

# 2020-12-02 CM - File Lineage

## Date

02 Dec 2020

## Attendees

- [Chris Grote](#)
- [Mandy Chessell](#)
- [Maryna Strelchuk](#)
- [@Ben Ryan](#)
- [Chris Replogle](#)
- [David Radley](#)
- [@Ernie Ostic](#)
- [Graham Wallis](#)
- [Jakub Moravec](#)
- [@Jerome Nilmeier](#)
- [@William Bittles](#)
- [Lena Woolf](#)
- [Nigel Jones](#)
- [Jim Spohrer](#)

## Goals

- Share use cases and design of Egeria's support for lineage of file processing

## Discussion items

Time	Item	Who	Notes
5mins	Welcome	All	
45 mins	File Lineage	Mandy	<p>Attachment reviewed: <a href="#">lineage.drawio</a></p> <ul style="list-style-type: none"><li>• Unique challenge with files is that the Asset you're interested in could be at various levels:<ul style="list-style-type: none"><li>◦ could be just a file itself (without much concern for the folder it appears within)</li><li>◦ or it could be all of the contents within a folder that is more of interest (eg. if the folder contains something like rolling logs, or a number of files where each contains a daily snapshot of information, etc) (<code>DataFolder</code>)</li><li>◦ important to treat these distinctly for various reasons:<ul style="list-style-type: none"><li>▪ behaving differently in lineage, eg. avoiding showing a fanning-out of many daily snapshot files when it's really one holistic dataset that happens to have daily snapshots within it (in the case of a <code>DataFolder</code>)</li><li>▪ need different connector types to be able to read their contents differently (one directly reads files, the other needs to combine the contents of all of the contents of the files)</li></ul></li><li>◦ but also to ensure that the <code>DataFolder</code> actually extends <code>FileFolder</code> (so that a given instance is actually both, and can therefore be consumed in both ways depending on the user and their needs in accessing it)</li></ul></li><li>• Discussed classification of elements (like files) that may be deleted but still need to be present in the lineage<ul style="list-style-type: none"><li>◦ for example, a landing file that is picked up by a process and moved elsewhere – we need to know about the landing file (even though it's been deleted) to show the lineage all the way back to its ultimate source</li><li>◦ we believe there is a relatively common industry term to represent this concept: <code>Tombstone</code></li><li>◦ however, we believe this could be considered a sensitive trigger term, and therefore are in favour of some other term to classify such elements in lineage: suggesting we go for <code>Memento</code></li></ul></li></ul>
10 mins	AOB	All	--

## Action items