

Skróty klawiszowe dostępne w edytorach kolorowanych

Skrót	Znaczenie	Skrót	Znaczenie
[Ctrl] + [Up]	Przewinięcie o linię w górę	[Ctrl] + 2	Idź do zakładki 2
[Ctrl] + [Down]	Przewinięcie w dół	[Ctrl] + 3	Idź do zakładki 3
[Ctrl] + [PgDown]	Przejdź do końca ekranu	[Ctrl] + 4	Idź do zakładki 4
[Ctrl] + [PgUp]	Przejdź do początku ekranu	[Ctrl] + 5	Idź do zakładki 5
[Ctrl] + [Home]	Przejdź do początku edytora	[Ctrl] + 6	Idź do zakładki 6
[Ctrl] + [End]	Przejdź do końca edytora	[Ctrl] + 7	Idź do zakładki 7
[Ins]	Przełącz tryb wstawianie/wpisywanie	[Ctrl] + 8	Idź do zakładki 8
[Ctrl] + [Ins]	Kopiuj zaznaczony fragment	[Ctrl] + 9	Idź do zakładki 9
[Shift] + [Del]	Wytnij zaznaczony fragment	[Shift] + [Ctrl] + 0	Ustaw zakładkę 0
[Shift] + [Ins]	Wklej ze schowka	[Shift] + [Ctrl] + 1	Ustaw zakładkę 1
[Ctrl] + [BkSp]	Usuń ostatnie słowo	[Shift] + [Ctrl] + 2	Ustaw zakładkę 2
[Alt] + [BkSp]	Undo (cofnięcie operacji)	[Shift] + [Ctrl] + 3	Ustaw zakładkę 3
[Shift] + [Alt] + [BkSp]	Redo (ponowienie cofniętej operacji)	[Shift] + [Ctrl] + 4	Ustaw zakładkę 4
[Shift] + [Ctrl] + I	Odsunięcie zaznaczonego bloku	[Shift] + [Ctrl] + 5	Ustaw zakładkę 5
[Shift] + [Ctrl] + U	Dosunięcie zaznaczonego bloku	[Shift] + [Ctrl] + 6	Ustaw zakładkę 6
[Ctrl] + M	Przełamanie linii	[Shift] + [Ctrl] + 7	Ustaw zakładkę 7
[Ctrl] + H	Wstaw linię	[Shift] + [Ctrl] + 8	Ustaw zakładkę 8
[Ctrl] + T	Usuń słowo	[Shift] + [Ctrl] + 9	Ustaw zakładkę 9
[Ctrl] + G	Usuń linię	[Shift] + [Ctrl] + N	Tryb wierszowego zaznaczania (std.)
[Shift] + [Ctrl] + Y	Usuń do końca linii	[Shift] + [Ctrl] + C	Tryb kolumnowego zaznaczania
[Ctrl] + 0	Idź do zakładki 0	[Shift] + [Ctrl] + L	Tryb zaznaczania pełnych linii
[Ctrl] + 1	Idź do zakładki 1	[Shift] + [Ctrl] + B	Dopasuj nawias

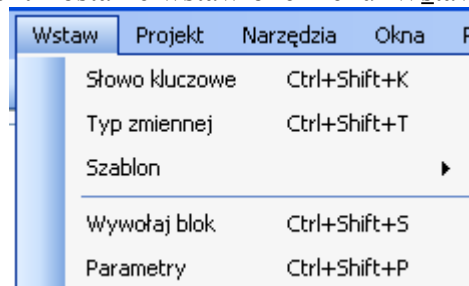
Nowe ułatwienia dostępne w edytorze pojedynczym kolorowanym

Od wersji 1.0.0.9 dostępne są kolejne dwie nowe możliwości przyspieszenia edycji kodu źródłowego napisanego w języku ST podczas używania edytora pojedynczego kolorowanego.

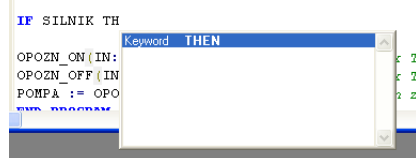
Po uruchomieniu edytora kodu języka ST, przed menu 'Projekt' zostanie wstawione menu 'Wstaw', widoczne na rysunku z prawej części strony.

Zawiera ono następujące pozycje:

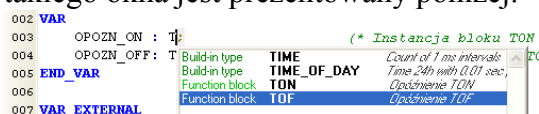
- Słowo kluczowe,
- Typ zmiennej,
- Szablon (menu rozwijalne),
- Wywołaj blok,
- Parametry.



Po wybraniu pierwszego z nich (opcjonalnie za pomocą skrótu klawiszowego [Ctrl]+[Shift]+K) pojawi się okno widoczne poniżej zawierające wszystkie słowa kluczowe języka ST. Wyświetlane są jedynie dopasowane do bieżącego wpisywanego słowa.

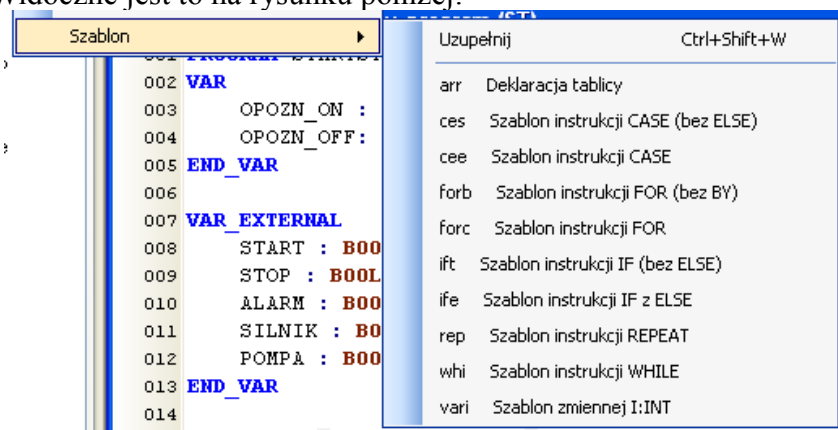


Podobnie działa drugie z menu 'Typ zmiennej'. Po wybraniu tej opcji wyświetli nam się lista dopasowanych nazw typów z wyjaśnionym rodzajem oraz krótkim komentarzem (dyrektywa \$COMMENT). Przykład użycia takiego okna jest prezentowany poniżej.



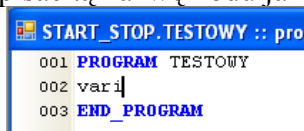
Nieco inną zasadę działania mają szablony. Standardowo CPDev udostępnia 9 szablonów, które użytkownik może zmieniać a także tworzyć własne. Szczegóły tych czynności będą omówione później. Z zasady szablony stosuje się do wpisywania stałych fragmentów kodu w bardzo szybki sposób. Edytor ten umożliwia dwie metody wywoływania szablonów. Pierwsza dla początkujących użytkowników i druga dla dłużej pracujących ze środowiskiem CPDev.

Pierwsza metoda opiera się na wyborze z menu właściwego szablonu oznaczonego kodem skrótu i małym opisem. Widoczne jest to na rysunku poniżej:



Po wybraniu dowolnej z pozycji znajdujących się pod kreską separatora w menu w miejscu kursora zostanie wstawiony kod skojarzony z daną pozycją, a kursor zostanie przeniesiony do pozycji wyznaczonej przez tworzącego szablon.

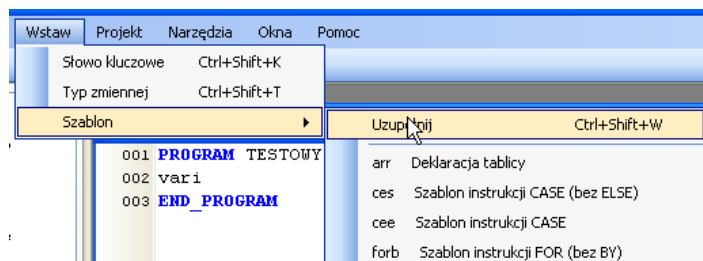
Druga metoda jest nieco szybsza z uwagi na skrót klawiszowy, czyli możliwość wykonania bez użycia myszy. Wymagania jakie są stawiane tej metodzie to znana nazwa kodu skrótu (widoczna w menu jako pierwszy człon). Należy wpisać tę nazwę kodu jak na rysunku poniżej,



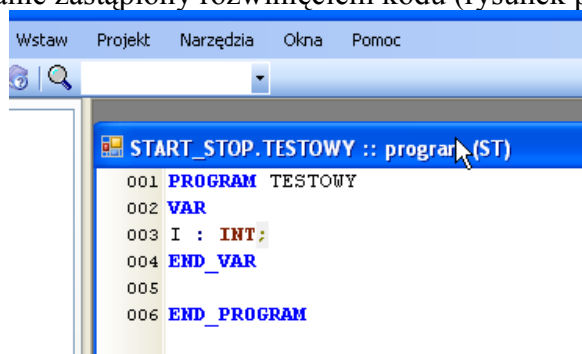
```

START_STOP.TESTOWY :: program
001 PROGRAM TESTOWY
002 var1
003 END_PROGRAM
    
```

a następnie wybrać z menu (mało ekonomiczne podejście) pozycję 'Uzupełnij' jak na rysunku poniżej.



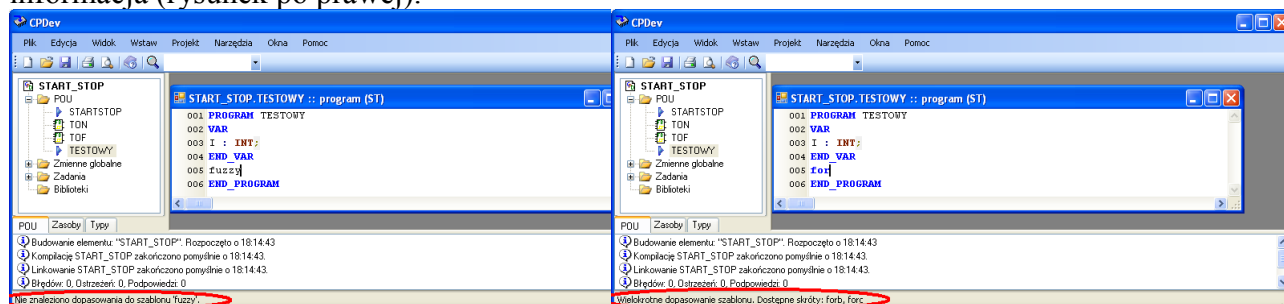
Alternatywnie można (a nawet trzeba, aby było szybciej) użyć skrótu [Ctrl]+[Shift]+W. Dotychczasowy skrót zostanie zastąpiony rozwinięciem kodu (rysunek poniżej).



```

START_STOP.TESTOWY :: program (ST)
001 PROGRAM TESTOWY
002 VAR
003 I : INT;
004 END_VAR
005
006 END_PROGRAM
    
```

Wpisanie niepełnej nazwy skrótu spowoduje wyszukanie skrótów rozpoczynających się od wpisanego ciągu. Gdy liczba dopasowanych jest równa 1 to zostaje wykonane autouzupełnianie, gdy nie znaleziono dopasowań to sygnalizowane jest to komunikatem w pasku statusu (rysunek po lewej otoczony czerwoną obwiednią), gdy znaleziono więcej dopasowań to wyświetla się stosowna informacja (rysunek po prawej).



FIXED: Użycie drugiej metody powoduje zastąpienie zawartości bieżącej linii kodem z szablonu !

W przypadku gdy CPDev został instalowany/aktualizowany z inną nazwą użytkownika niż obecnie używana, to należy uzupełnić brakujący element w konfiguracji aplikacji dla tego konta użytkownika poprzez uruchomienie programu cpdev.exe z następującymi parametrami:

```
cpdev.exe --gu-set-conf "CPDev.ST.STColorized.TemplateFile=sC:\Program Files\CPDev\Szablony\Edytor_ST.xml"
```

Niestety w chwili obecnej nie ma tej opcji wymienionej wśród okien konfiguracji, i obecnie (wersja 1.0.0.9) jest to jedyna metoda na dopisanie brakującej konfiguracji. Istnieje jeszcze druga, ale działa tylko w trybie serwisowym, który też się uaktywnia parametrem linii poleceń.

Analogicznie następuje zmiana pliku z szablonami na inny. Nowy plik szablonów zostanie wczytany przy tworzeniu nowego okna z edytorem pojedynczym kolorowanym.

Konfiguracja zawartości pliku szablonów

Plik szablonów edytora jest standardowym plikiem XML, którego struktura zostanie omówiona w tej części dokumentacji technicznej.

Plik można edytować dowolnym edytorem tekstowym, posiadającym kodowanie UTF-8 (np. Notatnik w Windows 2000 lub późniejszy), lub innym narzędziem do modyfikacji plików XML.

Cały dokument zawarty jest w elementach CPDEV (podobnie jak wszystkie inne XML-owe pliki pakietu). W drzewie CPDEV dołączony jest element ST_TEMPLATES, który zawiera szablony do kodów w języku ST. Każdy szablon jest zapisany między znacznikami <ITEM> oraz </ITEM> i symbolizuje pojedynczą pozycję w menu 'Szablony'. Element ITEM zawiera następujące podelementy:

SHORT_KEY – skrót, czyli nazwa którą można użyć wpisując z klawiatury rozkaz rozwinięcia szablonu.

NAME – rozbudowany element zawierający nazwę ustawienia w różnych językach. Regionalne nazwy definiowane są w elementach LCID, atrybutami ich są kody języków (value) oraz domyślność (default) w przypadku używania nie zaimplementowanego języka. CPDev wybiera również tą nazwę, gdy kody języka są identyczne, a nie zgadzają się tylko kody ustawień regionalnych; ten wybór ma pierwszeństwo po właściwym dopasowaniu, a przed domyślnością. W zawartości elementu LCID wpisujemy nazwę opisową (opcjonalnie nazwa może być otoczona sekcją CDATA).

CURSOR_POS – jednoznacznie definiuje tekst, który ma posłużyć na położenie kursora po użyciu szablonu. Taki tekst musi się pojawić tylko 1 raz w elemencie CODE.

CODE – rozwinięcie w postaci kodu które zostanie wstawione w miejscu występowania kursora. Standardowo otoczone są one sekcjami CDATA.

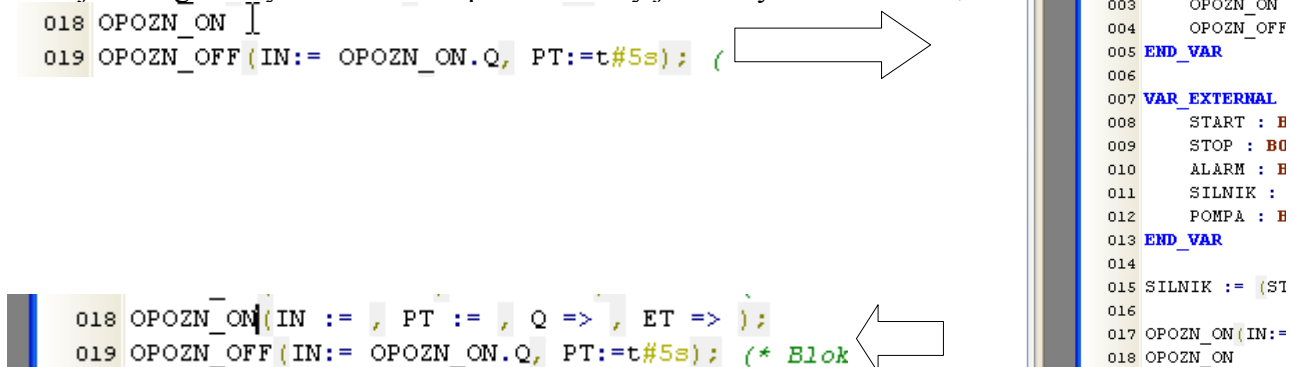
Poniższy przykład pokazuje w jaki sposób jest zdefiniowane pojedynczy szablon w pliku:

```
<ITEM>
  <SHORT_KEY>forc</SHORT_KEY>
  <NAME>
    <LCID value="0x0409" default="1">Template FOR instruction</LCID>
    <LCID value="0x0415" default="0">Szablon instrukcji FOR</LCID>
  </NAME>
  <CURSOR_POS>|</CURSOR_POS>
  <CODE>
    <![CDATA[FOR | := TO BY DO
END_FOR]]>
  </CODE>
</ITEM>
```

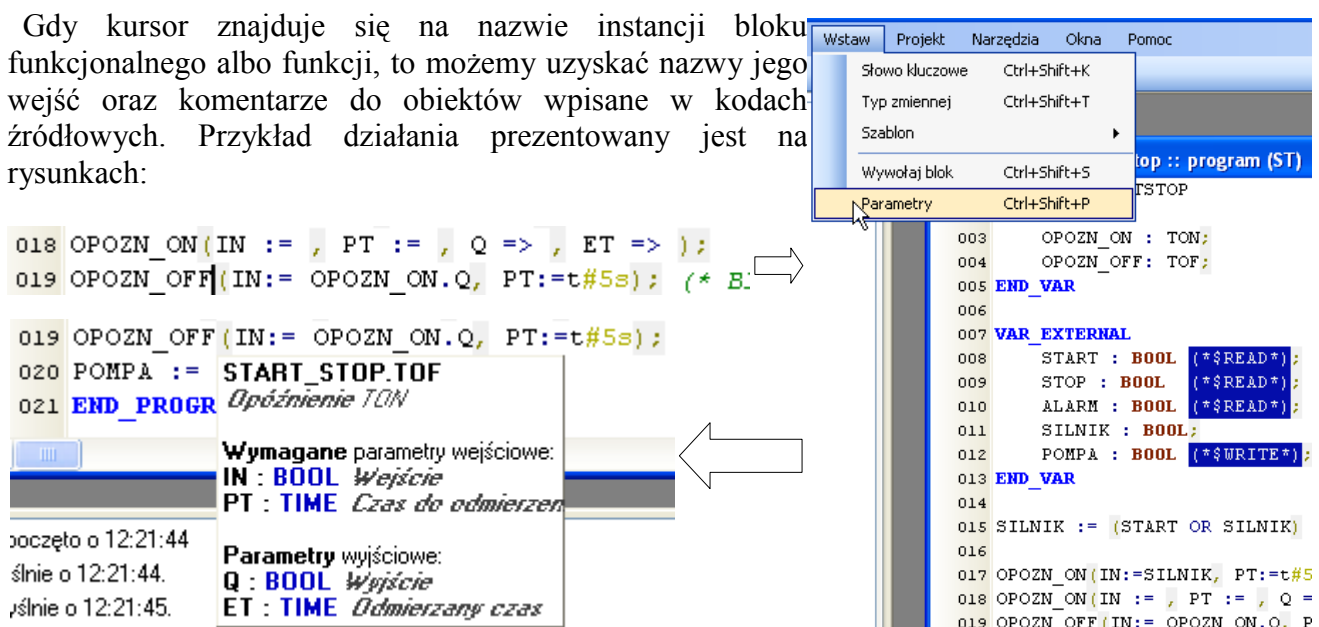
Nowości w wersji 1.0.0.9

Poniższe ułatwienia pozwalają na szybkie zorientowanie się w aktualnie edytowanym kodzie źródłowym jakie wejścia/wyjścia ma dany blok funkcjonalny czy funkcja oraz na szybkie utworzenie wywołania bloku funkcjonalnego.

Po wpisaniu nazwy instancji bloku funkcjonalnego (gdy znamy tylko fragment nazwy, to w celu jej uzupełnienia możemy posłużyć się dobrze już znaną podpowiadzią kontekstową) wybieramy z menu 'Wstaw Wywołaj blok' lub używamy skrótu klawiszowego [Ctrl]+[Shift]+S, a otrzymamy wstawiony szablon z wejściami oraz wyjściami bloku funkcjonalnego. Przykład działania prezentowany jest na rysunkach:



Rozwinięcie wywołania bloku funkcjonalnego składa się z nawiasu '(', nazw wejść bloku funkcjonalnego po których następuje ciąg podstawienia ':='. Parametry opcjonalne mają po podstawieniu wpisaną domyślną wartość. Po wyjściach bloku funkcjonalnego występują podstawienia '=>', które w razie zbyteczności mogą zostać usunięte przez programistę. Kolejne parametry rozdzielane są przecinkiem. Rozwinięcie kończy znak nawiasu ')' i średnik ';'.
Gdy kursor znajduje się na nazwie instancji bloku funkcjonalnego albo funkcji, to możemy uzyskać nazwy jego wejść oraz komentarze do obiektów wpisane w kodach źródłowych. Przykład działania prezentowany jest na rysunkach:



BUG: Z powodu błędu w kontrolce pakietu SynEdit szerokość tej podpowiedzi jest stała, co może powodować że zbyt długie komentarze nie zmieszczą się w obszarze wyświetlającym.

Okno podpowiedzi parametrów bloku funkcjonalnego zawiera:

- pełną (długą) nazwę bloku funkcjonalnego,
- komentarz,
- nazwy parametrów wejściowych wraz z ich typami oraz komentarzami,

- nazwy parametrów wejściowo-wyjściowych z ich typami oraz komentarzami,
- nazwy parametrów wyjściowych.

Wejściowe parametry opcjonalne mają również podane wartości domyślne. Podpowiedź znika po naciśnięciu klawisza [Esc], lub kliknięciu myszą.