

# CodeRush for Roslyn Shortcuts

## Clipboard

Feature	Shortcut	Context
Smart Copy	<b>Ctrl+C</b> or <b>Ctrl+Insert</b>	No selection, caret on token
Smart Cut	<b>Ctrl+X</b> or <b>Shift+Delete</b>	No selection, caret on token

## Defining Selections

Feature	Shortcut	Context
Selection Increase	<b>Num+</b> or <b>Ctrl+W</b>	Editor has focus
Selection Decrease	<b>Num-</b> or <b>Ctrl+Shift+W</b>	Editor has focus
Select CamelCase Left	<b>Alt+Shift+Left</b>	Editor has focus
Select CamelCase Right	<b>Alt+Shift+Right</b>	Editor has focus

## Wrapping Selections\*

Feature	Shortcut	Context
Comment/Uncomment	<b>/</b> or <b>'</b> (VB)	Multi-line selection
Block begin/end	<b>B</b> or <b>Shift+[</b>	Multi-line selection
try/catch	<b>C</b>	Multi-line selection
try/finally	<b>F</b>	Multi-line selection
try/catch/finally	<b>T</b>	Multi-line selection
Parens	<b>Shift+0</b> or <b>Shift+9</b>	Any selection
Not Parens	<b>Shift+1</b>	Any selection

\*Enable in the CodeRush Shortcuts options page. Search for "Embed" and enabled shortcuts as desired.

## Code Generation & Refactoring

Feature	Shortcut	Context
Refactor/Declare	<b>Ctrl+`</b> or <b>Num 0</b>	Editor has focus
Expand Template	<b>Space</b> or <b>Tab</b>	Caret after template
Cycle Scope Up	<b>Alt+Up</b>	At member/type declaration
Cycle Scope Down	<b>Alt+Down</b>	At member/type declaration
Duplicate Line	<b>Shift+Enter</b>	Editor has focus
Camel Delete Left	<b>Alt+Backspace</b>	Caret in CamelCase token
Camel Delete Right	<b>Alt+Delete</b>	Caret in CamelCase token

### Want to change a shortcut?

It's easy. From the CodeRush menu, select **Shortcuts** to bring up the Shortcuts options page.

## Creating Types

Element	Template
Class	<b>c</b>
Interface	<b>i</b>
Struct	<b>s</b>
Enum	<b>e</b>
Abstract Class	<b>a</b>
Delegate	<b>d</b>
Exception	<b>x</b>
Test Fixture	<b>t</b>
Test Method	<b>t</b>
Create Constructor	<b>cc</b>

To create **members** and **variables**, use *type mnemonics* (see next page).

## Navigation

Feature	Shortcut	Context
Highlight References	<b>Ctrl+Alt+U</b>	Caret on identifier
Show All References	<b>Shift+F12</b>	Caret on identifier
Tab to Next Reference	<b>Tab</b>	Caret inside identifier
Tab to Previous Reference	<b>Shift+Tab</b>	Caret inside identifier
Contextual Jump To	<b>Ctrl+Alt+N</b> or <b>Num Del</b>	Editor has focus
Jump to Symbol	<b>Ctrl+Shift+Q</b> or <b>Num Enter</b>	Editor has focus
Jump to File	<b>Ctrl+Alt+F</b>	Available anywhere
Drop Marker	<b>Alt+Home</b>	Editor has focus
Collect Marker	<b>Escape</b> or <b>Alt+End</b>	Editor has focus, marker dropped
Swap Markers	<b>Alt+Shift+Home</b>	Editor has focus, marker dropped
Toggle Bookmark	<b>Ctrl+Alt+{Number}</b>	Editor has focus
Go to Bookmark	<b>Alt+{Number}</b>	Editor has focus
Toggle Next Bookmark	<b>Ctrl+Num+</b>	Editor has focus
Go to Recent Bookmark	<b>Ctrl+Num-</b>	Editor has focus
Camel Case Nav Left	<b>Alt+Left</b>	Caret on CamelCase token
Camel Case Nav Right	<b>Alt+Right</b>	Caret on CamelCase token

## Debugging

Feature	Shortcut	Context
Toggle Temporary Breakpoint	<b>Alt+F9</b>	Debugging, editor has focus
Step Into Member	<b>Ctrl+Shift+F11</b>	Debugging, caret on member reference
Show Expression Explorer	<b>Alt+Down</b>	Debugging, caret inside expression
Toggle Debug Visualizer	<b>Ctrl+Alt+D</b>	Debugging, editor has focus
Refresh/Evaluate Expression	<b>Ctrl+R</b>	Debugging, editor has focus

## Test Runner

Feature	Shortcut	Context
Run active test	<b>Ctrl+T, R</b>	Caret inside test method
Debug active test	<b>Ctrl+T, D</b>	Caret inside test method
Run all tests in file	<b>Ctrl+T, F</b>	Inside file containing a test class
Show unit test runner	<b>Ctrl+T, T</b>	Available anywhere
Step Into Member	<b>Ctrl+Shift+F12</b>	Debugging, caret on reference

## CodeRush Templates

Expand the templates listed here with the **spacebar** or the **Tab** key (depending on your settings). The **User Guide** shows how to create your own custom templates.

### Flow

Element	Template
For loops	<b>fr</b>
For Each	<b>fe</b>
While loop	<b>w</b>
Do While	<b>dw</b>
Switch	<b>sw</b>
Select	<b>se</b>
Throw new exception	<b>tne</b>
Using statement	<b>u</b>
Block begin/end	<b>b</b>

### If/else

Element	Template
If statement	<b>if</b>
If not ...	<b>ifn</b>
Else	<b>l</b>
Else If ...	<b>lf</b>

### Try

Element	Template
try/catch	<b>tc</b>
try/finally	<b>tf</b>
try/catch/finally	<b>tcf</b>

# Members and Variables

Element	Shortcut*
Variable (parameter, local, field)	V or v
New Instance	N or n
Read-only Fields	O or o
Methods	M or m
Properties	P or p
Read-only	R or r
Write-only	W or w
Auto-implemented	A or a
Type Reference	t
Constants	q
Typecast (in code blocks)	c
Class	c

\*Uppercase letters expand to **static** members.

# Type Shortcuts



## System

Type	Shortcut
Boolean	b
Byte	by
Char	c
DateTime	d8
Decimal	de
Double	d
EventArgs	ea
Exception	x
Guid	g
Int16	sh
Int32	i
Int64	l
IntPtr	ip
Object	o
Single	si
String	s
Type	ty
UInt32	u

## System.Drawing

Type	Shortcut
Bitmap	bt
Brush	br
Color	cl
Font	fn
Graphics	gx
Pen	pn
Point	p
PointF	pf
Rectangle	r
RectangleF	rf
Size	z
SizeF	zf

## System.Windows

Type	Shortcut
DependencyObject	do
DependencyProperty	dp
Point	p
Rect	r
Size	z
Thickness	th
Vector	vi

## System.Windows.Media

Type	Shortcut
Brush	br
Color	cl
FontFamily	ff
GeometryDrawing	gd
GeometryGroup	gg
GradientBrush	gb
HitTestResult	htr
LinearGradientBrush	lgb
Matrix	mx
MatrixTransform	mt
MediaPlayer	mp
Pen	pn
RotateTransform	rt
ScaleTransform	sct
SkewTransform	skt
TransformGroup	tg
TranslateTransform	tt
Visual	vi
VisualBrush	vb

## System.IO

Type	Shortcut
StreamReader	sr
StreamWriter	sw

Follow the templates above with a Type shortcut. For example, on an empty line inside a class or method, “**nb**” followed by **space bar** creates a **new** initialized **Boolean** instance. **m** by itself creates a method with no return value.

## System.Net

Type	Shortcut
FileWebRequest	fwq
FileWebResponse	fwr
HttpWebRequest	hwq
HttpWebResponse	hwr
WebRequest	wq
WebResponse	wr

## System.Data

Type	Shortcut
DataRow	dr
DataRowView	drv
DataSet	ds
DataTable	dt
DataRowView	dv

## System.Data.SqlClient

Type	Shortcut
SqlClientPermission	qcp
SqlCommand	qc
SqlCommandBuilder	qcb
SqlConnection	qn
SqlDataAdapter	qda
SqlDataReader	qdr
SqlParameter	qp
SqlParameterCollection	qpc
SqlTransaction	qt

## System.Threading

Type	Shortcut
ManualResetEvent	mre
Monitor	mo
Thread	t
ThreadStart	ts

## XPO

Type	Shortcut
AggregateOperand	xao
BetweenOperator	xbwo
BinaryOperator	xbo
FunctionOperator	xfo
GroupOperator	xgo
InOperator	xio
UnitOfWork	xu
Session	xss
UnaryOperator	xuo
XPCollection	xcl

## Microsoft.SharePoint

Type	Shortcut
SPContentType	spct
SPContext.Current.Site	spcs
SPContext.Current.Web	spcw
SPField	spf
SPList	spl
SPListItem	spli
SPSite	sps
SPWeb	spw

## System.Xml

Type	Shortcut
XmlAttribute	xa
XmlComment	xc
XmlDocument	xd
XmlDocumentType	xdt
XmlElement	xe
XmlException	xx
XPathDocument	xpd
XmlNode	xn
XmlSerializer	xs
XmlText	xt
XmlTextReader	xtr
XmlTextWriter	xtw
XmlWriter	xw

## Other types

Type	Shortcut
{Active Type}	/
{Paste}	\
GregorianCalendar	gc
Regex	rx
StringBuilder	sb

## System.Collections

Type	Shortcut
ArrayList	a
Hashtable	h
NameValueCollection	nvc
Queue	q
Stack	k
StringCollection	sc

## Generic Collections – one parameter

Type	Shortcut
BindingList	bl
Collection	c
Comparer	cmp
EqualityComparer	ec
ICollection	ic
IEnumerable	ie
LinkedList	ll
LinkedListNode	lln
List	l
Queue	q
ReadOnlyCollection	roc
Stack	k
XPCollection	xcl

Follow with a “.” and then the shortcut for the type of the parameter. For example, “**nl.i**” creates a new List of integers.

## Generic Collections – two parameters

Type	Shortcut
Dictionary	d
KeyValuePair	kvp
SortedDictionary	sd
SortedList	sl

Follow these shortcuts with a “.” and then the shortcuts for the types of the parameters separated by a comma. For example, “**nd.s,i**” creates a new Dictionary keyed by strings and holding integers.

## Tips

- **Shift+Space** suppresses template expansion
- **t, f, & n** for true, false & null.
- **r, rt, rf, and rn** return values.
- Inside methods, templates like **fe, fri, c{TypeShortcut}, st, sf, and sn** work intelligently with identifiers copied to the clipboard.
- Use **{TypeShortcut}.e** to get a **.Empty** reference for a struct (like **s.e** or **p.e**).