Dashboard schemas that are stored on CAST Storage Service/PostgreSQL **instance "B"**
		• **restore** the schemas on CAST Storage Service/PostgreSQL **instance "A"**
		
		See **Maintenance activities for CAST Storage Service and PostgreSQL** for more information about backing up and restoring schemas.

✓	**Access to resources**	The server on which you are running the **migration.bat** transformation tool explained in **Step 1**, must have access to: • the **existing Delivery folder** configured in the CAST Management Studio • the **Delivery folder configured for the target AIP Node** in to which the application(s) will be imported • the **CAST Storage Service/PostgreSQL instance** hosting the schemas for the Applications to transform
✓	**Configure required third-party source code**	If the source code of the Application to import uses **third-party source code** (such as .NET Assemblies, or Maven repos), ensure you configure AIP Console to use the third-party code from the SAME location as defined in the CAST Management Studio. See the following pages: • **Configuring source code delivery for .NET** • **Configuring source code delivery for Maven**
✓	**Enable the path matching feature**	Ensure you enable the **Path matching option for source code delivery rescan**. This will help to ensure that the source code you upload for the base-line analysis/snapshot after importing the Application is consistent.

Definition: for the rest of this page, the term "**AIP Node"** will refer to the AIP Node installed on the server-host where the Applications to be imported have previously been managed with the CAST Management Studio.

Step 1 - Copy and transform the Applications in the existing Delivery folder to the new Delivery folder

CAST AIP Console uses a different format of **Delivery folder** (since version 1.13) than the format that is used in legacy deployments of CAST AIP. Therefore, before an application can be imported into AIP Console, the files stored in the existing Delivery folder relative to the Applications to be imported must be transformed into the format that the AIP Console can understand.

This step is typically designed to transform, in one go, all existing/source Delivery folder data for all specified applications that need to be imported into AIP Console:

- If any Applications to be imported into AIP Console have their data stored in different existing/source Delivery folders, this step should be repeated for each existing/source Delivery folder configured in CAST Management Studio.
- If the AIP Nodes managed by AIP Console each use their own dedicated Delivery folder rather than a shared/common Delivery folder (not recommended) this step should be repeated for each dedicated Delivery folder configured in each AIP Node

This step therefore serves two purposes:

- it will first **copy** the data for the selected Applications stored in the **existing Delivery folder configured in CAST Management Studio** into the new Delivery folder configured for the target AIP Nodes.
- it will then **transform** these files into a structure that AIP Console can understand
- the **existing** Delivery folder remains unchanged

Where should I action this step?

The server on which you are running the this step, must have access to:

- the **existing Delivery folder** configured in the CAST Management Studio
- the **new Delivery folder** configured for the target AIP Node in to which the application(s) will be imported
- the **CAST Storage Service/PostgreSQL instance** hosting the schemas for the Applications to transform

The files/scripts described in **Step 1** are available as a **downloadable extension**. The **latest release of this extension** is delivered with each major or minor release of AIP Console on EACH AIP Node. If in doubt, CAST recommends downloading and using the tool to ensure that you have the latest release of the files/scripts. The instructions below assume you are using the files/scripts provided on each AIP Node - if you are using the files/scripts provided as an extension, simply change the path to the file that is specified.

Modify config.json

Locate `%PROGRAMFILES%\CAST\AipConsole\AipNode\bin\dmt-migration-tool\config.json` on your **AIP Node** or use the file provided in the **downloadable extension**. This file will be provided out-of-the-box as follows:

```
[
  {
    "delivery_folder_location": "C:\\CASTMS\\Delivery",
    "destination_delivery_folder_location": "",
    "log_file_location": "",
    "css_servers": [
      {
        "dbname": "postgres",
        "host": "localhost",
        "port": "2282",
        "username": "operator",
        "password": "CRYPTED2:90B1A6EC1618661401B724DB5AC34595"
      }
    ],
    "applications": []
  }
]
```

Edit this file with a text editor and enter the information as described below:

`delivery_folder_location`	Enter the location of the **Delivery folder defined in the CAST Management Studio** for the Application (s) you want to transform. See note below about the **required path syntax** for the config.json file.
`destination_delivery_folder_location`	Enter the location of the **Delivery folder defined for the AIP Node.** This can be an **empty folder** if you are importing into an AIP Node that has been recently added to AIP Console. For example: `"destination_delivery_folder_location": "S:\\Delivery",` The transformation tool will: • copy the Applications specified in `applications` into the `destination_delivery_folder_location` • the transformation of these Applications will then occur in `destination_delivery_folder_location` • `delivery_folder_location` will remain unchanged for roll back purposes See note below about the **required path syntax** for the config.json file.
`log_file_location`	Enter a full path to a folder that will be used to store a log file called **delivery.folder.migration.tool.logrecord** that will contain log for the transformation process, for example: `"log_file_location": "D:\\temp",` See note below about the **required path syntax** for the config.json file.
`css_servers`	Check that the `css_servers` details correspond to the **CAST Storage Server/PostgreSQL instance** which is used to host the CAST AIP schemas for the Applications you are transforming.

	The encrypted key in the `password` parameter corresponds to the default password for the **operator user.** If your **CAST Storage Server/PostgreSQL instance** uses different credentials, you can generate a new **encryption key** for the **password** entry, please see **Using the aip-encryption-tool to encrypt credentials**.
`applications`	Enter the names of the applications you want to include in the transformation process, separated by a **comma** and **space** if there is more than one. All Applications must be located in the same `delivery_folder_location`. CAST's recommended **best practice** if you want to transform multiple applications is to **only** specify those applications that will be imported (in **Step 3**) into the **same AIP Node**. `"applications": ["app1", "app2", "app3"]`

All paths specified in **config.json** must:

- use **either double backslashes or single forward slashes**. Single backslashes will cause an error when migration.bat is run.
- be **valid (i.e. accessible)** from the server on which you run migration.bat (see below).

Paths can refer to network shares or local paths.

Run migration.bat

Run `%PROGRAMFILES%\CAST\AipConsole\AipNode\bin\dmt-migration-tool\migration.bat` on your **AIP Node**. This will copy the chosen Applications from the existing Delivery folder configured in CAST Management Studio into the AIP Node Delivery folder and transform the structure into a format that AIP Console **1.13.x** requires.

You can check the `log_file_location` to ensure the transformation and move process functioned correctly.

Step 2 - Apply configuration changes

Restart the AIP Node and AIP Console to ensure all changes are taken into account:

- If the AIP Node / AIP Console are installed as a **Windows Service**, restart the services
- If the AIP Node / AIP Console are running only using the **batch files**, close the CMD windows to stop the processes, then restart them using the following files:

```
AIP Node: %PROGRAMFILES%\CAST\AipConsole\AipNode\tools\runAipNode.bat
AIP Console: %PROGRAMFILES%\CAST\AipConsole\AipConsole\tools\runAIPConsole.bat
```

Step 3 - Launch the import batch file to import the application(s)

This step performs the import of the applications so that AIP Console is made aware that the applications are now being managed in the specific AIP Node. This process can only be performed **one Application at a time** and should be actioned **on the server hosting the AIP Node** into which you want to import the Application. If you need to import more than one Application, you will need to **repeat** the process described below for each Application, for example using a batch script.

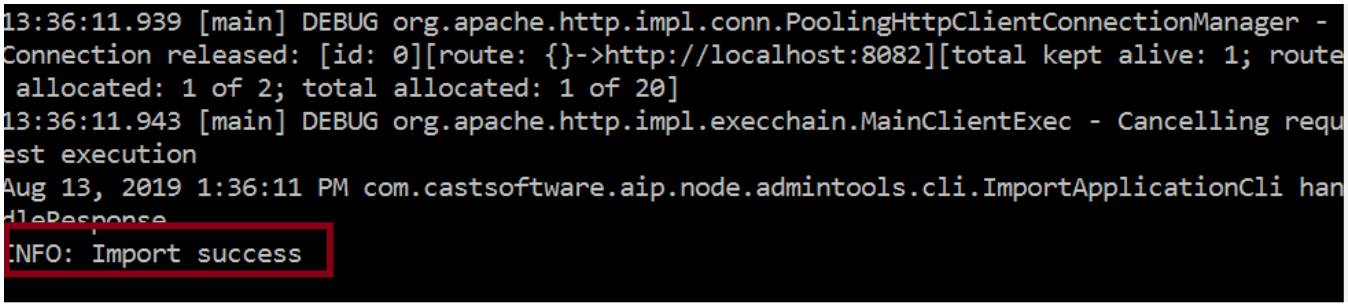
To do so, locate the following batch file:

```
%PROGRAMFILES%\CAST\AipConsole\AipNode\admin\importApplication.bat
```

Run this batch from a **CMD** window using the following arguments - see **Available CLI arguments**:

```
importApplication.bat -apiHost "<http://localhost:8082>" -appName "<applicationName>" -mngtSchemaName
"<mngt_schemaName>_mngt" -token "<AIP Node token>"
```

A successful import will result in the following message displayed in the CMD console:



Login to the AIP Console - the imported application will now be listed with any others, all versions and snapshots will be available. If not, then please refresh the AIP Console page. Repeat the process for any additional applications you need to import.

Available CLI arguments

All argument data must be enclosed in quote marks: "....".

Arguments	Default value	Required?	Description
-appName	-	✓	Specifies the **Application** to be imported. This is case-sensitive and must match the name of the Application as defined in the CAST Management Studio.
-apiHost	http://localhost:8082	✗	Specifies the **server/port URL** for the AIP Node. If omitted, the default value is used.
-mngtSchemaName	-appName + _mngt	✗	Specifies the name of the **Management Service schema** in which the Application to be imported is currently managed. If omitted, the default value will be created using a combination of the value of `-appName` + `_mngt`. For example if `-appName` = MEUDON, then `-mngtSchemaName` = MEUDON_MNGT
-token	-	✓	Specifies the **access token** for the AIP Node - this is generated during the installation of the AIP Node package - see **AIP Node package - back-end installation**). If you do not have a record of it, it can be found in the `<installation>\AipNode\data\aip-node-app.properties` file on the line `spring.liquibase.parameters.baseToken`.

Step 4 - Analyze and generate a snapshot of the new version of the application

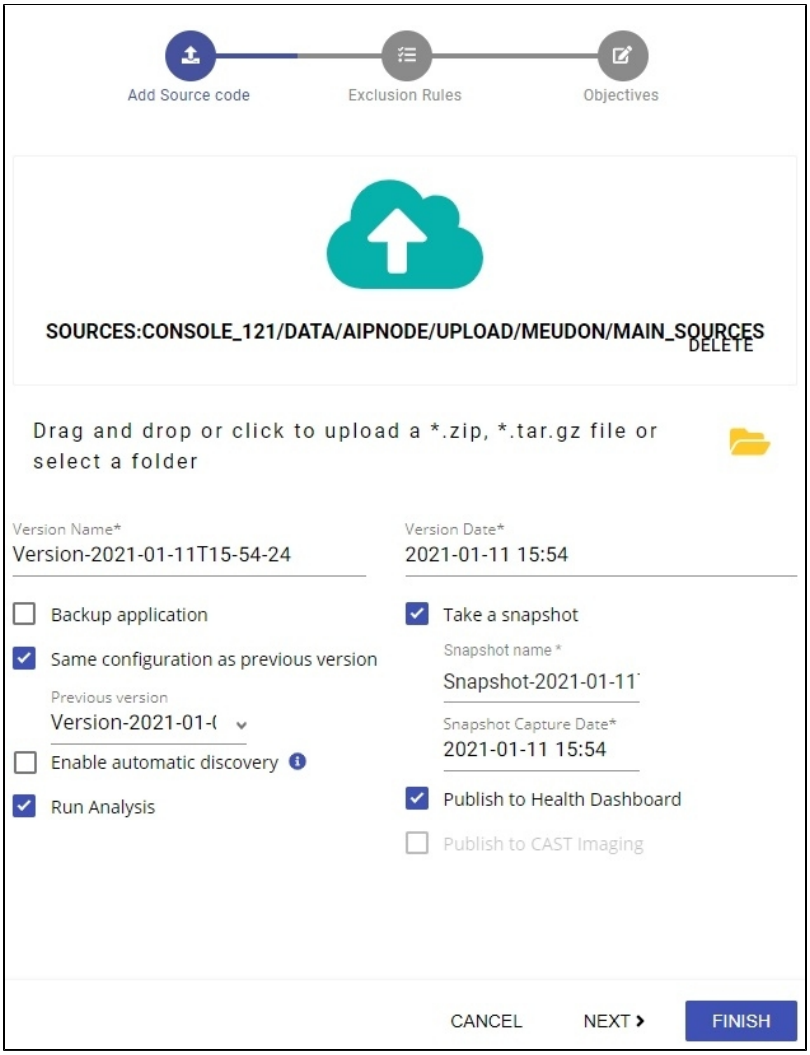
If the stability of the results is an absolute must, CAST highly recommends running a consistency snapshot. There should be no differences in analysis results as long as the same underlying release of CAST AIP is being used to perform the analysis/snapshot and the consistency snapshot will validate this.

AIP Console requires that source code is delivered in a **ZIP file** or in a **source code folder.** To structure it properly, follow the instructions in the **section below.** Also as stated in the **prerequisites**, ensure that you enable the **path matching feature** to help identify missing or altered source code during the base-line code analysis following the import of the Application - see **Path matching option for source code delivery rescan**.

Once the ZIP is created or the code added to the source code folder, **add a new version** for your imported Application and ensure you:

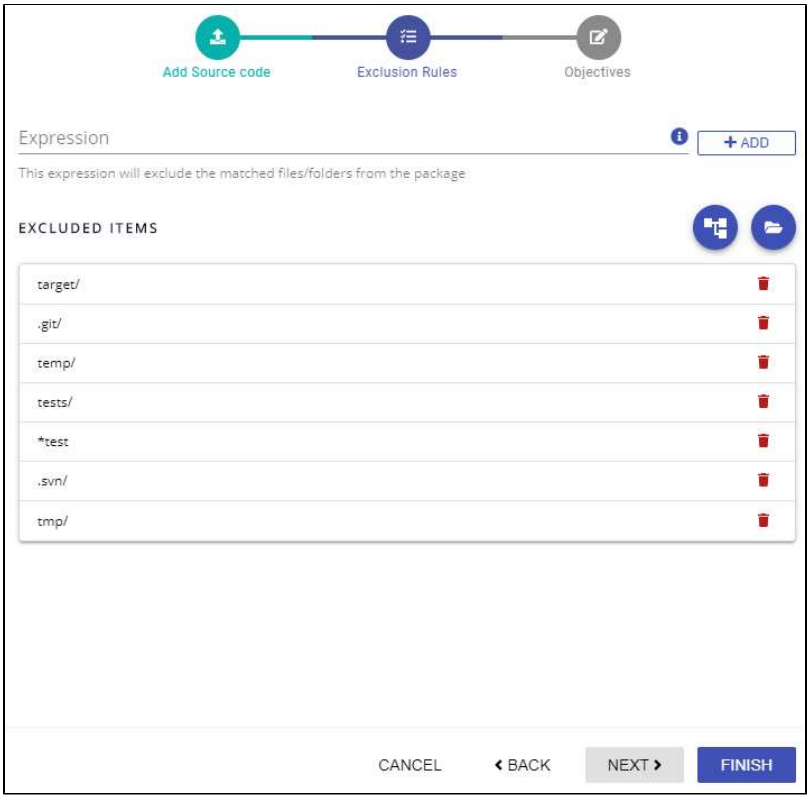
- **activate** the **Same configuration as previous version** and **Run analysis/Take a snapshot** options (see **Standard rescan - add a new Version - deliver code - generate snapshot** for more information)
- **deactivate** the **Enable automatic discovery** option (see **Standard rescan - add a new Version - deliver code - generate snapshot** for more information)

Click to enlarge



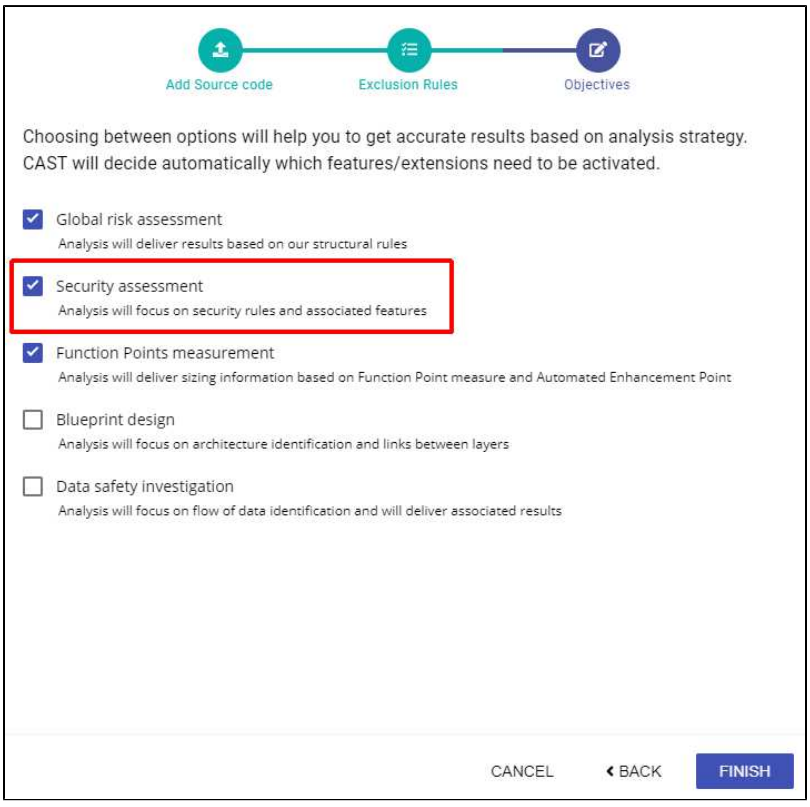
Click **NEXT** to define any folder exclusions (usually this is not necessary):

Click to enlarge



Click **NEXT** to define any **Objectives**. If User Input Security options were **enabled** in the imported Application in CAST Management Studio, ensure that you enable the **Security assessment** option (this will enable User Input Security):

Click to enlarge



Click **FINISH** to generate the consistency snapshot.

Code delivery structure

When the original source code is configured in **one single parent folder** with sub-folders corresponding to each Delivery Manager Tool package, for example:

```
R:/Source/AppName
                |----DMT------|
                        |---- Analyzed (source code)
                        |---- DB
                        |-----helping folder1---(with subfolders)
                        |-----helping folder2---(with subfolders)
```

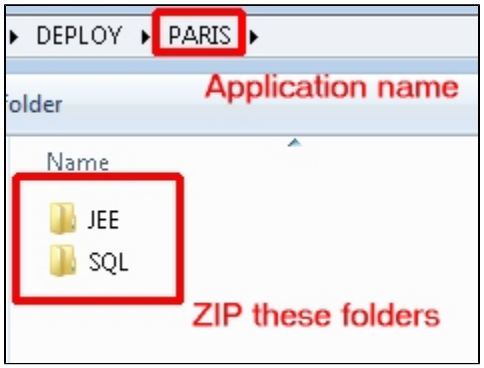
Then you should copy all folders under "DMT" and paste into a new folder as shown below. Finally:

- if using a ZIP file, zip the folders in the "temp" folder - but do not ZIP the "temp" folder, nor create any intermediary folders. The **Path matching option for source code delivery rescan** will then detect the differences in the uploaded ZIP file.
- or upload the contents of the temp folder into your source code delivery folder.

```
D:/temp
        |-----Analyzed (source code)
        |-----DB
        |-----helping folder1--(with subfolders)
        |-----helping folder2--(with subfolders)
```

Alternative method - use the DEPLOY/<application_name> folder

Alternatively, if your **DEPLOY/<application_name>** folder still contains the **deployed source code** (i.e. you have not manually removed it following the most recent snapshot), you can re-use contents of this folder and use it for the new version you are delivering via the AIP Console:



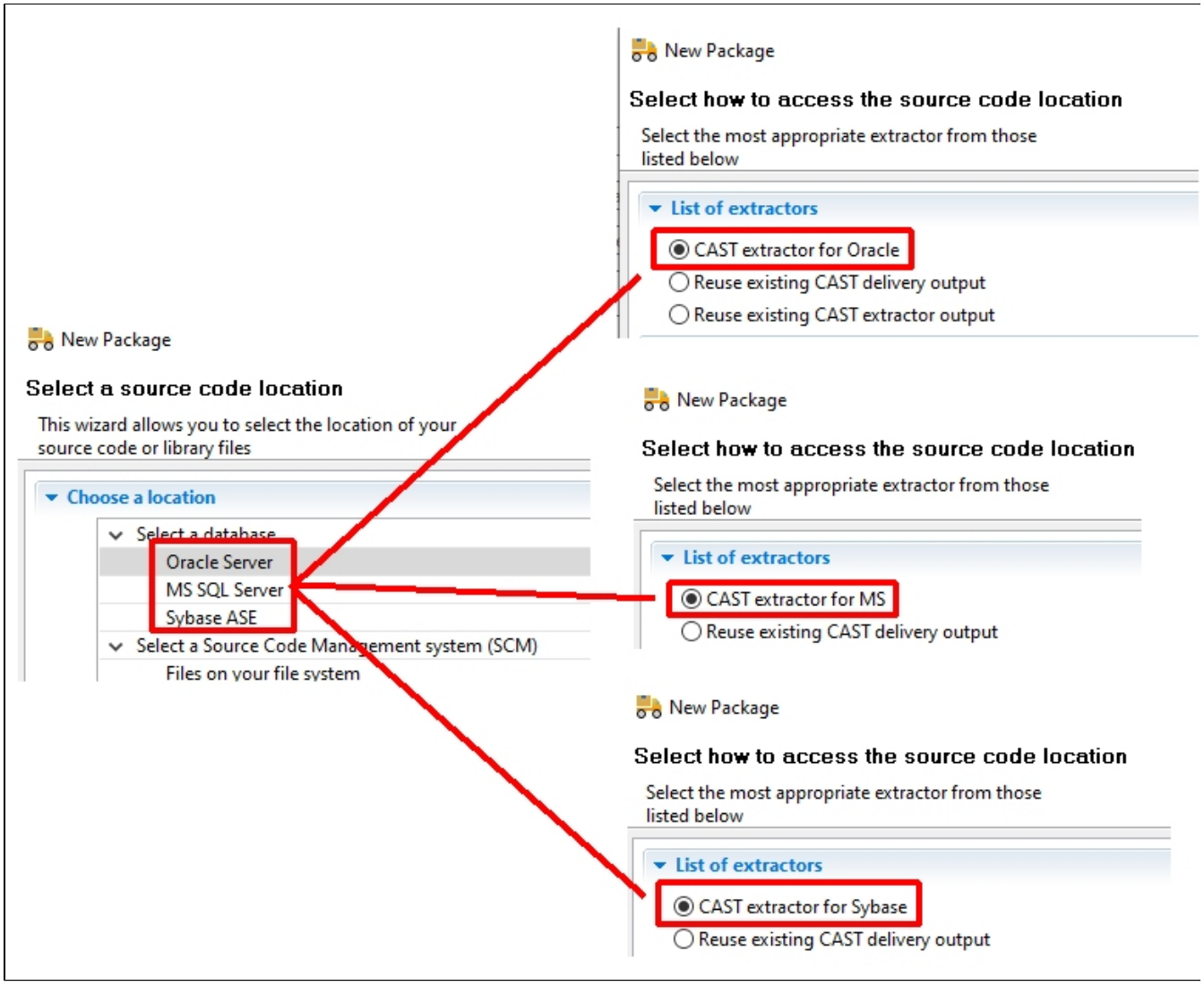
When using this method, there should be no differences in the source code paths that AIP Console already has for the imported Application version, and therefore the **Path matching option for source code delivery rescan** should not detect any differences.

Note about "live" SQL database packages created in the CAST Delivery Manager Tool

If the imported Application contains source code packages for SQL databases such as Oracle Server, Microsoft SQL or Sybase ASE created in the CAST Delivery Manager Tool (DMT) using the "live" database extractor built into the DMT then you will need to include the **uaxdirectory/uax files** or the **.castextraction file** in the source code ZIP file or the source code delivery folder. You can obtain these files either by running the standalone **CAST Database Extractor** again on your target database, or by locating the files in the **DEPLOY/<application_name>** folder. When code is delivered for the new version, CAST AIP Console will detect that the "live" SQL database packages are missing (assuming that the **Path matching option for source code delivery rescan** is enabled) and display the missing packages dialog. When this occurs, you must delete any paths that correspond to these "live" SQL database packages.

The "live" SQL database packages would have been created in the CAST Delivery Manager Tool using the following options:

Click to enlarge



Troubleshooting

Missing Analysis Units

If you configured "**Folder on your file system**" packages with subfolder selections in the CAST Delivery Manager Tool, you will loose your Analysis Units after import into AIP Console and the generation of a consistency snapshot. This case is not supported for now.

Add Version in fails during copying and attaching packages step

When delivering your new version of source code, the process may fail during the copying and attaching packages step. The corresponding error in the log file is as follows:

```
Error: Child task on exception
  The following command is too long to be properly processed by the operating system: '<very long path>'
Return value: 1000
ERR: YYYY-MM-DD HH:mm:SS: Exception occurred while running the job
com.castsoftware.aip.node.jobs.JobInterruptedException: Failed to execute DMT command. Exit value: 1000
```

This occurs when the path used by the command exceeds 8190 characters. One possible workaround is to move the installation of AIP Core on the affected to AIP Node to a shorter path, for example C:\AIP - this will reduce the number of characters in the command line. Then update the following file on the AIP Node to match the new location of AIP Core and restart the AIP Node:

```
%PROGRAMDATA%\CAST\AipConsole\AipNode\aip-node-app.properties
```

For example

```
# ============================
# CAST AIP Installation folder
# ----------------------------
cast.ms.home=C:/AIP
```

Technical notes

Additional dependent files

The vast majority of Application analyses employ various dependent files to ensure that the analysis runs smoothly or to provide additional configuration, for example:

- JAR/archive/XML/properties/classpath files/folder etc.
- DLM (Dynamic Link Manager) configuration files
- Architecture Checker models
- etc.

These additional files will automatically be taken into account when you run an analysis or generate a snapshot for an Application that you have imported into the AIP Console - however, these files will not be visible in the AIP Console configuration screens. If you do need to see these files in the AIP Console, they will need to be uploaded and added manually.

Existing Reference Pattern configurations

Any **Reference Pattern configurations** that exist in CAST Management Studio will not be made available in the **Application - Config - Reference Finder** screen in AIP Console (available from 1.22). You should instead continue to use CAST Management Studio to manage these existing Reference Patterns.

Information about storage folder locations (Delivery, Deploy, LISA, LTSA and Log)

Background information:

- For Applications that were originally defined in CAST Management Studio, the associated **Delivery/Deploy/LISA/LTSA/Log** folders would have been defined in CAST Management Studio during the initial startup. These folder paths are stored in the **CMS_PREF_SOURCES** table in the application's **_MNGT** schema and the locations can be modified using the **Preferences > Platform Settings option in CAST Management Studio** or using the **castglobalsettings.ini** file located at the root of the AIP Core installation.
- For Applications defined in AIP Console from scratch, the **CMS_PREF_SOURCES** table in the Application's **_MNGT** schema is also used to store the location of the associated **Delivery/Deploy/LISA/LTSA/Log** folders. The default locations of these folders are described in **Configure AIP Node storage folder locations - optional - v. 1.x** and are configured using the **%PROGRAMDATA%\CAST\AipConsole\AipNode\aip-node-app.properties** on the AIP Node or using the **castglobalsettings.ini** file located at the root of the AIP Core installation.

When an application that was originally created and managed in CAST Management Studio is subsequently imported into AIP Console, the paths defined in the **CMS_PREF_SOURCES** table in the application's **_MNGT** schema will be **re-used** for the associated **Deploy/LISA/LTSA/Log** folders, therefore these locations will be re-used for all subsequent analyses - this is the expected behaviour. The exception to this rule is where any of the values in CMS_PREF_SOURCES are **empty**, in which case, the value in the **castglobalsettings.ini** file located at the root of the AIP Core installation is used (if there are no custom locations defined in the .ini file, then the default locations defined in the .ini file are used). With regard to the **Delivery** folder: the files are physically moved to the location that is defined during the import process.

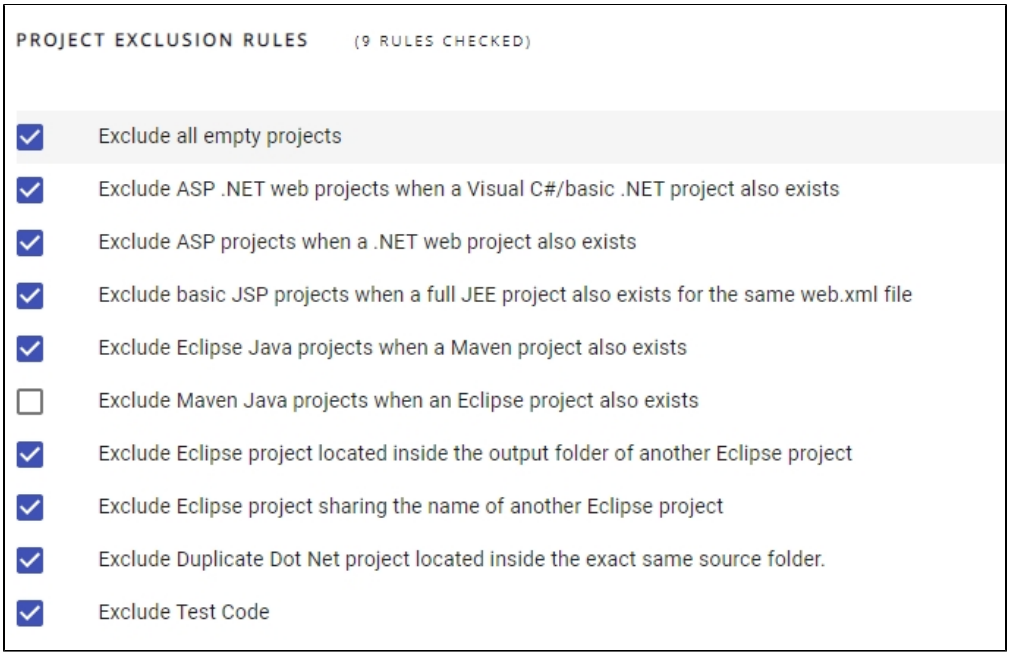
Post import of your Application into AIP Console, you may therefore find that folders defined in CAST Management Studio may be being used during analyses. This is as expected, however, you may wish to review the paths used and update them if necessary. You can find more information in **Configure AIP Node storage folder locations - optional - v. 1.x**, however, whilst it is technically possible to change the **folder** locations once an Application has been imported, please be aware of the following if you choose to do so:

- Changing the **Delivery** folder location is NOT supported and will break your existing Application - you can define this location during the import process
- Changing the **Deployment** folder will cause added/deleted objects to be recorded in subsequent analyses/snapshots
- **Logs** and **LISA/LTSA** folders can be changed with no impact (existing items will remain in the old location)

Project Exclusion Rules

Applicable when importing Applications into AIP Console 1.26

In AIP Console 1.26, **Project Exclusion Rules** have been introduced - you can see more about this in **Standard onboarding - add a new Version - deliver code - generate snapshot**:



When importing into AIP Console from CAST Management Studio, the following applies:

- When the Version contains only one file based package, Project Exclusion Rules from that specific package will be applied to all packages.
- When the Version contains multiple file based packages:
  - the first Package whose Project Exclusion rules differ from the default options (see image above) will be used and applied to all packages
  - if no file based package is present then the default options (see image above) will be used and applied to all packages