

# ABAP Quickfix for S4 Conversions: An Eclipse ADT Plugin

ABAPConf 2022

---

Lawrence Cabac

lc@leuc.de

<https://leuc.de/abapquickfixs4c>

07.12.2022

ADventas Consulting GmbH [lawrence.cabac@adventas.de](mailto:lawrence.cabac@adventas.de) <https://adventas.de>

## ① Introduction

About Me

Company

## ② Motivation

## ③ Creating an ADT Plugin

Features of the ABAP QuickFix for S4 Conversions Plugin

Examples

## ④ Conclusion

# Introduction

---

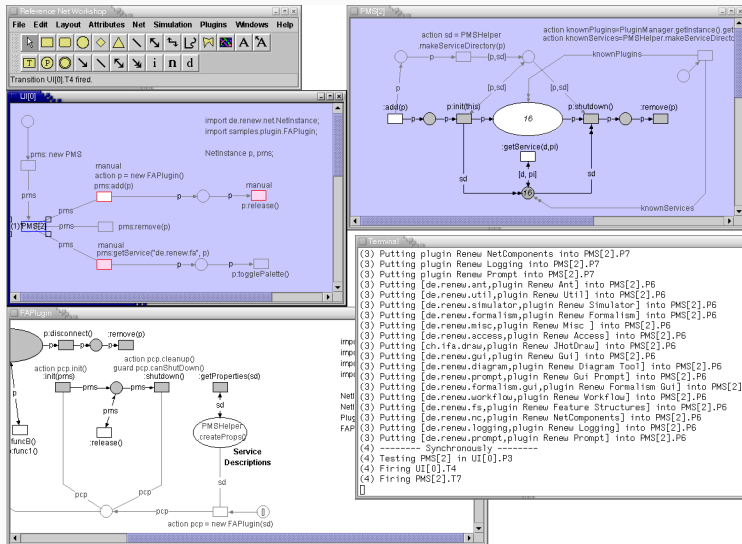
## Background: University of Hamburg, Theoretical Computer Science

- Distributed AI: Modeling Dynamic Software Architectures.
  - Multi-Agent Systems
  - Plugin Systems
- Petri Nets
  - Semantics: Formal  $\rightarrow$  Verification / Operational  $\rightarrow$  Execution
  - Reference Nets (Object-Orientation, full Java Integration)
- Development IDE: Renew [[www.renew.de](http://www.renew.de)] +200 Plugins

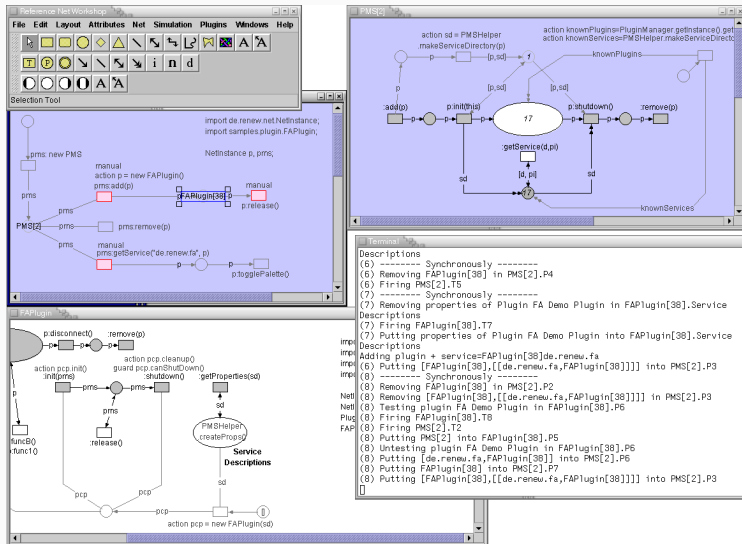
## Current Employment: ADventas Consulting GmbH [[www.adventas.de](http://www.adventas.de)]

- Since 2017: ABAP Development with Eclipse in GTM, MM, SD, FI, ...

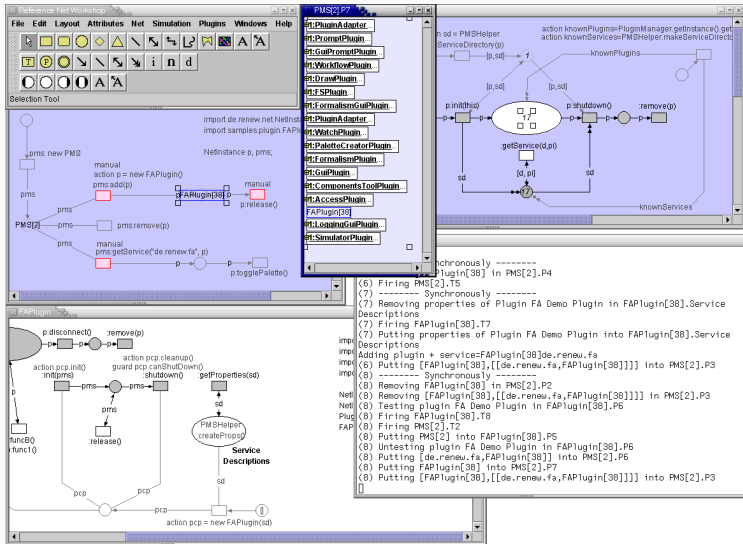
# Example Plugin Management System Model

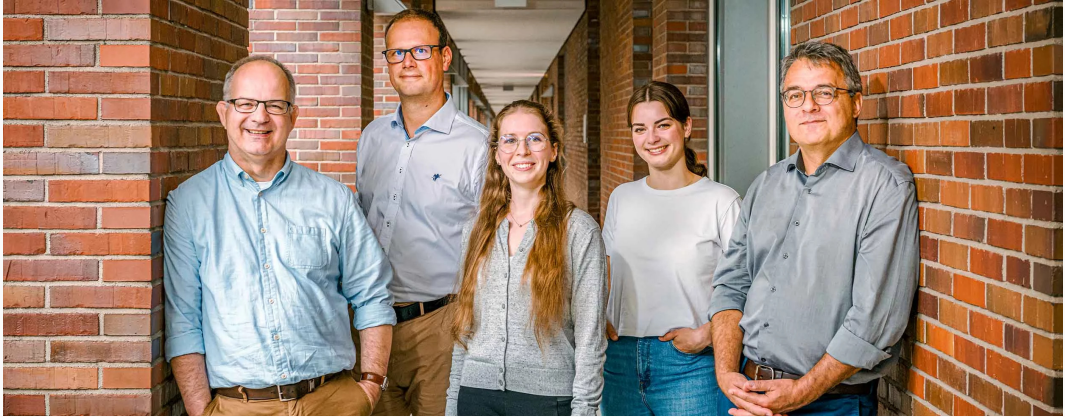


# Example Plugin Management System Model



## Example Plugin Management System Model

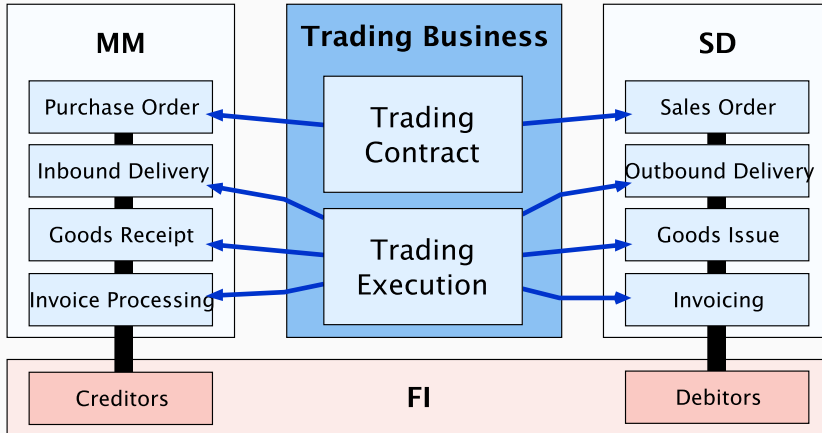




**Figure 4:** ADventas Team



# Global Trade Management



**Figure 5:** Trading Contract and Trading Execution Workbench

After: [help.sap.com]

# Motivation

---

# Project: S4 Conversion Projects

## Challenges

Frequently recurring tasks: adaptation SQL statements.

- Select statements without *order by* clause.
- Non-unique *select single* statements
  - ATC warning: *select single is possibly not unique*.
  - Convert select (w/o single) statement and add *up to 1 rows*.
- Tables with changed key in S4.
  - Introduction of a single UUID key field.
  - Order sequence is not always clear.
  - Sometimes the old key fields have changed.
  - *Order by* sequences have to be provided explicitly.

```
select single * from vbfa into ls_vbfa
where vbelv = ls_vbak-vbeln and vbtyp_n = 'J'.
```

Listing 1: Select single statement

```
select single p~lgort from vepo as p
into @<pos>-lgort
where p~vbeln = @<pos>-vbeln
and p~werks = @<pos>-werks.
```

Listing 2: Select single statement

# ABAP QuickFix Plugin (Łukasz Pęgiel 2022): StatementAssistRegex

**Package** com.abapblog.adt.quickfix.assist.syntax.statements

## Class StatementAssistRegex

All Implemented Interfaces:

IAssist, IAssistRegex

### Method Summary

Methods inherited from interface com.abapblog.adt.quickfix.assist.syntax.statements.IAssist

canAssist, getAssistIcon, getAssistLongText, getAssistShortText, getChangedCode, getReplaceLength, getStartOfReplace

Methods inherited from interface com.abapblog.adt.quickfix.assist.syntax.statements.IAssistRegex

getMatchPattern, getReplacePattern

# First Steps: Move-corresponding

## Matching Pattern

```
(?s)(\s*)move-corresponding\s+(.*)\s+to\s+(.*)"
```

## Replacement Pattern

```
$3 = corresponding #( $2 )
```

## Replaced Code

```
* move-corresponding komwbhi to wbhi.  
wbhi = corresponding #( komwbhi ).
```

Listing 3: Before QuickFix

## Creating an ADT Plugin

---

## Plugin (Cabac et al. 2005)

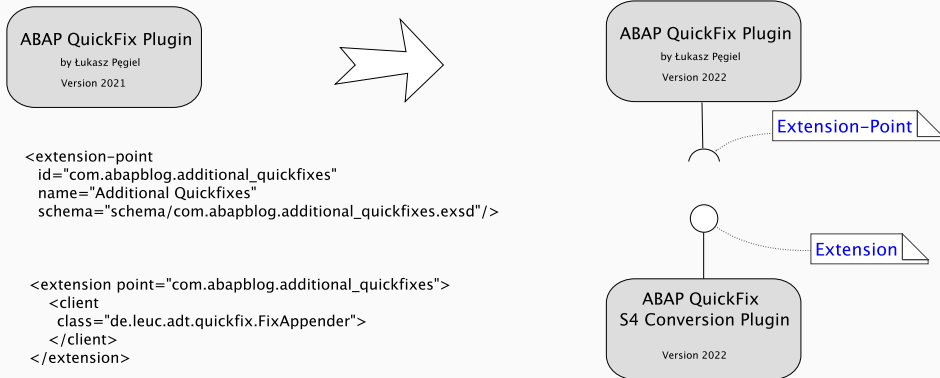
A plugin is a component that can be added to a (plugin) system. It extends a specified extensible functionality and follows a defined lifecycle.

## Plugin in Eclipse

- Jar-file or folder called bundle (cf. OSGi)
- Declaration of dependencies, version, etc.
- Declaration may contain extension-points: i.e. plugins can be extensible



# ABAP QuickFix Plugin: Extension Point



**Figure 6:** Extension Point and Extension

(Łukasz Pegiel 2022) (Lawrence Cabac 2022) (Eclipse 2022)

## Extension-Point

- Extension Schema (Additional Quickfixes)
- Interface (or Class) (IFixAppender)
- Export packages
- Implementation (add quick fixes to list of available quick fixes)

## Extension

- Declare requirements (Additional Quickfixes)
- Implement interface (`public class FixAppender implements IFixAppender`)

## Approach / Necessities

- Matching pattern (find relevant statements).
- Replacement pattern (Regex).
- Conversion to new syntax (using “@” and “,”).
- Format output.
- Optional (Preferences):
  - Provide newest syntax (2021).
  - Comment out original.
  - Provide change marker.
  - Configurable *order by* lists.

# Syntax

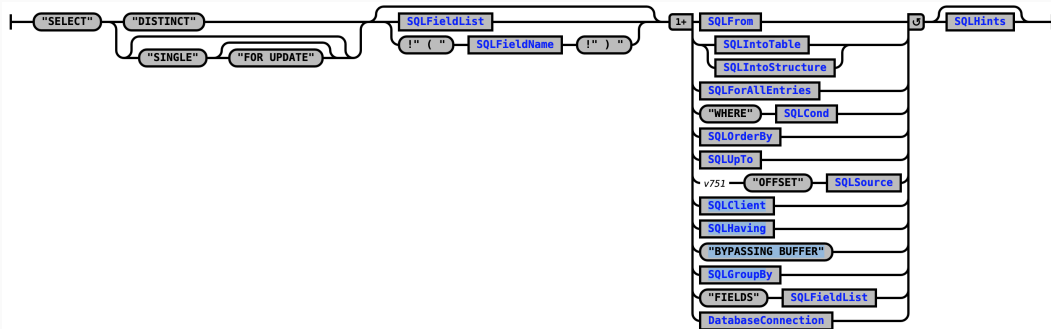


Figure 7: Syntax Diagram: Select Statement [syntax.abaplint.org]

## Matching Pattern

```
1 (?<select>select)\s+(?<single>single)\s+(?<fields>.*)
2   (?: (?: (?<from>from) (?<table>.*)) |
3       (?: (?<into>into) (?<variable>.*)) |
4       (?: (?<where>where) (?<condition>.*))) {3}
```

## Replacement Pattern

```
1 ${select} ${fields}
2   ${from} ${table}
3   ${into} ${variable}
4   up to 1 rows
5   ${where} ${condition}
6   order by primary key.
4 endselect
```

## Formatting

- field list → add commas
- variables → add arobase ('@')
- line feeds
- indentation

# Implementation Details

## Preferences

type filter text

- > General
- ✓ ABAP Development
  - Activation
  - Cross Trace
  - Debug
  - > Editors
  - > Graphical Tools
  - Profiling
  - Project Explorer
  - Quick Fix
  - Quick Fix S4**
  - Quick Fix S4 - Order-By List

### Quick Fix S4

Settings for ABAP QuickFix Plugin S4

Comments and Indentations.

- ☒ Use new style (2021)
- ☒ Add change comment line
- ☒ Comment out original

Indentations in select statements

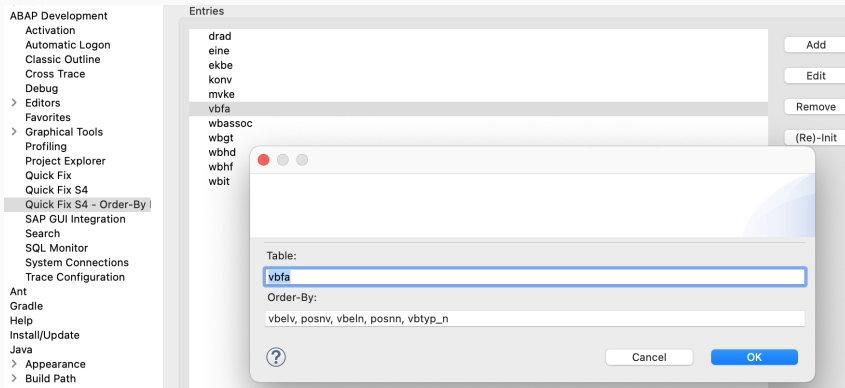
Comment text for changes

**Figure 8:** Preferences: 2021 style, comment line, commented original



# Implementation Details

## Preferences: Order-By List



**Figure 9:** Preferences: list of order by sequences

## In Addition / Spin-Off

- Formatting change select single into new style.
- Formatting change select/endselect into new style.
- Move-corresponding → corresponding #

## Example: VBFA (Document Flow - SD)

```
select single * from vbfa into ls_vbfa
  where vbelv = ls_vbak-vbeln and vbtyp_n = 'J'.
```

Listing 4: Before QuickFix

```
select *
  from vbfa
  into @ls_vbfa
  up to 1 rows
  where vbelv = @ls_vbak-vbeln
        and vbtyp_n = 'J'
  order by vbelv, posnv, vbeln, posnn, vbtyp_n.
endselect.
```

Listing 5: After QuickFix

## Example: WBGT (Generic Document Information - GTM)

```
SELECT SINGLE tkonn tkonn_ex tposn
  INTO (gs_komwbgt-tkonn, gs_komwbgt-tkonn_ex, gs_komwbgt-tposn)
  FROM wbgd WHERE doc_type EQ gs_komwbgt-doc_type AND
                vbeln      EQ g_vbeln          AND
                posnr      EQ g_posnr.
```

Listing 6: Before QuickFix

```
SELECT
  FROM wbgd
  FIELDS tkonn, tkonn_ex, tposn
  WHERE doc_type EQ @gs_komwbgt-doc_type
        AND vbeln EQ @g_vbeln
        AND posnr EQ @g_posnr
  ORDER BY doc_type, vbeln, posnr, posnr_sub, gjahr
  INTO (@gs_komwbgt-tkonn, @gs_komwbgt-tkonn_ex, @gs_komwbgt-tposn)
  UP TO 1 ROWS.
ENDSELECT.
```

Listing 7: After QuickFix

## Example WBIT (Association Item - GTM)

```
select single tkonn tposn
  from wbit into (tkonn, tposn)
    where tkonn = tkonn
      and tposn = tposn.
```

Listing 8: Before QuickFix

```
select tkonn, tposn
  from wbit
  into (@tkonn, @tposn)
  up to 1 rows
  where tkonn = @tkonn
    and tposn = @tposn
  order by doc_type, doc_nr, doc_year, doc_bukrs, item, sub_item.
endselect.
```

Listing 9: After QuickFix

## Example WBIT (new style – 2021)

```
select single tkonn tkonn_ex tposn
into (gs_komwbgt-tkonn , gs_komwbgt-tkonn_ex , gs_komwbgt-tposn )
from wbgd where doc_type eq gs_komwbgt-doc_type
and vbeln eq g_vbeln and posnr eq g_posnr.
```

Listing 10: Before QuickFix

```
select
from wbgd
fields tkonn , tkonn_ex , tposn
where doc_type eq @gs_komwbgt-doc_type
and vbeln eq @g_vbeln
and posnr eq @g_posnr
order by primary key
into (@gs_komwbgt-tkonn , @gs_komwbgt-tkonn_ex , @gs_komwbgt-tposn )
up to 1 rows.
endselect.
```

Listing 11: After QuickFix

## Conclusion

---

- APAP Quick Fix S4 Conversion Plugin is available.
- Specialized quick-fix for conversion projects (select single).
- Formatting of SQL statements.



# Lessons Learned

- Regular expressions work great for simple changes.
- Regular expressions can lead to surprising result with complex statement.
- Unit-Tests are always useful.
- Changing of upstream API can be surprising as well.

- Add formatting options – make Formatter exchangeable.
- Add order-by for `select/endselect`.
- Switch to Maven build process.
- Provide code signing.

## References

---

Cabac, L., M. Duvigneau, D. Moldt, and H. Rölke (2005).

**Modeling dynamic architectures using nets-within-nets.**

In *Proceedings of the 26th International Conference On Application and Theory of Petri Nets*, LNCS, Miami. Springer-Verlag.

Eclipse (2022).

**Eclipse marketplace.**

<https://marketplace.eclipse.org/>.

Lawrence Cabac (2022).

**ABAP quickfix S4 conversions plugin.**

<https://github.com/lc-leuc/ABAPQuickFixS4Conversion> <https://leuc.de/abapquickfixs4c/>.

Łukasz Pęgiel (2022).

**ABAP quickfix plugin.**

<https://github.com/fidley/ABAPQuickFix>.

# Influence SAP ADT

Show Camel Case in Package Explorer: <https://influence.sap.com>  
CDS view Abapdoc in Element Info view: <https://influence.sap.com>

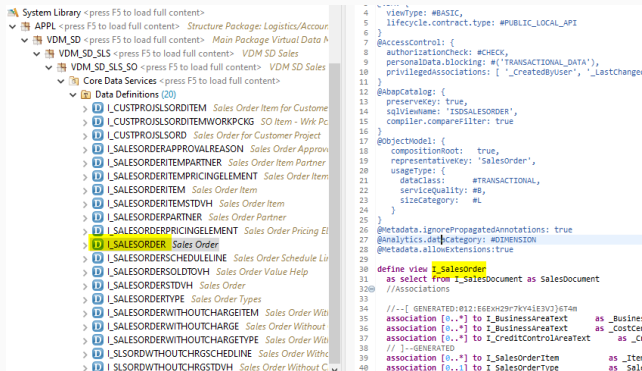


Figure 10: Package Explorer shows only upper case.