Bridge 7 Migration Guide

In order to migrate to Bridge 7, find below a summary of all migration issues you need to keep in mind.

Software Versions

MagicDraw Version

E2E BUILDER 7 is designed for MagicDraw UML 18.0. If you need to migrate BUILDER projects from older MagicDraw versions, refer to Migrating Builder Projects to MagicDraw 18.

E2E Software

BRIDGE 7 contains many new features and improvements that lead to dependencies between Builder, Analyzer and Bridge versions. Please make sure that you update all components to version 7. You cannot e.g. deploy services that have been compiled with BRIDGE 7 to an older E2E Bridge.

Update all software components to BRIDGE 7, BRIDGE first, then BUILDER and ANALYZER.

Apache

With BRIDGE 7 we updated Apache to version 2.4. Refer to the Apache pages for more information on behavior changes. If you have changed the proxy templates of your BRIDGE installation, you will need to merge your modifications with the new templates provided with BRIDGE 7.

If you have modified the proxy templates as described on Configuring the Proxy, your proxy template modifications will be kept after a BRIDGE update. The system templates (folder <your Bridge data directory>/proxies/**system_templates**), however, will be updated to the new Apache version. Your changes may not fulfill the needs of Apache 2.4, and you may need to re-apply your changes to the new templates. In this case, copy the newly provided templates from folder **system_templates** to folder **templates** and merge your changes to the new template.

Java

In order to run the BRIDGE installer, you need a specific Java version: since Bridge 7.2.0 use Java version **11** or newer.

BRIDGE License

To use BRIDGE 7, you need a new BRIDGE license - please contact our support for more information. The BRIDGE installer will check your Bridge license. When updating a BRIDGE to a new version (e.g. BRIDGE 6 to BRIDGE 7), you need to apply the new license **before** updating the system.

BRIDGE Certificates

BRIDGE 7 uses Tomcat 9 or newer. This means that the BRIDGE will no longer accept certificates using the insecure MD5 algorithm. **Before** updating your BRIDGE, you need to update the BRIDGE certificate, and the proxy certificate with new ones using e.g. SHA256.
How to do this is described on Managing the Bridge Certificate and Managing Bridge Proxy Server Certificates pp.

Service Repositories

Migrating to BRIDGE 7 does not mean you have to migrate all of your service at once:

> ✅ All old repositories of BRIDGE 5.1 and BRIDGE 6.0 that have been compiled with older compilers, will still run on BRIDGE 7.

However, services that have been compiled with BUILDER 7 can only be run on a BRIDGE 7.

BUILDER / Compiler

Migration Checklist
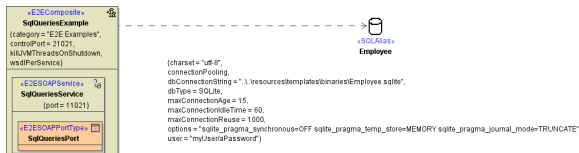
✅	ToDo	Documentation Link
	Check compatibility with Node.js 8 (before updating the BRIDGE)	
	Check certificates, if using secure algorithm	- Managing the Bridge Certificate - Managing Bridge Proxy Server Certificates
Updating the BRIDGE		

As already stated above, you do not need to migrate your services to BRIDGE/BUILDER 7 at once, but if you want to migrate a service, mind the following:

- Change your service regarding the items mentioned below:
    - Component Diagram
    - Local Variable Re-declaration
    - DateTime Parsing
- If a service uses libraries, you need to also compile and re-import these libraries - from bottom up. Otherwise the main service may not start or throw unexpected errors.

Old Component Diagrams

With introduction of BRIDGE 6 in 2013, we deprecated the old component and deployment diagram and replaced them by a new, simplified component diagram:

As of BRIDGE 7, we removed the support for the old component and deployment diagram from the E2E Compiler - only models with the new component diagram will compile. Before compiling such a model with E2E Compiler 7, you need to replace the old component and deployment diagrams by the simplified ones as described on The Components Wizard. If your service contains more than one deployment, please refer to Defining More Than One Deployment.

> ⓘ Other changes regarding the compiler are downward compatible. This means that your services will still compile and run, but you may get warnings regarding deprecated features or options, e.g. the <<FileSystemAlias>> has been replaced by a <<DirectoryAlias>> and a <<FileAlias>>. You can still use the old service implementation for the time being, but you should rework it it near future. The outdated features/options may be removed in future versions.

Local Variable Re-declaration

As of Builder 6.0.29 / 7.2.0, the Compiler reports an error, if a local variable is re-declared within Action Script. You will need to change your implementation, if the Compiler points you to an implementation like this.

Refer to Compiler Reports Error: Local variable is re-declared for more information.

DateTime Parsing

With Runtime 2018.2, we have rewritten the parsing of DateTime expressions. The old implementation silently ignored if a DateTime pattern did not match the given string and tried to parse it anyway. This lead to undefined results for not-matching strings.

With the new implementation, the parseDateTimeExpression() Operation expects that the given string conforms to the given format. If the string does not match with the given format, the xUML Runtime will raise an exception (FUDSM/9, see System Errors) and not parse the string. This is the correct behavior in this case.
If conformity to the format is not guaranteed, you can use the match() Operation on the string to check.

These Runtime changes will only be enabled for services that have been compiled with Builder 7.0.0-beta2 or newer. In this case, you need at least to check the logs for FUDSM/9 when testing the service. We recommend to search your model for DateTime parsing and implement a match check with subsequent error handling.

Node.js

With BRIDGE 7, we updated the integrated Node.js to version 8 (Carbon, 8.9.1). If you are running own Node.js services, please make sure that they are Node.js 8 compatible.

REST

	Order a new BRIDGE license and apply it	Request a new BRIDGE license
	Install Java 11	
	Merge proxy template modifications with new templates	Configuring the Proxy
	Run the BRIDGE installer	BRIDGE Installation Guide
Migrating the xUML Services		
	Install the BUILDER	
	Migrate BUILDER projects to MagicDraw 18	Migrating Builder Projects to MagicDraw 18
	Compile all used libraries from bottom up	
	Replace old component diagram	The Components Wizard
	Check for date time parsing issues	DateTime Parsing
	Check, if REST needs to output pretty JSON	• REST Service Reference • REST Adapter Reference

With BRIDGE 7, we introduced a new option **jsonCompact** on the <<RESTAlias>> and <<REST service>> components (see REST Service Reference and REST Adapter Reference). With this option, you can control the generation of JSON: pretty or compact.

If you compile a service using REST with a BRIDGE 7 compiler, this option defaults to **true** (= compact JSON). This is in contrast to BRIDGE 6, where pretty JSON was generated in absence of this option. Set this option to **false**, if you need pretty JSON to be generated.