

Using the Resource Database to adapt to session timeouts

**G J Boris Allan, Joseph Allen, and Siu Wan
Westat, USA**

IBUC 2020, Limassol, Cyprus

A note on Blaise versions

When this original paper was written for the March/April 2020 IBUC, we were using Blaise 5.6.x and so some of the details in that paper have been superseded later versions of Blaise. Given time constraints, and the novelty of our new working environments (as exemplified by this virtual IBUC) we have not been able address all possible changes.

For example, this presentation is being finished when the current versions are Blaise 5.7.9 and Blaise 5.8.3, and we have production versions of instruments using a variety of Blaise 5.7.x versions. Some problems we highlight have been addressed by later Blaise versions.

This means that (generally) our examples are from Blaise 5.6.x and do not reflect the latest state-of-affairs, however, we think that this approach is generic and not overly dependent on a Blaise version.

Some ways of stopping and restarting a web page

- › Users close a web page by selecting X top right on the browser or selecting X on the relevant browser tab. We will term this “X-out”. The tab is lost and there is no currently-visible URL. This is likely to happen if the client does not want a “Save and Exit” button.
- › Users refresh the information on a page for some reason. Sometimes selecting refresh is accidental. By refreshing, the currently-visible URL is relaunched.
- › Somebody has made a copy of the web-page URL and the page is closed. Later the web page is relaunched using the copy of the URL.

All three forms of restart involve security issues and checks for restart are discussed in our earlier paper. Each check results in a resource database Quit command.

Session data and the survey database (1)

- › On starting a Blaise 5 web interview, the server creates a server-session database by copying data from the survey database. Data collected from web pages modify this session database.
- › At the end of an interview, or after a Quit action from the resource database, the data in the session database are copied to the survey database, and the session database is closed.
- › To X-out of a web page means that the Blaise server does not know that the user has disconnected (it is not proactive) – the session database is left open, to be closed after a timeout period.
- › Proactively, our management system (MS) could monitor the status of the remote user, know when an X-out happened, and know that the session database should be closed.

Session data and the survey database (2)

- › If there has been an X-out, the user might try to return to the interview legitimately via our MS. However, if the return is too swift, the Blaise server might not have acknowledged a timeout and still have an open session database.
- › After a correct MS login, the MS (using the API) starts an interview, setting the MgtSys.IsLaunched field to Yes – which fails. We cannot start because there is a locked session database: the session data are still live because not enough time has elapsed for a timeout.

Session data and the survey database (3)

- › The integrity of the Blaise database has the highest priority so allowing a user (however well-qualified) to remove a lock had to be considered carefully. Automatically locking a record is a good default.
- › In some situations record-locking can create issues, and further discussions with the Blaise developers began at this point. The integrity of the Blaise database is (as ever) paramount but now API users have been given the ability to remove a record lock.

Session data and the survey database (4)

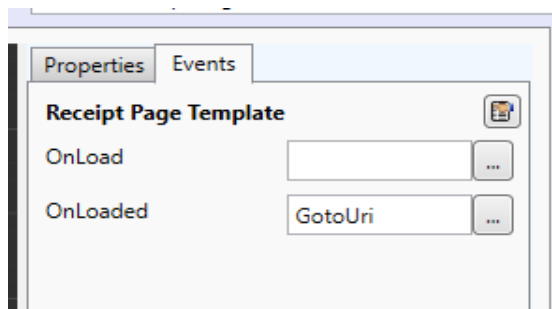
- › Another approach we considered was using the API in the MS to see the original session instance, and then closing that instance using the Quit method of the resource database.
- › The MS (via the API) would then loop waiting for the session to close, and launch a new session on that close.
- › There were problems with the time taken for the original session to close – it was very unpredictable, from a second or two to multiple minutes. Not reliable.

Session data and the survey database (5)

- › In our normal resource-database security processing, whenever we start a new interview session we compare the previous session GUID to the new session GUID. If the two session GUIDs are the same, we know that the previous session was not stopped (say, an X-out). We return to the MS using a return URL stored in the session data, whilst issuing a Quit and Save for the session.
- › When a session times out the session data are saved to the survey database, and the session database exists no longer. After a timeout we cannot access session data such as ExitURL. We have no indication that this is possible by some other technique.
- › Blaise generates an error that says there has been a server timeout (using a *generic* error-page template) and what we do is create a *specific* error-page template for timeout.

Ending an interview: Receipt Page Templates

- This is the way we would like all interviews to end. We get a referral to the MS by means of the OnLoaded event, which uses a field reference to ExitURL.
- Here is a description of the action, using the Expression language of the resource database for the Onloaded event of the template:
`<ReceiptPageTemplate OnLoaded="{Action GotoUri({Expression ExitURL.ValueAsText},{Values 'Target=_parent'})}">`



Ending an interview: Abort Page Templates (1)

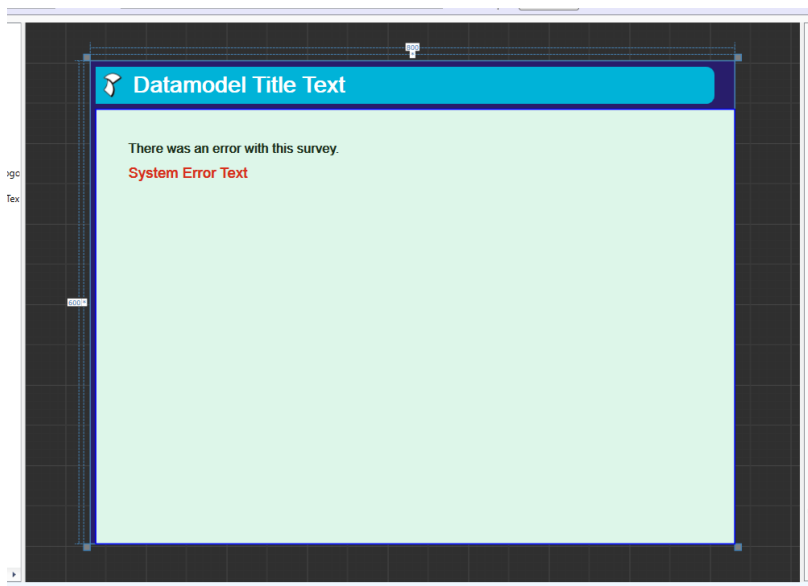
- › In the resource database, the instruction to Quit (used quite frequently) opens an Abort page. In our system, the Default abort page does not use a field reference but uses a server variable named URL – for more flexibility.
- › `<AbortPageTemplate OnLoaded="{Action GotoUri({Expression ServerVariables.GetString('URL')},{Values 'target=_parent'}})">`
- › This is directly akin to the receipt page OnLoaded event.

Ending an interview: Error Page Templates (1)

- If an instrument encounters a “system” error, such as a record locked by another user, then an error is shown. The list of system error codes is long, and includes:
 - InvalidKeyValue,
 - RecordExists,
 - ServerAccessDenied,
 - SessionExpired,
 - BrowserNotSupported,
 - SurveyNotActive,
 - RecordNotFound

Ending an interview: Error Page Templates (2)

- You will probably be aware of these error pages because of the Default error-page template, and its red system error text:



Ending an interview: Error Page Templates (3)

- › In general, we do not change the default error page template because we do not want the user to have any easy way back to the survey MS, as the user could be an intruder testing ways into the instrument (hacking).
- › We do, however, have a special TimeOut error-page template for one specific error – in the list of error codes the code named “SessionExpired”.
- › In the Error Page Templates list of the resource database we have a new template named “TimeOut”, which is first in the list of error templates. This means that Blaise 5 looks at this template first and then decides if the template is applicable for the current session error.

The Timeout error page template: Version 1 (1)

- Looking at the underlying expressions for this version of the Timeout error template, two interesting aspects are OnLoaded and ErrorCodes; and our problem is the hardcoded value of the URL:
- `<ErrorPageTemplate OnLoaded="{Action GotoUri({'https://www.westat.com', {Values 'target=_parent'}})}" ErrorCodes="SessionExpired">`
- We cannot use server variables or field references for our GotoURI because the session expired and, with the session expiration, the variable references also expired (and we cannot keep them alive).
- Only errors that are "SessionExpired" are recognized by this template, and so – as TimeOut is first in the list of error page templates – whenever the session expires this page is selected.

The Timeout error page template: Version 1 (2)

› ErrorCode

Error Page Template

ApplicabilityCondi



ApplyAtRuntime



Comment



DesignHeight

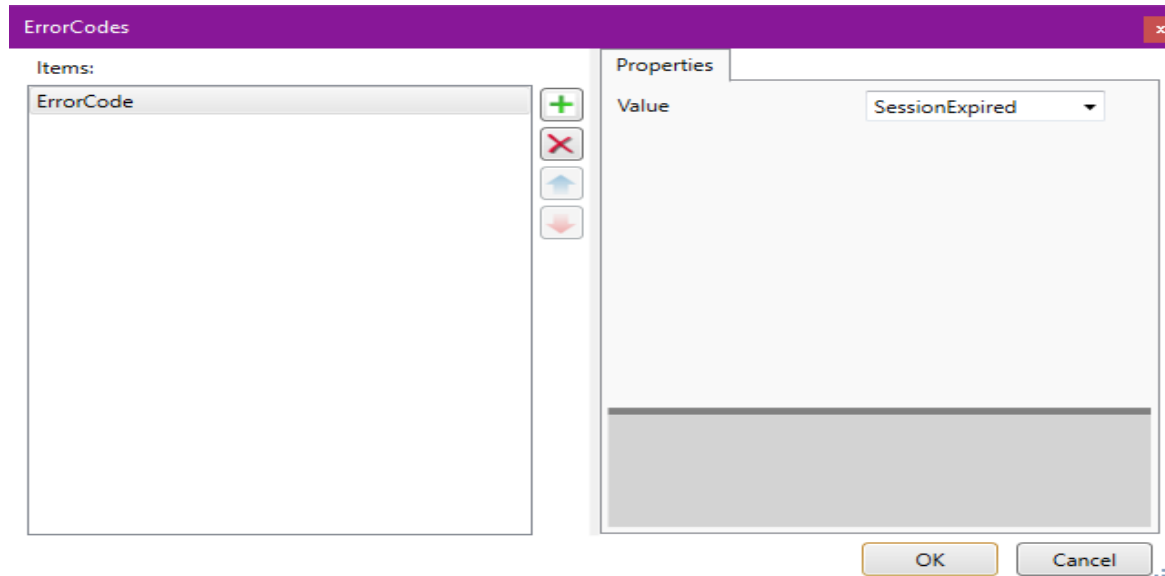
DesignWidth

ErrorCodes



The Timeout error page template: Version 1 (3)

› SessionExpired:



The Timeout error page template: Version 2 (1)

- › In an attempt to give more control over where we send the user after a session has expired, we needed to start a new session about which we had clear knowledge, and then act on information in that new session (a bit like our action with Quit/Abort) to send the user to a web page.
- › If we hardcode the web-page location in the OnLoad event for TimeOut, whenever the location changes then we have to make the change and reinstall the survey.
- › The new session we create is back on the Blaise server and is attached to a “dummy” record/case we know as “*”.

The Timeout error page template: Version 2 (2)

- › The * case is part of the main survey database, and we can assign (via the API) appropriate values to fields in MgtSys.
- › To start the survey with * in the current survey database, we use the OnLoad event not OnLoaded (client-side) because the action we want cannot be selected for that event. Here is the expression:
- ›

```
<ErrorPageTemplate OnLoad= "{Action StartSurvey({Expression  
State.InstrumentId}, {Values '*'},{','})}"  
ErrorCodes="SessionExpired">
```
- › When we load the TimeOut page, we run the StartSurvey action for the current survey (which is based on the resource database State.InstrumentId variable, not lost with the session).

The Timeout error page template: Version 2 (3)

- This technique seemed to work during development but then we put it through more extensive testing. What happened in practice is that the previous session seemed (sometimes) to be still alive and not ended, so that the previous key value was replaced by *, but the rest of the data from the previous key value remained. We decided for the pretest to hardcode the return URI.
- After that decision was made, Blaise 5.6.9.8022 was released and in the change log we read that there had been a problem with Action StartSurvey, very relevant to our experience. We did not change anything because we were live.