

Delivery folder split

- [Introduction](#)
 - [Why is this split required?](#)
 - [New delivery folder structure](#)
 - [When is the split actioned?](#)
 - [Situations that require manual migration before upgrading the Node package](#)
- [Manual Delivery folder split process](#)
 - [Step 1 - Ensure Console and all Nodes are stopped](#)
 - [Step 2 - Modify config.json](#)
 - [Step 3 - Run migration.bat](#)
- [What next?](#)



Summary: This step describes the process of splitting the Delivery folder when upgrading Console/Node packages to v. **1.13 from an older release.**



If you are already using Console/Node v. 1.13, then **please skip this step.**

Introduction

In Console **1.13** the format of the **Delivery folder** assigned to each Node was changed so that each Application now has its own **dedicated Delivery folder within the parent Delivery folder** (previously, the parent Delivery folder was global to all Applications).

Why is this split required?

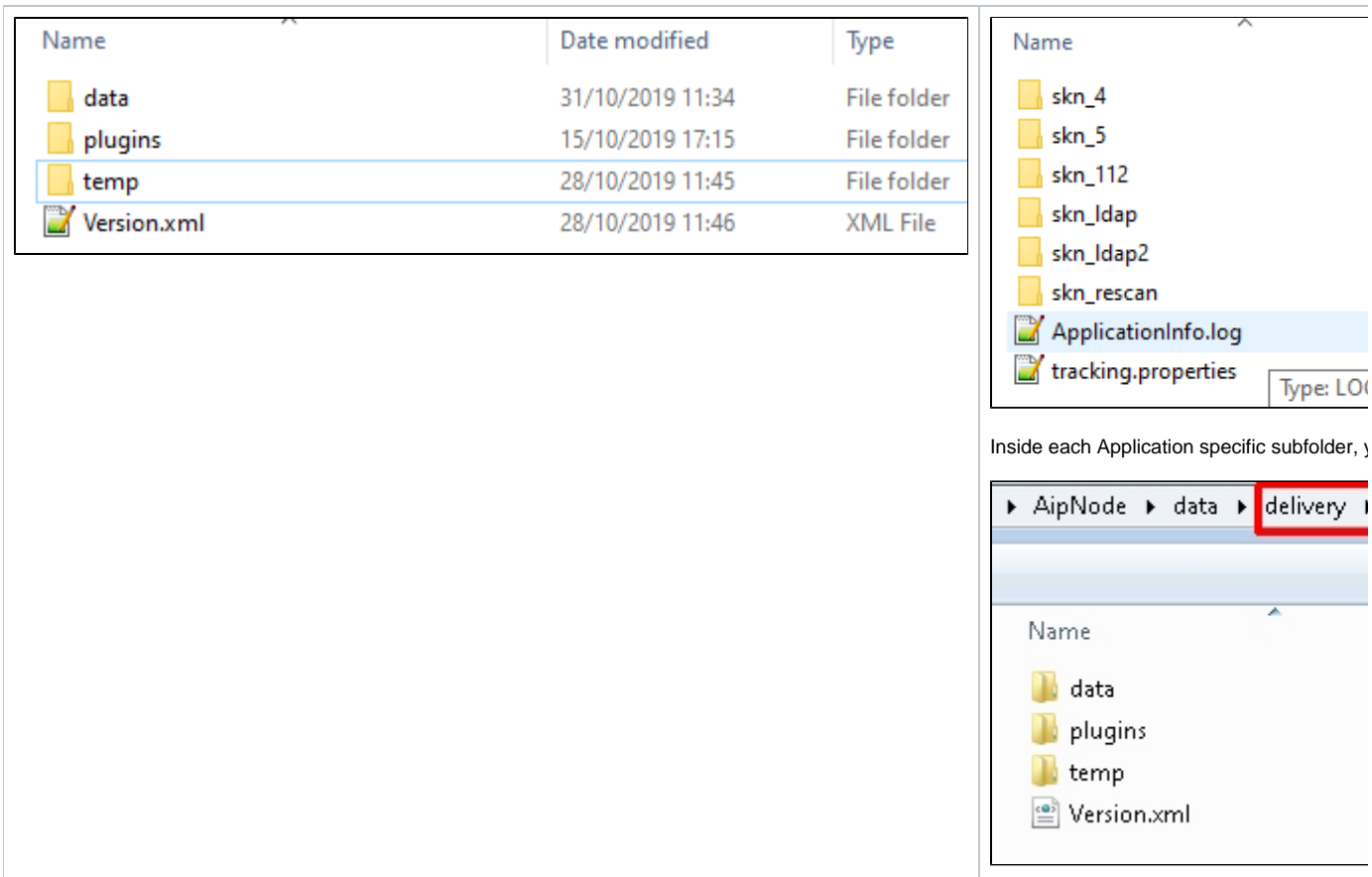
- To allow each Application to have dedicated (DMT) plugins and to therefore avoid a "collision" of plugins across the Applications.
- Possibility to have multiple versions of plugins running on different Applications
- Easier to backup and restore the Delivery folder for one single Application

New delivery folder structure

The designated Delivery folder location will become the "parent" of all the "application delivery folders":

Before the split is actioned

Post split - each folder corresponds to



When is the split actioned?

The process of "splitting" the Delivery folder is actioned automatically during an **upgrade of the Console and Node packages to 1.13** as follows:

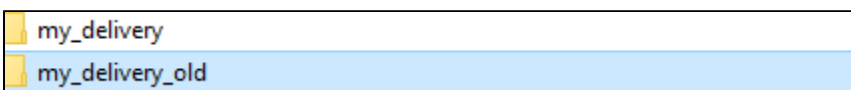
- When the **Node installer** detects that the Node's assigned Delivery folder is using the "old" structure, the split tool will be launched automatically:

Click to enlarge

```

Administrator: DMT Migration Tool
2019-12-18 13:48:37.099 - INFO --- Connection with info :postgres:localhost:2282:operator
2019-12-18 13:48:37.192 - INFO --- connection is generated with :org.postgresql.jdbc4.Jdbc4Connection@2f0a87b3
2019-12-18 13:48:37.192 - INFO --- Started schema update .
2019-12-18 13:48:37.190 - INFO --- Schema updated with no of rows :1
2019-12-18 13:48:37.190 - INFO --- Completed DB update
2019-12-18 13:48:37.200 - INFO --- START Tasks for Application : ApplicationInfoBean(Id=32, name=skn_4, uuid=acde064d-63
4b-46a0-9bf4-80f47cf8ab76)
2019-12-18 13:48:37.200 - INFO --- TASK : start copying application folder
2019-12-18 13:48:37.207 - INFO --- TASK : start copying other directories
2019-12-18 13:48:38.170 - INFO --- Started DB update :
2019-12-18 13:48:38.171 - INFO --- Connection with info :postgres:localhost:2282:operator
2019-12-18 13:48:38.251 - INFO --- connection is generated with :org.postgresql.jdbc4.Jdbc4Connection@319b92f3
2019-12-18 13:48:38.252 - INFO --- Started schema update .
2019-12-18 13:48:38.259 - INFO --- Schema updated with no of rows :1
2019-12-18 13:48:38.259 - INFO --- Completed DB update
2019-12-18 13:48:38.259 - INFO --- START Tasks for Application : ApplicationInfoBean(Id=37, name=skn_5, uuid=cd4819bc-01
35-448d-b686-52f9d6f8607d)
2019-12-18 13:48:38.260 - INFO --- TASK : start copying application folder
2019-12-18 13:48:38.267 - INFO --- TASK : start copying other directories
2019-12-18 13:48:39.225 - INFO --- Started DB update :
2019-12-18 13:48:39.226 - INFO --- Connection with info :postgres:localhost:2282:operator
2019-12-18 13:48:39.313 - INFO --- connection is generated with :org.postgresql.jdbc4.Jdbc4Connection@fcd6521
2019-12-18 13:48:39.314 - INFO --- Started schema update .
2019-12-18 13:48:39.319 - INFO --- Schema updated with no of rows :1
2019-12-18 13:48:39.319 - INFO --- Completed DB update
2019-12-18 13:48:39.320 - INFO --- START Tasks for Application : ApplicationInfoBean(Id=39, name=skn_ldap2, uuid=e7708b9
2-d666-4e01-99ec-a24667ea9cf8)
2019-12-18 13:48:39.321 - INFO --- TASK : start copying application folder
2019-12-18 13:48:39.340 - INFO --- TASK : start copying other directories
  
```

- The installation continues after the split tool has successfully executed. If an error occurs, you will be notified.
- Post upgrade, you will find the same Delivery folder with the new structure and a backup of the original one:



- Existing Applications will continue to work in Console as before.

Situations that require manual migration before upgrading the Node package

You will need to manually execute the split tool **before upgrading the Node package** in the following situations:

1. Where you have **multiple Nodes** with **multiple CAST Storage Services/PostgreSQL instances**, but **only one shared Delivery folder**
2. Where you are using **a mapped drive** for your Delivery folder such as R:\

Manual Delivery folder split process

The Delivery folder split process requires a dedicated tool - **Delivery Folder Migration Tool - 1.0**. This tool can be found on **CAST Extend**. It can be run on any server which has access to the Delivery folder and the associated CAST Storage Service/PostgreSQL instances.

Step 1 - Ensure Console and all Nodes are stopped

Before you start, ensure that Console and all Nodes are not running by stopping the relevant services.


Step 2 - Modify config.json


Locate `config.json` 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	Leave this path empty. This ensures that the split is actioned in <code>delivery_folder_location</code> . A backup copy of the existing Delivery folder (with the suffix _old) will be created alongside the existing Delivery folder.
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: <pre>"log_file_location": "D:\\temp",</pre> See note below about the required path syntax for the config.json file.

<p>css_servers</p>	<p>Check that the <code>css_servers</code> details correspond to the CAST Storage Server/PostgreSQL instances which are used to host the AIP schemas for the Applications you are transforming. You can specify more than one CAST Storage Server/PostgreSQL instance if you are using a shared Delivery folder and the Applications in it are managed on different CAST Storage Services. For example:</p> <pre data-bbox="613 279 1481 737"> "css_servers": [{ "dbname": "postgres", "host": "my_postgres1", "port": "2282", "username": "operator", "password": "CRYPTED2:90B1A6EC1618661401B724DB5AC34595" }, { "dbname": "postgres", "host": "my_postgres2", "port": "2282", "username": "operator", "password": "CRYPTED2:90B1A6EC1618661401B724DB5AC34595" }], </pre> <div data-bbox="613 804 1481 940" style="border: 1px solid #ccc; padding: 5px;"> <p> The encrypted key in the <code>password</code> 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 <code>password</code> entry, please see Using the aip-encryption-tool to encrypt credentials.</p> </div>
<p>applications</p>	<p>Leave this field blank. This ensures that all Applications in your Delivery folder are included in the split process.</p>

 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.

For example, to split the Delivery folder located at `Z:\delivery`, with two CAST Storage Service/PostgreSQL instances associated to the Applications in that Delivery folder, use the following configuration:

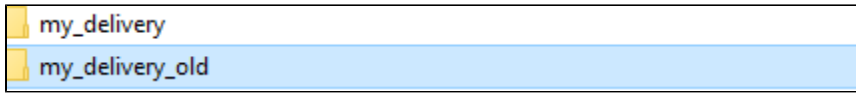
```

[
  {
    "delivery_folder_location": "Z:\\delivery",
    "css_servers": [
      {
        "dbname": "postgres",
        "host": "localhost",
        "port": "2282",
        "username": "operator",
        "password": "CastAIP"
      },
      {
        "dbname": "postgres",
        "host": "other_host",
        "port": "2282",
        "username": "operator",
        "password": "CastAIP"
      }
    ],
    "applications": []
  }
]

```

Step 3 - Run migration.bat

Run `migration.bat` provided in the [downloadable extension](#). This will split the existing Delivery folder into the format required by Console **1.13.x**. In addition, a backup of the existing Delivery folder will be created along side the current Delivery folder for roll back purposes.



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

What next?

Do not restart the Console and Node packages. Now move on to the next step in the upgrade process: [Console upgrade](#).