Adding the missing audit trail to R

Magnus Mengelbier
Managing Director

Topics

- Audit Trails
- Alternative approaches
- Tracking Changes
- Generating the Audit Trail
- Conclusion

Audit Trail

- Broader scope for and use of R with Life Sciences
- Scope and use brings expectations on compliance and regulatory
- Concept of Audit Trail established since the 1990's
  - Regulations
  - Directives
  - Guidance & Guidelines

  } Independent record of who did what when
- R as a file-based application … Audit Trail tracks files that are created, updated and deleted

Alternatives

- Redefine R functions
  - Add audit trail methods to functions that write and delete files
  - Excludes any writes and deletions performed using external routines, e.g. c/c++, Python, Java, etc
- File system auditing and monitoring utilities
  - Records actions on the files system
  - Conflicts with IT security processes
  - Difficult to isolate R-related processes
- Repositories
  - Solution external to the R environment
  - Introduces new and complex compliance questions, challenges and issues
- Super log
  - Extend with logging functions to capture additional audit details

Tracking Changes

```
before <- list.files( ".", recursive = TRUE, include.dirs = FALSE)

#### <---- execute R program here

after  <- list.files( ".", recursive = TRUE, include.dirs = FALSE)

created <- after[ ! after %in% before ]
deleted <- before[ ! before %in% after ]
```

	Prior	After
Created		x
Updated	x	x
Deleted	x	

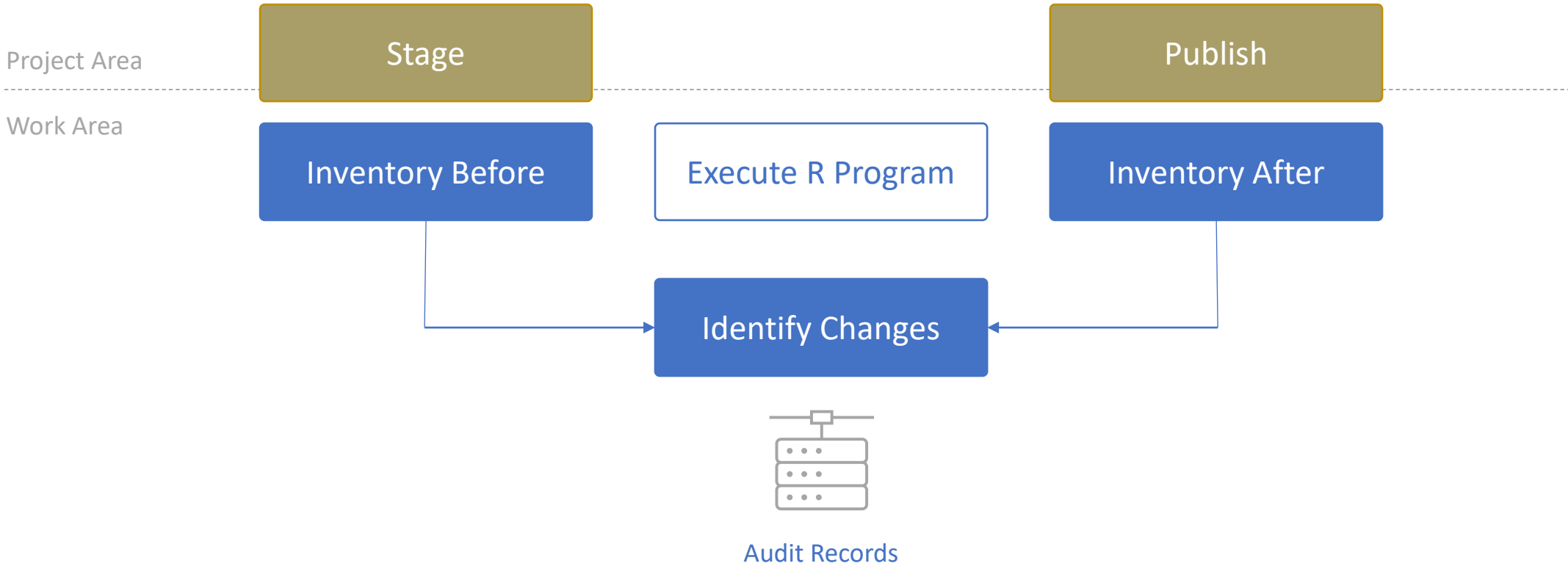
Tracking Changes: Adding Updates

```
sapply( list.files( ".", recursive = TRUE, include.dirs = FALSE),
        function(x) {
           digest::digest( x, file = TRUE, algo = "sha256")
        }, USE.NAMES = TRUE )
```

	Prior	After
Created		x
Updated	{digest}	{digest}
Deleted	x	

- Message Digest / Hash / Checksum is a derived value on the file content
- Different values indicates content is different
- Updated file = Different digest value before and after program execution

Audit Method



Executing R Programs: A Simple Approach Using Annotations

Annotate the program

```
# ---  rexec controls
# rexec-clone /path/to/project/analysis
# rexec-inputs /path/to/project1/analysis/data adsl adae
# rexec-inputs /path/to/project2/analysis/data adsl adae
# rexec-outputs /path/to/project/analysis/data
```

Run the program

```
cxlib_rexec( "program.R" )
```

In Summary

- Audit Trails defined in industry regulations, directives and guidelines
- Different approaches inherits different complexities and challenges
- Audit Method
  - Plain R
  - Minimal set of additional packages
  - Containers and traditional server-based environment architectures
  - Any Operating System
  - Compatible with interactive R environments
- Extendable to other languages

Magnus Mengelbier
magnus.mengelbier@limelogic.com

Limelogic AB
Jungmansgatan 12
SE-211 11 Malmö
Sweden

www.limelogic.com
github.com/limelogic