Use case 3:
Reusing curation data in build process

Automated use of curation data

- **Integrating** curated data into a **build process**.
- For every source file that is **actually compiled into a binary**, the compliance information is extracted from the SPDX tag:value file **via checksum**.
- This information is combined and now contains only those licenses that must be considered for distributuion of the binary.
- In addition, all **files without a matching checksum** are listed.
- These must be **checked individually**.

Example: BusyBox

1. List all compiled source files:
   a) Use compiler flags

```
# gcc -MMD file.list [TARGET]
```

or

   b) Build binary with debug information and extract source files from symbol table

```
# gdb -ex “info sources” [BINARY] >file.list
```

Example: BusyBox

1. List all compiled source files.
2. Create checksums for all files.
3. Pick File info (for each file) from SPDX file and assemble:

```
##File
FileName: busybox-1.35.0.tar.bz2/busybox-1.35.0.tar/busybox-1.35.0/coreutils/timeout.c
SPDXID: SPDXRef-item160248981
FileChecksum: SHA1: ad066d3cdf9f0c525cd9857d4b3cb26cd49dc6c2
[...]
LicenseConcluded: LicenseRef-GPL-2.0
[...]
FileCopyrightText: <text> Copyright (C) 2005-6, Roberto A. Foglietta
<me@roberto.foglietta.name> </text>
```

Example: BusyBox

1. List all compiled source files.
2. Create checksums for all files.
3. Pick File info (for each file) from SPDX file and assemble:

```
if sed -n "/[FileName]/,/##File/p" [SPDX] | grep -q $checksum
then
  sed -n "/[FileName]/,/##File/p" [SPDX] >>assembled.spdx
fi
```

Example: BusyBox

1. List all compiled source files.
2. Create checksums for all files.
3. Pick File info (for each file) from SPDX file and assemble.

```
if sed -n "/[FileName]/,/##File/p" [SPDX] | grep -q $checksum
then
  sed -n "/[FileName]/,/##File/p" [SPDX] >>assembled.spdx
```

4. List changed files:

```
else
  echo [FileName] >>changed-files.list
fi
```

Example: BusyBox

1. List all compiled source files.
2. Create checksums for all files.
3. Pick File info (for each file) from SPDX file and assemble.
4. List changed files.
5. Get unique ConcludedLicenses and add respective license texts to *assembled.spdx*:

```
LicenseID: LicenseRef-GPL-2.0
LicenseName: GNU General Public License v2.0 only
ExtractedText: <text> GNU General Public License, version 2
[...]
```

Example: BusyBox

1. List all compiled source files.
2. Create checksums for all files.
3. Pick File info (for each file) from SPDX file and assemble.
4. List changed files.
5. Get unique ConcludedLicenses and add respective license texts to *assembled.spdx*:

```
for i in ${grep LicenseConcluded assembled.spdx | cut -d' ' -f 2- | sort -u}
do
 echo $i >>licenses.list
 sed -n "\&LicenseID: $i$&,\&LicenseID:&p" [SPDX] >>assembled.spdx
done
```

Example: BusyBox

1. List all compiled source files.
2. Create checksums for all files.
3. Pick File info (for each file) from SPDX file and assemble.
4. List changed files.
5. Get unique ConcludedLicenses and add respective license texts to *assembled.spdx*.

→ **Should of course be scripted and automated!**

Example: BusyBox – result

- List of **all source files that were compiled into the binary**.
- List of **all source files that were changed** w.r.t. to the curated data.
- List of **all licenses of compiled files** that are present in the curated data.
- **SPDX tag:value file of compiled files** that are present in the curated data including concluded licenses, copyright notices and license texts.