

Uniface Debugger Quick Reference Card



Starting Debugger (Client Side)

Application process	Uniface Debugger
idf /deb	(none - automatically started)
uniface.exe /deb APS	(none - automatically started)
uniface.exe /deb=tcp:+13001:nowait APS (Starts immediately does not wait for Debugger)	udbg.exe /deb=tcp:+13001:create (Halts app if already started)
uniface.exe /deb= tcp:+13001:wait APS (Waits for start of Debugger)	udbg.exe /deb=tcp:+13001:create
uniface.exe /deb=tcp:+13001:create APS (Waits for start of Debugger)	udbg.exe /deb=tcp:+13001:nowait or udbg.exe /deb=tcp:+13001:wait

TCP:+13001 used in the table above can be qualified by the machine name and can be a different unused port setting, for example:

TCP:mymachine.compuware.com+13001 or
TCP:mymachine+13006

Debugging Commands

Go Shift+Ctrl+G	Run without debugging. To switch control back to the Debugger, choose Debug->Halt
Step Into Shift+Ctrl+S	Execute line with debugging
Step Over Shift+Ctrl+J	Execute line without debugging and go to next line
Step Out Shift+Ctrl+Q	Execute code in current component and stop at next statement in previous component
Run to cursor	Run and stop at the next break- point or current cursor position
Skip	Do not execute line and go to next line
Halt	Revive the Debugger
Toggle Breakpoint Shift+Ctrl+B	Mark or release a breakpoint
Clear All Breakpoints	Remove all breakpoints for this session.
Edit Breakpoints	Remove/add breakpoints

Debugging Remote and Web services

Set up these options via the 'urouter.asn' assignment file, according to the following table:

UServer process	Uniface Debugger
userver.exe /deb (Enables UServer to be 'debuggable')	udbg.exe /deb= <i>ProcessID</i> <i>ProcessID</i> from Task Manager
userver.exe /deb=tcp:+13001:nowait (Starts immediately, does not wait for Debugger)	udbg.exe /deb=tcp:+13001:create (Halts UServer)
userver.exe /deb=tcp:+13001:wait (Waits for Debugger)	udbg.exe /deb=tcp:+13001:create (Halts UServer)
userver.exe /deb=tcp:+13001:create (Waits for Debugger)	udbg.exe /deb=tcp:+13001:nowait or udbg.exe /deb=tcp:+13001: wait

Debugging the Uniface Server (UServer) on a remote server is achieved by connecting over a TCP connection from a remote machine.

Debugging the UServer process includes the additional option to connect to an existing running process using the system process ID that can be determined via the Microsoft Task Manager. However, this requires you to be logged on to the machine with the same user ID that starts or started the UServer process. Connecting with a different user ID will cause the connection to fail.

Common Field Validation Errors

0105	Not allowed to change primary key field
0119	Illegal ValRep value
0120 to 139	Incorrect subfields, format, or syntax
0140	Validation failed for field
0150	Requested number of "&" and " " operators not supported

Component Activation Errors

- 50 Signature descriptor for the current component not found
- 51 The identifier of the compiled component does not match the identifier in the signature descriptor
- 52 Protocol error (wrong sequence of operations)
- 53 An error occurred while copying the occurrences of an entity parameter to occurrences of the activated operation
- 54 An error occurred while copying occurrences of the entity parameter in the activated operation to occurrences of the component instance
- 55 or -56 Parameter error
- 57 The named instance cannot be found in the component pool
- 58 The named component cannot be found
- 59 No definition found for operation
- 60 An attempt was made by an instance other than the current modal form instance to start an operation other than the EXEC operation
- 61 The entity specified as an entity parameter is a dummy entity
- 62 The entity specified as an entity parameter must be the same entity as that specified in the operation
- 104 The path to the remote component is not specified in the Name Server assignment file
- 154 An instance with this name already exists
- 155 An error occurred while creating an instance
- 164 The instance is in the process of being deleted
- 1105 The instance name provided is not valid
- 1120 The operation name provided is not valid
- 1122 Wrong number of arguments
- 1123 Wrong number of parameters
- 1411 `activate` was performed on a modal form that is already in edit mode and that has an empty Execute trigger (an implicit `edit`)

IDE compile options

Compiling your application using the option `/nodebug` renders the application 'un-debuggable'. The debugger will not show any Proc code or watch data.

\$IOPRINT Settings

Set either in the application assignment file with `$ioprint` or within the Debugger Settings option, causes Uniface to put information in the message frame or, alternatively, in a file identified by the `$PUTMESS_LOGFILE` setting, according to the number supplied. To obtain multiple options, add together the option numbers according to the following table:

<code>\$ioprint 1</code>	Store sequence messages
<code>\$ioprint 2</code>	Connector function calls
<code>\$ioprint 4</code>	Return values from Fetch and Select connector function call
<code>\$ioprint 8</code>	Description block from Open connector function call
<code>\$ioprint 16</code>	Descriptions from <code>where</code> and <code>order by</code> clauses
<code>\$ioprint 32</code>	Generated SQL (if available)
<code>\$ioprint 64</code>	All system calls Uniface sends to operating system
<code>\$ioprint 128</code>	All calls to UOBJ plus the data sent

<code>\$ioprint 256</code>	All calls to a descriptor in a URR, ULANA, or USYSANA file
<code>\$ioprint 512</code>	All activates
<code>\$ioprint 1024</code>	Network messages
<code>\$ioprint 2048</code>	PNG image loader information

Extended Trace (XTrace)

Alternate Proc tracing option that, when enabled via the following settings within the assignment file, causes Uniface to write to the log file every Proc statement in order of execution. In addition, if the optional `$PROC_TRACING_ADDITION` setting is used, then additional information can be prefixed to each Proc statement.

```
$PROC_TRACING=TRUE
$PROC_TRACING_ADDITION="Status:%%$status"
$PROC_LOGFILE=C:\BNT\upda\work\xtrace_local.log
```

Proc Profiling

Used to obtain statistics and timing metrics on Proc execution; profiling logs each Proc statement executed together with execution timings in milliseconds (1/1000 of a second). In addition, timing on the execution of each module/trigger is also recorded.

The columns of data can be separated by one of four separators, which facilitates the use of other software, such as Microsoft Excel, to create custom timing or performance reports.

```
$PROC_PROFILING=TRUE
$PROC_PROFILING_SEPARATOR=comma
$PROC_LOGFILE=C:\BNT\upda\work\Profile_local.log
```
