

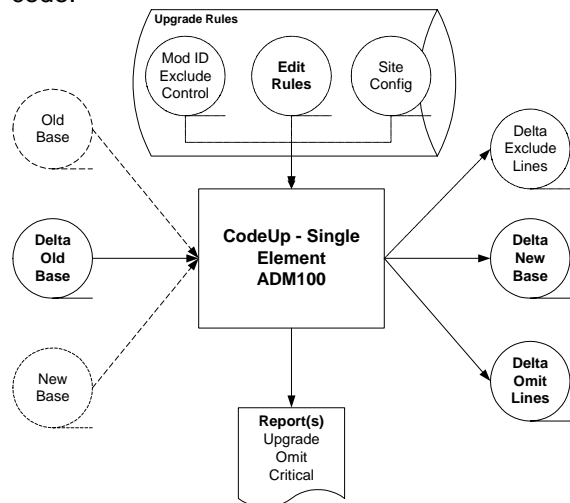
# CodeUp

Automation in Upgrading of Customized Procedural Code

Virtually every mainframe software vendor's application today is distributed in releases, where each release introduces new functionality and support for new hardware and/or software. For example, this release approach is used very strictly with credit card vendor software to reflect the semiannual compliance changes imposed by Visa and MasterCard – and the approach is similar in other industries.

Two primary mechanisms adapt vendor software to an individual installation: parameterization and extensions to base software: so called custom code. Parameters entered via online screens do not result in procedural code. Extensions to base software typically branch off the base software into procedural COBOL or Assembler code to fulfill a function unique to an installation (e.g. for specific customer or local government requirements).

The transition from one vendor release to the next is easier if the base code and the custom code are kept in separate libraries. Custom code is stored as 'delta deck' to the base code. The upgrade process uses the current base code, the new version of the base code and the delta decks as input to produce modified delta decks that will work in conjunction with the new version of base code.



Optional inputs allow exclusion of identifiable modifications from the upgrade process and provide a string substitution list for changed data tags.

Some vendors support the upgrade process with a utility application, typically relying on unchanged line sequence numbering.

Faced with the task of upgrading, customers can choose between three options:

- Delay implementation of the new release. This risks falling out of compliance and maintenance. New vendor functions are not available.
- Manual upgrade. This approach uses already scarce IT resources on repetitive and error-prone work at high cost for an unproductive effort.
- **CodeUp**. This approach automates the upgrade task to a very high degree, is quick, reliable and repeatable for upgrade after upgrade.

Note that use of change management products does not address this issue since they work with a granularity of a whole code object rather than individual lines of code or statements.

## Dramatically cuts time and effort:

- Automates upgrade up to 100% (typically manual intervention less than 10 - 20% compared to manual upgrade depending on degree of structural difference)
- Frees IT staff for development work
- Uses matching techniques from DNA-string pattern recognition, including fuzzy matches

## Supports wide area of functions:

- Shows all matching code strings
- Identifies all non-matching code strings
- Inserts delta decks in proper place even when changes occur for line numbers, paragraph names, copybook names, data tag names, working-storage fields or in white space (space, comma, etc.)
- Supports stacked delta decks
- Upgrades whole libraries in one pass-thru
- Supports merged code as well as delta decks
- **New with Release 1.8:**
  - Enhanced condition reporting and comparisons
  - Optional XML formatted output
  - Additional Return Codes for identifying groups of New Modifications after upgrade requiring unique handling

### Supports incremental approach:

- Allows inclusion/exclusion of selected delta decks
- Creates spill file for future upgrade
- Allows environment-specific definition of edit substitution rules
- Edit rules can be extended recursively

### Schematic:

The following chart illustrates the relationship between old base/old modification and new base/new modification. An excerpt of the CodeUp report is displayed.

#### OLD BASE

```
120100 A-PARA.
120200     MOVE F1 TO FLD1.
120300     MOVE F2 TO FLD2.
120400     MOVE F3 TO FLD3.
120500*
```

#### OLD MODIFICATION

```
120320     MOVE F21 TO FLD21.
```

#### NEW BASE

```
120100 A-PARA.
120200     MOVE F0 TO FLD0.
120300     MOVE F1 TO FLD1.
120400     MOVE F2 TO FLD2.
120500     MOVE F3 TO FLD3.
120600*
```

#### NEW MODIFICATION

```
120420     MOVE F21 TO FLD21.
```

#### CodeUp REPORT

```
OLD BASE LINES           5
NEW BASE LINES           6
OLD LINES NOT FOUND IN NEW BASE  0
NEW LINES NOT FOUND IN OLD BASE  1
LARGEST MATCHED BLOCK     4
LARGEST DELETED BLOCK     0
LARGEST INSERTED BLOCK    1
PERCENTAGE OF CHANGED LINES 9%
UPGRADE CONFIDENCE HIGH
```

#### C000

#### FOUND INSERTED LINES:

```
Old Base ABOVE :      OB->120300     MOVE F2 TO FLD2.
----->              INS>120320     MOVE F21 TO FLD21.
Old Base BELOW :      OB->120400     MOVE F3 TO FLD3.
Upper Anchor MAPPED TO NB->120400     MOVE F2 TO FLD2.
   Output INSERTION as  INS>120420     MOVE F21 TO FLD21.
Lower Anchor MAPPED TO NB->120500     MOVE F3 TO FLD3.
```

### Configuration:

- Mainframe MVS

### Support:

- CodeUp is very easy to use and comes fully and extensively documented.
- Two-day training classes are available from ALDOMEG upon request.
- Service to perform upgrades turn-key by ALDOMEG is available upon request.
- Maintenance agreement is available for updates after warranty and new releases.
- Warranty: 30 days after delivery

### Ordering Information:

Email: [sales@aldomeg.com](mailto:sales@aldomeg.com)  
Phone: ++1-407-389-2481  
Shipment on CD-ROM  
Delivery by courier available

### About ALDOMEG:

ALDOMEG has offices in Orlando, FL and representatives in Australia and Brazil. ALDOMEG is well recognized for a series of system and cross-application tools.

For more information, call ALDOMEG at the number below or visit the ALDOMEG website at [www.aldomeg.com](http://www.aldomeg.com).



### ALDOMEG, Inc.

115 W. Gore St.  
Orlando, FL 32806, U.S.A.  
Phone: ++1-407-389-2481