



**Scaling Digital Twin Experiences 402**  
**Extending Configurations within the Experience Server**

**Copyright © 2021 PTC Inc. and/or Its Subsidiary Companies. All Rights Reserved.**

User and training guides and related documentation from PTC Inc. and its subsidiary companies (collectively "PTC") are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed software user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the PTC copyright notice and any other proprietary notice provided by PTC. Training materials may not be copied without the express written consent of PTC. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes. Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC.

UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.

PTC regards software piracy as the crime it is, and we view offenders accordingly. We do not tolerate the piracy of PTC software products, and we pursue (both civilly and criminally) those who do so using all legal means available, including public and private surveillance resources. As part of these efforts, PTC uses data monitoring and scouring technologies to obtain and transmit data on users of illegal copies of our software. This data collection is not performed on users of legally licensed software from PTC and its authorized distributors. If you are using an illegal copy of our software and do not consent to the collection and transmission of such data (including to the United States), cease using the illegal version, and contact PTC to obtain a legally licensed copy.

**Important Copyright, Trademark, Patent, and Licensing Information:** See the About Box, or copyright notice, of your PTC software.

**UNITED STATES GOVERNMENT RIGHTS**

PTC software products and software documentation are “commercial items” as that term is defined at 48 C.F.

1. 2.101. Pursuant to Federal Acquisition Regulation (FAR) 12.212 (a)-(b) (Computer Software) (MAY 2014) for civilian agencies or the Defense Federal Acquisition Regulation Supplement (DFARS) at 227.7202-1(a) (Policy) and 227.7202-3 (a) (Rights in commercial computer software or commercial computer software documentation) (FEB 2014) for the Department of Defense, PTC software products and software documentation are provided to the U.S. Government under the PTC commercial license agreement. Use, duplication or disclosure by the U.S. Government is subject solely to the terms and conditions set forth in the applicable PTC software license agreement.

PTC Inc., 121 Seaport Blvd, Boston, MA 02210 USA

**Introduction**

In our previous example (401), we took our two configurations of our quadcopter, and we placed the 3d model content in the Experience Server Content Deliver Service (CDS).

In this example, we will take a leap forwards and look into how you can leverage the CDS for content, the IRS for helping to manage the configurations of content i.e. which combination of assets go together, and bring these into an experience which actually has very little idea of what it is viewing – it could be a quadcopter or a bicycle – it simply receives a description of the configured item and it creates the appropriate AR view.

Firstly, to help the user identify the different models we are going to use an Image Target. Let’s imagine our model quadcopter is delivered in a nice presentation box, and the box has a photo of the model on the front. A prospective customer, wishing to see what the model looks like, can scan the box and will see an Augmented 3d model appear.

Our cover photos will be used to create Image targets which are used to attach the 3D augmented model to the presentation box – the 3d model will appear to be sitting on top of the box.

<picture here showing the basic layout – static content in CDS, experience in ES, thingmarks calling out specific configs>

**301.1 Creating the content package**

Following on from the previous example, we will start first by preparing our content.

1. Make a new directory called “**sdte402**”
2. Inside here create the following folders
   1. “models”
   2. “images”
   3. “targets”
   4. “WEB-INF”
3. The “models” and “WEB-INF” folders are identical to the previous example – you can copy the files into this new folder structure.
4. Edit the metadata.json file, and amend the version number and description e.g.

{

"version": "1.0.1",

"title": {

"en": "Quadcopters"

},

"description": {

"en": "Models and targets of various quadcopters for exercise 400"

}

}

1. In the “images” folder, do this
2. In the “targets” folder,
3. Your folder structure should look like this :-

**+ images**

**+ models**

**+ targets**

**+ WEB-INF**

1. As described in example 401, create a zip file that contains the 4 directories and contents. The zipfile should retain the same name as before e.g. **sdte400.zip**
   1. **Uploading content to the CDS**
2. In example 401, we created a representation and named It sdte400. We now need to update the representation with the new assets. For this operation, we ill perform an update (PUT) operation againt the CDS webservice. With the zip file you created in step .7 above, run the following commandline

curl -X PUT -u %uname%:%passwd% -k -H "X-Requested-With: XMLHttpRequest" -F "File=@**sdte400.zip**" -H "Content-Type:multipart/form-data" %server%/ExperienceService/content/reps/sdte400

A helper batch script “create\_cds\_rep.bat” has been provided; this takes the zipfile name as a parameter.

> update\_cds\_rep sdte400.zip

1. Your repository should have PVZ and JSON files for both quadcopter models when complete. To check, you can run the following command

curl -u %uname%:%passwd% -k %server%/ExperienceService/content/reps

A helper batch script “list\_reps.bat” file is provided.

> list\_cds\_reps

You should see something like this as the response

{"totalCount":1,"items":[{"name":"sdte400","createdby": "YOU","createdon":"DATE", "modifiedby":"YOU”,"modifiedon":"DATE",**"url":"https://YOURSERVER/ExperienceService/content/reps/sdte400"**,"metadata":{"version":"1.0.1","title":{"en":"Quadcopters"}, "description":{"en":" Models and targets of various quadcopters for exercise 400"}}}]}

Note that the version number and description will have been updated to reflect the contents of the new version of metadata.json

1. Make a note of the url property – this is the location that our new representation is stored. We will use this in our experience.

"url":"https://YOURSERVER/**ExperienceService/content/reps/sdte400**"

**302.3 Update Your Vuforia Studio Experience**

Now that your configuration data has been stored inside ThingWorx, your Vuforia Studio experience needs to be edited to accept these changes.

1. Open ScalingDigitalTwinExperiences402 in Vuforia Studio.
2. Add two new app parameters guide and target.
3. Add a 3dimage, name it photo.
4. Bind app parameter guide to the photo ‘resource’
5. Click to add a filter, in here enter the following

return '/ExperienceService/content/reps/sdte400/targets/' + value;

1. Delete the spatial target
2. Next, create a tml widget. Name it dynamicImageTarget

The tml widget allows the programmer to create content at the DOM level.

In this example, we are going to create an image tracking target

1. Click on the ‘edit’ button top open the TML widget edit panel. In this panel, enter the following:-

<twx-dt-target

Id = "imagetarget"

guide-src = "{{app.params.guidesrc}}"

src = "{{app.params.imagesrc}}"

x = "0" y = "0" z = "0"

rx = "-90" ry = "0" rz = "0"

istracked = "false" >

</twx-dt-target>

What is this doing? Well, we are declaring an AR tracking target; the database describing this target is src={{app.params.imagesrc}}, and the guide-view is {{app,params.guidesrc}}. The {{variable}} syntax says “replace the value of {{variable}} here. We will see where this variable comes from shortly.

1. Your Vuforia Studio experience has now been updated to its proper state. Click **Publish** to publish your updated experience.
2. A Vuforia Studio project with the new changes added in this section named ScalingDigitalTwinExperiences402 can be found in GitHub. **Note**: As with the last tutorial, this project is meant to be used as a reference material for the project file unless you have changed your template mapping.

**302.4 Further investigation**

Some important facts to note

1. The folder structure outlined in 302.2 - this was just one example of the type of content you can choose to store. The main pre-requisite is the WEB\_INF flder containing the medata.json file. You can create other folders to help manage your content, for example you might have images and perhaps some pdf documents that are linked to the model – create “images” and “documents” folders and place the content in these. Once complete, zip the complete structure up as shown, and upload to the CDS.
2. In this first example, we constructed the path to our final model using a simple parameter that is passed as the experience is launched. In the next tutprial we ill look at how we can use the IRS mapping service to take one launch parameter – the configuration ID – and return multiple assets directly from the CDS storage.