



A Signify Developer's guide to debugging your own (or someone else's) code

A photograph of a desk setup, partially obscured by a red overlay. It features a computer monitor, a keyboard, and a small sign that reads "NEVER SETTLE".

Before we start

“If debugging is the process of removing software bugs, then programming must be the process of putting them in.”

- Edsger Dijkstra

Overview



- **Always** assume there is a bug in your code.
- Start with what you know. And work from there.
- Don't panic, be solution-oriented.
- Do you have enough information?
- Can you replicate the bug?
- VERY IMPORTANT: **Replicate the bug first** before starting to change code.
- DURING DEVELOPMENT: Thank QA for finding bugs in your code.



Where do I start?

- Server-side error: start with the stack trace. (.../SignifyErrorlog.axd).
- Database error (could be spotted in the stack trace): identify SQL code causing issue (SQL Profiler).
- Display (look and feel) error: analyse styles used on page (Chrome DevTools).
- Client-side script error: analyse scripts used on page (Chrome DevTools).
- API calls: (Fiddler, Chrome DevTools).
- Errors when loading a Partial View: (Chrome DevTools).

Server-side error

- Look for the last time in the stack trace that it touched the system's code.

System.Data.SqlClient.SqlException

The DELETE statement conflicted with the REFERENCE constraint "FK_pwNodeTargetAudienceOrgNodes_orgNodes". The conflict occurred in database "SignifyHR_MasterQA", table "dbo.pwNodeTargetAudienceOrgNodes", column 'orgNodeId'. and The statement has been terminated.


```
System.Web.HttpUnhandledException (0x80004005): Exception of type 'System.Web.HttpUnhandledException' was thrown. ----> System.Data.SqlClient.SqlException (0x80131904): The DELETE statement conflicted with the RE
end
The statement has been terminated.
at System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction)
at System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction)
at System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHasConnectionLock, Boolean asyncClose)
at System.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, TdsParserStateObject stateObj, Boolean& dataReady)
at System.Data.SqlClient.SqlCommand.FinishExecuteReader(SqlDataReader ds, RunBehavior runBehavior, String resetOptionsString, Boolean isInternal, Boolean forDescribeParameterEncryption, Boolean shouldCacheFor
at System.Data.SqlClient.SqlCommand.RunExecuteReaderTds(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, Boolean async, Int32 timeout, Task& task, Boolean asyncWrite, Boolean inRetr
at System.Data.SqlClient.SqlCommand.RunExecuteReader(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, String method, TaskCompletionSource`1 completion, Int32 timeout, Task& task, Bo
at System.Data.SqlClient.SqlCommand.InternalExecuteNonQuery(TaskCompletionSource`1 completion, String methodName, Boolean sendToPipe, Int32 timeout, Boolean& usedCache, Boolean asyncWrite, Boolean inRetry)
at System.Data.SqlClient.SqlCommand.ExecuteNonQuery()
at Microsoft.ApplicationBlocks.Data.SqlHelper.ExecuteNonQuery(SqlConnection connection, CommandType commandType, String commandText, SqlParameter[] commandParameters) in D:\Program Files\Signify Software\Signi
at Microsoft.ApplicationBlocks.Data.SqlHelper.ExecuteNonQuery(String connectionString, CommandType commandType, String commandText, SqlParameter[] commandParameters) in D:\Program Files\Signify Software\Signi
at SQL.ExecuteNonQuery(String spName, SqlParameter[] parms) in D:\Program Files\Signify Software\SignifyHR_Build\Web\App\Framework\SQL.cs:line 226
at Signify.HumanResources.Organization.Unit.DeleteUnit() in D:\Program Files\Signify Software\SignifyHR_Build\Web\App\HumanResources\Organization\Unit.cs:line 503
at HumanResources.Organization_Admin_PlanDetail.DeleteUnit() in D:\Program Files\Signify Software\SignifyHR_Build\Web\App\HumanResources\Organization\Admin\PlanDetail.aspx.cs:line 869
at System.Web.UI.WebControls.LinkButton.OnClick(EventArgs e)
at System.Web.UI.WebControls.LinkButton.RaisePostBackEvent(String eventArgument)
at System.Web.UI.WebControls.LinkButton.System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(String eventArgument)
```

- Identify inputs used by Client/QA, if required for debugging.
- Replicate the issue consistently on local. **IMPORTANT: make sure your setup on local is the same as the client's.**
- Start to change the code using the same inputs as QA/Client to verify the solution



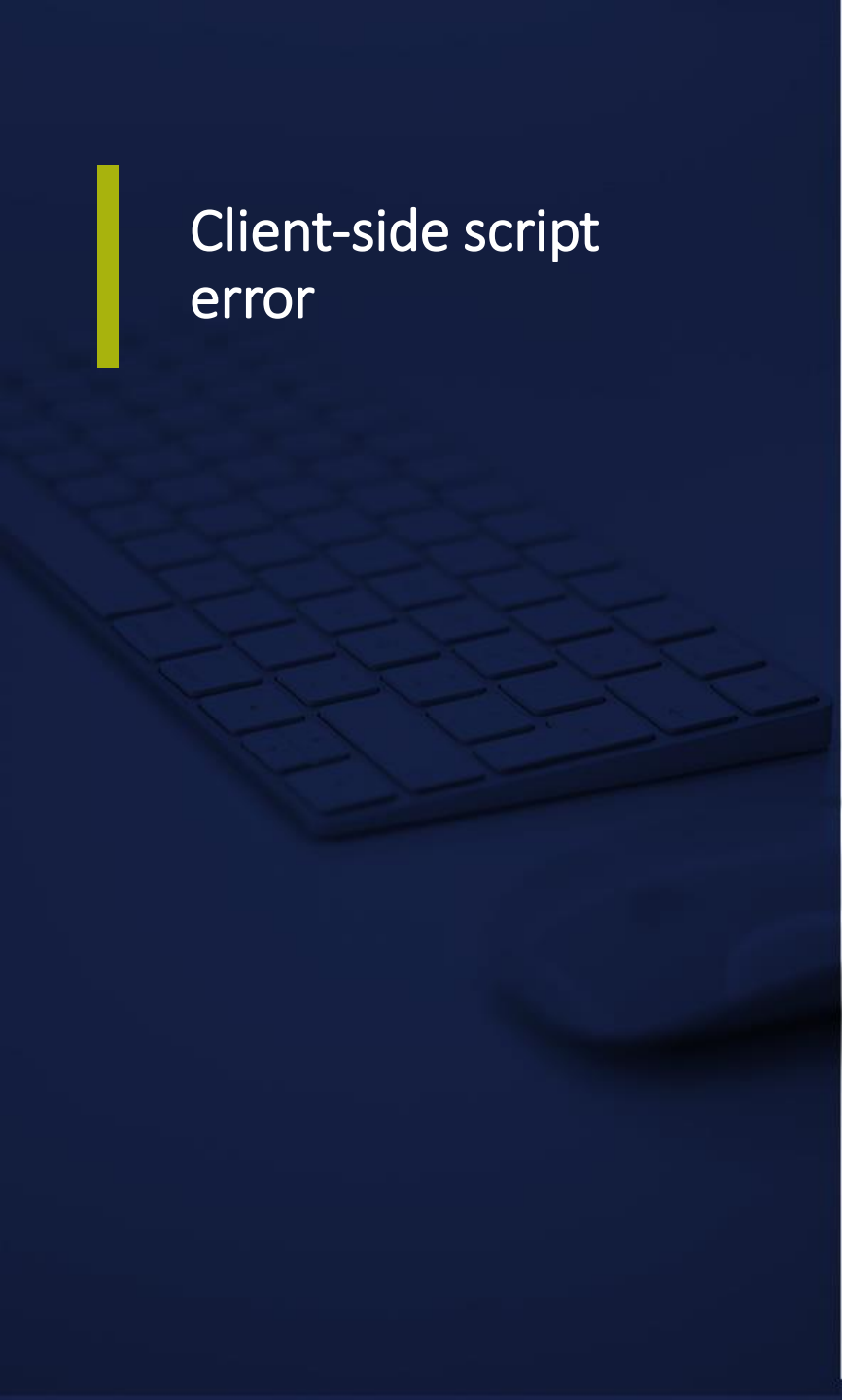
Database error

- Use SQL Profiler to identify query/stored procedure causing error.
- Run query in SQL Server Management Studio with parameters.
- **TIP:** Use PRINT statements if
 1. Procedure is complex and has many points of failure
 2. Dynamic SQL is used.
- **TIP:** Duplicates?
 1. Remove joins.
 2. Add joins one at a time to determine which one causes duplicates.
- **TIP:** Poor performance?
 1. Check that **With(nolock)** table hint is used in selects.
 2. Do not use in-line user defined functions when selecting 1000+ records
 3. Reports – filter results as early as possible, use temp tables wisely.
 4. Query Execution Plan



Display (look and feel) error

- Use DevTools in Chrome (Inspect/Ctrl+Shift+I).
- Find origin of styling.
- Change HTML markup.
- Preview changes.



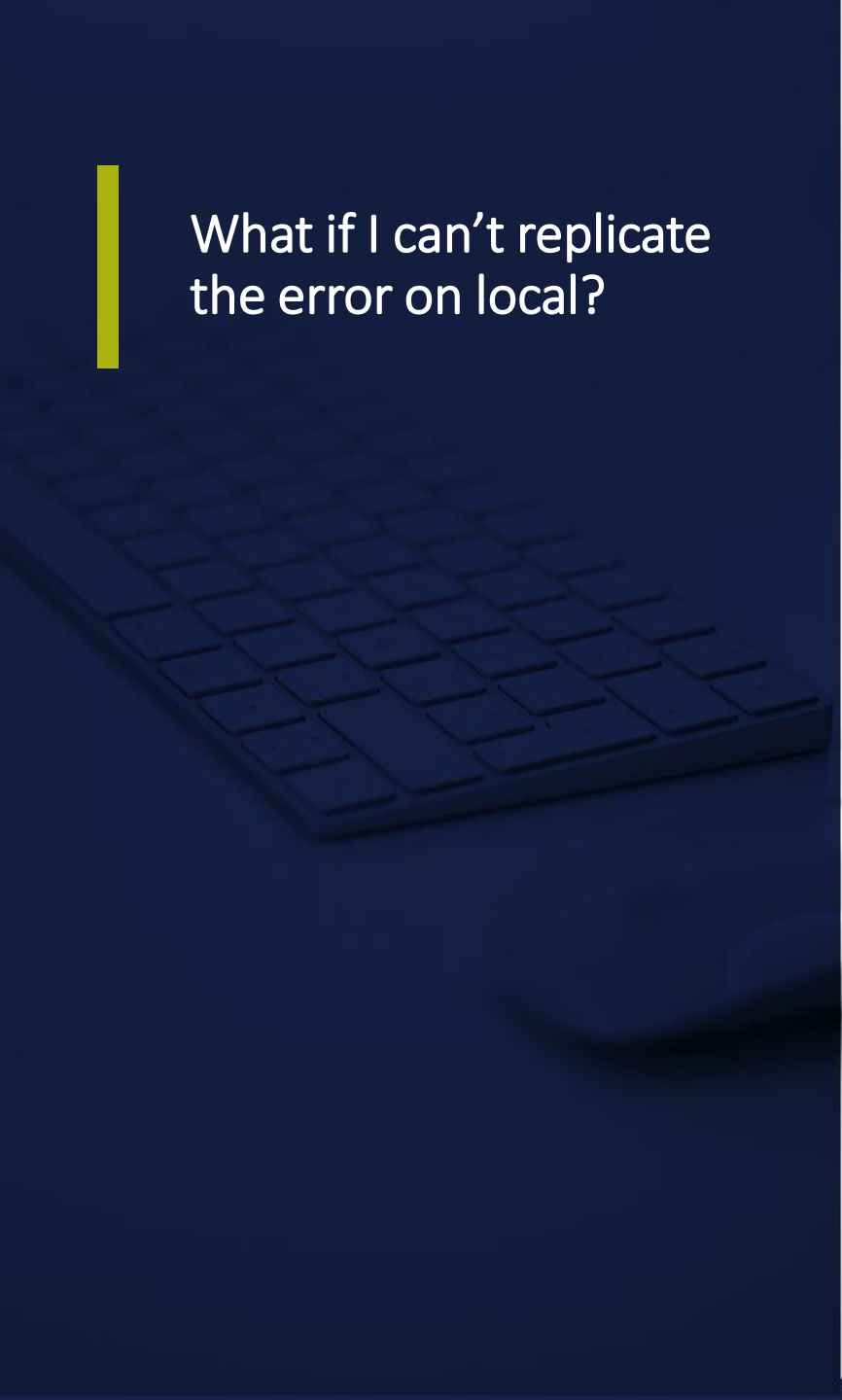
Client-side script error

- Analyse scripts used on page (Chrome DevTools).
<https://developers.google.com/web/tools/chrome-devtools/javascript/>
 1. Find HTML markup for selected control in **Elements**
 2. Place breakpoint in **Sources**
 3. Run query in **Console**
- When debugging javascript on a partial view, add “debugger” to script in **Sources** – javascript is added dynamically when partial is loaded.
- **SOAPUI** for SOAP and REST testing.
- **Fiddler** <http://www.telerik.com/fiddler>
 1. Debug API calls by building request to simulate a post / get.



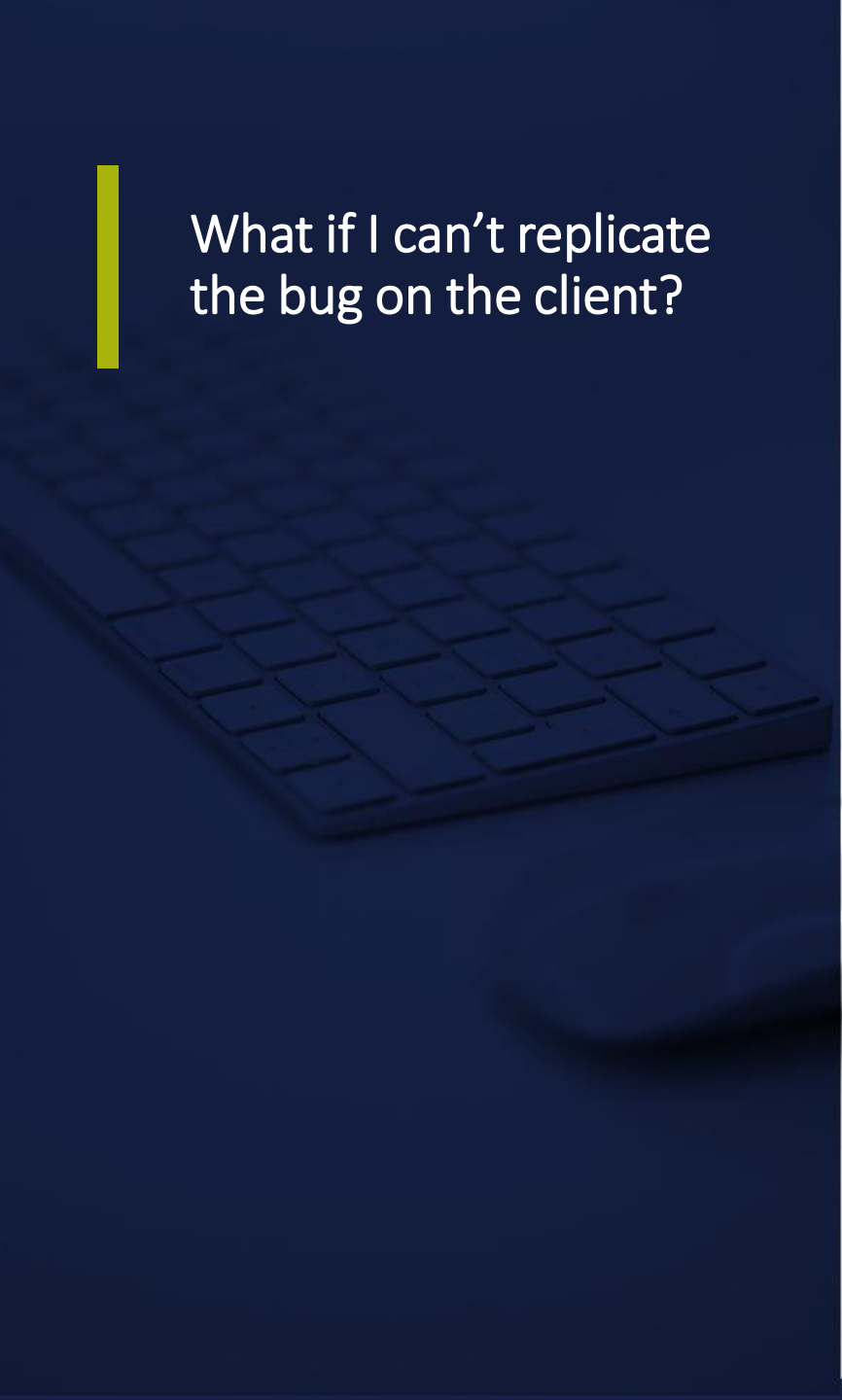
Errors when loading a Partial View

- Errors on Partials are consumed.
- Use the Network tab to see the actual error.
- Follow Server Side error steps from this point onwards.



What if I can't replicate the error on local?

- Get the client's database (ask the client's permission first) .
- **IMPORTANT:** if the client's database is visible from the Signify network or while logged onto their VPN, your connection string can point directly to the database without bring the database to Signify. BUT, take great care not to influence the client's system while this is done.
- Restore the database on THESEUS\SQL2019.
- Reset your repository to the client's version (available on Master Builder).
- Add breakpoints in your code where the error is experienced (according to the stack trace).
- Follow the steps to replicate (supplied by QA).



What if I can't replicate the bug on the client?

- Check UserLog table to identify browser used by user when error occurred.
Some bugs are browser or browser-version specific.
- As a last resort, ask permission from the client to contact the user directly.
Hold a Teamviewer session with the user and let them talk you through the steps they follow to replicate the issue.

A vertical white bar is positioned to the left of the 'Quotes' text. The background of the slide features a red-tinted image of a desk with a computer monitor, keyboard, and a small sign that says 'NEVER SETTLE'.

Quotes

- “When debugging, novices insert corrective code. Experts remove defective code.” – **Richard Pattis**
- “Sometimes it pays to stay in bed on Monday, rather than spending the rest of the week debugging Monday’s code.” – **Dan Salomon**
- “Only half of programming is coding. The other 90% is debugging.” – **Unknown**
- “Weeks of coding can save hours of planning.” - **Unknown**



The End