



[@ldomeg Home](#) / Products

## CodeUp Frequently Asked Questions (FAQ)

Version 1.021216

**Q1:** What does CodeUp do?

**A:** CodeUp upgrades changes (modifications) to software releases. Such upgrades typically become necessary when a new base software release or version becomes available. A well-known example: compliance releases in the credit card industry, i.e. the half-yearly changed mandated by Visa and MasterCard.

**Q2:** How is customized code related to base software release code?

**A:** Customizations need to be integrated in the correct place as shown in the example below fixing a trivial mistake.

```
100100      IF A > B
100110          MOVE C TO D
100120      ELSE IF A < B
100130          MOVE D TO C
100140          END-IF
100150      END-IF.
```

The condition A = B is not accounted for. Let's change this with a modification:

```
100135          ELSE MOVE 1 TO D
```

The resultant code reads:

```
100100      IF A > B
100110          MOVE C TO D
100120      ELSE IF A < B
100130          MOVE D TO C
100135          ELSE MOVE 1 TO D
100140          END-IF
100150      END-IF.
```

**Q3:** What are the implications when base software code changes?

**A:** The aforementioned changes (customizations) to base software releases need to be replicated in the new software release. Customizations are placed in the correct place using line numbers. Let's assume that due to other changes, the respective lines in the new base software release have slightly changed line numbers, i.e.

```
100110      IF A > B
100120          MOVE C TO D
100130      ELSE IF A < B
100140          MOVE D TO C
100150          END-IF
100160      END-IF.
```

The resultant code – using the unchanged customization line (i.e. without CodeUp) – will now read:

```
100110      IF A > B
100120          MOVE C TO D
100130      ELSE IF A < B
100135          ELSE MOVE 1 TO D
100140          MOVE D TO C
```

```

100150          END-IF
100160          END-IF.

```

Whilst this resultant code is obviously wrong, it is not too bad: it results in a syntactically incorrect program and the compile step will provide an error message. In many instances, though, the wrongly merged customization still produces a syntactically correct program – and the error is detected only during testing – or in production!

**Q4:** Give an example for such more difficult issue!

**A:** See the following example:

```

100100          IF A > 200
100110              MOVE 2 TO B
100120              ADD 1 TO CTR2
100130          ELSE IF A > 300
100140              MOVE 3 TO B
100150              ELSE IF A > 4
100160                  MOVE 4 TO B
100170                  ADD 1 TO CTR4
100180              END-IF
100190          END-IF
100200          END-IF.

```

A custom code line amends the forgotten increment for counter 3

```

100145          ADD 1 TO CTR3

```

resulting in the following correct program snippet:

```

100100          IF A > 200
100110              MOVE 2 TO B
100120              ADD 1 TO CTR2
100130          ELSE IF A > 300
100140              MOVE 3 TO B
100145              ADD 1 TO CTR3
100150              ELSE IF A > 4
100160                  MOVE 4 TO B
100170                  ADD 1 TO CTR4
100180              END-IF
100190          END-IF
100200          END-IF.

```

Assume that a new software release shifted the line numbers just so slightly to

```

100120          IF A > 200
100130              MOVE 2 TO B
100140              ADD 1 TO CTR2
100150          ELSE IF A > 300
100160              MOVE 3 TO B
100170              ELSE IF A > 4
100180                  MOVE 4 TO B
100190                  ADD 1 TO CTR4
100200              END-IF
100210          END-IF
100220          END-IF.

```

The unchanged custom code line (i.e. without CodeUp) now produces the following code snippet

```

100120          IF A > 200
100130              MOVE 2 TO B
100140              ADD 1 TO CTR2
100145              ADD 1 TO CTR3
100150          ELSE IF A > 300
100160              MOVE 3 TO B

```

```

100170          ELSE IF A > 4
100180                MOVE 4 TO B
100190                ADD 1 TO CTR4
100200                END-IF
100210          END-IF
100220    END-IF .

```

which is syntactically correct (wherefore the compiler will not show an error), but logically wrong (counter 3 is incremented for the wrong condition).

**Q5:** What about minor logical changes from one software release to the next?

**A:** CodeUp performs logical pattern matching (not just physical pattern matching), and thus treats the following three expressions as equivalents:

010100	03 A PICTURE X(2).	010100	03 A PIC X(02).
100100	ELSE IF A > 300	100100	ELSE
		100110	IF A > 300
200100	IF A EQUALS TO B	200100	IF A = B

Note that automated identification tools show these logically equivalent expressions as 'different' and rely on the programmer to provide the assessment of 'equivalent'.

**Q6:** What if a new release or version of base software introduces a new naming convention, e.g. for data tag names?

**A:** CodeUp provides an ancillary input file (edit list) allowing to provide pairs of expressions, e.g. FILE1, INPUT-FILE with the following result:

Original`	Is treated as
010100 SELECT FILE1 ...	010100 SELECT INPUT-FILE ...
100100 OPEN INPUT FILE1	100100 OPEN INPUT INPUT-FILE
200100 READ FILE1 INTO	200100 READ INPUT-FILE INTO
200110 FILE1-RECORD ...	200110 INPUT-FILE-RECORD ...

**Q7:** What options exist to perform such upgrades?

**A:** There are three methods: manual upgrade, automated identification tools, and automated upgrade tools.

- Manual insertion of customized lines is cumbersome, time-consuming and error-prone. Also it uses valuable resource and skilled resources for a largely menial task. However, there are occasions when such skill is needed – and in the absence of tools, programmers are assigned to upgrade everything manually – complex and simple alike.
- Automated identification typically compares old and new version of base code using line numbers as identification, although some more advanced methods sometimes are used. Differences are identified, speeding up the actual upgrade process, which is still performed manually.
- An automated upgrade tool such as CodeUp not only identifies the difference in base software, but also adjusts the line number for every modified code line, thus reducing manual effort to a minimum.

**Q8:** Does this correspond to Artificial Intelligence?

**A:** No, although some might like to call it that. CodeUp uses effective pattern-matching techniques and uses paragraph names (amongst other things) as pointers. However, CodeUp does not attempt to identify the meaning of a code modification. Consequently, there are instances where CodeUp will identify a difference, but not propose a line change. In these instances, a programmer will perform the adaptation manually.

**Q9:** What percentage of manual intervention is required when using CodeUp?

- A:** Obviously, there is no unique and absolute number. The following serve as guidelines, proven in a few installations:
- Typical software release upgrade, such as a compliance release upgrade: automation is virtually 100%.
  - Upgrade from one software version to another version with significant structural changes: automation should be better than 80 or 85%, comparing total effort for a manual upgrade to total effort for a CodeUp assisted upgrade.

**Q10:** How can CodeUp be acquired?

- A:** There are three ways: as a service, as a license or in a hybrid version of service and license.
- Under a service agreement, @ldomeg or one of its distributors performs the upgrade and deliver either upgraded modifications, compiled upgraded modifications or tested upgraded programs depending on a customer's request.
  - Under a license agreement, customer's staff performs the upgrade using CodeUp. @ldomeg or its distributors provide training and on-going support.
  - The hybrid version is a combination of the aforementioned: a pilot project is performed by @ldomeg or its distributor (preferably with customer's participation) and customer staff performs the rest of the upgrade having licensed CodeUp

**Q11:** Is CodeUp limited to certain software vendors' code?

- A:** CodeUp has no dependencies that limit its use to particular industries or software vendors – as long as base software is using COBOL or Assembler in a mainframe environment. Whilst there are no inherent limitations to other platforms such as AS/400 or UNIX, CodeUp itself has not been tested for these environments. Contact @ldomeg for further information.

**Q12:** What assumptions does CodeUp rely on?

- A:** There are a few: Code written in Assembler or COBOL; base software is periodically updated as new releases and/or versions; the new and the old base software are structurally similar; modifications together with old base software are functionally operational.

**Q13:** Are there restrictions for CodeUp use?

- A:** There are some restrictions related to CodeUp use. Such restrictions and remedies are:
- CodeUp is intended for a customer's own and owned software/software modifications. Third party use (e.g. operating a service where multiple customer's modifications are upgraded) require a special arrangement. Contact @ldomeg or its distributor.

**Q14:** How much does CodeUp cost?

- A:** Price depends on the type of arrangement sought (service, license, hybrid) and also is based on installation size. Benefits are:
- Repetitive use of license: after one -time investment saving continue for years
  - Break-even typically after one or two upgrades
  - Alternate use for skilled resources
  - Speed of upgrade – especially important for compliance upgrades in the credit card industry

**Q15:** How is CodeUp different from other products on the market?

**A:** Other products fall in the category of automated identification tools leaving a large amount of the task to a programmer.

**Q16:** What support is available during the course of an upgrade project?

**A:** @ldomeg and/or its distributor provide phone support. Contact @ldomeg or its distributor.

**Q17:** What contractual agreements are available?

**A:** Services are covered by a service agreement, CodeUp licenses by a software license agreement. Upgrades in a hybrid fashion use a service agreement and a license agreement. Although, a CodeUp license is a perpetual license, there is a maintenance agreement available for on-going maintenance support; maintenance agreement also entitles a customer to periodic new CodeUp releases.

**Q18:** Does customized code have to be kept in separate libraries or files?

**A:** CodeUp works in both instances:

- If modifications are kept separate from base software code, the result will be an upgraded version of the modification in a separate file.
- If modifications are integrated with base software code, CodeUp will produce newly integrated code, i.e. the adapted modifications are merged with the new base software code.

[@ldomeg Home](#) / Products