sgcWebBroker 1.5

Table of Contents

Introduction

sgcWebBroker is a package that improves WebBroker technology, adding the following features:

- **Session Management** using **session id, cookies or automatic.**
- Support for **session timeout**.
- **Async Events** using AJAX.
- Automatically **serves files** on a local directory.
- Events to handle **New Sessions**, **Async Events**...
- **Support** from **JQueryMobile Framework**.

Main components available :

- **TsgcWebBroker Application:** Non-visual component, it's used to manage user sessions and async events.
- **TsgcWebBrokerJQMTemplate:** Non-visual component, used to enhance HTML forms using JQueryMobile, touch events...

Overview

IDE Editions

Delphi supported IDE

- Delphi 7
- Delphi 2007
- Delphi 2009
- Delphi 2010
- Delphi XE
- Delphi XE2
- Delphi XE3
- Delphi XE4
- Delphi XE5
- Delphi XE6
- Delphi XE7
- Delphi XE8
- Delphi 10
- Delphi 10.1
- Delphi 10.2
- Delphi 10.3

Installation

Delphi

1. Unzip the files included into a directory {$DIR}

2. From Delphi:

Add the directory where the files are unzipped {$DIR} to the Delphi library pat under Tools, Environment options, Directories

All Delphi/C++Builder Versions

Add the directory {$DIR}\source to the library path

For specific Delphi/C++Builder version

Delphi 7 : Add the directory {$DIR}\libD7 to the library path
Delphi 2007 : Add the directory {$DIR}\libD2007 to the library path
Delphi 2009 : Add the directory {$DIR}\libD2009 to the library path
Delphi 2010 : Add the directory {$DIR}\libD2010 to the library path
Delphi XE : Add the directory {$DIR}\libDXE to the library path
Delphi XE2 : Add the directory {$DIR}\libDXE2\$(Platform) to the library path
Delphi XE3 : Add the directory {$DIR}\libDXE3\$(Platform) to the library path
Delphi XE4 : Add the directory {$DIR}\libDXE4\$(Platform) to the library path
Delphi XE5 : Add the directory {$DIR}\libDXE5\$(Platform) to the library path
Delphi XE6 : Add the directory {$DIR}\libDXE6\$(Platform) to the library path
Delphi XE7 : Add the directory {$DIR}\libDXE7\$(Platform) to the library path
Delphi XE8 : Add the directory {$DIR}\libDXE8\$(Platform) to the library path
Delphi 10 : Add the directory {$DIR}\libD10\$(Platform) to the library path
Delphi 10.1 : Add the directory {$DIR}\libD10_1\$(Platform) to the library path

Delphi 10.2 : Add the directory {$DIR}\libD10_2\$(Platform) to the library path
Delphi 10.3 : Add the directory {$DIR}\libD10_3\$(Platform) to the library path

3. From Delphi

Choose

File, Open and browse for the correct Packages\sgcWebBroker.groupproj (First compile sgcWebBrokerX.dpk and then install dclsgcWebBrokerX.dpk)

packages files for Delphi

sgcWebBrokerD7.groupproj	: Delphi 7
sgcWebBrokerD2007.groupproj	: Delphi 2007
sgcWebBrokerD2009.groupproj	: Delphi 2009
sgcWebBrokerD2010.groupproj	: Delphi 2010
sgcWebBrokerDXE.groupproj	: Delphi XE
sgcWebBrokerDXE2.groupproj	: Delphi XE2
sgcWebBrokerDXE3.groupproj	: Delphi XE3
sgcWebBrokerDXE4.groupproj	: Delphi XE4
sgcWebBrokerDXE5.groupproj	: Delphi XE5
sgcWebBrokerDXE6.groupproj	: Delphi XE6
sgcWebBrokerDXE7.groupproj	: Delphi XE7
sgcWebBrokerDXE8.groupproj	: Delphi XE8
sgcWebBrokerD10.groupproj	: Delphi 10
sgcWebBrokerD10_1.groupproj	: Delphi 10.1
sgcWebBrokerD10_2.groupproj	: Delphi 10.2
sgcWebBrokerD10_3.groupproj	: Delphi 10.3

4. Demos

All demos are available in subdirectory Demos.

History

[*] : Bug
[+] : New
[-] : Deleted
[/] : Breaking changes

1.5: 2019 April

[+] : Added support for Rad Studio 10.3 Rio

1.4: 2018 May

[+] : Added support for Rad Studio 10.1 Berlin
[+] : Added support for Rad Studio 10.2 Tokyo

1.3: 2015 October

[+] : Added support for Rad Studio 10
[+] : Added support for Rad Studio XE8

1.2: 2014 October

[+] : Added support for Rad Studio XE6
[+] : Added support for Rad Studio XE7
[+] : Added support to JQueryMobile 1.4

1.1: 2014 January

[+] : Added support from JQueryMobile Framework

[*] : Fixed Bug Getting URL

1.0: 2013 October

[+] : First Version

QuickStart

QuickStart

Let's start with a basic example where we need to create a WebBroker Project.

1. Create a new WebBroker Project.

2. Drop a TsgcWebBrokerApplication in a TWebModule.

3. Set your ISAPI directory using DocumentRoot

4. Set Session Tracking method, by default, is set to automatic.

5. Create a new UserSession Class where you can store all properties, methods... specific for each session.

```
TsgcUserSession = class
private
  FRegistered: Boolean;
  FUser: String;
public
  property Registered: Boolean read FRegistered write
FRegistered;
  property User: String read FUser write FUser;
end;
```

6. After a new session is created, create an instance of UserSession Class and assign to UserSession.Data.

```
procedure
sgcWebBrokerApplication1AfterNewSession(Sender:
TObject; UserSession: TsgcWebBrokerUserSession);
begin
  UserSession.Data := TsgcUserSession.Create;
end;
```

7. On each client request, you can access to your UserSession Class doing a simple Cast to UserSession

```
procedure
sgcWebBrokerApplication1BeforeDispatch(Sender:
TObject; Request: TWebRequest; Response: TWebResponse;
UserSession: TsgcWebBrokerUserSession; var Handled:
Boolean);
begin
```

```
  if TsgcUserSession(UserSession).Registered then
    DoSomething
  else
    DoNothing;
end;
```

8. If you want to handle AJAX requests, you need to handle OnAsyncRequest Event.

From Client Side, you can send parameters to server, using this javascript code:

```
sgcAsync('test', '&client_text=1234);
```

From Server Side, you can get client params using "Params" property.

```
procedure  sgcWebBrokerApplication1AsyncRequest(Sender:
TObject;   Method,   Params:   string;   UserSession:
TsgcWebBrokerUserSession;        var        Response:
TsgcWBJSResponse;
var
  oParams: TStringList;
  vClientText: String;
begin
  oParams := TStringList.Create;
  Try
    oParams.Delimiter := '&';
    oParams.StrictDelimiter := True;
    oParams.DelimitedText := aParams;

    if Method = 'test' then
      vClientText := oParams.Values['client_text'];
  Finally
    oParams.Free;
  End;
end;
```

Components

TsgcWebBrokerApplication

TsgcWebBrokerApplication implements Server User Sessions Management and can handle Async Events sent by browser clients. Follow next steps to configure this component:

1. Drop a TsgcWebBrokerApplication component in the TWebDataModule(you can drop only ONE).

2. Set Session Management type: by cookies, url or automatic.

3. You can handle the following events:

OnBeforeDispatch: used to send the response to an HTTP client Request or to handle any message.

OnAfterDispatch: used to send the response to an HTTP client Request or to handle any message if the response message hasn't been sent.

OnBeforeNewSession: every time a new request without session is received, this event is fired, using Accept variable you can Accept if a new session is created or not.

OnAfterNewSession: every time a new session is created, this event is fired. Here you can initialize user session properties.

```
procedure
TWebModule3.sgcWebBrokerApplication1AfterNewSession(Se
nder: TObject;
  UserSession: TsgcWebBrokerUserSession);
begin
  UserSession.Data := TsgcUserSession.Create;
end;
```

OnAsyncRequest: every time a client sends an AJAX request to server, this event is fired. Response variable is used to send a response to browser client.

```
procedure
TWebModule3.sgcWebBrokerApplication1AsyncRequest(Sende
r: TObject;
    Method,      Params:      string;      UserSession:
TsgcWebBrokerUserSession; var
    Response: TsgcWBJSResponse);
var
  oParams: TStringList;
begin
  oParams := TStringList.Create;
  Try
    oParams.Delimiter := '&';
    oParams.StrictDelimiter := True;
    oParams.DelimitedText := aParams;
    if   (oParams.Values['username']   <>   '')   and
(oParams.Values['password'] = '1234') then
    begin
      GetUserSession(FUserSession).Registered := True;
      GetUserSession(FUserSession).User              :=
oParams.Values['username'];
      FResponse.ShowMessage('Authorized    Access    to
user: ' + oParams.Values['username']);
      FResponse.Refresh;
    end
    else
      FResponse.ShowMessage('Verify      your      your
user/password and try again');
  Finally
    oParams.Free;
  End;
end;
```

Properties

DocumentRoot: if defined, every time a new request is received, component will search inside directory specified and if file requested is found, it will serve automatically.

Session Tracking: if you need to handle user sessions, you can specify which method is used to track User Sessions:

stCookies: uses cookies to handle user sessions, requires that browser clients have cookies enabled.

stURL: uses session id as query string parameters to handle user sessions.

stAuto: this is the default mode, first tries to use cookies, if cookies are not allowed, then uses url session id.

stNone: session management is disabled.

SessionTimeout: number of minutes before a session is closed.

SessionTimeoutURL: if defined, every time a session is closed, it sends browser client to a specified URL.

Sessions: returns a list of available user sessions.

UserSession: returns latest User Session processed.

Methods

CloseSession: Close a Session from Server Side.

AJAX

If you need to send an Ajax request to server, you need to call to sgcAsync Javascript method:

```
sgcAsync(method, params delimited by &);
```

Example1: send an async request without params:

```
<script>
  sgcAsync('servertime', '');
</script>
```

Example2: send an async request with 2 params:

```
<script>
  sgcAsync('rpc', '&number1=10&number2=20');
</script>
```

JQueryMobile

TsgcWebBrokerJQMTemplate

This is the Main component used to render JQueryMobile Styles on WebBroker.

You must drop this component on a form and link to TsgcWebBrokerApplication and a PageProducer.

This component has several Sub-components like navigation bars, list views, dialogs... all are grouped on Properties:

- **Enhance:** custom properties to enhance standard html components.
- **Safari:** browser properties allowing full-screen, scalable, splash-screen images...
- **HeaderBar:** header bar where you can define up to two buttons (example: home, help...), title...
- **FooterBar:** footer bar where you can define up to two buttons, footer text...
- **HeaderNavBar:** header navigation bar with up to five buttons, usually used for navigation between pages.
- **FooterNavBar:** footer navigation bar with up to five buttons, usually used for navigation between pages.
- **HeaderListView:** header list view, custom list view positioned on top of the page.
- **FooterListView:** footer list view, custom list view position on bottom of the page.
- **LeftSlidePanel:** left slide panel, allowing to build custom navigation panels.
- **RightSlidePanel:** right slide panel, allowing to build custom navigation panels.
- **Dialog:** build custom modal dialogs.
- **PopUp:** build custom on context popups.

TsgcWebBrokerJQMResources

This component is used to dispatch javascript and css files needed from JQueryMobile.

Drop this component on a form and assign to a TsgcWebBrokerApplication component.

You can select which JQueryMobile API do you want to use:

- JQueryMobile 1.2
- JQueryMobile 1.4

Properties

TsgcWebBrokerJQMTemplate | Enhance

This components allows to define properties to enhance standard Intraweb Components:

- **Enable:** if true, enhances standard intraweb components.
- **DataInline:** By default, all buttons in the body content are styled as block-level elements so they fill the width of the screen. However, if you want a more compact control that is only as wide as the text and icons inside or If you have multiple controls that should sit side-by-side on the same line, enable this property.
- **DataMini:** if enabled, creates a more compact version of controls.
- **Theme:** select one theme of the list.
- **TableBodyTheme:** select one theme of the list, applies to body grid.
- **TableColumnPriority:** default priority of grid columns.
- **TableColumnToggle:** if enabled, selectively hides columns at narrower widths as a sensible default but also offers a menu to let users manually control which columns they want to see. This mode is better for financial data tables that have compact values and need to maintain comparisons across columns and rows of data. It can also be used for building things like product comparison tables.
- **TableColumnText:** if not defined is "Columns..."
- **TableHeaderTheme:** select one theme of the list, applies to header grid.

TsgcWebBrokerJQMTemplate | Bar

HeaderBar is a toolbar at the top of the page that usually contains the page title text and optional buttons positioned to the left and/or right of the title for navigation or actions. Headers can optionally be positioned as fixed so they remain at the top of the screen at all times instead of scrolling with the page.

FooterBar is very similar to the header in terms of options and configuration. The primary difference is that the footer is designed to be less structured than the header to allow more flexibility, so the framework doesn't automatically reserve slots for buttons to the left or right as it does in headers.

- **Enabled:** if enabled, it shows toolbar.
- **Fixed:** enable in situations where the toolbar is a global navigation element, you may want it to appear fixed so it doesn't scroll out of view.
- **Fullscreen:** works just like the fixed mode except that the toolbars overlay the page content, rather than reserving a place in the document when not in fixed mode. This is useful for immersive apps like photo or video viewers where you want the content to fill the whole screen and toolbars can be hidden or summoned to appear by tapping the screen. Keep in mind that the toolbars in this mode will sit over page content so this is best used for specific situations.
- **Text:** Text of the toolbar.
- **Theme:** select one of the list.
- **Left/Right Button:**
  - **Enabled:** if enabled, it shows button.
  - **Active:** if enabled, shows button as active.
  - **Icon:** select one of the list.
  - **IconPos:** select icon position: centered, no icon, left...
  - **URL:** define url properties.

You can capture which button is selected, using OnAsyncBarButtonClick Event.

```
procedure   OnAsyncBarButtonClick(Sender:   TObject;   Id:
string);
begin
  if Id = 'sgcWBJQMHeaderBarLeftButtonURL' then
    // ... user selects Left Header Button
  else if Id = 'sgcWBJQMHeaderBarLeftButtonURL' then
    // ... user selects Right Header Button
```

```
end;
```

TsgcWebBrokerJQMTemplate | NavBar

sgcJQueryMobile has a NavigationBar widget that is useful for providing up to 5 buttons with optional icons in a bar, typically within a header or footer.

- **Enabled:** if enabled, it shows navigation bar.
- **Position:** select where navigation bar is positioned: header, content or footer.
- **Theme:** select one of the list.
- **Buttons:**
    - **Enabled:** if enabled, it shows button.
    - **Active:** if enabled, shows button as active.
    - **Icon:** select one of the list.
    - **IconPos:** select icon position: centered, no icon, left...
    - **URL:** define url properties.

You can capture which button is selected, using OnAsyncBarButtonClick Event.

```
procedure   OnAsyncBarButtonClick(Sender:   TObject;   Id:
string);
begin
  if Id = 'sgcWBJQMHeaderNavBarButton1URL' then
    // ... user selects First Header Button
  else if Id = 'sgcWBJQMHeaderNavBarButton2URL' then
    // ... user selects Second Header Button
end;
```

TsgcWebBrokerJQMTemplate | ListView

Lists are used for data display, navigation, result lists, and data entry so sgcJQueryMobile includes a wide range of list types and formatting examples to cover most common design patterns.

- **Enabled:** if enabled, it shows listview.
- **Divider:** List items can be turned into dividers to organize and group the list items.
- **DividerTheme:** select one theme of the list.
- **AutoDividers:** A listview can be configured to automatically generate dividers for its items, the text used to create dividers is the uppercased first letter of the item's text.
- **Header:** Header Customized Text.
- **Footer:** Footer Customized Text.
- **Search:** adds a filter with a simple client-side search feature.
- **SearchTheme:** select one theme of the list.
- **Read-Only:** if enabled, contents aren't linked.
- **Items:** is a TStringList where you can put a simple title, html text, links...

You can capture user selected item using OnAsyncListViewClick

```
procedure   OnAsyncListViewClick(Sender:   TObject;   Name:
string; Item: Integer);
begin
  if Item = 0 then
    // ... user selects first item
  else if Item = 1 then
    // ... user selects second item
end;
```

TsgcWebBrokerJQMTemplate | Dialog

Define custom Dialog Properties.

- **Enabled:** if true, dialog is enabled.
- **Title:** Title message.
- **Text:** Body message.
- **HeaderTheme:** select one theme of the list. Applies to header dialog.
- **ContentTheme:** select one theme of the list. Applies to content dialog.
- **NoCloseButton:** if true, enables close button.
- **Button1:** if defined, first button text.
- **Button2:** if defined, second button text.
- **Button3:** if defined, third button text.

You can capture Button clicked by user, using OnAsyncDialogClick, it returns Button Text as a parameter.

```
procedure OnAsyncDialogClick(Sender: TObject; Button: String);
begin
  if Button = 'Yes' then
    // ... user select yes button
  else if Button = 'No' then
    // ... no select no button
end;
```

TsgcWebBrokerJQMTemplate | PopUp

Here you can define a customized menu context.

- **Enabled:** if enabled, shows popup.
- **Text:** message text.
- **Menu:** shows list view items.

If you want to show PopUp, you must use ShowPopUp method. Use ClosePopUp if you wan to close an opened PopUp.

TsgcWebBrokerJQMTemplate | SlidePanel

sgcJQueryMobile has a SlidingPanel widget that is useful for provide a Navigation Menu on the left or the right of the page.

- **Enabled:** if enabled, it shows sliding panel.
- **Header:** Header Custom Text Message.
- **Footer:** Footer Custom Text Message.
- **DisplayMode:** Panels can display in three different ways
    - **Reveal:** The panel shifts the viewport over open, partially obscuring part of the page when it pushes it out of view.
    - **Push:** The panel behaves just like pan at narrow widths but at wider viewports (above 600 pixels), the content re-flows to make room for the panel so it can stay open permanently.
    - **Overlay:** The panel slides over the content like a standard menu.

To show SlidingPanel use these methods: ShowLeftSlidePanel or ShowRightSlidePanel.

If you need to Close a Panel, you can use: CloseLeftSlidePanel or CloseRightSlidePanel.

TsgcWebBrokerJQMTemplate | Safari

On this component you can set browser view properties.

- **Enabled:** if true, enables custom properties.
- **FullScreen:** the web application runs in full-screen mode.
- **IconURL:** default icon url
- **InitialScale:** The initial scale of the viewport as a multiplier. The default is calculated to fit the webpage in the visible area
- **PhoneDetection:** Enables or disables automatic detection of possible phone numbers in a webpage in Safari on iOS .
- **Scalable:** if enabled, user can zoom page, otherwise, it can't.
- **SplashScreenUrl:** image url.
- **Style:** This property has no effect unless you first enable FullScreen,

Events

TsgcWebBrokerJQMTemplate | Orientation Events

- **OnAsyncOrientationChange:** Triggers when a device orientation changes (by turning it vertically or horizontally). When bound to this event, the callback function has one argument, the event object. The event object contains an orientation property equal to either "portrait" or "landscape".

TsgcWebBrokerJQMTemplate | Touch Events

- **OnAsyncTouchTap:** Triggers after a quick, complete touch event.
- **OnAsyncTouchTapHold:** Triggers after a held complete touch event.
- **OnAsyncTouchSwipe:** Triggers when a horizontal drag of 30px or more (and less than 75px vertically) occurs within 1 second duration.
- **OnAsyncTouchSwipeLeft:** Triggers when a swipe event occurred moving in the left direction.
- **OnAsyncTouchSwipeRight:** Triggers when a swipe event occurred moving in the right direction.

Reference

WebBroker

Web Broker components (located on the Internet tab of the Tool palette) enable you to create event handlers that are associated with a specific Uniform Resource Identifier (URI). When processing is complete, you can programmatically construct HTML or XML documents and transfer them to the client. You can use Web Broker components for cross-platform application development.

More info:

http://docwiki.embarcadero.com/RADStudio/XE4/en/Using_Web_Broker
http://nickhodges.com/oldstuff/Delphi/Articles/DL180/dl180.htm

AJAX

Ajax is a group of interrelated web development techniques used on the client-side to create asynchronous web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. Data can be retrieved using the XMLHttpRequest object. Despite the name, the use of XML is not required (JSON is often used instead. See AJAJ), and the requests do not need to be asynchronous

More info

JQueryMobile

JQuery Mobile is an easy way to build sites and apps that are accessible on all popular smartphone, tablet and desktop devices.

The framework includes an Ajax navigation system that brings animated page transitions and a core set of UI widgets: pages, dialogs, toolbars, listviews, buttons with icons, form elements, accordions, collapsibles, and more.

jQuery Mobile has broad support for the vast majority of all modern desktop, smartphone, tablet, and e-reader platforms.

More Information

License

License

eSeGeCe Components End-User License Agreement

eSeGeCe Components ("eSeGeCe") End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and the Author of eSeGeCe for all the eSeGeCe components which may include associated software components, media, printed materials, and "online" or electronic documentation ("eSeGeCe components"). By installing, copying, or otherwise using the eSeGeCe components, you agree to be bound by the terms of this EULA. This license agreement represents the entire agreement concerning the program between you and the Author of eSeGeCe, (referred to as "LICENSER"), and it supersedes any prior proposal, representation, or understanding between the parties. If you do not agree to the terms of this EULA, do not install or use the eSeGeCe components.

The eSeGeCe components is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The eSeGeCe components is licensed, not sold.

The eSeGeCe components is a freeware. You may evaluate it for free and You can use the eSeGeCe components for commercial purpose. If you want SOURCE CODE you need to pay the registration fee. You must NOT give the license keys and/or the full editions of eSeGeCe (including the DCU editions and Source editions) to any third individuals and/or entities. And you also must NOT use the license keys and/or the full editions of eSeGeCe from any third individuals' and/or entities'.

1. GRANT OF LICENSE

The eSeGeCe components is licensed as follows:

(a) Installation and Use.

LICENSER grants you the right to install and use copies of the eSeGeCe components on your computer running a validly licensed copy of the operating system for which the eSeGeCe components was designed [e.g., Windows 95, Windows NT, Windows 98, Windows 2000, Windows 2003, Windows XP, Windows ME, Windows Vista, Windows 7].

(b) Royalty Free.

You may create commercial applications based on the eSeGeCe components and distribute them with your executables, no royalties required.

(c) Modifications (Source editions only).

You may make modifications, enhancements, derivative works and/or extensions to the licensed SOURCE CODE provided to you under the terms set forth in this license agreement.
(d) Backup Copies.
You may also make copies of the eSeGeCe components as may be necessary for backup and archival purposes.
2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS
(a) Maintenance of Copyright Notices.
You must not remove or alter any copyright notices on any and all copies of the eSeGeCe components.
(b) Distribution.
You may not distribute registered copies of the eSeGeCe components to third parties. Evaluation editions available for download from the eSeGeCe official websites may be freely distributed.
You may create components/ActiveX controls/libraries which include the eSeGeCe components for your applications but you must NOT distribute or publish them to third parties.
(c) Prohibition on Distribution of SOURCE CODE (Source editions only).
You must NOT distribute or publish the SOURCE CODE, or any modification, enhancement, derivative works and/or extensions, in SOURCE CODE form to third parties.
You must NOT make any part of the SOURCE CODE be distributed, published, disclosed or otherwise made available to third parties.
(d) Prohibition on Reverse Engineering, Decompilation, and Disassembly.
You may not reverse engineer, decompile, or disassemble the eSeGeCe components, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.
(e) Rental.
You may not rent, lease, or lend the eSeGeCe components.
(f) Support Services.
LICENSER may provide you with support services related to the eSeGeCe components ("Support Services"). Any supplemental software code provided to you as part of the Support Services shall be considered part of the eSeGeCe components and subject to the terms and conditions of this EULA.
eSeGeCe is licensed to be used by only one developer at a time. And the technical support will be provided to only one certain developer.
(g) Compliance with Applicable Laws.
You must comply with all applicable laws regarding use of the eSeGeCe components.
3. TERMINATION
Without prejudice to any other rights, LICENSER may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the eSeGeCe components in your possession.

4. COPYRIGHT

All title, including but not limited to copyrights, in and to the eSeGeCe components and any copies thereof are owned by LICENSER or its suppliers. All title and intellectual property rights in and to the content which may be accessed through use of the eSeGeCe components is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All rights not expressly granted are reserved by LICENSER.

5. NO WARRANTIES

LICENSER expressly disclaims any warranty for the eSeGeCe components. The eSeGeCe components is provided "As Is" without any express or implied warranty of any kind, including but not limited to any warranties of merchantability, non-infringement, or fitness of a particular purpose. LICENSER does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the eSeGeCe components. LICENSER makes no warranties respecting any harm that may be caused by the transmission of a computer virus, worm, time bomb, logic bomb, or other such computer program. LICENSER further expressly disclaims any warranty or representation to Authorized Users or to any third party.

6. LIMITATION OF LIABILITY

In no event shall LICENSER be liable for any damages (including, without limitation, lost profits, business interruption, or lost information) rising out of "Authorized Users" use of or inability to use the eSeGeCe components, even if LICENSER has been advised of the possibility of such damages. In no event will LICENSER be liable for loss of data or for indirect, special, incidental, consequential (including lost profit), or other damages based in contract, tort or otherwise. LICENSER shall have no liability with respect to the content of the eSeGeCe components or any part thereof, including but not limited to errors or omissions contained therein, libel, infringements of rights of publicity, privacy, trademark rights, business interruption, personal injury, and loss of privacy, moral rights or the disclosure of confidential information.

Index