START

Code or data
modification

Is the modification
imply a migration? — **No** → No migration to write

Yes

Does migration require multi steps?

Depending on the answer to this question, the migration could be complex and must require to be split into multiple steps.

Writing a migration

Schema ← Which kind of migration? → **Data**

Writing the migration queries

Is my data applicable for all environments, for all contexts?

Generic and/or infrastructural data → Which kind of data?

1 Finding the migration revision number

Contextual data

Writing the migration queries

Yes ← Does the migration contain queries for multi components?

Separate queries by components and chronology

4 Finding the next free file name for the context

No

2 Apply the name for the migration parts

3 Apply the name for the migration script

5 Put the annotation for script sequence ordering

Does migration need one more step? — **Yes**

No

Commiting the migration files

Migration ready to be tested

START

Code or data
modification

Does the
modification imply
a migration?

No → No
migration
to write

Yes

Does migration
require multi steps?

Depending on the answer to this question, the migration could be complex and must be split into multiple steps.

0

Writing rules

Does migration
need one more
step?

Yes

No

Commiting the
migration files

Migration
ready to be
tested

START

Code or data
modification

Is the modification
imply a migration?

No → No
migration
to write

Yes → Does migration
require multi steps?

Depending on the answer to this question, the migration could be complex and must require to be split into multiple steps.

Writing a migration

1 Finding the
migration revision
number

Writing the
migration queries

Schema ← Which kind of
migration? → Data

**Generic and/or
infrastructural data**

Which kind of
data?

Is my data applicable for all environments, for all contexts?

Is the migration
contains queries for
multi component?

Yes → Separate queries by
components and
chronology

No

Contextual data

Writing the
migration queries

3 Apply the name for
the migration script

2 Apply the name for
the migration parts

4 Finding the next
free file name for
the context

5 Put the annotation
for script sequence
ordering

Yes ← Does migration
need one more
step?

No

Commiting the
migration files

→ Migration
ready to be
tested

START

Code or data modification

Is the modification imply a migration? — **No** → No migration to write

Yes

Writing a migration

Which kind of migration? — **Data** → Which kind of data?

Schema

Generic or infra. data

Contextual data

Writing generic migration queries

Writing contextual migration queries

Commiting the migration files

Migration script ready

1 — Finding the migration revision

Where was the modification done?
- Branch / Release → Is the modification merged in Trunk?
  - No → Merge the modification into the Trunk
  - Yes → Get the revision number of the merge
- Trunk → Get the revision number of the modification

Merge the modification into the Trunk → Get the revision number of the merge

This part is important to allow following the modifications and matching with migration scripts relative to the commits.

Revision number retrieved

2 — Migration part naming

Which component is concerned?
- Server Library → Go to the migration directory of SLIB-resources
- Server → Go to the migration dir of <component>-ear

3 Name the migration script

Do not forget that the queries will be run from SLIB to Server in chronological order based on <-n> convention for a certain revision.

Is there another part for the migration?

Splitting finished

3 — Build revision name

Is a script already present for the revision?
- Yes → Is there a script named r<revnum>-n.sql?
  - No → Rename the script to r<revnum>-1.sql
  - Yes → Name the script as r<revnum>-<n+1>.sql
- No → Name the script to r<revnum>.sql

Naming complete

5 — Script ordering

Does the script require specific ordering?
- No →
- Yes → Where must the script be placed?
  - Before → Put in script header -- @before <comp> <rnum>
  - At → Put in script header -- @at <comp> <rnum>
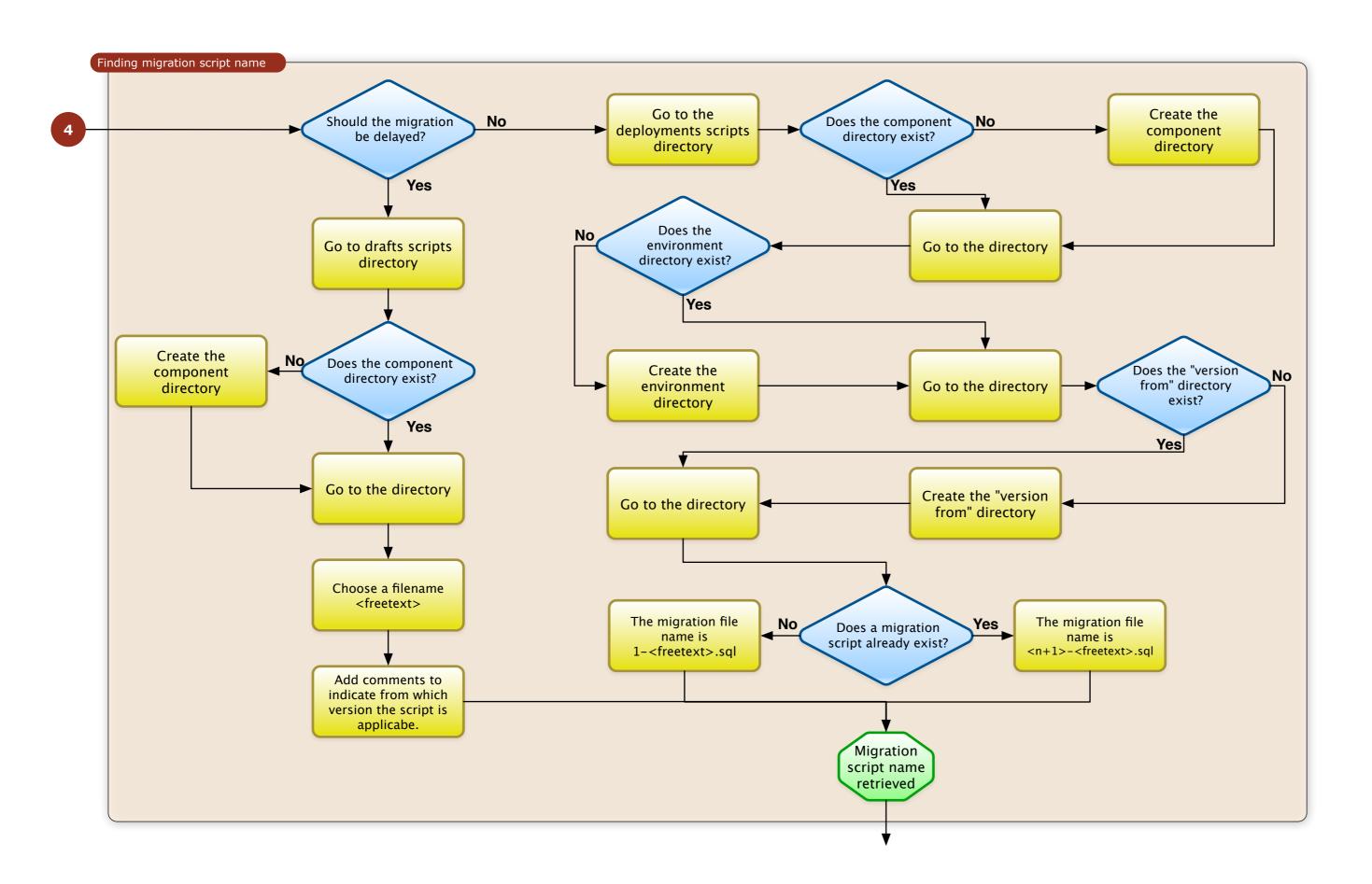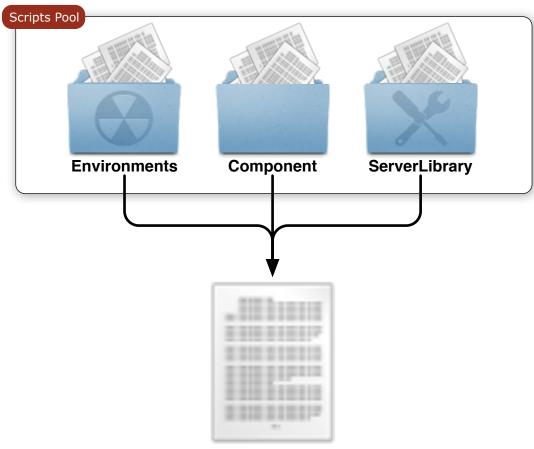 - After → Put in script header -- @after <comp> <rnum>

These annotations allow to position a script for the migration.

Script ordering done

4 — Finding migration script name

Go to the deployments scripts directory

Does the directory for component exist?
- No → Create component directory → Go to the directory
- Yes → Go to the directory

Does directory for environment exist?
- No → Create environment directory → Go to the directory
- Yes → Go to the directory

Does the directory for "version from" exist?
- No → Create the version from directory → Go to the directory
- Yes → Go to the directory

Does a migration script already exist?
- No → The migration file name is 1-<freetext>.sql
- Yes → The migration file name is <n+1>-<freetext>.sql

Migration script name retrieved

Finding migration script name

4 → Go to the deployments scripts directory → Does the directory for the environment exist? — **No** → Create the environment directory

Does the directory for the environment exist? — **Yes** → Go to the directory

Create the environment directory → Go to the directory

Go to the directory → Does the directory for the version from exist? — **No** → Create the directory of the current version deployed

Does the directory for the version from exist? — **Yes** → Go to the directory

Create the directory of the current version deployed → Go to the directory → Does the directory for component exist? — **No** → Create the directory of the component

Does the directory for component exist? — **Yes** → Go to the directory

Create the directory of the component → Go to the directory

Go to the directory → Does a migration script already exist? — **No** → The migration file name is 1-<freetext>.sql

Does a migration script already exist? — **Yes** → The migration file name is <n+1>-<freetext>.sql

The migration file name is 1-<freetext>.sql → Migration script name retrieved

The migration file name is <n+1>-<freetext>.sql → Migration script name retrieved

Finding migration script name

Should the migration be delayed?

No → Go to the deployments scripts directory → Does the component directory exist?

No → Create the component directory

Yes → Go to the directory

Does the environment directory exist?

No → Create the environment directory → Go to the directory → Does the "version from" directory exist?

Yes

No → Create the "version from" directory → Go to the directory

Yes → Go to the directory

Does a migration script already exist?

No → The migration file name is 1-<freetext>.sql

Yes → The migration file name is <n+1>-<freetext>.sql

Should the migration be delayed?

Yes → Go to drafts scripts directory → Does the component directory exist?

No → Create the component directory → Go to the directory

Yes → Go to the directory → Choose a filename <freetext> → Add comments to indicate from which version the script is applicabe.

Migration script name retrieved

Environments   **Component**   **ServerLibrary**

Final Script