

GMac Command Names	C# Member Name	C# Member Signatures	Service
	Ipr.Root	GMac.GMacAST.AstRoot Root { get; }	The GMacAST root node
	Ipr.Progress	TextComposerLib.Progress.ProgressComposer Progress { get; }	The progress reporting object
	Ipr.Output	GMac.GMacScripting.GMacScriptOutput Output { get; }	The script output object
	Ipr.Output.Log	TextComposerLib.Linear.LinearComposer Log { get; }	The script log object
	Ipr.LocalVariables	IEnumerable<GMac.GMacAST.Symbols.AstLocalVariable> LocalVariables { get; }	A list of all GMac local variables in the main body of the script
	Ipr.Commands	IEnumerable<GMac.GMacAST.Commands.AstCommand> Commands { get; }	A list of all GMac commands in the main body of the script
Reset, rst	Ipr.Reset()	void Reset(GMac.GMacAST.Symbols.AstSymbol symbol) void Reset(GMac.GMacAST.Symbols.AstFrame symbol) void Reset(GMac.GMacAST.Symbols.AstNamespace symbol) void Reset(string symbolName)	Reset main scope (a namespace or a frame) for GMac commands execution
Open, opn	Ipr.OpenScope()	void OpenScope(GMac.GMacAST.Symbols.AstSymbol symbol) void OpenScope(GMac.GMacAST.Symbols.AstNamespace symbol) void OpenScope(GMac.GMacAST.Symbols.AstFrame symbol) void OpenScope(string symbolName) void OpenScope(params string[] symbols) void OpenScope(params GMac.GMacAST.Symbols.AstSymbol[] symbols)	Open a scope for GMac symbols referencing
Close, cls	Ipr.CloseScope()	void CloseScope() void CloseScope(GMac.GMacAST.Symbols.AstNamespace symbol) void CloseScope(GMac.GMacAST.Symbols.AstFrame symbol) void CloseScope(GMac.GMacAST.Symbols.AstSymbol symbol) void CloseScope(string symbolName) void CloseScope(params string[] symbolNames) void CloseScope(params GMac.GMacAST.Symbols.AstSymbol[] symbols)	Close a scope \all scopes for GMac symbols referencing
Namespace, ns	Ipr.Namespace()	GMac.GMacAST.Symbols.AstNamespace Namespace(string symbolName)	Find a namespace GMacAST node given the namespace name
Frame, fm	Ipr.Frame()	GMac.GMacAST.Symbols.AstFrame Frame(string symbolName)	Find a frame GMacAST node given the frame name
BasisVector, bv	Ipr.BasisVector()	GMac.GMacAST.Symbols.AstFrameBasisVector BasisVector(string symbolName)	Find a basis vector GMacAST node given the basis vector name
Multivector, mv	Ipr.FrameMultivector()	GMac.GMacAST.Symbols.AstFrameMultivector FrameMultivector(string symbolName)	Find a frame multivector type GMacAST node given the multivector type name
Subspace, ss	Ipr.Subspace()	GMac.GMacAST.Symbols.AstFrameSubspace Subspace(string symbolName)	Find a subspace GMacAST node given the subspace name
Constant, ct	Ipr.Constant()	GMac.GMacAST.Symbols.AstConstant Constant(string symbolName)	Find a constant GMacAST node given the constant name
Structure, st	Ipr.Structure()	GMac.GMacAST.Symbols.AstStructure Structure(string symbolName)	Find a structure GMacAST node given the structure name
Macro, mc	Ipr.Macro()	GMac.GMacAST.Symbols.AstMacro Macro(string symbolName)	Find a macro GMacAST node given the macro name
Variable, lv	Ipr.LocalVariable()	GMac.GMacAST.Symbols.AstLocalVariable LocalVariable(string symbolName)	Find a local variable GMacAST node given the local variable name
Symbol, sy	Ipr.Symbol()	GMac.GMacAST.Symbols.AstSymbol Symbol(string symbolName)	Find a GMac symbol GMacAST node (namespace, frame, structure, macro, etc.) given the symbol name
Type, ty	Ipr.GMacType()	GMac.GMacAST.AstType GMacType(string symbolName)	Find a GMac type GMacAST node given the type name
ValueAccess, vla	Ipr.ValueAccess()	GMac.GMacAST.Expressions.AstDatastoreValueAccess ValueAccess(string valueAccessName) GMac.GMacAST.Expressions.AstDatastoreValueAccess ValueAccess(string valueAccessName, GMac.GMacAST.AstType requiredType)	Construct a GMacAST datastore value access object based on the given string
TypeOf, tyo	Ipr.GMacTypeOf()	GMac.GMacAST.AstType GMacTypeOf(string valueAccessName)	Return the GMacAST type node of the given datastore value access
ValueExists, vie	Ipr.ValueAccessExists()	System.Boolean ValueAccessExists(string valueAccessName)	Returns true if the given datastore value access exists
Expression, expr	Ipr.Expression()	GMac.GMacAST.Expressions.AstExpression Expression(string exprText)	Compile the given string into a GMacAST expression
Declare, declr	Ipr.Declare()	void Declare(string varName, string varTypeName) void Declare(string varName, GMac.GMacAST.IAstObjectWithType varType)	Declare a Gmac local variable with a given type
Assign, asn	Ipr.Assign()	void Assign(string lhsName, string rhsExprText) void Assign(string lhsName, GMac.GMacAST.IAstObjectWithExpression rhsExpr) void Assign(string lhsName, string lhsTypeName, string rhsExprText) void Assign(string lhsName, GMac.GMacAST.IAstObjectWithType lhsType, string rhsExprText) void Assign(string lhsName, string lhsTypeName, GMac.GMacAST.IAstObjectWithExpression rhsExpr) void Assign(string lhsName, GMac.GMacAST.IAstObjectWithType lhsType, GMac.GMacAST.IAstObjectWithExpression rhsExpr) void Assign(GMac.GMacAST.Expressions.AstDatastoreValueAccess lhsValueAccess, string rhsExprText) void Assign(GMac.GMacAST.Expressions.AstDatastoreValueAccess lhsValueAccess, GMac.GMacAST.IAstObjectWithExpression rhsExpr)	Assign a value of a given expression to a GMac local variable or to part of a local variable.
Execute, exec	Ipr.Execute()	void Execute(string commandText)	Execute a set of GMac commands given as a GMacDSL code string
ValueOf, vlo	Ipr.ValueOf()	GMac.GMacAST.Expressions.AstValue ValueOf(string valueAccessName)	Finds the GMacAST value associated with a datastore value access
Evaluate, eval	Ipr.Evaluate()	GMac.GMacAST.Expressions.AstValue Evaluate(GMac.GMacAST.Expressions.AstExpression expr) GMac.GMacAST.Expressions.AstValue Evaluate(string exprText)	Evaluate the given string into a GMacAST expression
SubspaceToMultivector, ss2mv	Ipr.SubspaceToMultivector()	GMac.GMacAST.Expressions.AstValueMultivector SubspaceToMultivector(string subspaceName, string templateText) GMac.GMacAST.Expressions.AstValueMultivector SubspaceToMultivector(string subspaceName, TextComposerLib.StringSequenceTemplate varNameTemplate) GMac.GMacAST.Expressions.AstValueMultivector SubspaceToMultivector(string subspaceName, Func<int,string> basisBladeToVarName) GMac.GMacAST.Expressions.AstValueMultivector SubspaceToMultivector(string subspaceName, Func<GMac.GMacAST.Symbols.AstFrame,int,string> basisBladeToVarName) GMac.GMacAST.Expressions.AstValueMultivector SubspaceToMultivector(string subspaceName, Func<GMac.GMacAST.Symbols.AstFrameBasisBlade,string> basisBladeToVarName)	Construct a symbolic multivector given a subspace
ComputeToExpr, cte	Ipr.ComputeToExpr()	Wolfram.NETLink.Expr ComputeToExpr(string codeText)	Use Mathematica to evaluate the given string into a symbolic Expr object
ComputeToString, cts	Ipr.ComputeToString()	string ComputeToString(string codeText)	Use Mathematica to evaluate the given string into a string value
ComputeToInputForm, ctif	Ipr.ComputeToInputForm()	string ComputeToInputForm(string codeText)	Use Mathematica to evaluate the given string into a string in input form
ComputeToOutputForm, ctof	Ipr.ComputeToOutputForm()	string ComputeToOutputForm(string codeText)	Use Mathematica to evaluate the given string into a string in output form
ComputeToTypeset, ctt	Ipr.ComputeToTypeset()	System.Drawing.Image ComputeToTypeset(string codeText)	Use Mathematica to evaluate the given string into a string in typeset form
ComputeToImage, cti	Ipr.ComputeToImage()	System.Drawing.Image ComputeToImage(string codeText)	Use Mathematica to evaluate the given string into an image
AsString, as	Ipr.AsString()	string AsString(GMac.GMacAST.Expressions.AstValueMultivector value) string AsString(GMac.GMacAST.Expressions.AstValue value)	Describe the given GMacAST value as a string
Store, str	Ipr.Output.Store()	GMac.GMacScripting.GMacScriptOutputItem Store(string title, string description, string item) GMac.GMacScripting.GMacScriptOutputItem Store(string title, string item) GMac.GMacScripting.GMacScriptOutputItem Store(string title, string description, System.Drawing.Image item) GMac.GMacScripting.GMacScriptOutputItem Store(string title, System.Drawing.Image item) GMac.GMacScripting.GMacScriptOutputItem Store(string title, string description, GMac.GMacAST.Expressions.AstValue item) GMac.GMacScripting.GMacScriptOutputItem Store(string title, GMac.GMacAST.Expressions.AstValue item)	Store the given GMacAST value, string, or image in the Output object of the script
	Ipr.Substitute()	string Substitute(string text, params object[] items)	Substitute the given items inside a text containing numbers between {} delimiters; similar to String.Format() method
IncIndent, ii	Ipr.Output.Log.IncreaseIndentation()	TextComposerLib.Linear.LinearComposer IncreaseIndentation() TextComposerLib.Linear.LinearComposer IncreaseIndentation(string indent)	Increase text indentation of the Log object
DecIndent, di	Ipr.Output.Log.DecreaseIndentation()	TextComposerLib.Linear.LinearComposer DecreaseIndentation()	Decrease text indentation of the Log object
Append, ap	Ipr.Output.Log.Append()	TextComposerLib.Linear.LinearComposer Append(string text) TextComposerLib.Linear.LinearComposer Append<T>(T text)	Append text to the last line of the Log object
AppendLine, apl	Ipr.Output.Log.AppendLine()	TextComposerLib.Linear.LinearComposer AppendLine() TextComposerLib.Linear.LinearComposer AppendLine(string text) TextComposerLib.Linear.LinearComposer AppendLine<T>(T text)	Append text and a line break to the last line of the Log object
AppendNewLine, apnl	Ipr.Output.Log.AppendNewLine()	TextComposerLib.Linear.LinearComposer AppendNewLine() TextComposerLib.Linear.LinearComposer AppendNewLine(string text) TextComposerLib.Linear.LinearComposer AppendNewLine<T>(T text)	Append a new line of text to the Log object
AppendAtNewLine, apantl	Ipr.Output.Log.AppendAtNewLine()	TextComposerLib.Linear.LinearComposer AppendLineAtNewLine() TextComposerLib.Linear.LinearComposer AppendLineAtNewLine(string text) TextComposerLib.Linear.LinearComposer AppendLineAtNewLine<T>(T text)	Append text to an empty line at the end of the Log object
AppendLineAtNewLine, apantl	Ipr.Output.Log.AppendLineAtNewLine()	TextComposerLib.Linear.LinearComposer AppendLineAtNewLine() TextComposerLib.Linear.LinearComposer AppendLineAtNewLine(string text) TextComposerLib.Linear.LinearComposer AppendLineAtNewLine<T>(T text)	Append a full line of text to an empty line at the end of the Log object
SetShortcuts, ssc	Ipr.SetShortcuts()	void SetShortcuts(string methodName, params string[] shortcuts)	Change the shortcut command names associated with some C# method
ResetShortcuts, rsc	Ipr.ResetShortcuts()	void ResetShortcuts()	Reset all shortcut command names to their defaults