

- Abs(number)
- ACos(number)
- AddSOAPRequestHeader(webservice,namespace,name,value,[mustUnderstand=false])
- AddSOAPResponseHeader(namespace,name,value,[mustUnderstand=false])
- AjaxLink(url)
- AjaxOnLoad(functionName)
- ApplicationStop()
- ArrayAppend(array,value,[merge=false])
- ArrayAvg(array)
- ArrayClear(array)
- ArrayContains(array,value)
- ArrayContainsNoCase(array,value)
- ArrayDelete(array,value,[indicateExists=false])
- ArrayDeleteAt(array,index)
- ArrayDeleteNoCase(array,value,[indicateExists=false])
- ArrayEach(array,callback)
- ArrayFilter(array,callback)
- ArrayFind(array,value)
- ArrayFind(array,callback)
- ArrayFindAll(array,value)
- ArrayFindAll(array,callback)
- ArrayFindAllNoCase(array,value)
- ArrayFindNoCase(array,value)
- ArrayFirst(array)
- ArrayGetMetaData(array)
- ArrayInsertAt(array,position,value)
- ArrayIsDefined(array,index)
- ArrayIsEmpty(array)
- ArrayLast(array)
- ArrayLen(array)
- ArrayMap(array,callback)
- ArrayMax(array)
- ArrayMin(array)
- ArrayNew(dimension,[isSynchronized=true])
- ArrayPrepend(array,value)
- ArrayReduce(array,callback,[initialValue=""])
- ArrayResize(array,size)
- ArraySet(array,start,end,value)
- ArraySetMetaData(array,metadata)
- ArraySlice(array,offset,[length=-1])
- ArraySort(array,callback)
- ArraySum(array,[ignoreEmpty=false])
- ArraySwap(array,position1,position2)

- `ArrayToList(array,[delimiter=,])`
- `Asc(string)`
- `ASin(number)`
- `Atn(number)`
- `BinaryDecode(string,encoding)`
- `BinaryEncode(binaryData,encoding)`
- `BitAnd(number1,number2)`
- `BitMaskClear(number,start,length)`
- `BitMaskRead(number,start,length)`
- `BitMaskSet(number,mask,start,length)`
- `BitNot(number)`
- `BitOr(number1,number2)`
- `BitSHLN(number,count)`
- `BitSHRN(number,count)`
- `BitXor(number1,number2)`
- `BooleanFormat(value)`
- `CacheGet(id,region)`
- `CacheGetAllIds(region,[includeExpired=true])`
- `CacheGetEngineProperties()`
- `CacheGetMetadata(id,[objectType=OBJECT,region=""])`
- `CacheGetProperties([region=""])`
- `CacheGetSession(objectType,[isKey=false])`
- `CacheIdExists(id,[region=""])`
- `CachePut(id,value,[timeSpan="",idleTime="",region="",throwOnError=false])`
- `CacheRegionExists(region)`
- `CacheRegionNew(region,[properties="",throwOnError=true])`
- `CacheRegionRemove(region)`
- `CacheRemove(id,[throwOnError=false,region="",exact=true])`
- `CacheRemoveAll([region=""])`
- `CacheSetProperties(properties,[region=""])`
- `CallStackDump(output)`
- `CallStackGet()`
- `Canonicalize(input,restrictMultiple,restrictMixed,[throwOnError=false])`
- `Ceiling(number)`
- `CharsetDecode(string,encoding)`
- `CharsetEncode(binaryData,encoding)`
- `Chr(number)`
- `CJustify(string,length)`
- `Compare(string1,string2)`
- `CompareNoCase(string1,string2)`
- `Cos(number)`
- `CreateDate(year,month,day)`
- `CreateDateTime(year,[month=1,day=1,hour=0,minute=0,second=0])`

- CreateDynamicProxy(cfc,interfaces)
- CreateObject(type,className)
- CreateObject(type,className,context,server,action,domain,username,password)
- CreateObject(type,className,context,server)
- CreateObject(type,className, PortName,server)
- CreateObject(type,className, PortName, Locale)
- CreateObject(type,className,context, Locale)
- CreateObject(type,className, Assembly,server)
- CreateObject(type,className, Assembly, Locale)
- CreateObject(type,className,context,server,DotnetPort)
- CreateObject(type,className,context,server,action)
- CreateObject(type,className,context,locale,action)
- CreateObject(type,className,context,locale,DotnetPort)
- CreateObject(type,className,PortName,server,DotnetPort)
- CreateObject(type,className,PortName,server,action)
- CreateObject(type,className,PortName,locale,action)
- CreateObject(type,className,PortName,locale,DotnetPort)
- CreateObject(type,className,Assembly,server,DotnetPort)
- CreateObject(type,className,Assembly,server,action)
- CreateObject(type,className,Assembly,locale,action)
- CreateObject(type,className,Assembly,locale,DotnetPort)
- CreateObject(type,className,context,server,DotnetPort)
- CreateObject(type,className,context,server,action)
- CreateObject(type,className,context,locale,action)
- CreateObject(type,className,context,locale,DotnetPort)
- CreateObject(type,className,PortName,server,DotnetPort,protocol,secure)
- CreateObject(type,className,PortName,server,action,protocol,secure)
- CreateObject(type,className,PortName,locale,action,protocol,secure)
- CreateObject(type,className,PortName,locale,DotnetPort,protocol,secure)
- CreateObject(type,className,Assembly,server,DotnetPort,protocol,secure)
- CreateObject(type,className,Assembly,server,action,protocol,secure)
- CreateObject(type,className,Assembly,locale,action,protocol,secure)
- CreateObject(type,className,Assembly,locale,DotnetPort,protocol,secure)
- CreateODBCDate(date)
- CreateODBCDateTime(date)
- CreateODBCTime(date)
- CreateTime(hour,[minute=0,second=0])
- CreateTimeSpan(days,hours,minutes,seconds)
- CreateUUID()
- CSRFGenerateToken([key="",forceNew=false])
- CSRFVerifyToken(token,[key=""])
- DateAdd(datepart,number,date)
- DateCompare(date1,date2,[datePart=s])

- DateConvert(conversionType,date)
- DateDiff(datepart,date1,date2)
- DateFormat(date,[mask=dd-mmm-yy])
- DatePart(datePart,date)
- DateTimeFormat(date,timezone,[mask=dd-MMM-yyyy HH:nn:ss])
- Day(date)
- DayOfWeek(date)
- DayOfWeekAsString(dayOfWeek,[locale=defaultLocale])
- DayOfYear(date)
- DaysInMonth(date)
- DaysInYear(date)
- DE(string)
- DecimalFormat(number)
- DecodeForHTML(string)
- DecodeFromURL(string)
- DecrementValue(number)
- Decrypt(string,key,[algorithm=CFMX\_COMPAT,encoding=UU,IV="",iterations=0])
- Decrypt(string,key,[algorithm=CFMX\_COMPAT,encoding=UU,Salt="",iterations=0])
- Decrypt(string,key,[algorithm=CFMX\_COMPAT])
- DecryptBinary(binaryData,key,[algorithm=CFMX\_COMPAT,prefix="",iterations=0])
- DeleteClientVariable(name)
- Deserialize(string,type,useCustomSerializer)
- DeserializeJSON(json,[strictMapping=true,useCustomSerializer=true])
- DeserializeXML(string,[useCustomSerializer=true])
- DirectoryCopy(source,destination,[recurse=false,filter=""])
- DirectoryCreate(path)
- DirectoryDelete(path,[recurse=false])
- DirectoryExists(path)
- DirectoryList(path,[recurse=false,listInfo=path,filter="",sort="",type=""])
- DirectoryRename(oldPath,newPath)
- DollarFormat(number)
- DotNetToCFTYPE(variableName)
- Duplicate(object)
- EncodeFor(type,value)
- EncodeForCSS(string,[canonicalize=false])
- EncodeForDN(string,[canonicalize=false])
- EncodeForHTML(string,[canonicalize=false])
- EncodeForHTMLAttribute(string,[canonicalize=false])
- EncodeForJavaScript(string,[canonicalize=false])
- EncodeForLDAP(string,[canonicalize=false])
- EncodeForURL(string,[canonicalize=false])
- EncodeForXML(string,[canonicalize=false])
- EncodeForXMLAttribute(string,[canonicalize=false])

- EncodeForXPath(string,[canonicalize=false])
- Encrypt(string,key,[algorithm=CFMX\_COMPAT,encoding=UU,IV=""])
- Encrypt(string,key,[algorithm=CFMX\_COMPAT,encoding=UU,Salt=""])
- EncryptBinary(binaryData,key,[algorithm=CFMX\_COMPAT,IV="",iterations=0])
- EncryptBinary(binaryData,key,[algorithm=CFMX\_COMPAT,Salt="",iterations=0])
- EntityDelete(entity)
- EntityLoad(entityName,[id="",unique="",options=""])
- EntityLoad(entityName,[id="",Order="",options=""])
- EntityLoad(entityName,[Filter="",unique="",options=""])
- EntityLoad(entityName,[Filter="",Order="",options=""])
- EntityLoadByExample(sampleEntity,[unique=false,matchCriteria=""])
- EntityLoadByPK(entity,id)
- EntityMerge(entity)
- EntityNew(entityName,[properties="",ignoreExtras=false])
- EntityReload(entity)
- EntitySave(entity,[forceInsert=false])
- EntityToQuery(entity,[name=""])
- Evaluate(expression)
- ExceptionKeyExists(exception,key)
- Exp(number)
- ExpandPath(path)
- FileAppend(filePath,data,charset)
- FileClose(file)
- FileCopy(source,destination)
- FileDelete(filePath)
- FileExists(filePath)
- FileGetMimeType(file,[strict=true])
- FileGetMimeType(filePath,[strict=true])
- FileIsEOF(file)
- FileMove(source,destination)
- FileOpen(filePath,[mode=read,charset="",seekable=false])
- FileRead(filePath,[charset=""])
- FileRead(filePath,[bufferSize=""])
- FileReadBinary(filePath)
- FileReadLine(file)
- FileSeek(file,position)
- FileSetAccessMode(filePath,mode)
- FileSetAttribute(filePath,attribute)
- FileSetLastModified(filePath,date)
- FileSkipBytes(file,skipCount)
- FileUpload(destination,[fileField="",mimeType="",onConflict=Error,strict=true])
- FileUploadAll(destination,[mimeType="",onConflict=Error,strict=true,continueOnError=false,errorVariable=""])

- `FileWrite(filePath,data,[charset=""])`
- `FileWriteLine(file,data)`
- `Find(substring,string,[start=1])`
- `FindNoCase(substring,string,[start=1])`
- `FindOneOf(set,string,[start=1])`
- `FirstDayOfMonth(date)`
- `Fix(number)`
- `Floor(number)`
- `FormatBaseN(number,radix)`
- `GeneratePBKDFKey(algorithm,passphrase,salt,iterations,keysize)`
- `GenerateSecretKey(algorithm,[keysize=-1])`
- `getAllTemplateCachelds(key)`
- `GetApplicationMetadata()`
- `GetAuthUser()`
- `GetBaseTagData(tagName,[level=1])`
- `GetBaseTagList(caller)`
- `GetBaseTemplatePath()`
- `GetClientVariablesList()`
- `GetComponentMetadata(path)`
- `GetContextRoot()`
- `GetCpuUsage([time=1000])`
- `GetCurrentTemplatePath()`
- `GetDirectoryFromPath(path)`
- `GetEncoding(scope)`
- `GetException(object)`
- `GetFileFromPath(filepath)`
- `GetFileInfo(filepath)`
- `GetFreeSpace(path)`
- `getFromTemplateCache(id,key)`
- `GetFunctionCalledName()`
- `GetFunctionList()`
- `GetGatewayHelper(gatewayID)`
- `GetHttpRequestData([includeBody=true])`
- `GetHttpTimeString([date=""])`
- `GetLocale()`
- `GetLocaleDisplayName([locale=defaultLocale,inlocale=""])`
- `GetLocalHostIP()`
- `GetMetadata(value,extendedInfo)`
- `GetMetricData(mode)`
- `GetPageContext()`
- `GetPrinterInfo([printer=""])`
- `GetPrinterList()`
- `GetProfileSections(path,[encoding=""])`

- `GetProfileString(path,section,entry,[encoding=""])`
- `GetReadableImageFormats()`
- `GetSafeHTML(string,[policyFile="",throwOnError=false])`
- `GetSOAPRequest([webservice=""])`
- `GetSOAPRequestHeader(namespace,name,[asXML=false])`
- `GetSOAPResponse(webservice)`
- `GetSOAPResponseHeader(webservice,namespace,name,[asXML=false])`
- `GetSystemFreeMemory()`
- `GetSystemTotalMemory()`
- `GetTempDirectory()`
- `GetTempFile(path,prefix)`
- `GetTemplatePath()`
- `GetTickCount()`
- `GetTimeZoneInfo()`
- `GetToken(string,index,[delimiter=""])`
- `GetTotalSpace(path)`
- `GetUserRoles()`
- `GetVFSMetadata(fileSystemType)`
- `GetWriteableImageFormats()`
- `Hash(string,[algorithm=CFMX_COMPAT,encoding="",additionalIterations=1])`
- `HMack(message,key,[algorithm="",encoding=""])`
- `Hour(date)`
- `HTMLCodeFormat(string,[version=2.0])`
- `HTMLEditFormat(string,[version=2.0])`
- `IIf(condition,expression1,expression2)`
- `ImageAddBorder(image,thickness,[color="",type=constant])`
- `ImageBlur(image,[blurRadius=3])`
- `ImageClearRect(image,x,y,width,height)`
- `ImageCopy(image,x,y,width,height,[dx=-999,dy=-999])`
- `ImageCreateCaptcha(height,width,text,[difficulty=low,fonts="",fontSize=24])`
- `ImageCrop(image,x,y,width,height)`
- `ImageDrawArc(image,x,y,width,height,startAngle,arcAngle,[filled=false])`
- `ImageDrawBeveledRect(image,x,y,width,height,[raised=false,filled=false])`
- `ImageDrawCubicCurve(image,x1,y1,ctrlx1,ctrly1,ctrlx2,ctrly2,x2,y2)`
- `ImageDrawLine(image,x1,y1,x2,y2)`
- `ImageDrawLines(image,xcoords,ycoords,[isPolygon=false,filled=false])`
- `ImageDrawOval(image,x,y,width,height,[filled=false])`
- `ImageDrawPoint(image,x,y)`
- `ImageDrawQuadraticCurve(image,x1,y1,ctrlx,ctrly,x2,y2)`
- `ImageDrawRect(image,x,y,width,height,[filled=false])`
- `ImageDrawRoundRect(image,x,y,width,height,arcWidth,arcHeight,[filled=false])`
- `ImageDrawText(image,string,x,y,[attributeCollection=""])`
- `ImageFlip(image,[transpose=VERTICAL])`

- ImageGetBLOB(image)
- ImageGetBufferedImage(image)
- ImageGetExifMetadata(image)
- ImageGetEXIFTag(image,tagname)
- ImageGetHeight(image)
- ImageGetIptcMetadata(image)
- ImageGetIPTCTag(image,tagname)
- ImageGetMetadata(image)
- ImageGetWidth(image)
- ImageGrayscale(image)
- ImageInfo(image)
- ImageMakeColorTransparent(image,color)
- ImageMakeTranslucent(image,percent)
- ImageNegative(image)
- ImageNew(source,[width=-1,height=-1,type=rgb,canvasColor=000000])
- ImageOverlay(image1,image2,[rule=SRC\_OVER,alpha=1.0])
- ImagePaste(image1,image2,x,y)
- ImageRead(path)
- ImageReadBase64(data)
- ImageResize(image,width,height,[interpolation=highestquality,blurFactor=1])
- ImageRotate(image,angle,[x=0,y=0,interpolation=nearest])
- ImageRotateDrawingAxis(image,angle,[x=0,y=0])
- ImageScaleToFit(image,fitWidth,fitHeight,[interpolation=highestquality,blurFactor=1])
- ImageSetAntiAliasing(image,[antialias=on])
- ImageSetBackgroundColor(image,color)
- ImageSetDrawingColor(image,color)
- ImageSetDrawingStroke(image,[attributeCollection=""])
- ImageSetDrawingTransparency(image,percent)
- ImageSharpen(image,[gain=1.0])
- ImageShear(image,shear,[direction=HORIZONTAL,interpolation=nearest])
- ImageShearDrawingAxis(image,shx,shy)
- ImageTranslate(image,xTrans,yTrans,[interpolation=nearest])
- ImageTranslateDrawingAxis(image,x,y)
- ImageWrite(image,destination,[quality=0.75])
- ImageWriteBase64(image,destination,format,[inHTMLFormat=false,overwrite=true])
- ImageXORDrawingMode(image,color)
- IncrementValue(number)
- InputBaseN(string,radix)
- Insert(substring,string,position)
- Int(number)
- InvalidateOauthAccessToken(token,type)
- invoke(object,methodName)
- invoke(object,methodName,arguments)



- `isArray(value,[dimension=-999])`
- `isBinary(value)`
- `isBoolean(value)`
- `isClosure(value)`
- `isCustomFunction(value)`
- `isDate(value)`
- `isDateObject(value)`
- `isDDX(value)`
- `isDebugMode()`
- `isDefined(value)`
- `isFileObject(value)`
- `isImage(value)`
- `isImageFile(value)`
- `isInstanceOf(value,type)`
- `isIPv6(value)`
- `isJSON(value)`
- `isLeapYear(value)`
- `isLocalHost(value)`
- `isNumeric(value)`
- `isNumericDate(value)`
- `isNull(value)`
- `isObject(value)`
- `isPDFArchive(value,standard)`
- `isPDFFile(value)`
- `isPDFObject(value)`
- `isQuery(value)`
- `isSafeHTML(value,[policyFile=""])`
- `isSimpleValue(value)`
- `isSOAPRequest()`
- `isSpreadsheetFile(value)`
- `isSpreadsheetobject(value)`
- `isStruct(value)`
- `isUserInAnyRole(value)`
- `isUserInRole(value)`
- `isUserLoggedIn()`
- `isValid(type,value,[pattern=""])`
- `isValid(type,value,min,max)`
- `isValidOauthAccesstoken(value,type)`
- `isWddx(value)`
- `isXml(value, ParserOptions)`
- `isXmlAttribute(value)`
- `isXmlDoc(value)`
- `isXmlElement(value)`

- XmlNode(value)
- XmlRoot(value)
- JavaCast(type,variable)
- JSStringFormat(string)
- LCase(string)
- Left(string,count)
- Len(object)
- ListAppend(list,value,[delimiter=,])
- ListChangeDelims(list,newDelimiter,[delimiter=,includeEmptyFields=false])
- ListContains(list,substring,[delimiter=,includeEmptyFields=false])
- ListContainsNoCase(list,substring,[delimiter=,includeEmptyFields=false])
- ListDeleteAt(list,position,[delimiter=,includeEmptyFields=false])
- ListEach(list,callback,[delimiter=,includeEmptyFields=false])
- ListFilter(list,callback,[delimiter=,includeEmptyFields=false])
- ListFind(list,value,[delimiter=,includeEmptyFields=false])
- ListFindNoCase(list,value,[delimiter=,includeEmptyFields=false])
- ListFirst(list,[delimiter=,includeEmptyFields=false])
- ListGetAt(list,position,[delimiter=,includeEmptyFields=false])
- ListInsertAt(list,position,value,[delimiter=,includeEmptyFields=false])
- ListLast(list,[delimiter=,includeEmptyFields=false])
- ListLen(list,[delimiter=,includeEmptyFields=false])
- ListMap(list,callback,includeEmptyFields,[delimiter=,])
- ListPrepend(list,value,[delimiter=,])
- ListQualify(list,qualifier,[delimiter=,elements=ALL,includeEmptyFields=false])
- ListReduce(list,callback,[initialValue="",delimiter=,includeEmptyFields=false])
- ListRemoveDuplicates(list,[delimiter=,ignoreCase=false])
- ListRest(list,[delimiter=,includeEmptyFields=false])
- ListSetAt(list,position,value,[delimiter=,includeEmptyFields=false])
- ListSort(list,callback,[Delimiter=""])
- ListSort(list,sortType,[sortOrder=asc,delimiter=,includeEmptyFields=false,localeSensitive=false])
- ListToArray(list,[delimiter=,includeEmptyFields=false,multiCharacterDelimiter=false])
- ListValueCount(list,value,[delimiter=,])
- ListValueCountNoCase(list,value,[delimiter=,])
- LJustify(string,length)
- Log(number)
- Log10(number)
- LSCurrencyFormat(number,[type=local,locale=defaultLocale])
- LSDDateFormat(date,[mask=medium,locale=defaultLocale])
- LSDDateTimeFormat(date,[mask=dd-MMM-yyyy HH:nn:ss,locale=defaultLocale,timezone=""])
- LSEuroCurrencyFormat(currency,[type=local,locale=defaultLocale])
- LSIsCurrency(currency,[locale=defaultLocale])
- LSIsDate(date,[locale=defaultLocale])
- LSIsNumeric(number,[locale=defaultLocale])

- `LSNumberFormat(number,[mask="",locale=defaultLocale])`
- `LSParseCurrency(currency,[locale=defaultLocale])`
- `LSParseDateTime(date,[locale=defaultLocale,format=""])`
- `LSParseEuroCurrency(currency,[locale=defaultLocale])`
- `LSParseNumber(number,[locale=defaultLocale])`
- `LSTimeFormat(time,[mask=short,locale=defaultLocale])`
- `LTrim(string)`
- `Max(number1,number2)`
- `Mid(string,start,count)`
- `Min(number1,number2)`
- `Minute(date)`
- `Month(date)`
- `MonthAsString(month,[locale=defaultLocale])`
- `Now()`
- `NumberFormat(number,[mask=""])`
- `ObjectEquals(clientobject,originalobject)`
- `ObjectLoad(object)`
- `ObjectLoad(path)`
- `ObjectSave(object,filePath)`
- `ORMClearSession(datasource)`
- `ORMCloseAllSessions()`
- `ORMCloseSession(datasource)`
- `ORMEvictCollection(entityName,collectionName,primaryKey)`
- `ORMEvictEntity(entityName,primaryKey)`
- `ORMEvictQueries(cacheName,datasource)`
- `ORMExecuteQuery(hql,[params="",queryOptions=""])`
- `ORMExecuteQuery(hql,[params="",unique=false,queryOptions=""])`
- `ORMFlush(datasource)`
- `ORMFlushAll()`
- `ORMGetSession(datasource)`
- `ORMGetSessionFactory(datasource)`
- `ORMIndex(entityName)`
- `ORMIndex(Instance)`
- `ORMIndexPurge(entityName)`
- `ORMReload()`
- `ORMSearch(searchQuery,entityName,[fields="",optionMap=""])`
- `ORMSearchOffline(searchQuery,entityName,fieldsToSelect,[fields="",options=""])`
- `ParagraphFormat(string)`
- `ParameterExists(param)`
- `ParseDateTime(date,[format=""])`
- `Pi()`
- `PrecisionEvaluate(expression)`
- `PreserveSingleQuotes(string)`

- Quarter(date)
- QueryAddColumn(query,columnName,array,[columnType=""])
- QueryAddRow(query,data)
- QueryAddRow(query,number)
- QueryConvertForGrid(query,page,pageSize)
- QueryDeleteColumn(query,column)
- QueryDeleteRow(query,row)
- QueryEach(query,callback)
- QueryExecute(sql,[params="",queryOptions=""])
- QueryFilter(query,callback)
- QueryGetResult(query)
- QueryGetRow(query,rowNumber)
- QueryKeyExists(query,column)
- QueryMap(query,callback,[schema=""])
- QueryNew(columnNames,[columnTypes="",data=""])
- QueryReduce(query,callback,[initialValue=""])
- QuerySetCell(query,columnName,value,[row=-1])
- QuerySort(query,callback)
- QuotedValueList(queryColumn,[delimiter=,])
- Rand(algorithm)
- Randomize(number,[algorithm=CFMX\_COMPAT])
- RandRange(number1,number2,[algorithm=CFMX\_COMPAT])
- REEscape(regex)
- REFind(regex,string,[start=1,returnsubexpressions=false,scope=ONE])
- REFindNoCase(regex,string,[start=1,returnsubexpressions=false,scope=ONE])
- ReleaseComObject(com)
- REMatch(regex,string)
- REMatchNoCase(regex,string)
- RemoveCachedQuery(sql,datasource,[params="",region=QUERY])
- RemoveChars(string,start,count)
- RepeatString(string,count)
- Replace(string,substring,replacement,[scope=ONE])
- Replace(string,substring,callback,[scope=ONE])
- ReplaceList(string,list1,list2,[delimiterList1=,,delimiterList2=,,includeEmptyFields=false])
- ReplaceList(string,list1,list2,[delimiterList=""])
- ReplaceListNoCase(string,list1,list2,[delimiterList1=,,delimiterList2=,,includeEmptyFields=false])
- ReplaceNoCase(string,substring,replacement,[scope=ONE])
- ReplaceNoCase(string,substring,callback,[scope=ONE])
- REReplace(string,regex,substring,[scope=ONE])
- REReplaceNoCase(string,regex,substring,[scope=ONE])
- RestDeleteApplication(directory)
- RestInitApplication(directory,[serviceMapping="",options=""])
- RestSetResponse(response)

- Reverse(string)
- Right(string,count)
- RJustify(string,length)
- Round(number)
- RTrim(string)
- RunAsync(callback,[timeOut=1000])
- Second(date)
- SendGatewayMessage(gatewayID,data)
- Serialize(object,type,[useCustomSerializer=true])
- SerializeJSON(data,[queryFormat=struct,useSecureJSONPrefix=false,useCustomSerializer=false])
- SerializeXml(xml,[useCustomSerializer=true])
- SessionGetMetadata()
- SessionInvalidate()
- SessionRotate()
- SetEncoding(scope,encoding)
- SetLocale(locale)
- SetProfileString(path,section,entry,value,[encoding=""])
- SetVariable(name,value)
- Sgn(number)
- Sin(number)
- Sleep(duration)
- SpanExcluding(string,set)
- SpanIncluding(string,set)
- SpreadsheetAddAutofilter(spreadsheet,autofilter)
- SpreadsheetAddColumn(spreadsheet,data,[datatype=""])
- SpreadsheetAddColumn(spreadsheet,data,startRow,startColumn,insert,[datatype=""])
- SpreadsheetAddFreezePane(spreadsheet,freezeColumn,freezeRow,column,row)
- SpreadsheetAddImage(spreadsheet,imagePath,anchor)
- SpreadsheetAddInfo(spreadsheet,properties)
- SpreadsheetAddPagebreaks(spreadsheet,rowbreaks,columnbreaks)
- SpreadsheetAddRow(spreadsheet,data,[row=-1,column=0,insert=true,datatype=""])
- SpreadsheetAddRows(spreadsheet,data,[insert=true])
- SpreadsheetAddRows(spreadsheet,data,[startRow=-1,startColumn=0,insert=true,datatypes="",insertColumnHeader=false])
- SpreadsheetAddSplitPane(spreadsheet,col,row,splitColumn,splitRow,[position=upper\_left])
- SpreadsheetCreateSheet(spreadsheet,[sheetName=""])
- SpreadsheetDeleteColumn(spreadsheet,column)
- SpreadsheetDeleteColumns(spreadsheet,columns)
- SpreadsheetDeleteRow(spreadsheet,row)
- SpreadsheetDeleteRows(spreadsheet,rows)
- SpreadsheetFormatCell(spreadsheet,format,row,column)
- SpreadsheetFormatCellRange(spreadsheet,format,startRow,startColumn,endRow,endColumn)
- SpreadsheetFormatColumn(spreadsheet,format,column)

- SpreadsheetFormatColumns(spreadsheet,format,columns)
- SpreadsheetFormatRow(spreadsheet,format,row)
- SpreadsheetFormatRows(spreadsheet,format,rows)
- SpreadsheetGetCellComment(spreadsheet,row,column)
- SpreadsheetGetCellFormula(spreadsheet,row,column)
- SpreadsheetGetCellValue(spreadsheet,row,column)
- SpreadsheetGetColumnCount(spreadsheet,[sheetName=""])
- SpreadsheetGetColumnCount(spreadsheet,[sheetNumber=""])
- SpreadsheetInfo(spreadsheet)
- SpreadsheetMergeCells(spreadsheet,startRow,endRow,startColumn,endColumn)
- SpreadsheetNew(sheetname,[isXmlformat=false])
- SpreadsheetRead(filePath,[sheetName=all])
- SpreadsheetRead(filePath,[sheetNumber=all])
- SpreadsheetReadBinary(spreadsheet)
- SpreadsheetRemoveSheet(spreadsheet,sheetName)
- SpreadsheetSetActiveSheet(spreadsheet,sheetName)
- SpreadsheetSetActiveSheetNumber(spreadsheet,sheetNumber)
- SpreadsheetSetCellComment(spreadsheet,comment,row,column)
- SpreadsheetSetCellFormula(spreadsheet,formula,row,column)
- SpreadsheetSetCellValue(spreadsheet,value,row,column,[datatype=""])
- SpreadsheetSetColumnWidth(spreadsheet,column,width)
- SpreadsheetSetFooter(spreadsheet,leftFooter,centerFooter,rightFooter)
- SpreadsheetSetHeader(spreadsheet,leftHeader,centerHeader,rightHeader)
- SpreadsheetSetRowHeight(spreadsheet,row,height)
- SpreadsheetShiftColumns(spreadsheet,start,[columns=1])
- SpreadsheetShiftColumns(spreadsheet,start,end,[columns=1])
- SpreadsheetShiftRows(spreadsheet,start,end,[rows=1])
- SpreadsheetWrite(spreadsheet,filePath,[password="",overwrite=false,autosize=true])
- SpreadsheetWrite(spreadsheet,filePath)
- Sqr(number)
- StoreAddACL(url,acl)
- StoreGetACL(url)
- StoreGetMetadata(url)
- StoreSetACL(url,acl)
- StoreSetMetadata(url,metadata)
- StripCr(string)
- StructAppend(struct1,struct2,[overwriteFlag=true])
- StructClear(struct)
- StructCopy(struct)
- StructCount(struct)
- StructDelete(struct,key,[indicateExists=false])
- StructEach(struct,callback)
- StructFilter(struct,filter)

- StructFind(struct,key)
- StructFindKey(struct,key,[scope=ONE])
- StructFindValue(struct,key,[scope=ONE])
- StructGet(path)
- StructGetMetaData(struct)
- StructInsert(struct,key,value,[overwrite=false])
- StructIsEmpty(struct)
- StructKeyArray(struct)
- StructKeyExists(struct,key)
- StructKeyList(struct,[delimiter=,])
- StructMap(struct,callback)
- StructNew([type="",sortType=text,sortOrder=asc,localeSensitive=false])
- StructReduce(struct,callback,[initialValue=""])
- StructSetMetaData(struct,metadata)
- StructSort(struct,[sortType=text])
- StructSort(struct,[Callback])
- StructToSorted(struct,[sortType=text,sortOrder=asc,path="",localeSensitive=false])
- StructUpdate(struct,key,value)
- Tan(number)
- ThreadJoin([threadName="",timeout=0])
- ThreadTerminate(threadName)
- TimeFormat(time,[mask=hh:mm tt])
- ToBase64(string,[encoding=""])
- ToBase64(BinaryData,[encoding=""])
- ToBinary(data)
- ToScript(CFVariable,JSVariableName,[outputFormat=true,ASFormat=false])
- ToString(value,[encoding=""])
- TransactionCommit()
- TransactionRollback(savepoint)
- TransactionSetSavepoint(savepoint)
- Trim(string)
- UCase(string)
- URLDecode(string,[charset=utf-8])
- URLEncodedFormat(string,[charset=utf-8])
- URLSessionFormat(url)
- Val(string)
- ValueArray(query,columnName)
- ValueList(queryColumn,[delimiter=,])
- VerifyClient()
- Week(date,[calendar=""])
- Wrap(string,limit,[strip=false])
- WriteBody(value)
- WriteOutput(string,[encodeFor=""])

- `WSGetAllChannels([channel="",clustered=false])`
- `WSGetSubscribers(channel,[clustered=false])`
- `WSPublish(channel,message,[filterCriteria="",clustered=false])`
- `WSSendMessage(message)`
- `XmlChildPos(xmlNode,childName,index)`
- `XmlElemNew(xmlNode,namespace,[childName=""])`
- `XMLFormat(string,[escape=false])`
- `XmlGetType(xml)`
- `XmlNew([caseSensitive=false])`
- `XmlParse(xmlString,[caseSensitive=false,validator=""])`
- `XmlSearch(xmlNode,xPath,[params=""])`
- `XmlTransform(xml,xsl,[params=""])`
- `XMLValidate(xmlString)`
- `Year(date)`
- `YesNoFormat(value)`