Contents

- Short introduction of the R peach syntax
  - Minimalistic syntax and optional parameters
- Preparation
  - Installing the rJava and techila packages
- Creating a test Project
- Peach feature demos based on request
  - Snapshotting
  - Job input files
  - ...
Introduction of the R peach syntax

- Simple interface for creating Projects

- Typically the R peach syntax will define:
  - The name of the function that will be called
  - The name of the R-script that will be sourced on the Worker
  - Input arguments for the executable function
  - Files that will be transferred to the participating Workers
  - Number of Jobs
  - Location of the gmk directory
Introduction of the R peach syntax

- Simple interface for creating Projects
- Typically the R peach syntax will define:
  - The name of the function that will be called
  - The name of the R-script that will be sourced on the Worker
  - Input arguments for the executable function
  - Files that will be transferred to the participating Workers
  - Number of Jobs
  - Location of the gmk directory

```r
peach(files = list("gridification_dist.r"), funcname = "gridification_dist", ...)
```

```r
Worker
source("gridification_dist.r")
...
result <- gridification_dist()
...
result <- 1+1
```
Introduction of the R peach syntax

- Simple interface for creating Projects

- Typically the R peach syntax will define:
  - The name of the function that will be called
  - The name of the R-script that will be sourced on the Worker
  - Input arguments for the executable function
  - Files that will be transferred to the participating Workers
  - Number of Jobs
  - Location of the gmk directory

```
peach(funcname="parameters_dist", params=list(multip,"<param>"), peachvector=1:jobs,..)
result <- parameters_dist(multip,jobidx)
```
Introduction of the R peach syntax

- Simple interface for creating Projects
- Typically the R peach syntax will define:
  - The name of the function that will be called
  - The name of the R-script that will be sourced on the Worker
  - Input arguments for the executable function
  - Files that will be transferred to the participating Workers
  - Number of Jobs
  - Location of the gmk directory

```r
peach(funcname="datafiles_dist", params=list(param"), peachvector=1:jobs, datafiles=list("datafile.txt"),
```
Introduction of the R peach syntax

- Example of a complete R peach syntax:

```r
peach(funcname="functionName",
    files=list("exampleScript.R"),
    params=list("<param>"араметrer=10),
    datafiles=list("datafile.txt"),
    peachvector=1:10,
    gmkroot="../../../../../")
```

1. `# Function executed on Workers`
2. `# Script sourced on Workers`
3. `# Input arguments`
4. `# File transferred to Workers`
5. `# Number of Jobs set to 10`
6. `# Path of the 'gmk' directory`
Introduction of the R peach syntax

- Peach features can be enabled by adding more parameters:

```r
peach(funcname="functionName",
    files=list("exampleScript.R"),
    params=list("<param>",parameter=10),
    datafiles=list("datafile.txt"),
    peachvector=1:10,
    gmkroot="../..../..", 
    Rversion="2101",
    stream = TRUE,
    callback="cbFun")
```

# Function executed on Workers
# Script sourced on Workers
# Input arguments
# File transferred to Workers
# Number of Jobs set to 10
# Path of the 'gmk' directory
# Runtime Bundle version
# Enable streaming
# Execute 'cbFun' once for
# each result.
Preparation

- Accessing the peach interface requires two R packages
  - rJava (available in repositories)
  - techila (included in the Techila Grid Management Kit)

- Installing rJava package
  1. Launch R
  2. Install from R with the following command:
     ```
     install.packages("rJava")
     ```

- Installing techila package
  1. Change your directory in R to "<full path>/gmk/grid/R"
  2. Install with command:
     ```
     install.packages("techila",repos=NULL,type="source")
     ```
Creating a test Project

1. Change your current working directory in R to:
   
   `<full path>/gmk/examples/R/Tutorial/1_gridification`

2. Source the `run_gridification.r` script:
   
   ```r
   source("run_gridification.r")
   ```

3. Create the test Project:
   
   ```r
   result <- run_gridification(5)
   ```

4. When prompted, enter your password

5. A status bar will be displayed containing Project information
Requests on what features to demonstrate

- Techila Grid Management Kit contains examples of:
  - Streaming & Callback
  - Job Input Files
  - How to include your own libraries
  - Managing result files with the filehandler
  - Iterative Projects
  - And more...