

	PdfParser	
Reset	Reset(), FilterClear(), SectionsClear()	
Move	Left(), Right(), Up(), Down(), Above(), Below()	
Find	Find("test", "Test", "test2") FindPrev("total") Find(x=>x.Bold && x.Text.MatchesPattern("tot*"))	FindRegex(new Regex("[0-9]*")) FindPattern("chapter topic*") FindPage(12)
Filter	Filter(x=>x.Italic && x.Text.Length > 3) FilterText("total", "amount") FilterTextPattern("Total of*section") FilterIndex(54, 71)	FilterWindow(1, 1, 100, 100) FilterOffset(0, 0, 100, 50) FilterPage(3, 6) FilterEnclosedArea()
Token	Text, Font, Bold, Italic, Page, X, Y, Height, Width, DataType, Index	ToNumber(), ToDate()
Table / Grid	WhiteSpaceLimit, MaxRowHeight, MultiPage, IncludeUnmatchedCells	HeaderSeparatedByLine, ColumnsSeparatedByLines, RowsSeparatedByLines, ColumnBorders[]
Bookmarks	SetBookmark("pos1") GoBookmark("pos1")	DeleteBookmark("pos1") DeleteAllBookmarks()
Sections	Split(x=>x.Height==16) Split(x=>x.Height==16, x=>x.Text.Contains("Total"))	FilterSections(x=>x.Text.MatchesPattern("Chapt*")) FilterSections(Sections[0].Children)
	DataSetParser	
Current	Text, X, Y, Sheet, SheetName	
Find	Find("sometext") FindPrev("total", "Total") FindPattern("where is *")	FindSheet(6) FindSheet("Summary") FindSheetPattern("Summary of *")
Move	Left(), Right(), Up(), Down() - first non-empty Left(steps) - move specific number of cells	Above(), Below() - left-most non-empty cell
Table / Grid	Table() - table from current position Table(int headerRows) Table(int width, height) Table(int left, top, width, height, headerRows) Table(bool left, top, width, height, headerRows)	Grid() - same parameters as table (rowHeader=0) Table(2) Table(4, 6) - table 4x6 Table(2,0,12,5) - table 14x5, moving 2 cells left Table(false, false, true, true, 2)
	Table Extensions	
Columns	SelectColumns(2, 4, 6) SelectColumns("name", "price", "quantity") SelectColumnsPattern("total*", "name description") SelectColumns(x=>ColumnHeader.Contains("Name"))	DeleteColumns(1, 2, 5) DeleteColumns("field2", "field4") DeleteColumnsPattern("total*", "name description") DeleteColumns(x=>ColumnHeader.Contains("Amount"))
	SelectColumnsAs(allowMissingColumn, "f1", "name", "f2", "amount") SelectColumnsAs(allowMissingColumn, addMissingColumnAsEmpty, "f1", "name", "f2", "amount")	
	NameColumns("name", "price", "amount")	AddColumn("Amount", "0.0", typeof(string))
Rows	SelectRows("Convert(amount, System.Double)>0") SelectRows(x=>x[0].ToString().Contains("total")) SelectRowsContainingTextPattern("quant???")	DeleteRowsContainingText("Total", "total") DeleteRows(x=>x[0].ToString() == "")
Join	DataTable dt = dt1.Join(dt2, "name", "name") LeftJoin, FullJoin	dt = dt1.Union(dt2) - by column order dt = dt1.Union(dt2, true) - by column names
Group	DataTable dt = dt1.Distinct("name") table.Rollup("amount")	table.Transpose() table.Reverse()
Format	table.ToNumber(4, 5, 8) table.ToNumber("Amount", "Price")	table.ToDate(1, 4, 5) table.ToDate("event date")
Update	table.Update(x=>x[2]=x[0].ToString() + x[1].ToString()) table.Update(x=>x[1]="", x=>x[0].ToString().Contains("Total")) table.ReplaceText("Total", "") //replace text in all table cells	
Check	bool correctTable = table.HeaderMatches("name", "quantity", "amount") bool correctTable = table.ContainsColumns("name", "amount") bool correctTable = table.ContainsColumnsPattern("name description", "amount*") bool empty = table.HasEmptyColumns() bool empty = table.HasEmptyRows() bool contains = table.HasText("amount")	