

Optional	Use	Decl.	Assign v.	Decl. w
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

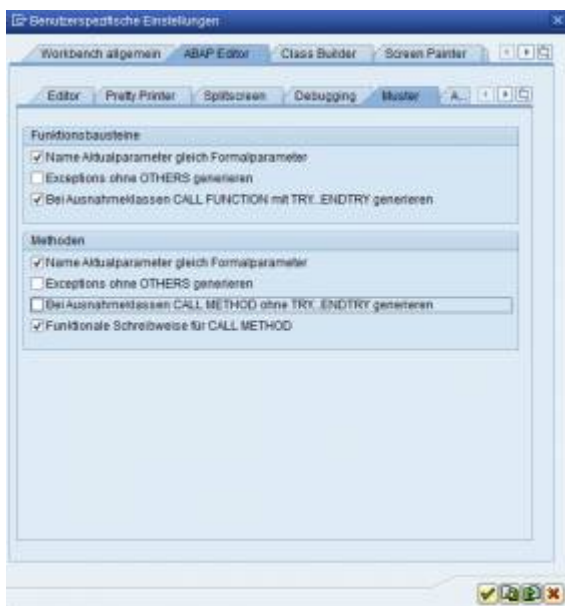
# Hovitaga Advanced Pattern Enhancement

Wer in ABAP programmiert hat mit Sicherheit schon den Befehl „Muster einfügen“ im Editor verwendet. Hierüber lassen sich in erster Linie die Aufrufe für Funktionsbausteine und Methoden in den Quelltext einfügen. Gerade bei Funktionsbausteinen ist dies ein unerlässliches Hilfsmittel, denn die Syntaxprüfung prüft bei Funktionsbausteinen nicht, ob ein falscher Parameter verwendet wird.

Die ungarische SAP-Entwicklerfirma [Hovitaga](#) hat für das Einfügen von Mustern ein kleines kostenloses AddOn entwickelt. Doch der Reihe nach.

## Einfügen von Mustern

Über das Menü *Hilfsmittel - Einstellungen* im Editor können einige Einstellungen für das Einfügen von Mustern vorgenommen werden:



## Muster einfügen SAP Standard

Das Einfügen von Mustern erfolgt über den Druckknopf *Muster* im Editor. Es erscheint ein Popup in dem man auswählen muss, für welchen Befehl man ein Muster einfügen möchte. Wir entscheiden uns für die Option *ABAP Objects Muster* und gelangen auf ein weiteres Popup:

OO Anweisungsmuster

Call Method

Instanz	GR_GRID
Klasse/Interface	CL_GUI_ALV_GRID
Methode	SET_TABLE_FOR_FIRST_DISPLAY

Create Object

Instanz	
Klasse	

Raise Event

Klasse/Interface	
Ereignis	

Raise Exception

Exceptionklasse	
-----------------	--

✓ ✗

Nachdem ich mich an dieser Stelle häufig ärgere, dass in der Suchhilfe zur Methode alle Methoden der Klasse angezeigt werden und keine Unterscheidung zwischen PRIVATE, PROTECTED oder PUBLIC gemacht wird, endet hier auch die Funktionalität im SAP Standard. Je nach Release wird vor dem Aufruf der Methode noch die Datendeklaration eingefügt.

## Muster einfügen Hovitaga style

Dort, wo SAP aufhört, fängt die Funktionalität von Hovitaga an. An der Stelle, wo normalerweise der Code in den Quelltext eingefügt wird, erscheint erneut ein Popup:



Hier kann durch verschiedene Funktionen der Aufruf im Quelltext gesteuert werden. Anstatt im Coding nicht benötigte Parameter zu löschen, kann im Dialog bereits entschieden werden, welche Parameter benötigt werden. Auf Wunsch wird auch die Datendeklaration für den jeweiligen Parameter eingefügt. Für Tabellenparameter kann auch ein entsprechender Arbeitsbereich deklariert werden. Hovitaga folgt hierbei den üblichen Namenskonventionen:

- S = Struktur
- T = Tabelle
- V = Variable
- R = Referenz auf ein Objekt

Durch einen Schalter kann zwischen globaler und lokaler Datendeklaration gewählt werden, was sich in der Namenskonvention durch die Verwendung von G oder L auswirkt.

In einer Vorschau wird der einzufügende Aufruf angezeigt. Man sieht sofort, wie sich die Änderungen von z.B. „Remove Unused Lines“ auswirken.

## **Fazit**

Ich möchte nicht soweit gehen, dass es ein unentbehrliches Tool für den ABAP-Programmierer ist. Allerdings finden sich sinnvolle Funktionen in dieser Erweiterung, die das Leben durchaus leichter machen können. Es gibt also eine klare Tricktresor-Empfehlung!

## **Download**

[Hovitaga Advanced Pattern Enhancement - Product page and Download](#)