



Debugging and Monitoring



Topics

- Debugging in Service Studio
 - During Debugging
 - Inspecting Variables
 - Debugging a Producer Module
 - Testing / Debugging Mobile Apps
- Monitoring in OutSystems
 - Service Center Logs

Troubleshooting applications

Code is subject to many errors during execution

Troubleshooting focuses on finding those errors so they can be fixed

Troubleshooting is generally a combination of:

- Debugging
 - Suspends code execution at given points
 - Analyze the value of variables
 - Developer can execute the code step by step
- Monitoring
 - Analyzing and correlating system logs
 - OutSystems records events in logs
 - Errors, slow queries, web service calls, ...

A full-page background image featuring an astronaut in a white spacesuit floating in the black void of space, surrounded by stars. The astronaut is holding a silver laptop in their lap. The word "Debugging" is overlaid in white text on the left side of the astronaut.

Debugging

Debugging in Service Studio

Place breakpoints on elements that generate code

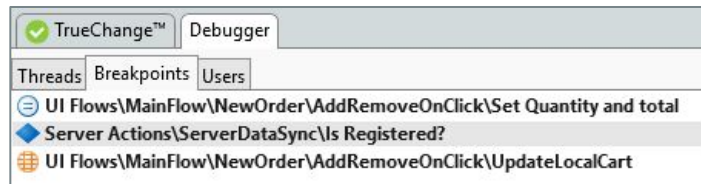
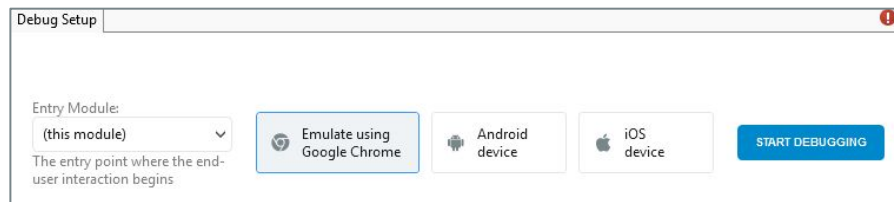
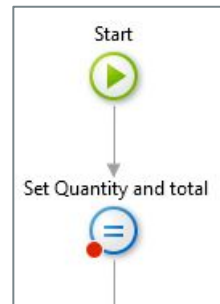
- Server and Client Actions

Start the debugger before opening the app in the browser

- Service Studio registers the breakpoints
- Debugging can be done using the browser emulator or the device

Breakpoints are listed in the Debug panel

- Can be temporarily disabled
- Can be removed



During Debugging

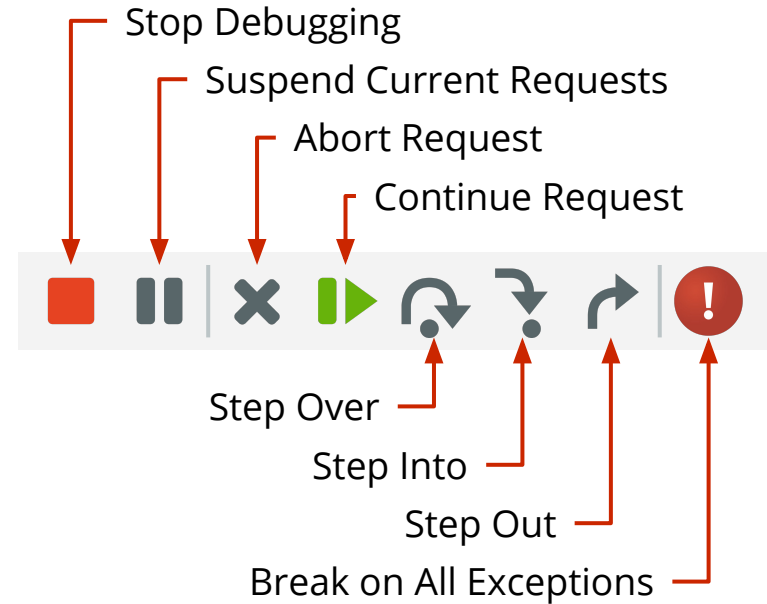
The following commands are available once code execution stops at a breakpoint

Regular debugging operations

- Stop, Continue, Suspend, Abort
- Step Over, Step Into, Step Out

Break on All Exceptions forces the debugger to suspend the execution of threads when exceptions occur

- Handled or unhandled

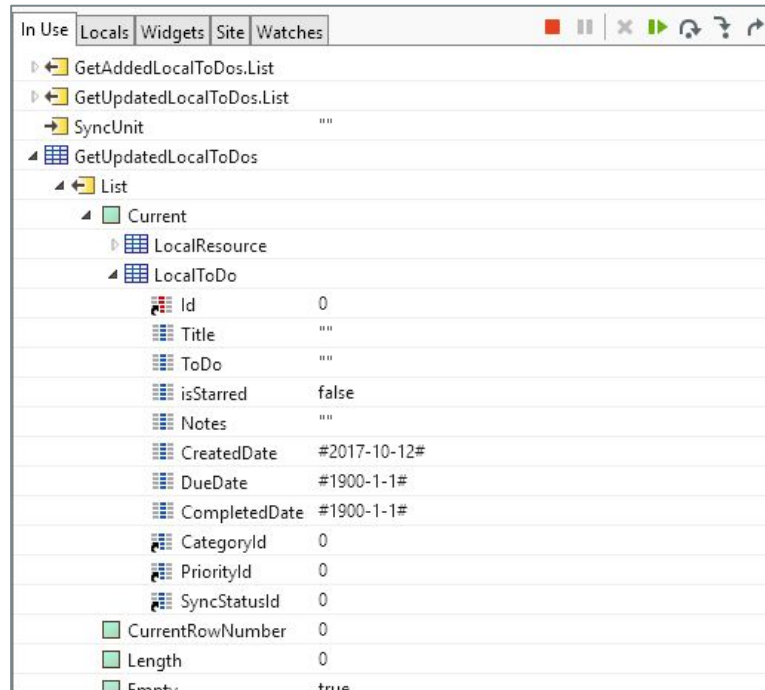


Inspecting Variables during Debugging

Inspect the values of variables when code execution is stopped

- **In Use** by the current code statement
- **Locals** to the current execution scope
- Stored in this Screen's **Widgets**
- Global to the **Site** (i.e. application)
- Selected by the developer for quick access to always be under **Watch**

Watches are always displayed, even if out of scope



Debugging a Producer Module

By default, breakpoints stop execution of requests made to that module *directly*

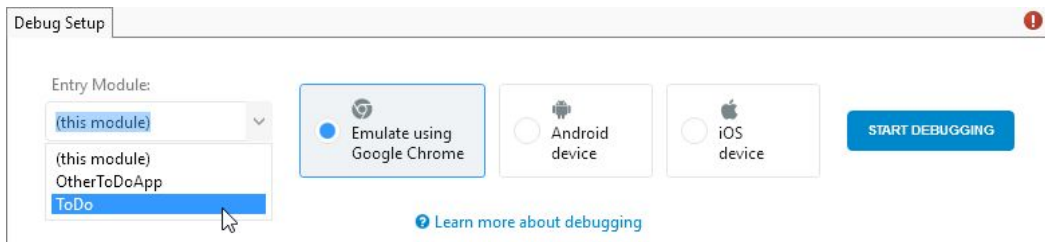
- Requests to *https://myserver.com/MyApp/* will stop on breakpoints inside MyApp

It is possible to stop the execution in another module's breakpoints

- Request to a consumer can stop in a producer
- e.g. Requests to *https://myserver.com/ToDo/* to stop on ToDo_Core breakpoints

To stop the execution in ToDo_Core, ToDo must be selected as entry module

- Requests start in the consumer module, but can stop on Core's breakpoints
- This can be done in any producer module of the application



Debugging Options

On Mobile apps there are three debug options:



Emulate using
Google Chrome



Android device



iOS device

Outsystems Now can also be used on Android to
debug mobile apps running on the device

Testing Mobile Apps

- Google Chrome emulation allows to quickly test mobile apps
 - Inspect the source code
 - Does not test offline and plugin behaviors
 - OutSystems Now on Android is an alternative to debug the app on the device
- Android and iOS Debugging
 - Requires installing the native generated app
 - Connect the device with an USB Cable
 - Allows to better troubleshoot the performance and user experience
 - Offline scenarios
 - Plugin interaction
 - https://success.outsystems.com/Documentation/11/Developing_an_Application/Troubleshooting_Applications/Debugging_Applications

A full-page background image featuring an astronaut in a white spacesuit floating in the black void of space, surrounded by stars. The astronaut is holding a silver laptop in front of their chest. The word "Monitoring" is superimposed in white text over the astronaut's torso.

Monitoring

Monitoring in OutSystems

OutSystems tracks the occurrence of many events in distinct **Logs**

Logged events include situations for which diagnostics may be required:

- Errors
- Database query durations
- Web Service calls
- Mobile requests
- Native app generation

Monitoring is the act of analyzing and correlating these logs



Service Center Logs

Service Center provides log reports

- **Errors**
- **General** audits
- **Mobile** requests
- **Integration** calls
- **Extension** calls
- Overall **Environment Health**
- ... and more

Reports can be filtered by:

- application
- time window
- message content

and can be exported to Excel

The screenshot displays the OutSystems Service Center interface. The top navigation bar is red with the OutSystems logo and tabs for Factory, Monitoring (selected), Administration, and Analytics. Below this, a breadcrumb trail shows the path: Development > Errors > General > Web Requests > Mobile Requests > Service Actions > Integrations > Extensions > Timers > Emails > Processes > Mobile Apps > Environment Health > Security. The main content area is titled 'Error Log' and includes a sub-header: 'List of Error events logged by both the system and the applications including the exceptions raised by them. Select an eSpace ...'. Below this is a filter section with dropdowns for 'Application' (set to 'OSMDB') and 'eSpace' (set to '(All)'), and input fields for 'From' and 'To' dates. There are also buttons for 'Filter' and 'Reset'. Below the filter section is a table of error logs. The table has columns for 'Time of Log', 'eSpace', 'Message', 'Module', and 'Server'. The first row shows an error from 2018-10-16 21:56:05 in the OSMDB eSpace, with the message 'Request timed out.' and details for Global ACW16PRDP11. The second row shows an error from 2018-10-16 00:21:03 in the OSMDB eSpace, with a detailed message about a FOREIGN KEY constraint conflict in the 'dbo.OSUSR_9J3_PERSONROLE2' table, and details for ACW16PRDP11. The third row shows an error from 2018-10-11 09:40:55 in the OSMDB eSpace, with the message 'The file 'OSMDB\webscreen.aspx' does not exist.' and details for Global ACW16PRDP11. Each row has a 'Detail' link.

Time of Log	eSpace	Message	Module	Server
2018-10-16 21:56:05	OSMDB	Request timed out.	Global	ACW16PRDP11
2018-10-16 00:21:03	OSMDB	The INSERT statement conflicted with the FOREIGN KEY constraint "OSFRK_OSUSR_9J3_PERSONMOVIEROLE1_OSUSR_9J3_PERSONROLE2_PERSONROLEID". The conflict occurred in database "OutSystems_OS11", table "dbo.OSUSR_9J3_PERSONROLE2", column "ID". The statement has been terminated.	ACW16PRDP11	
2018-10-11 09:40:55	OSMDB	The file 'OSMDB\webscreen.aspx' does not exist.	Global	ACW16PRDP11

Log Detail

Clicking on the 'Detail' of an Error log shows extensive information on the logged event

Shows details like

- Message text
- Call Stack

Important for debugging an application during development

outsystems Factory Monitoring Administration Analytics

OS11 Training Errors General Web Requests Mobile Requests Service Actions Integrations Extensions Timers Emails Processes Mobile Apps Environment Health Security

Error Detail

[Back to Log](#)

Id: 0d6fe141-66d0-49a3-be95-8aade5864f2
Time of Log: 2018-10-16 00:21:03
eSpace: OSMDb
Tenant: Users
User: (23)
Session Id: o452nffleizm4j3ddvwmrm
Server: ACW16PRDP11
Module:

Message: The INSERT statement conflicted with the FOREIGN KEY constraint "OSFRK_OSUSR_9J3_PERSONMOVIEROLE1_OSUSR_9J3_PERSONROLE2_PERSONROLEID". The conflict occurred in database "OutSystems_OS11", table "dbo.OSUSR_9J3_PERSONROLE2", column "ID". The statement has been terminated.

Environment Information:
eSpaceVer: 0 [Id=100, Publ=9, CompiledWith=11.0.108.0]
RequestUrl: https://os11training.outsystems.net/OSMDb/AddMovieParticipant.aspx (Method: POST)
AppDomain: LM\W3VC\1\ROOT\OSMDb-219-131841184259997830
FilePath: E:\OutSystems\Platform_Server\running\OSMDb-496610101\AddMovieParticipant.aspx
ClientIp: 172.16.154.4
Locale: en-US
DateFormat: yyyy-MM-dd
PID: 8164 [womp], Started="09/10/2018 07:48:00", Priv=1138Mb, Virt=2116625Mb
TID: 80
Thread Name: NET: 4.0.30319.42000

The INSERT statement conflicted with the FOREIGN KEY constraint "OSFRK_OSUSR_9J3_PERSONMOVIEROLE1_OSUSR_9J3_PERSONROLE2_PERSONROLEID". The conflict occurred in database "OutSystems_OS11", table "dbo.OSUSR_9J3_PERSONROLE2", column "ID". The statement has been terminated.

Stack:
at System.Data.SqlClient.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseAction)
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.SqlDataReader.TryConsumeMetaData()
at System.Data.SqlClient.SqlDataReader.get_MetaData()
at System.Data.SqlClient.SqlCommand.FinishExecuteReader(SqlDataReader ds, RunBehavior runBehavior, String resetOptionsString, Boolean isInternal, Boolean forDescribeParameterEncryption)
at System.Data.SqlClient.SqlCommand.RunExecuteReaderTds(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, Boolean async, Int32 timeout, Task& task, Boolean& usedCache, Boolean asyncWrite, Boolean inRetry)
at System.Data.SqlClient.SqlCommand.RunExecuteReader(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream, String method, TaskCompletionSource`1 completion, Int32 timeout, Task& task, Boolean& usedCache, Boolean asyncWrite, Boolean inRetry)
at System.Data.SqlClient.SqlCommand.ExecuteReader(CommandBehavior behavior, String method)
at OutSystems.HubEdition.DatabaseProvider.SqlServer.ExecutionService.ExecutionService.ExecuteReader(DbCommand cmd)
at OutSystems.HubEdition.DatabaseProvider.SqlServer.ExecutionService.ExecutionService.ExecuteScalar(DbCommand cmd)
at OutSystems.Internal.Db.Command.ExecuteScalar(String description, Boolean isApplication, Boolean skipLog, Boolean applyTransformationsToParameters)
at ssOSMDb.ExtendedActions.CreatePersonMovieRole(HeContext heContext, RCPersonMovieRoleRecord inParamSource, Int64 outParamId)
at ssOSMDb.Flows.FlowMainFlow.ScmAddMovieParticipant.CommandSave(HeContext heContext)

[Submit Feedback](#)

Summary

- Debugging in Service Studio
 - During Debugging
 - Inspecting Variables
 - Debugging a Producer Module
 - Testing / Debugging Mobile Apps
- Monitoring in OutSystems
 - Service Center Logs



**Debugging and
Monitoring
Thank You!**