Debuggowanie w VS2010

Rafał Zarębski

Breakpointy

- F5 – Start Debugging/Continue
- F9 – Place breakpoint
- F10 – Step over
- F11 – Step into

Breakpointy

Set Next Statement

- Ustawia następna instrukcje która będzie wykonana
- Można cofać program, pomijać fragmenty kodu
- Ctr-Shift-F10 lub drag the arrow

Breakpointy

Etykiety dla breakpointów

- Pozwalają porządkować breakpointy w większych projektach
- Można filtrować breakpointy i wykonywać na nich operacje (okno breakpoints)

Breakpointy

Conditional Breakpoint

- Program zatrzyma się jeśli spełniony będzie jakiś zadany warunek
- Intellisense przy zadawaniu warunku
- Is true, Is Changed

Breakpointy

- Hit Count: sprawdza ile razy dany breakpoint został napotkany
- When hit: Pozwala wypisać wiadomość w output lub uruchomić makro
- Export/Import, format xml

Data Tip

- Podczas debugowania daje możliwość podglądania wartości zmiennych.
- Pozwala oglądanie nie tylko zmiennych typów prostych ale również złożonych obiektów
- Można „przypinać” sobie Data Tipy tak aby pracować na nich
- Można dodawać komentarze

Data Tip

- Można zmieniać wartości zmiennych podczas działania programu
- Import/Export, format xml

Watches

- Kolejny sposób na analizowanie wartości zmiennych
- Do 4 okien z watchami
- Okno Locals zawiera watche dla zmiennych lokalnych
- Okno Autos zawiera watche dla zmiennych ważnych z punktu widzenia obecnie wykonywanego polecenia

Watches

- Zakładać watch można albo klikając na kod, albo ręcznie wpisując wyrażenie
- Działa Intellisense
- Quick-Watch

Immediate Window

- Okno w którym w każdej chwili można wykonać kawałek kodu. W przypadku kiedy aplikacja zatrzymana jest w trakcie wykonywania można edytować zmienne w programie. Działa Intellisense.

Inne okienka

- Call Stack:
  Okienko pokazujące stos wywołań funkcji. Zawiera nazwę funkcji język, w którym została napisana oraz dodatkowe informacje między innymi nr linii.
- Registers: pokazuje zawartość rejestrów.
- Memory: pokazuje pamięć programu.

IntelliTrace

W porównaniu z tradycyjnymi debugerami debugowanie przy pomocy IntelliTrace, dostępne w programie Microsoft Visual Studio 2010 Ultimate, zapewnia ulepszony obraz aplikacji. Tradycyjne debugery pokazują stan aplikacji w czasie bieżącym oraz ograniczone informacje na temat zdarzeń, jakie miały miejsce w przeszłości. Należy zatem wywnioskować zdarzenia, które miały miejsce w przeszłości, na podstawie stanu aplikacji w czasie bieżącym, lub ponownie uruchomić aplikację, w celu odtworzenia minionych zdarzeń. Dzięki IntelliTrace można zobaczyć zdarzenia, które miały miejsce w przeszłości oraz kontekst ich wystąpienia. Redukuje to liczbę ponownych uruchomień, których wymaga się, w celu debugowania aplikacji oraz możliwość, że błąd nie zdoła się odtworzyć, gdy ponownie uruchomimy aplikację.

IntelliTrace

Przykłady zastosowań:

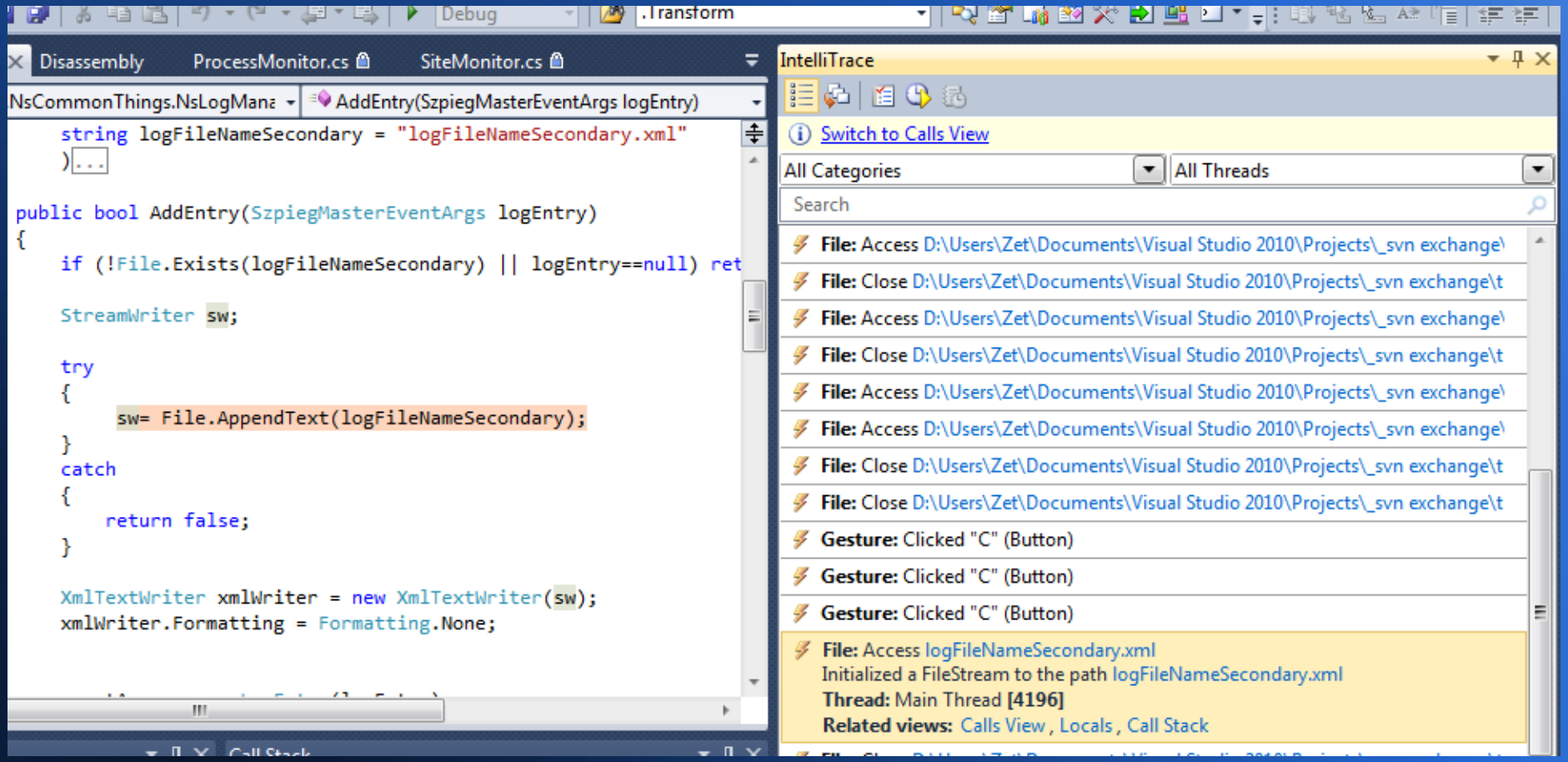
- Breakpointy
- Wyjątki
- Błędy nie powtarzalne

IntelliTrace

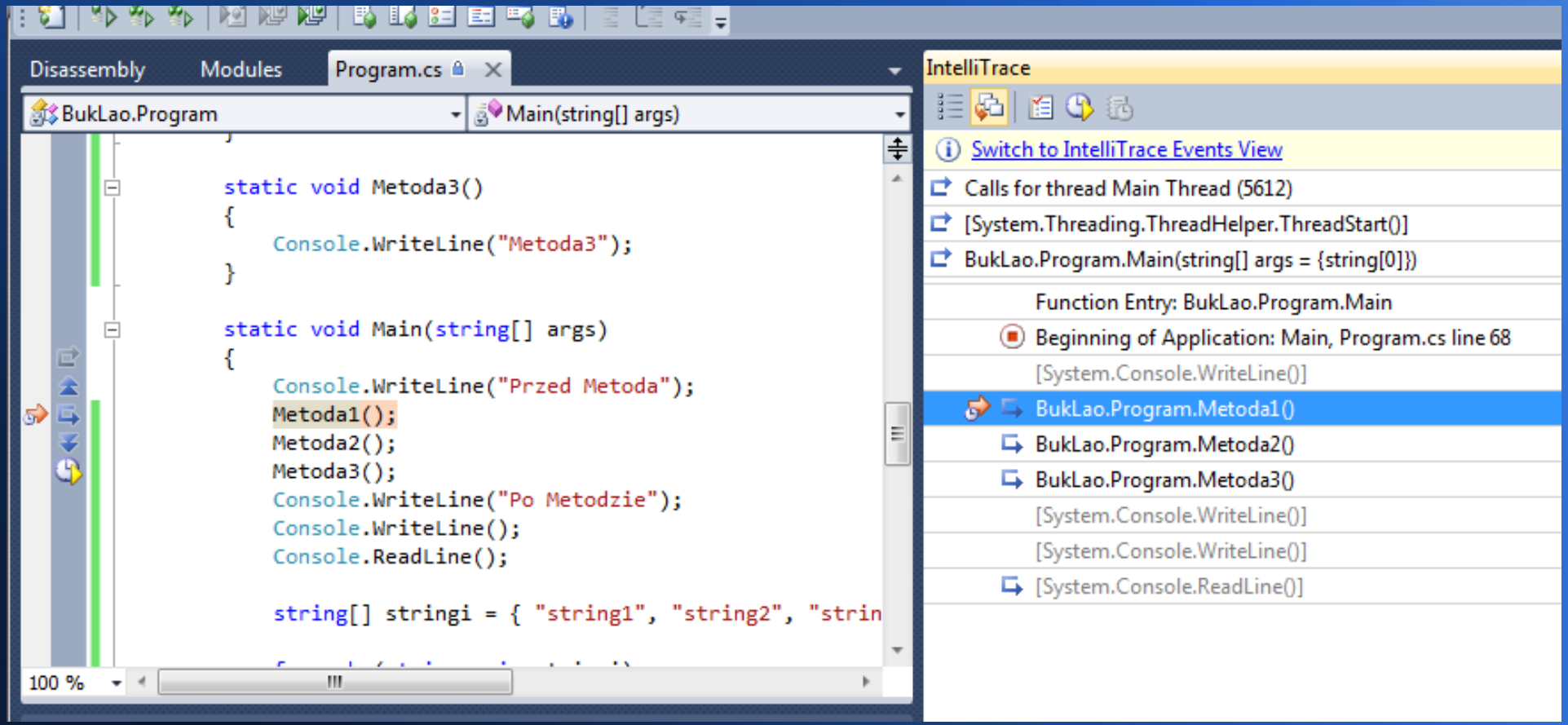
Zbieranie danych domyślnie włączone

- Zdarzenia debuggera
- Zdarzenia wyjątków
- Zdarzenia strukturowe

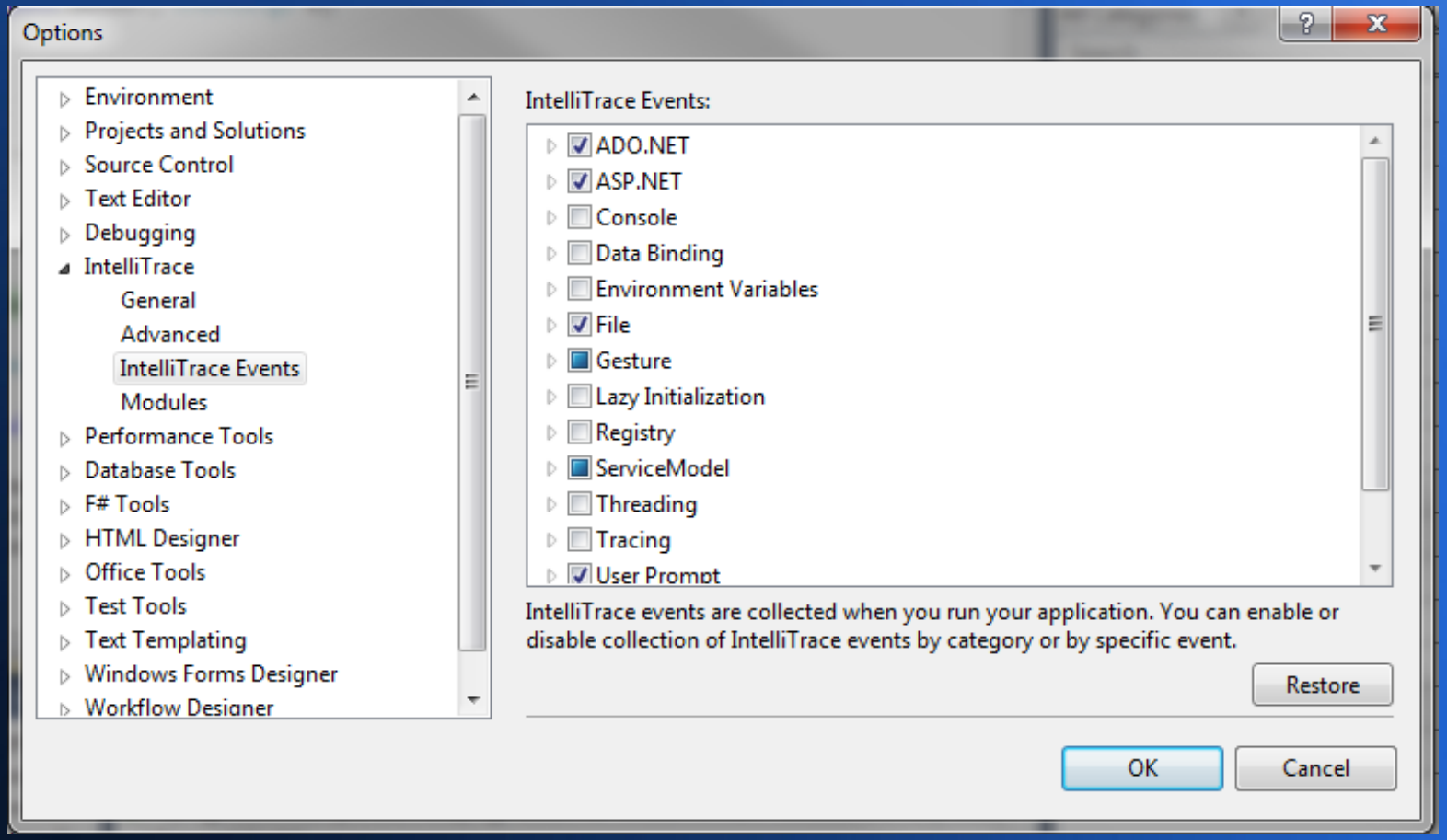
IntelliTrace



IntelliTrace



IntelliTrace



Obsługiwane aplikacje i scenariusze debugowania

IntelliTrace obsługuje debugowanie aplikacji Visual Basic oraz C#, które korzystają z .NET v. 2.0, 3.0, 3.5, lub 4. Możliwe jest debugowanie większości aplikacji, w tym aplikacji utworzonych przy użyciu ASP.NET, Windows Forms, WPF, Windows Workflow oraz WCF. IntelliTrace nie obsługuje debugowania C++, skryptów, ani innych języków. Debugowanie aplikacji F# obsługiwane jest eksperymentalnie.

KONIEC

- ok