

2.4.1.1. On-Premise and Factory Tag Naming Conventions

The Ab Initio Factory Migration team will use the following naming convention to determine the most recent Capital One production tag to use when extracting and converting Capital One projects:

<Project Name>_<n_major>.<n_minor>.00

Where:

- **<Project Name>** — The name of a production project.
- **<n_major>** — The major version number; can be an integer from **00** to **99**.
- **<n_minor>** — The minor version number; can be an integer from **00** to **99**.
- **00** — The final **00** in the production tag does not increment and is ignored.

<n_major> and **<n_minor>** both start at **00**, and **<n_minor>** is incremented first. When **<n_minor>** reaches **99**, **<n_major>** is incremented and **<n_minor>** starts again at **00**.

For example, using **ENTPUB** as the **<Project Name>**, and starting with a production tag value of **ENTPUB_01.95.00**, the tag values would increment as follows:

```
ENTPUB_01.95.00
ENTPUB_01.96.00
ENTPUB_01.97.00
ENTPUB_01.98.00
ENTPUB_01.99.00
ENTPUB_02.00.00
ENTPUB_02.01.00
ENTPUB_02.02.00
ENTPUB_02.03.00
ENTPUB_02.04.00
ENTPUB_02.05.00
```

2.4.1.2. Performing EME Merges and Integrations

The Capital One Factory Migration team will use standard Co>Operating System **air** commands to perform EME merges, integrations, and general EME maintenance tasks.

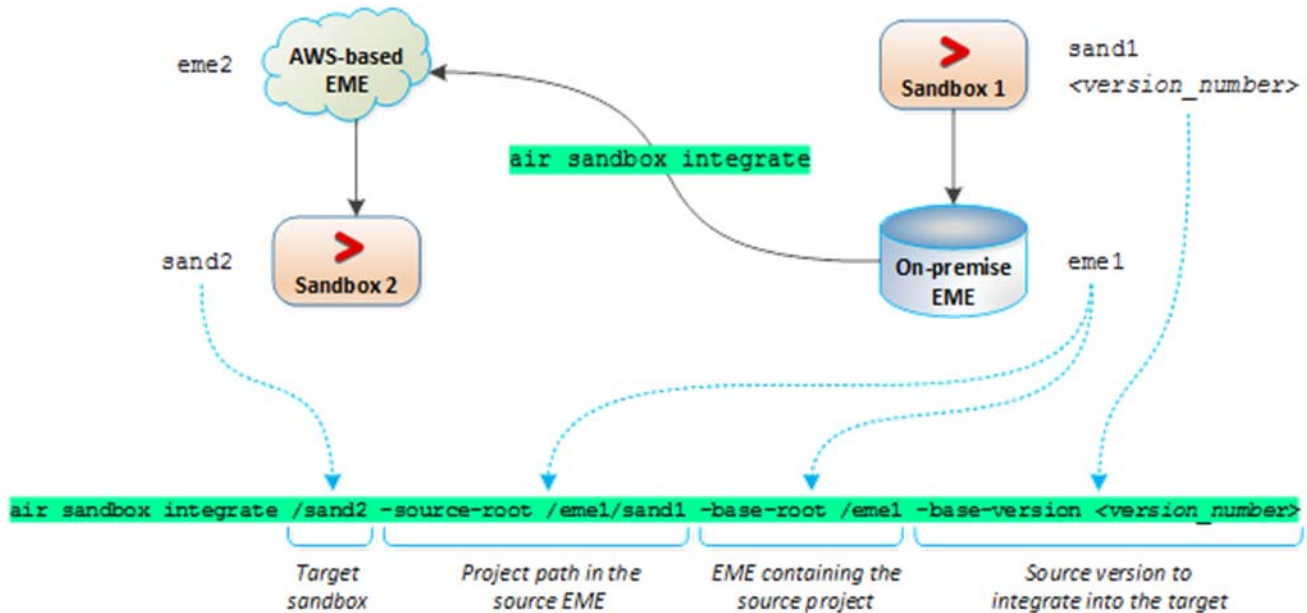
The following table shows some common Co>Operating System **air** commands that are relevant in the context of Factory Migration procedures. For usage examples and complete information about **air** commands, see “Command Reference” in the *Ab Initio Technical Repository* online help.

For an example of how to use the **air sandbox integrate** command to merge sandboxes across two EME repositories – for example, the on-premise EME and the Factory EME on AWS – see Section 2.4.1.3, “Example Integration Scenario.”

Command	Summary
air lock set	Lock modified objects before check-in.
air object changed	Reports on the state of an object in a checked-in sandbox with respect to its associated technical repository project.
air object changed -diff	View detailed changes per object.
air object dataset-associations	List cross-graph dataset dependencies (upstream & downstream).
air object field-lineage	List within-graph and cross-graph field lineage, including transforms (upstream & downstream).
air object rollback	Back out changes to a modified object.
air object status	Reports the state of an object in a sandbox.
air sandbox attach	Attach a converted sandbox to the AWS EME.
air sandbox diff	View object changes in a converted sandbox.
air sandbox import	Check in sandbox changes.
air sandbox info	Displays information about the state of the sandbox or files within it.
air sandbox integrate	Merges the changes from a source project version into a target sandbox using the specified base project version as the original starting point.
air sandbox merge	Merges three files — starting from a common base file — to combine changes made to the source and target files and produce a merged result in the destination file.
air sandbox resolve	Resolves a file left in the unresolved state as the result of a merge or an integrate action.

2.4.1.3. Example Integration Scenario

The following example shows how to use the **air sandbox integrate** command to merge changes in a sandbox in the on-premise EME to a corresponding sandbox in the Factory EME on AWS.



In this example, a source sandbox named `sand1` in an on-premise EME Technical Repository named `eme1` will be integrated into a target sandbox named `sand2` in an AWS-based Technical Repository named `eme2`. In addition, in this example, the version of the project in `sand1` that will be used as the basis for the integration action is `2595607`.

To perform this integration action, you would first check in all changes in both the source and target sandboxes, then use the **air sandbox integrate** command on the *target EME*, as follows:

```
air sandbox integrate /sand2 -source-root /eme1/sand1 -base-root /eme1 -base-version 2595607
```

The parameters used in this example are explained below.

Parameter	Description
<target_sandbox>	The target sandbox for the integration; the source sandbox will be integrated into this target sandbox. In this example, <target_sandbox> is <code>/sand2</code> . This target sandbox must already exist and be attached to the target EME.
-source-root	The full project path in the source EME; in this example, <code>/eme1/sand1</code> .
-base-root	The directory path for the source EME.
-base-version	The version of the source project to use as the basis for the integration.

Note: The **-source-root** and **-base-root** options are currently undocumented, and have been exposed by Ab Initio to Capital One expressly for the purposes of the Capital One Cloud Migration project. Please direct any questions about these options to your Ab Initio project lead.

2.4.1.4. Factory Moderator

Ab Initio recommends that Capital One establishes a **Factory Moderator** role in the Development/Factory team. The Factory Moderator would be responsible for monitoring, managing, and integrating any on-premise code changes during the factory migration.

2.4.1.5. Technical Advisory Group

Ab Initio recommends that Capital One establishes a **Technical Advisory Group** comprised of representatives from each consulting team, the Factory Development team, and Ab Initio developers. The Technical Advisory Group would provide input and make decisions regarding the overall Factory Project and any manual changes to workflows, conversions, SQL syntax that may be required.

2.4.1.6. D4 – Responsibility:

- Prepare an inventory analysis of artifacts per project using the Ab Initio metadata repository. Analysis will be provided to LOBs (D4)
- ECE Development environment should be ready (D4)
- Readiness of the data migration tool (D4)
- The Ab Initio technical repository (EME) will be replicated to ECE development environment (including all objects) (D4)

2.4.1.7. LOB – Responsibility:

- Application owners should freeze the code (LOB)

2.4.1.8. ECE Development – Cloud

D4 – Responsibility