

<code>initWithMutableData:(NSMutableData *)data forDebugging:(BOOL)debug languageEncoding:(DPSProgramEncoding)langEnc nameEncoding:(DPSNameEncoding)nameEnc textProc:(DPSTextProc)tProc errorProc:(DPSErrorProc)errorProc</code>	Initializes a newly allocated NSDPSContext that writes its output to data using the language and name encodings specified by langEnc and nameEnc. The callback functions tProc and errorProc handle text and errors generated by the context. If debug is YES, the output is given in human-readable form in which large structures (such as images) may be represented by comments.
<code>(BOOL)isDrawingToScreen</code>	Returns YES if the drawing destination is the screen.
<code>(NSMutableData *)mutableData</code>	Returns the receiver's data object.
<code>(DPSContext)DPSContext</code>	Returns the corresponding DPSContext.
<code>(void)flush</code>	Forces any buffered data to be sent to its destination.
<code>(void)interruptExecution</code>	Interrupts execution in the receiver's context.
<code>(void)notifyObjectWhenFinishedExecuting:(id <NSDPSContextNotification>)object</code>	

<code>(DPSContext *)errorProc</code>	Returns the context's error callback function.
<code>(DPSContext *)setErrorProc:(DPSContext *)proc</code>	Sets the context's error callback function to <code>proc</code> .
<code>(DPSContext *)setTextProc:(DPSContext *)proc</code>	Sets the context's text callback function to <code>proc</code> .
<code>(DPSContext *)textProc</code>	Returns the context's text callback function.
<code>(DPSContext *)printFormat:(NSString *)format,...</code>	Constructs a string from <code>format</code> and following string objects (in the manner of <code>printf()</code>) and sends it to the context's destination.
<code>(DPSContext *)printFormat:(NSString *)format arguments:(va_list)argList</code>	Constructs a string from <code>format</code> and <code>argList</code> (in the manner of <code>vprintf()</code>) and sends it to the context's destination.
<code>(DPSContext *)writeData:(NSData *)buf</code>	Sends the PostScript data in <code>buf</code> to the context's destination.
<code>(DPSContext *)writePostScriptWithLanguageEncodingConversion:(NSData *)buf</code>	Writes the PostScript data in <code>buf</code> to the context's destination as plain text, encoded tokens, or a binary object sequence, as necessary depending on the language encoding of the receiver.
<code>(DPSContext *)awaitReturnValues</code>	Waits for all return values from the result table.
<code>(DPSContext *)writeBOSArray:(const void *)data count:(unsigned int)items ofType:(DPSDefinedType)type</code>	Write an array to the context's destination as part of a binary object sequence. The array is taken from <code>data</code> and consists of <code>items</code> items of type <code>type</code> .
<code>(DPSContext *)writeBOSNumString:(const void *)data length:(unsigned int)count ofType:(DPSDefinedType)type scale:(int)scale</code>	Write a number string to the context's destination as part of a binary object sequence. The string is taken from <code>data</code> as described by <code>count</code> , <code>type</code> , and <code>scale</code> .
<code>(DPSContext *)writeBOSString:(const void *)data length:(unsigned int)bytes</code>	Write a string to the context's destination as part of a binary object sequence. The string is taken from <code>data</code> .
<code>(DPSContext *)writeBinaryObjectSequence:(const void *)data length:(unsigned int)bytes</code>	Write a binary object sequence to the context's destination. The sequence consists of <code>length</code> bytes (a count) of <code>data</code> .
<code>(DPSContext *)updateNameMap</code>	Updates the context's name map from the client library's name map.
<code>(DPSContext *)chainChildContext:(NSDPSContext *)child</code>	Links <code>child</code> (and all of its children) to the receiver as its child context that receives a copy of all PostScript code sent to the receiver.
<code>(DPSContext *)childContext</code>	Returns the receiver's child context, or <code>nil</code> if none exists.
<code>(DPSContext *)parentContext</code>	Returns the receiver's parent context, or <code>nil</code> if none exists.
<code>(DPSContext *)unchainContext</code>	Unlinks the child context (and all of its children) from the receiver's chained contexts.

id)setSynchronized:(BOOL)flag

id)setOutputTraced:(BOOL)flag

id)setSynchronized:(BOOL)flag

Returns whether the wait method is invoked each time the
of output to the server.

Causes the data (PostScript code, return values, etc.) flow
application's single context and the Display PostScript
diagnostic output.

Sets whether the wait method is invoked each time the rec
output to its destination.