

TBL Methods to Deploy Projects to Servers

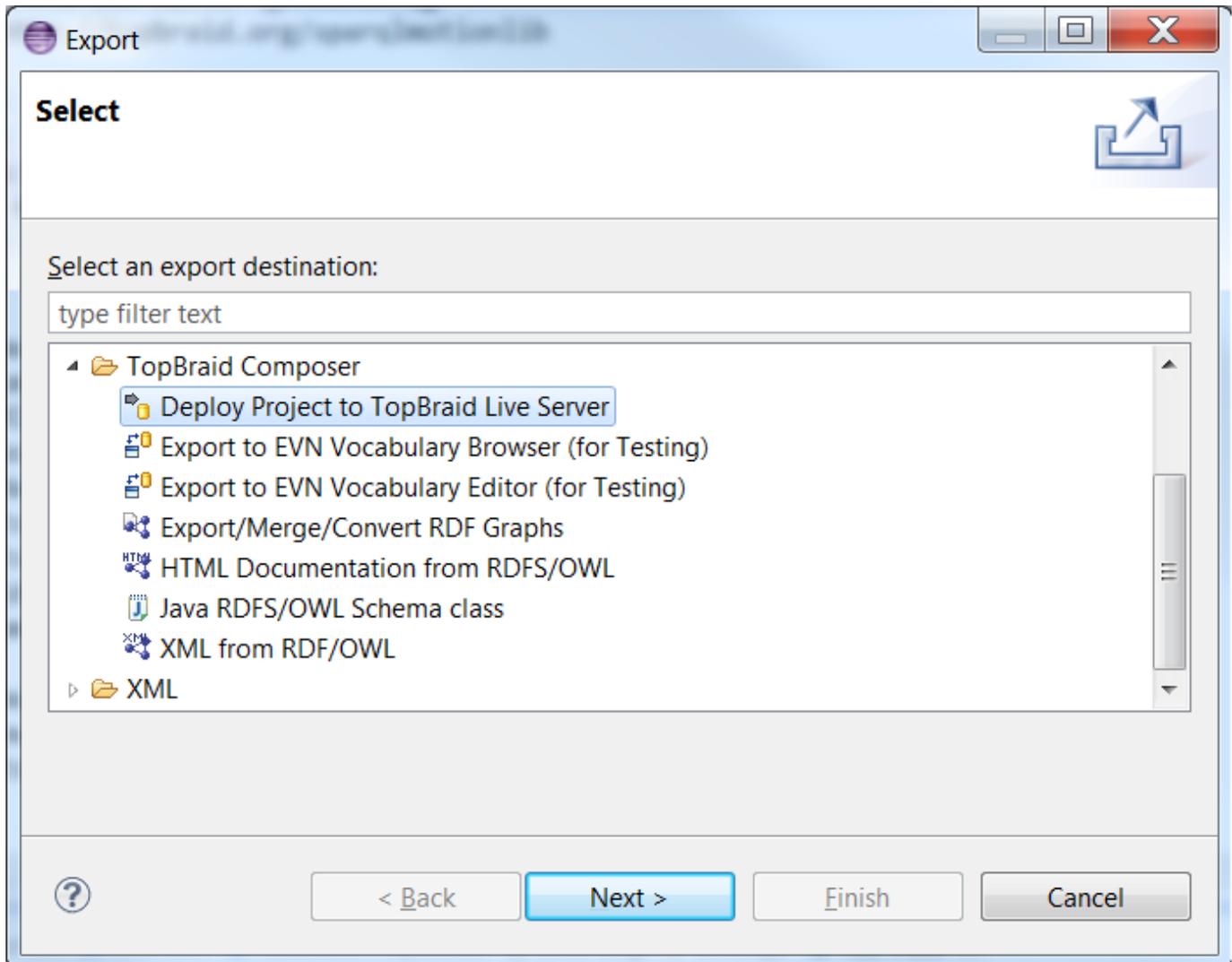
Page Contents

- 1 TopBraid Composer Export
- 2 Upload Project to Server
- 3 Send Projects between Servers
- 4 Appendix: Web Services For Deployment

TopBraid projects are workspace directories that contain multiple resources, such as asset collections, teamwork settings and working copies, and various configurations, etc. This section describes three methods for deploying projects to a TopBraid Suite server (i.e., EDG, EVN, or TBL) from three different project sources: (1) a TopBraid Composer IDE - Maestro Edition (TBC-ME), (2) a .zip file of a project (directory), or (3) another TopBraid server.

TopBraid Composer Export

Developers can design and test semantic content and web applications using TBC-ME, and then deploy those projects to a TopBraid (Live) server (i.e., EDG, EVN, TBL). TBC-ME has a custom **Export** wizard that deploys projects from its workspace into an existing TopBraid server. In TBC-ME, select a project (in a Project Explorer or Navigator view), and then either right-click the project or use **File** and select **Export... > TopBraid Composer > Deploy Project to TopBraid Live Server**.



Click next in this wizard:

Deploy Project to TopBraid Live Server

Server URL:

User name:

Password:

Overwrite existing project with the same name

Send necessary connection credentials

? < Back Next > Finish Cancel

Check the *Send necessary connection credentials* checkbox (to include the database credentials of the RDBMS connection in secured storage). Clicking **Finish** will zip up the entire selected project with all the RDBMS connection files, and other project files, and send it to the server specified in the 'Server URL'. And it will unzip it at the destination server in the workspace directory.

NOTE: This should only be used with projects that contain TTL files. If used with a repositories project checking the *Overwrite existing project with the same name* box will clear the database of triples before uploading. Please zip the project and use the Project Upload feature inside TBL instead.

Upload Project to Server

Users can click the 'Project Upload' link from the admin page.

First create a zip file for the project folder (including the .project file) in your system. And click the 'Choose file' button to select such zip file. It should upload the zip file into the server, and unzip it properly in its workspace directory. This method will not create any database credentials in the server.

Send Projects between Servers

TBL administrators can send projects from one TBL server to another one, e.g., from a test server to a production server. From the Admin page, select **Server Administration > Send Projects to Another Server**.

Select the list of projects to send, and also check the box whether you want to send the database triples also to the destination server. It essentially zip up the entire project from the source server and unzip it at the destination server workspace. And also create the appropriate triples for all the graphs involved at the destination server. This essentially will eliminate the steps in copying database information from one test server to the production server. And all the imported vocabulary are sent properly also. This functionality should create the vocabulary and its import properly at the destination server.

Appendix: Web Services For Deployment

We also provide REST web service to perform HTTP requests for the following tasks if you have such needs to do so. You can perform the followings using simple HTTP requests, this can help with initially setting up a server.

The following tables show the exact HTTP requests and parameters required to perform the specified tasks. The **bold** words are required system keyword, whereas the *italics* words (e.g. *xxx*) should be replaced with your specific details.

- **To create a new TBL Project**, use the `createProjectService` to create a new vocab by specifying a new vocab name and new vocab name space.

Send this http request to the destination server

Http request	GET request
Path	tbl/tbl/swp
Parameters:	
_viewClass	http://topbraid.org/teamwork#CreateProjectService
_snippet	true
projectType	http://tbl.topbraidlive.org/tblprojects#ProjectType
owlImports	http://www.w3.org/2004/02/skos/core
spinImports	http://topbraid.org/spin/tblconstraints
name	<i>\$(newVocabName)</i>
defaultNamespace	<i>\$(newVocabNameSpace)</i>

- **To load rdf file into the vocab** Use `importRDFFileService` to load a .ttl file into an existing vocab.

send this http request to the destination server

Http request	POST request
Path	tbl/tbl/swp
Parameters:(use Multipart-form data)	
_viewClass	http://topbraid.org/teamwork#ImportRDFFileService
_base	urn:x-evt-master:\$(vocabName)
projectGraph	urn:x-evt-master:\$(vocabName)
format	turtle
record	true
file	<i>\$(RDFFileName)</i> (should be the absolute path to that ttl file e.g. c:\Users\user1\Downloads\DBPediaCountryCapitals.sms.ttl, mime-type is application/octet-stream)

- **To deploy a project** Use `unzipProject` service to deploy a project (in .zip file) to a destination server.

create a zip file of the directory of the project first, and send this http request to the destination server

Http request	POST request
Path	tbl/tbl/unzipProject
Parameters: (use Multipart-form data)	
old_project	update
projectName	<i>\$(ProjectName)</i>
project	<i>\$(ProjectName.zip)</i> (the absolute path to that zip file location.

- **To delete a project** Use `deleteProject` service to delete a project in the server.

Http request	POST request
--------------	--------------

Path	tbl/tbl/deleteProject
Parameters:	
appName	<i>\${project name}</i>
delName	project

- **To send triples for a baseURI** Use `sendTriples` service to send triples for a baseURI from the source server to the destination server

send this http request to the source server, this will send triples from the source graph store to the destination server

Http request	POST request
Path	tbl/tbl/sendTriples
Parameters: (use Multipart-form data)	
graphURI	<i>\${baseURI}</i>
serverURL	<i>\${destURL}</i> (e.g. <code>http://black.tqinc.info:7080/tbl/tbl/</code>)
userName	<i>\${destUserName}</i>
password	<i>\${destPassword}</i>