

Yokogawa MEG Reader Toolbox

Revision 1.4

Specifications

1 November 2011

License Agreement

This software is available via the license described at the end of this document.

Notice

This software is written for MATLAB® version 7.5 (R2007b) onwards.
MATLAB is a registered trademark of The MathWorks, Inc.

The contents of this software are subject to change without prior notice as a result of continuing improvements to the software's performance and functions.

This software consists of following functions:

Category	Function name	Purpose
Read MEG data	getYkgwData	Get measurement data.
Read MEG header	getYkgwHdrSystem	Get information about system.
	getYkgwHdrChannel	Get information about channel.
	getYkgwHdrAcqCond	Get information about data acquisition condition.
	getYkgwHdrEvent	Get information about trigger event.
	getYkgwHdrCoregist	Get information about coregistration.
	getYkgwHdrDigitize	Get information about digitization.
	getYkgwHdrSubject	Get information about subject.
	getYkgwHdrBookmark	Get information about bookmark.
Read MRI	getYkgwMriHdr	Get information about header of MRI file (*.mri).
Others	getYkgwVersion	Get information about version of this toolbox.

getYkgwData

This function retrieves the measurement data of whole channels by the specified file path and sample range.

```
data = getYkgwData(
    filepath,
    start_sample,
    sample_length
);
```

Arguments:

<i>filepath</i>	string	[in] File path
<i>start_sample</i>	double	[in] Start sample or trial(frame) number for retrieving data. The start number corresponding to each acquisition type is as follows : - Continuous Raw : Start sample number for retrieving data. (0 origin) - Evoked Average : Start sample number for retrieving data. (0 origin) - Evoked Raw : Start frame number for retrieving data. (1 origin) When both <i>start_sample</i> and <i>sample_length</i> are omitted, you can get data of whole samples.
<i>sample_length</i>	double	[in] Sample length for retrieving data. The number of samples or trials(frames) corresponding to each acquisition type is as follows : - Continuous Raw : Number of samples for retrieving data. - Evoked Average : Number of samples for retrieving data. - Evoked Raw : Number of trials(frames) for retrieving data. When this parameter is omitted or is specified as 'Inf', you can get data from <i>start_sample</i> to the end of sample(frame).

Return values:

<i>data</i>	matrix(double)	[out] double matrix of measurement data. Row : number of channels(whole channel), Column : number of samples Unit of the each channel depends on channel type as follows: MagnetoMeter [Tesla] AxialGradioMeter [Tesla] PlanarGradioMeter [Tesla] ReferenceMagnetoMeter [Tesla] ReferenceAxialGradioMeter [Tesla] ReferencePlanarGradioMeter [Tesla] TriggerChannel [Volt] EegChannel [Volt] *This has already been reflected EEG gain EcgChannel [Volt] *This has already been reflected ECG gain EtcChannel [Volt] NullChannel [Volt]
-------------	----------------	--

getYkgwHdrSystem

This function retrieves information of the system.

```
system_info = getYkgwHdrSystem(
    filepath
);
```

Arguments:

<i>filepath</i>	string	[in] File path
-----------------	--------	----------------

Return values:

<i>system_info</i>	structure	[out] The structure of system information.
<i>.version</i>	double	Data version
<i>.revision</i>	double	Data revision
<i>.system_id</i>	double	System ID
<i>.system_name</i>	string	System name
<i>.model_name</i>	string	Model name

getYkgwHdrChannel

This function retrieves information about channel.

```
channel_info = getYkgwHdrChannel(filepath);
```

Arguments:

filepath string [in] File path

Return values:

channel_info structure [out] The structure of channel information.
.channel_count double The number of whole channels.
.channel structure array The detail information of channels. ('index 1' corresponds to 'channel 0')
.type double Channel type as follow table:

NullChannel	= 0;
MagnetoMeter	= 1;
AxialGradioMeter	= 2;
PlanarGradioMeter	= 3;
ReferenceMagnetoMeter	= 257;
ReferenceAxialGradioMeter	= 258;
ReferencePlanarGradioMeter	= 259;
TriggerChannel	= -1;
EegChannel	= -2;
EcgChannel	= -3;
EtcChannel	= -4;

.data structure The geometrical information of a channel. These fields is based on MEG device coordinate system. These fields of each channel type are as follows: See Figure.1 and Figure.2.

If channel type is AxialGradioMeter or ReferenceAxialGradioMeter (see Figure.3),

<i>.x</i>	double	x coordinate of inner coil position [meter]
<i>.y</i>	double	y coordinate of inner coil position [meter]
<i>.z</i>	double	z coordinate of inner coil position [meter]
<i>.zdir</i>	double	Sensor orientation from z-axis [degree]
<i>.xdir</i>	double	Sensor orientation from x-axis [degree]
<i>.baseline</i>	double	Baseline length [meter]
<i>.size</i>	double	Inner coil size [meter]
<i>.name</i>	string	Abbreviation name

If channel type is PlanarGradioMeter or ReferencePlanarGradioMeter (see Figure 4),

<i>.x</i>	double	x coordinate of inner coil position [meter]
<i>.y</i>	double	y coordinate of inner coil position [meter]
<i>.z</i>	double	z coordinate of inner coil position [meter]
<i>.zdir1</i>	double	Sensor orientation from z-axis [degree]
<i>.xdir1</i>	double	Sensor orientation from x-axis [degree]
<i>.zdir2</i>	double	Baseline orientation from z-axis [degree]
<i>.xdir2</i>	double	Baseline orientation from x-axis [degree]
<i>.baseline</i>	double	Baseline length [meter]
<i>.size</i>	double	Inner coil size [meter]

If channel type is MagnetoMeter or ReferenceMagnetoMeter,

<i>.x</i>	double	x coordinate of coil position [meter]
<i>.y</i>	double	y coordinate of coil position [meter]
<i>.z</i>	double	z coordinate of coil position [meter]
<i>.zdir</i>	double	Sensor orientation from z-axis [degree]
<i>.xdir</i>	double	Sensor orientation from x-axis [degree]
<i>.size</i>	double	Inner coil size [meter]
<i>.name</i>	string	Abbreviation name

If channel type is EegChannel or EcgChannel,

<i>.type</i>	double	Type
<i>.id</i>	double	ID
<i>.name</i>	string	Abbreviation name
<i>.gain</i>	double	Gain

If channel type is TriggerChannel or EtcChannel,

<i>.type</i>	double	Type
<i>.id</i>	double	ID
<i>.name</i>	string	Abbreviation name

If channel type is NullChannel, there is no field.

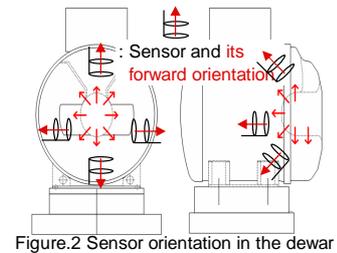
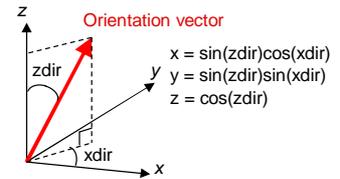


Figure.2 Sensor orientation in the dewar

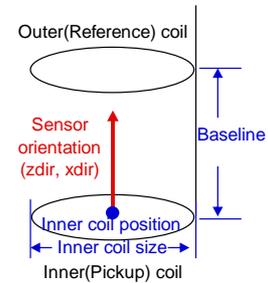


Figure.3 AxialGradioMeter parameter

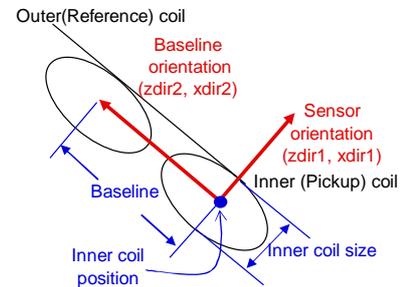


Figure.4 PlanarGradioMeter parameter

getYkgwHdrAcqCond

This function retrieves information about data acquisition condition.

```
acq_cond = getYkgwHdrAcqCond(
    filepath
);
```

Arguments:

filepath string [in] File path

Return values:

acq_cond structure [out] The structure of information about data acquisition condition.
.acq_type double Acquisition type
AcqTypeContinuousRaw = 1;
AcqTypeEvokedAve = 2;
AcqTypeEvokedRaw = 3;

If acquisition type is AcqTypeContinuousRaw,
.sample_rate double Sampling rate [Hz]
.sample_count double The number of samples which were actually acquired [sample]
.specified_sample_count double The number of samples which were specified before starting acquisition [sample]

If acquisition type is AcqTypeEvokedAve or AcqTypeEvokedRaw,
.sample_rate double Sampling rate [Hz]
.frame_length double Frame length (The number of samples per one trial) [sample]
.pretrigger_length double Pretrigger length (The number of samples before trigger per one trial) [sample]
.average_count double The number of trials(frames) which were actually acquired [trial]
.specified_average_count double The number of trials(frames) which were specified before starting acquisition [trials]
.multi_trigger structure The structure of multi trigger information.
.enable boolean Is multi trigger mode ? (true : multi trigger mode)
.count double Number of multi triggers
.list structure array List of multi triggers (If not multi trigger mode, this structure array is set to empty.)
.enable boolean Is current multi trigger set to enable ? (true : enable)
.code double Event code (1 origin)
.name string Event name
.average_count double The number of trials(frames) which were actually acquired [trial]
.specified_average_count double The number of trials(frames) which were specified before starting acquisition [trials]

getYkgwHdrEvent

This function retrieves information about trigger event.

```
event = getYkgwHdrEvent(
    filepath
);
```

Arguments:

filepath string [in] File path

Return values:

event structure array [out] The structure array of trigger event corresponding to each trial.
.sample_no double Sample number of current event (0 origin)
.code double Event code (1 origin)
.name string Event name

getYkgwHdrCoregist

This function retrieves information about coregistration.

```
coregist = getYkgwHdrCoregist(filepath);
```

Arguments:

filepath string [in] File path

Return values:

<i>coregist</i>	structure	[out] The structure of information about coregistration.
<i>.done</i>	boolean	Is coregistration done? (true : done)
<i>.mri_type</i>	double	MRI type NoMriFile = 0; NormalMriFile = 1; VirtualMriFile = 2;
<i>.mri_file</i>	string	File path of MRI file (*.mri)
<i>.hpi_file</i>	string	File path of HPI(Head Position Indicator) file (*.mrk)
<i>.meg2mri</i>	matrix(double)	4 x 4 matrix which transforms MEG device coordinate to MRI coordinate [meter] usage: $[xmri, ymri, zmri, 1]' = coregist.meg2mri * [xmeg, ymeg, zmeg, 1]'$
<i>.mri2meg</i>	matrix(double)	4 x 4 matrix which transforms MRI coordinate to MEG device coordinate [meter] usage: $[xmeg, ymeg, zmeg, 1]' = coregist.meg2mri * [xmri, ymri, zmri, 1]'$
<i>.hpi</i>	structure array	The structure array of HPI(Head Position Indicator)
<i>.meg_pos</i>	matrix(double)	HPI position $[x, y, z]$ on MEG device coordinate [meter]
<i>.mri_pos</i>	matrix(double)	HPI position $[x, y, z]$ on MRI coordinate [meter] (Before coregistration, this field is set to $[0,0,0]$)
<i>.label</i>	string	HPI label as follows: 'LPA' : Left PreAuricular 'RPA' : Right PreAuricular 'CPF' : Center PreFrontal 'LPF' : Left PreFrontal 'RPF' : Right PreFrontal
<i>.model</i>	structure	The structure of conductor model.
<i>.type</i>	double	Conductor model type UNKNOWN_MODEL = -1; NO_MODEL = 0; SPHERICAL_MODEL = 1; LAYERED_MODEL = 2;

If Conductor model type is SPHERICAL_MODEL,

<i>.cx</i>	double	x coordinate of spherical center position on MRI coordinate [meter]
<i>.cy</i>	double	y coordinate of spherical center position on MRI coordinate [meter]
<i>.cz</i>	double	z coordinate of spherical center position on MRI coordinate [meter]
<i>.radius</i>	double	radius of spherical conductor on MRI coordinate [meter]

If Conductor model type is LAYERED_MODEL,

<i>.ax</i>	double	Coefficient 'ax' of planar equation $'ax * x + ay * y + az * z = c'$
<i>.ay</i>	double	Coefficient 'ay' of planar equation $'ax * x + ay * y + az * z = c'$
<i>.az</i>	double	Coefficient 'az' of planar equation $'ax * x + ay * y + az * z = c'$
<i>.c</i>	double	Coefficient 'c' of planar equation $'ax * x + ay * y + az * z = c'$

getYkgwHdrDigitize

This function retrieves information of the digitization.

```
digitize = getYkgwHdrDigitize(filepath);
```

Arguments:

filepath string [in] File path

Return values:

<i>digitize</i>	structure	[out] The structure of information and points about digitization.
<i>.info</i>	structure	The structure of information about digitization.
<i>.digitizer_file</i>	string	File path of digitizer file
<i>.done</i>	boolean	Is matching done? (true : done)
<i>.meg2digitizer</i>	matrix(double)	4 x 4 matrix which transforms MEG coordinate to Digitizer coordinate.
<i>.digitizer2meg</i>	matrix(double)	4 x 4 matrix which transforms Digitizer coordinate to MEG coordinate.
<i>.point</i>	structure array	The structure of point data about digitization.
<i>.name</i>	string	Point name
<i>.x</i>	double	x-coordinate on digitizer coordinate [meter]
<i>.y</i>	double	y-coordinate on digitizer coordinate [meter]
<i>.z</i>	double	z-coordinate on digitizer coordinate [meter]

getYkgwHdrSubject

This function retrieves information of the subject.

```
subject = getYkgwHdrSubject(
                                     filepath
);
```

Arguments:

filepath string [in] File path

Return values:

<i>subject</i>		[out] The structure of subject information.
<i>.id</i>	string	ID
<i>.name</i>	string	Name
<i>.birthday</i>	string	Birthday
<i>.sex</i>	string	Sex
<i>.handed</i>	string	Handed

getYkgwHdrBookmark

This function retrieves information about bookmark.

```
bookmark = getYkgwHdrBookmark(
                                     filepath
);
```

Arguments:

filepath string [in] File path

Return values:

<i>bookmark</i>	structure array	[out] The structure array of bookmark information.
<i>.sample_no</i>	double	Sample number of bookmark
<i>.label</i>	double	Label of bookmark
<i>.comment</i>	string	Comment of bookmark

getYkgwHdrSource

This function retrieves information of the sources.

```
source = getYkgwHdrSource(
                                     filepath
);
```

Arguments:

filepath string [in] File path

Return values:

<i>source</i>	structure array	[out] The structure array of analyzed source information. Note : Sources are arranged in order of estimated time.
<i>.type</i>	double	Type of source DipoleModel = 1; DistributedSourceModel = 2;
<i>.time</i>	double	Analyzed Time [second] from 1970.1.1
<i>.sample_no</i>	double	Time sample index of source
<i>.channel_list</i>	row vector(double)	Channel number (0 origin) list which used to estimate
<i>.model</i>	structure	The structure of conductor model.
<i>.type</i>	double	Conductor model type UNKNOWN_MODEL = -1; NO_MODEL = 0; SPHERICAL_MODEL = 1; LAYERED_MODEL = 2;

If Conductor model type is SPHERICAL_MODEL,

<i>.cx</i>	double	x coordinate of spherical center position on MEG coordinate [meter]
<i>.cy</i>	double	y coordinate of spherical center position on MEG coordinate [meter]
<i>.cz</i>	double	z coordinate of spherical center position on MEG coordinate [meter]
<i>.radius</i>	double	radius of spherical conductor on MEG coordinate [meter]

If Conductor model type is LAYERED_MODEL,

<i>.ax</i>	double	Coefficient 'ax' of planar equation 'ax * x + ay * y + az * z = c'
<i>.ay</i>	double	Coefficient 'ay' of planar equation 'ax * x + ay * y + az * z = c'
<i>.az</i>	double	Coefficient 'az' of planar equation 'ax * x + ay * y + az * z = c'
<i>.c</i>	double	Coefficient 'c' of planar equation 'ax * x + ay * y + az * z = c'

<i>.algorithm</i>	structure	The structure of conductor algorithm.
<i>.magnetic_field_calc</i>	double	Algorithm of magnetic field calculation BiotSavartLaw = 1; SarvasLaw = 2; MagneticDipoleLaw = 3;
<i>.variable_restraint</i>	double	Algorithm of variable restraint NoRestraint = 0; PositionRestraint = 1; DirectionRestraint = 2; IntensityRestraint = 3;
<i>.optimization</i>	double	Algorithm of optimization GradientAlgorithm = 1; LeadFieldReconstructionAlgorithm = 2; ManualSetAlgorithm = 3; UserAlgorithm = 4;
<i>.filter</i>	structure	The structure of spectral filter setting.
<i>.hpf, .lpf</i>	structure	The structure of high-pass / low-pass filter setting.
<i>.enable</i>	boolean	Does this filter enable?
<i>.cutoff_frequency</i>	double	Cutoff frequency [Hz]
<i>.window_type</i>	double	Window type NoWindow = 0; HanningWindow = 1; HammingWindow = 2;
<i>.width</i>	double	Filter width
<i>.bpf, .bef</i>	structure	The structure of band-pass / band-eliminate filter setting.
<i>.enable</i>	boolean	Does this filter enable?
<i>.low_frequency</i>	double	Low frequency [Hz]
<i>.high_frequency</i>	double	High frequency [Hz]
<i>.window_type</i>	double	Window type
<i>.width</i>	double	Filter width
<i>.moveave</i>	structure	The structure of moving average setting.
<i>.enable</i>	boolean	Does this filter enable?
<i>.width</i>	double	Filter width
<i>.baseadj</i>	structure	The structure of baseline adjustment setting.
<i>.enable</i>	boolean	Does this filter enable?
<i>.type</i>	double	Type of baseline adjustment PretriggerBaselineAdjust = 0; PosttriggerBaselineAdjust = 1; AllRangeBaselineAdjust = 2; ExplicitBaselineAdjust = 3;
<i>.start_time</i>	double	Start time [millisecond]
<i>.end_time</i>	double	End time [millisecond]
<i>.gof</i>	double	Goodness-of-fit (GOF)
<i>.correlation</i>	double	Correlation Coefficiency
<i>.label</i>	double	Label
<i>.comment</i>	string	Comment
<i>.total_intensity</i>	double	Total intensity of sources
<i>.dipole_count</i>	double	Number of dipole sources
<i>.dipole_list</i>	structure array	The structure array of dipole sources
<i>.x</i>	double	x coordinate of dipole position on MEG coordinate [meter]
<i>.y</i>	double	y coordinate of dipole position on MEG coordinate [meter]
<i>.z</i>	double	z coordinate of dipole position on MEG coordinate [meter]
<i>.zdir</i>	double	Dipole orientation from z-axis [degree]
<i>.xdir</i>	double	Dipole orientation from x-axis [degree]
<i>.intensity</i>	double	Dipole intensity (moment) [Ampere Meter]

getYkgwMriHdr

This function retrieves header information of specified mri file (*.mri).

```
mri_header = getYkgwMriHdr(filepath);
```

Arguments:

filepath string [in] File path

Return values:

mri_header structure [out] The structure of mri header information.
.data_style double Data style (0 : DICOM, others : Polhemus)
.model structure The structure of conductor model.
.done boolean Is conductor model defined ? (true : defined)
.type double Conductor model type
 UNKNOWN_MODEL = -1;
 NO_MODEL = 0;
 SPHERICAL_MODEL = 1;
 LAYERED_MODEL = 2;

If Conductor model type is SPHERICAL_MODEL,

.cx double x coordinate of spherical center position on MRI coordinate [meter]
.cy double y coordinate of spherical center position on MRI coordinate [meter]
.cz double z coordinate of spherical center position on MRI coordinate [meter]
.radius double radius of spherical conductor on MRI coordinate [meter]

If Conductor model type is LAYERED_MODEL,

.ax double Coefficient 'ax' of planar equation 'ax * x + ay * y + az * z = c'
.ay double Coefficient 'ay' of planar equation 'ax * x + ay * y + az * z = c'
.az double Coefficient 'az' of planar equation 'ax * x + ay * y + az * z = c'
.c double Coefficient 'c' of planar equation 'ax * x + ay * y + az * z = c'

.hpi structure array The structure of point data about picked HPI.
.done boolean Is pick-up of a HPI point done ? (true : done)
.mri_pos double HPI position [x, y, z] on MRI coordinate [meter]
.label string HPI label as follows:
 'LPA' : Left PreAuricular
 'RPA' : Right PreAuricular
 'CPF' : Center PreFrontal
 'LPF' : Left PreFrontal
 'RPF' : Right PreFrontal

.image_parameter structure The structure of image parameters.
.intensity vector(double) 1 x 2 row vector, minimum and maximum of image values
.initial_color vector(double) 1 x 2 row vector, minimum and maximum of initial brightness
.color vector(double) 1 x 2 row vector, minimum and maximum of current brightness

.normalize structure The structure of normalized HEAD coordinate system (LPA(x-), RPA(x+), nasion(y+)). See Figure.5.
.done boolean Is HEAD coordinate system defined ? (true : defined)
.mri2normalize matrix(double) 4 x 4 matrix which transforms MRI coordinate to HEAD coordinate [meter]
 usage: [*xhead*, *yhead*, *zhead*, 1]' = *mri_header.normalize.mri2normalize* * [*xmri*, *ymri*, *zmri*, 1]'

.point structure array The structure of point data about HEAD fiducial points.
.done boolean Is pick-up of a HEAD fiducial point done ? (true : done)
.name string Name of HEAD fiducial points.
.x double x coordinate of a HEAD fiducial point on MRI coordinate [meter]
.y double y coordinate of a HEAD fiducial point on MRI coordinate [meter]
.z double z coordinate of a HEAD fiducial point on MRI coordinate [meter]

.besa_fiducial structure The structure of BESA fiducial information.
.point structure array The structure of point data about BESA fiducial points.
.done boolean Is pick-up of a BESA fiducial point done ? (true : done)
.x double x coordinate of a BESA fiducial point on MRI coordinate [meter]
.y double y coordinate of a BESA fiducial point on MRI coordinate [meter]
.z double z coordinate of a BESA fiducial point on MRI coordinate [meter]

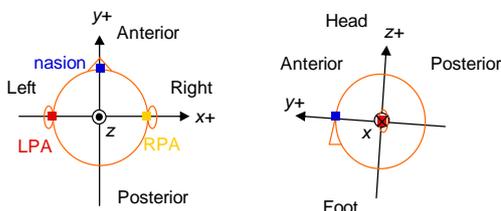


Figure.5 Normalized HEAD coordinate system

getYkgwVersion

This function retrieves version of this toolbox.

```
ykgw_ver = getYkgwVersion;
```

Arguments:

none

Return values:

<i>ykgw_ver</i>		[out] structure of toolbox version
<i>.version</i>	string	toolbox version : major.minor
<i>.major</i>	double	toolbox major version
<i>.minor</i>	double	toolbox minor version
<i>.revision</i>	double	toolbox revision version
<i>.build</i>	double	toolbox build version
<i>.date</i>	string	release date yyyy.mm.dd

License Agreement

Copyright (c) 2010-2011 YOKOGAWA Electric Corporation.
All rights reserved.

Yokogawa MEG Reader Toolbox is distributed subject to the following license conditions:

SOFTWARE LICENSE AGREEMENT*

Software: Yokogawa MEG Reader Toolbox

1. The "software", below, refers to Yokogawa MEG Reader Toolbox (in binary form and accompanying documentation). Each licensee is addressed as "you" or "Licensee."
2. The copyright holders shown above and their third-party licensors hereby grant Licensee a royalty-free nonexclusive license, subject to the limitations stated herein.
3. You may make a copy or copies of the software for use within your organization, if you meet the following conditions:
Copies must include the copyright notice and this software License Agreement in the documentation and/or other materials provided with the copy.
4. You may not modify a copy or copies of the software or any portion of it.
5. This software and data processed by this software may not be used for clinical and/or diagnostic purposes.
6. WARRANTY DISCLAIMER. THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND. THE COPYRIGHT HOLDERS, THEIR THIRD PARTY LICENSORS, AND THEIR EMPLOYEES:
(1) DISCLAIM ANY WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT,
(2) DO NOT ASSUME ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF THE SOFTWARE,
(3) DO NOT REPRESENT THAT USE OF THE SOFTWARE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS,
(4) DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL BE CORRECTED.
7. LIMITATION OF LIABILITY. IN NO EVENT WILL THE COPYRIGHT HOLDERS, THEIR THIRD PARTY LICENSORS, OR THEIR EMPLOYEES: BE LIABLE TO YOU AND/OR ANY PARTY FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR OTHER CONSEQUENTIAL DAMAGES, OR FOR EXEMPLARY, SPECIAL, PUNITIVE OR SIMILAR DAMAGES OF ANY KIND, WHETHER BASED ON CONTRACT, STRICT LIABILITY, TORT, WARRANTY (EXPRESS OR IMPLIED), OR ANY OTHER LEGAL GROUNDS FOR ANY USE OF THIS SOFTWARE, INCLUDING WITHOUT LIMITATION, ANY LOST PROFITS, LOSS OF INCOME, BUSINESS INTERRUPTION, LOSS OF USE, LOSS OR DESTRUCTION OF PROGRAMS OR OTHER DATA, LOSS OF AVAILABILITY AND THE LIKE ON YOUR INFORMATION HANDLING SYSTEM, OR OTHERWISE, EVEN IF THE COPYRIGHT HOLDERS HAS BEEN EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
8. The copyright holders reserve the right to change, or temporarily or permanently withdraw, information and/or contents contained in its own software, or to suspend or discontinue the services provided through this software at any time, without prior notice.
The copyright holders shall not be held liable for damage of any kind sustained by you for any reason, as a result of the copyright holders changing, or temporarily or permanently withdrawing, such information.

*This License Agreement is subject to change without notice.