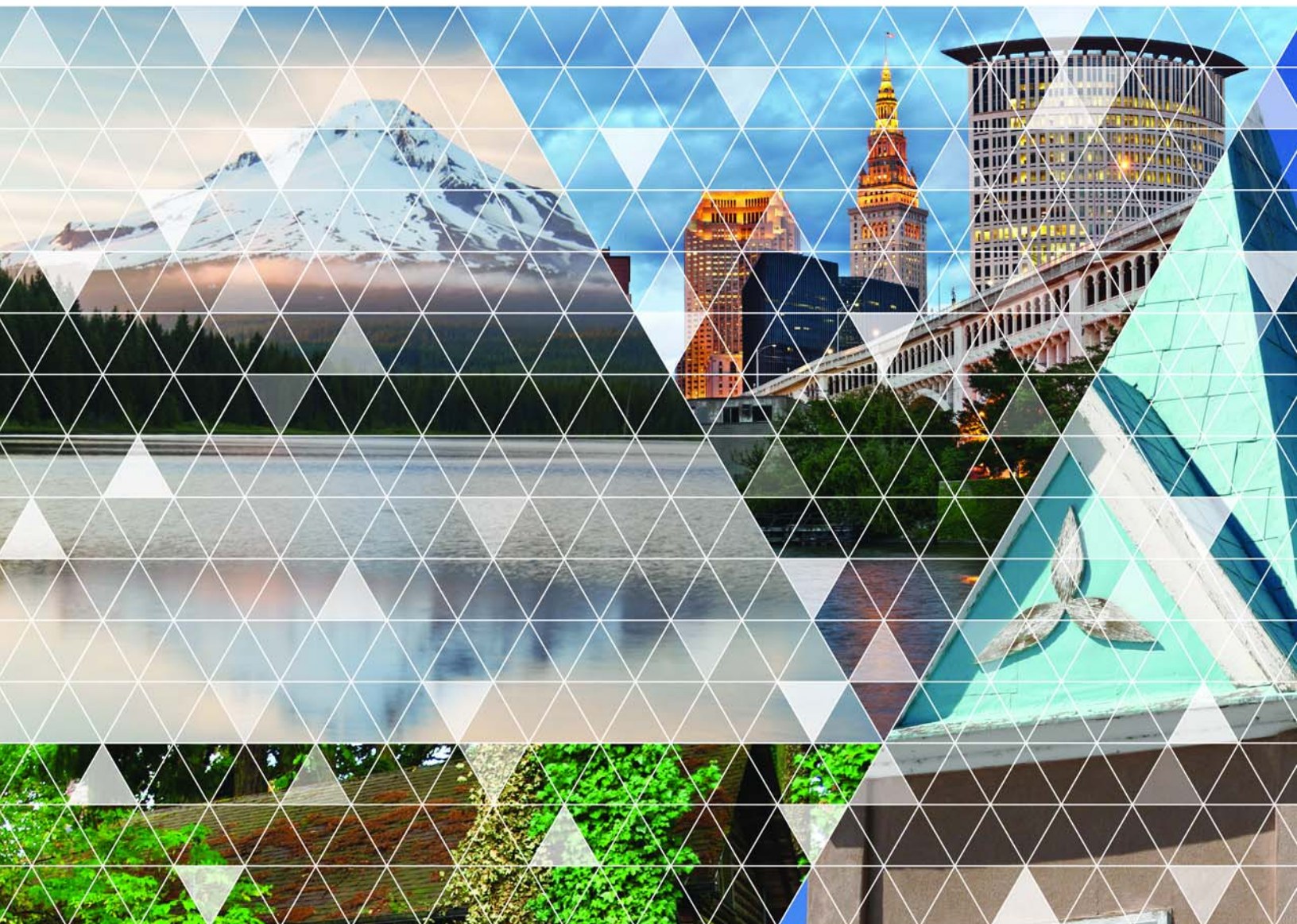


Accela Civic Platform®

Custom Portlets Development Guide



Accela Civic Platform Custom Portlets Development Guide

© 2017 Accela, Inc. All rights reserved.

Accela, tCivic Platform Accela logo, tCivic Platform Accela logo with Civic Platform “Government Software” notation, Accela Automation, Accela Asset Management, Accela Citizen Access, Accela Mobile Citizen Access, Accela ERS, Accela GIS, Accela IVR, Accela Land Management, Accela Licensing, Accela Mobile Office, Accela Public Civic Platform, tCivic Platform and Safety, Accela Service Request, Accela Wireless, Kiva DMS, Kiva Development Management System, 'PERMITS' Plus, SiteSync Civic Platform, Tidemark Advantage, Civic Platform, Civic Cloud, Civic Civic Platform, E-Boardroom, EnvisionConnect, Envista, GEOTMS, IQM2, Mediatraq, Minutetraq, PublicStuff, Trusted To Do More, Velocity Civic Platform, Vantage360, and tCivic Platform Accela logos, devices, product names, and service names are trademarks or service marks of Accela, Inc. Brava! Viewer is a trademark of Informative Graphics Corporation. Windows is a registered trademark of Microsoft Corporation. Acrobat is a trademark of Adobe Systems Incorporated. Portions copyright 2009 Civic Platforming-Lan 'digdog' Civic Platform and digdog software. All tCivic Platform company names, product names, and designs mentioned herein are trademarks of tCivic Platform or their respective owners.

Version 8.0.3.0.10

April 2017

Corporate Civic Platform Headquarters

2633 Camino Ramon
Suite 500
BisCivic Platform RancCivic Platform 3
San Ramon, CA 94583

Tel: (888) 722-2352

Fax: (925) 659-3201

www.accela.com

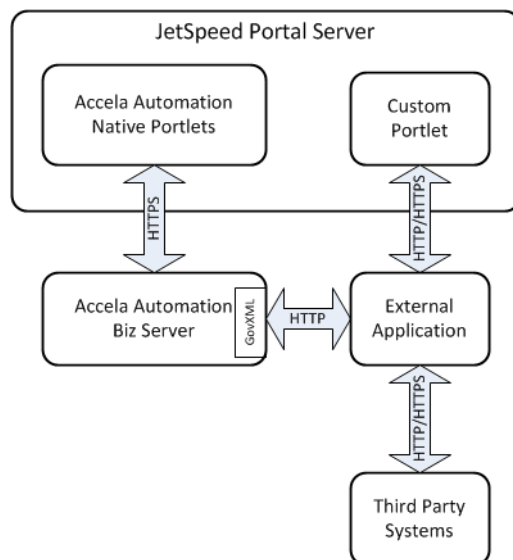
CHAPTER 1: INTRODUCTION

The information in this guide applies to both Civic Platform V360 and the new UI.

Custom portlets coexist with Civic Platform native portlets and run external applications. When you add a custom portlet to Civic Platform, you specify the URL to the external application you want to run inside your custom portlet.


External applications ([Figure 1: Custom Portlets Functional Diagram](#)) interface with the Civic Platform biz server or other third party systems. External applications that communicate with the Civic Platform biz server do so through GovXML and use HTTP only. External applications that communicate with third party systems can do so through any interface the third party system supports and can use HTTP or HTTPS.

Figure 1: Custom Portlets Functional Diagram



Civic Platform provides a default custom portlet, within which you can run your external application, or you can create your own custom portlet. You can add up to four instances of the default custom portlet to Civic Platform ([Figure 2: Sample Custom Portlet](#)) and run different external applications in each one.

Figure 2: Sample Custom Portlet



Tom Garrick
Inspector
ID: 10182
Department: Building
District: East Bay, CA

Today's Job

10=-00000-00002
InspReqOnly
Contact: Alfredo Castro
Tel: (510) 373-3863
Address: 1214 Oak St, Alameda, CA

11ACC-00000-00245
199 Final Structural
Contact: Evan Tisa
Tel: (510) 373-3853
Address: 400 Park St, Alameda, CA

Rejected List


11ACC-00000-00235
226 Meter Connection
Contact: Ben Bush
Tel: (510) 764-0296
Address: 2201 SOUTH Shore Cr, Alameda, CA

February Summary
Inspections: 5
Done: 4
To-do: 1

11ACC-00000-00231
190 Misc Structural
Contact: Warner Abbot
Tel: (510) 373-3802
Address: 920 Park St, Alameda, CA

11ACC-00000-0023W
199 Final Structural
Contact: Adela Baker
Tel: (510) 373-3833
Address: 2263 Santa Clara Ave, Alameda, CA

Map Satellite Hybrid



Map data ©2012 Google - [Terms of Use](#)

To create your own custom portlet from the beginning, you define portlet parameters in a .xreg file and deploy the .xreg file to the Civic Platform Jetspeed portal server. The Jetspeed portal server automatically loads all system and custom portlets, configured in files with the .xreg extension, during system startup. After system startup, Civic Platform displays available portlets in the Add Portlets dialog.

CHAPTER 2: DEVELOPMENT

Topics

- [Creating a Custom Portlet Configuration File](#)
- [Developing External Applications](#)

Creating a Custom Portlet Configuration File

To create your own custom portlet, you create a .xreg file and copy this file into the Civic Platform installation distribution. The .xreg file you create contains configuration information for your custom portlet.

Note: *Best practice is to create your own .xreg file for your custom portlet, and give it a unique name. Otherwise, if you only modify an existing .xreg file, your customizations can be overwritten during the next Civic Platform upgrade.*

Note: *The Civic Platform installation distribution provides default .xreg configuration files in the av.web\deploy\jetspeed.war\WEB-INF\conf\ folder. The accelaportlets.xreg file contains configuration information for most of the default Civic Platform portlets.*

To create and deploy a custom portlet configuration file

1. Create a new custom portlet file and save it with a .xreg extension.
2. Copy an existing portlet configuration, from one of the files in the av.web\deploy\jetspeed.war\WEB-INF\conf\ folder, into your custom portlet file.

Note: *An individual portlet configuration starts and ends with the <portlet entry> element.*

3. Modify the configuration information from the copied portlet to meet the needs of your custom portlet.

[Figure 1: Configuration information for the Default Custom Portlet](#) shows configuration information for the default custom portlet, excerpted from the accelaportlets.xreg file, and [Table 1: Description of Standard Configuration File Elements](#) describes standard elements used by the default custom portlet.

Figure 1: Configuration information for the Default Custom Portlet

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <registry>
3    </portlet-entry>
4    <portlet-entry name="Accela Custom Report" hidden="false" type="instance" parent=""
      application="false">
5      <security-ref parent="V360Security"/>
6      <meta-info>
7        <title>Accela Custom Report:en$$Accela Custom Report:zhAccela Custom Report:ar$$Accela
          Custom Report:de</title>
8        <description>Accela Custom Portlet</description>
9      </meta-info>
10     <classname>com.accela.av360.framework.jetspeed.portlet.AccelaIFramePortlet</classname>
11     <parameter name="source" value="/portlets/reports/reportShow.do?mode=show" hidden="true"
      cachedOnName="true" cachedOnValue="true">
12       <meta-info/>
13     </parameter>
14     <parameter name="height" value="1100" hidden="false" cachedOnName="true" cachedOnValue="true">
15       <meta-info/>
16     </parameter>
17     <parameter name="width" value="100%" hidden="false" cachedOnName="true" cachedOnValue="true">
18       <meta-info/>
19     </parameter>
20     <parameter name="frameborder" value="0" hidden="true" cachedOnName="true"
      cachedOnValue="true">
21       <meta-info/>
22     </parameter>
23     <parameter name="marginwidth" value="0" hidden="true" cachedOnName="true"
      cachedOnValue="true">
24       <meta-info/>
25     </parameter>
26     <parameter name="marginheight" value="0" hidden="true" cachedOnName="true"
      cachedOnValue="true">
27       <meta-info/>
28     </parameter>
29     <parameter name="name" value="Accela Custom Report" hidden="true" cachedOnName="true"
      cachedOnValue="true">
30       <meta-info/>
31     </parameter>
32     <parameter name="scrolling" value="no" hidden="true" cachedOnName="true" cachedOnValue="true">
33       <meta-info/>
34     </parameter>
35     <media-type ref="html"/>
36     <url cachedOnURL="false"/>
37     <category group="Jetspeed">V360</category>
38   </portlet-entry>
39 </registry>

```

Table 1: Description of Standard Configuration File Elements

Element	Parameters
portlet-entry	<p>Contains general portlet parameters.</p> <p>name: the portlet entry name.</p> <p>hidden: whether to display the portlet. The value of "true" hides the portlet. The value of "false" displays the portlet.</p> <p>type: the portlet entry type. Use "instance" to deploy the portlet into an instance of Civic Platform. This is the only applicable value for custom portlets.</p> <p>parent: the parent of the portlet to display. Leave empty if there is no parent.</p> <p>application: whether it is an application. The only allowed value is "false".</p>
security-ref	<p>A reference to the content in the accel-security.xreg file under the \conf directory.</p>

Table 1: Description of Standard Configuration File Elements

Element	Parameters
meta-info	<p>title: the name of the portlet to display in the Title column of the Add Portlets dialog. You can use different language codes to designate titles for different languages.</p> <p>description: the title of the portlet to display in the description column of the Add Portlets dialog.</p>
classname	The name of the Civic Platform java class that processes the portlet configuration information.
parameter names	<p>Contains parameters that control general portlet sizing and behavior.</p> <p>source: the “source” parameter which accesses source of the portlet to display</p> <p>height: the default height of the portlet to display</p> <p>width: the default width of the portlet to display.</p> <p>frameborder: the default size of the portlet frame to display</p> <p>marginwidth: the default width of the portlet margin to display</p> <p>marginheight: the default height of the portlet margin to display</p> <p>name: the name of the portlet to display</p> <p>scrolling: whether to display a scroll bar for the portlet</p>
media-type	Provides a reference to the media of the portlet to display. Keep the default HTML.
url	Determines whether the portlet supports cache on URL or not. Best practice is to set this to “true”.
category	Specifies the category of the portlet to display.

4. Save and backup your .xreg configuration file.
5. Deploy your custom portlet ([Chapter 1: Deployment on page 13](#)).

Developing External Applications

This section provides guidelines and a GovXML sample for external applications that run in a custom portlet.

Topics

- [Constructing URLs to External Applications](#)
- [Using Session Information](#)
- [Using GovXML for Standard Choice Requests \(Example\)](#)
- [Linking to Civic Platform Native Portlets](#)

Constructing URLs to External Applications

When you add a custom portal to Civic Platform, you specify the URL to the external application you want to run in the custom portlet ([Adding Custom Portlets to Your Console on page 14](#)). When a user accesses the custom portlet, Civic Platform constructs a URL, that includes session information, to the external application. For example, if the URL to an external application is:

```
https://webServer/Customportlet.jsp
```

Civic Platform might construct the following URL:

```
https://webServer/
Customportlet.jsp?ssoId=12345678&serviceProviderCode=Flag
staff&userId=admin&sessionId=711FD310DBF65593F178B3D4583C
2589.
```

The constructed URL can be shown as follows:

```
<my_URL>?ssoId<my_SSO_ID>&serviceProviderCode=<my_service
_provider_code>&userId=<my_user_ID>&sessionId=<my_session
_ID>
```

where:

my_URL	is the URL you specified when you added the custom portlet to Civic Platform.
my_SSO_ID	is the single sign on authentication ID for the Civic Platform session. This parameter corresponds to the <ApplicationState> element in GovXML.
my_service_provider_code	is the agency code of the current user. This parameter corresponds to the <ServiceProviderCode> element in GovXML.
my_user_ID	is the Civic Platform login account for the current user. This parameter corresponds to the <Username> element in GovXML.
my_session_ID	is the session information for the current user.

Using Session Information

Your external application can use GovXML to get session specific information that Civic Platform embeds in the URL ([Constructing URLs to External Applications on page 8](#)). Your external application can use this information to customize functionality exposed at the agency or user levels.

To get session information

```
String userId = request.getParameter("userId");
String serviceProviderCode =
request.getParameter("serviceProviderCode");
String ssoId = request.getParameter("ssoId");
String sessionId = request.getParameter("sessionId");
```

You can append this session information to specific GovXML API calls. For example:

To append session information to the getAvailableSessionKeys API

```
String sessionKeysUrl = new StringBuilder
("https://accela.sanramon.com:5443/portlets/
customPortletHelper.do?action=getAvailableSessionKeys");
httpUrl.append("acauth=").append(ssoId).("&userId=").append(
userId).append("&serviceProviderCode=").append(service
ProviderCode).append("&register=true");
```

To set up the HTTP client and execute the method

```
HttpClient client = new HttpClient();
GetMethod method = new
GetMethod(sessionKeysUrl.toString());
method.setRequestHeader("Cookie", "JSESSIONID=" +
sessionId);
method.setRequestHeader("User-Agent", "Mozilla/4.0 (
Acceal Automation )");
client.executeMethod(method);
```

To get the session keys

```
String sessionKeys = method.getResponseBodyAsString();
```

Here is the sample return result from the above script

```
["department","altID","addressID","providerNumber","asset
Type","trustAccountReceptID","capID","userId","assetSeq",
"setId","module","inspectionType","inspectionId","invoice
Id","parcelID","addressUID","transactionID","sourceNumber
","userGroup","assetGroup","ownerNumber","fullName","lang
uage","licenseSeqNo","contactSeqNumber"]
```

Using GovXML for Standard Choice Requests (Example)

The following examples show how to make a GovXML Standard Choice request and get a GovXML Standard Choice response.

To make a GovXML Standard Choice request

```
<?xml version="1.0" encoding="utf-8"?>
<GovXML xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.accela.com/schema/
AccelaOpenSystemInterfaceXML">
  <GetStandardChoices>
    <System>
      <XMLVersion>720</XMLVersion>
      <!--Get the service provider code with
request.getParameter("serviceProviderCode") -->
```

```

        <ServiceProviderCode>serviceProviderCode</
ServiceProviderCode>
<!--Get the user ID with request.getParameter("userId")
--!>
        <Username>userId</Username>
        <MaxRows>50</MaxRows>
        <StartRow>1</StartRow>
        <EndRow>0</EndRow>
        <TotalRows>0</TotalRows>
<!--Get the sso Id with request.getParameter("ssoId") --!>
        <ApplicationState>ssoId</ApplicationState>
        <Context>AccelaMobileOffice</Context>
        <LanguageID>en-US</LanguageID>
    </System>
    <standardChoiceType>CONDITIONS_OF_APPROVALS</
standardChoiceType>
</GetStandardChoices>
</GovXML>

```

To get a GovXML Standard Choice response

```

<?xml version="1.0" encoding="UTF-8"?>
<GovXML xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.accela.com/schema/
AccelaOpenSystemInterfaceXML">
    <GetStandardChoicesResponse>
        <System>
            <XMLVersion>720</XMLVersion>
            <ServiceProviderCode> serviceProviderCode </
ServiceProviderCode>
            <Username> userId </Username>
            <MaxRows>50</MaxRows>
            <StartRow>1</StartRow>
            <EndRow>1</EndRow>
            <TotalRows>1</TotalRows>
            <ApplicationState> ssoId </ApplicationState>
            <Context>AccelaMobileOffice</Context>
            <Error>
                <ErrorCode>0</ErrorCode>
            </Error>
            <LanguageID>en-US</LanguageID>
        </System>
    </GetStandardChoicesResponse>
</GovXML>

```

```

    <standardChoiceType>CONDITIONS_OF_APPROVALS</
standardChoiceType>
    <StandardChoices>
      <Enumeration>
        <Keys>
          <Key>yes</Key>
        </Keys>
        <IdentifierDisplay>yes</IdentifierDisplay>
      </Enumeration>
    </StandardChoices>
  </GetStandardChoicesResponse>
</GovXML>

```

Linking to Civic Platform Native Portlets

You can link to Civic Platform native portlets, or other custom portlets, from within a custom portlet. You can specify parameters in your links so that the linked to portlets show content according to the specified parameters.

Example Use Case

A custom portlet contains a list of licenses owned by a person. When a user clicks one of the license numbers, the license detail portlet opens with details for the selected license number.

Note: Access permissions for the linked-to portlet determines user access.

Table 2: Sample Links to Civic Platform Native Portlets provides sample links to data in the Record, Address, Parcel, Owner, Licensed Professional, Asset and Inspection portlets.

Table 2: Sample Links to Civic Platform Native Portlets

Data Type	Sample Link
Record Detail	\$server domain\$/portlets/cap/capsummary/ CapTabSummary.do?mode=tabSummary&serviceProviderCode=SACC O&ID1=12CAP&ID2=00000&ID3=000O4&requireNotice=YES&clearFor m=clearForm&module=Building&isFromCapList=false&isGeneralCAP= Y
Address Detail	\$server domain\$/portlets/address/ refAddressForm.do?value(mode)=view&refAddressId=1220&sourceNu mber=82&value(addressUID)=&SpecialFlag=no&isFromAddressList=fal se&module=Building
Parcel Detail	\$server domain\$/portlets/parcel/ refParcelForm.do?mode=viewForGIS&isFromParcelList=false&SpecialF lag=no&refParcelNumber=100001

Table 2: Sample Links to Civic Platform Native Portlets

Data Type	Sample Link
Owner Detail	<code>\$server domain\$/portlets/owner/ refOwnerForm.do?value(mode)=view&value(refOwnerNumber)=113215 9&value(sourceSeqNumber)=&value(ownerUID)=&isFromOwnerList=fal se&module=Building</code>
Contact Detail	<code>\$server domain\$/portlets/contact/ref/ refContactDetail.do?value(mode)=view&value(contactSeqNumber)=1&v alue(lookup)=false&isFromContactList=false&module=Building&autoFoc us=false</code>
Licensed Professional Detail	<code>\$server domain\$/portlets/professional/ref/ refProfessionalDetail.do?value(mode)=edit&value(licSeqNbr)=679544& module=Building&autoFocus=false</code>
Asset Detail	<code>\$server domain\$/portlets/ams/asset/ dataForm.do?mode=doShowAssetDetail&isRedirect=false&module=AM S&assetID=ROC&assetType=AAM&assetGroup=1</code>
Inspection Detail	<code>\$server domain\$/portlets/inspection/ inspectionDetail.do?mode=view&fromPage=inspectionDailyList&service ProviderCode=RENO&capID1=12CAP&capID2=00000&capID3=0000I& inspectionID=123905708&inspectionType=F769+Fire+Final&scheduled Date=02-07-2012&module=Building&autoFocus=false</code>

CHAPTER 3: DEPLOYMENT

Topics

- Deploying Custom Portlets
- Adding Custom Portlets to Your Console
- Opening Your Custom Portlet

Deploying Custom Portlets

To deploy your custom portlet

1. Copy your .xreg file to the `av.web\deploy\jetspeed.war\WEB-INF\conf\` folder in your Civic Platform installation distribution.
2. Restart the Civic Platform web server.

Your new portlet, with the name defined for the portlet-entry element in your configuration file, appears in the Add Portlets dialog.

Console Editor: Use the buttons and drop-down menus to customize your console.

Media Type : html

Edit Layout

Add Portlet

Add Reference

Add Page

Edit Properties

Page :

Category	All Portlets	Parent	All Parents	Filter
Add	Title	Description		
<input type="checkbox"/>	AA Classic	V360 AA Classic Portlet		
<input type="checkbox"/>	Accela Custom Portlet	Accela Custom Portlet		
<input type="checkbox"/>	Accela Custom Report	Accela Custom Portlet		
<input type="checkbox"/>	Accela News	Accela News		
<input type="checkbox"/>	Activities	V360 Activities Portlet		
<input type="checkbox"/>	Addresses	V360 Addresses Portlet		
<input type="checkbox"/>	Administration	V360 Administration Portlet		
<input type="checkbox"/>	Alerts	V360 Alerts Portlet		
<input type="checkbox"/>	ARW Reports	V360 ARW Reports Portlet		
<input type="checkbox"/>	Asset Condition Assessment	Asset Condition Assessment Portlet		
<input type="checkbox"/>	Assets	V360 Asset Management Portlet		
<input type="checkbox"/>	Batch Jobs	V360 Batch Jobs Portlet		
<input checked="" type="checkbox"/>	Building Permits	V360 Building Permits Portlet		
<input type="checkbox"/>	Bulletin Board	V360 Bulletin Board Portlet		
<input type="checkbox"/>	Calendar	V360 User Calendar Portlet		

Apply

Cancel

Next >>

A B C D E F G H I J K L M N O P Q R S T U V

Adding Custom Portlets to Your Console

You add custom portlets to Civic Platform the same way you add other Civic Platform native portlets.

To add a custom portlet

1. Click the **Console Options** link in the top right corner of Civic Platform.

Civic Platform displays the Console Editor with a list of existing main links.

2. Add a link to your custom portlet.

- If you want to add your custom portlet to an existing main link, click the name of the main link to which you want to associate your custom portlet.
- If you want to create a new main link for your custom portlet, click **Add Page**.

Civic Platform displays the main link details page.

The screenshot displays the 'Console Editor' interface. At the top, a blue header bar contains the text 'Console Editor: Use the buttons and drop-down menus to customize your console.' Below this, there are several buttons: 'Edit Layout', 'Add Portlet', 'Add Reference', 'Add Page', and 'Edit Properties'. The 'Media Type' is set to 'html'. The 'Page' field is empty. The 'Layout' dropdown menu is set to 'V360 One Column' and the 'Skin' dropdown menu is set to 'V360 Default Skin'. Below these dropdowns are 'Apply' and 'Cancel' buttons. At the bottom of the interface, there is a section titled 'Building Permits' with its own 'Apply' and 'Cancel' buttons.

3. Click **Add Portlet**.

Civic Platform displays a list of available portlets.

Console Editor: Use the buttons and drop-down menus to customize your console.

Media Type : html **Edit Layout** **Add Portlet** Add Reference Add Page **Edit Properties**

Page :

Category	All Portlets	Parent	All Parents	Filter
<input type="checkbox"/>	AA Classic		V360 AA Classic Portlet	
<input type="checkbox"/>	Accela Custom Portlet		Accela Custom Portlet	
<input type="checkbox"/>	Accela Custom Report		Accela Custom Portlet	
<input type="checkbox"/>	Accela News		Accela News	
<input type="checkbox"/>	Activities		V360 Activities Portlet	
<input type="checkbox"/>	Addresses		V360 Addresses Portlet	
<input type="checkbox"/>	Administration		V360 Administration Portlet	
<input type="checkbox"/>	Alerts		V360 Alerts Portlet	
<input type="checkbox"/>	ARW Reports		V360 ARW Reports Portlet	
<input type="checkbox"/>	Asset Condition Assessment		Asset Condition Assessment Portlet	
<input type="checkbox"/>	Assets		V360 Asset Management Portlet	
<input type="checkbox"/>	Batch Jobs		V360 Batch Jobs Portlet	
<input checked="" type="checkbox"/>	Building Permits		V360 Building Permits Portlet	
<input type="checkbox"/>	Bulletin Board		V360 Bulletin Board Portlet	
<input type="checkbox"/>	Calendar		V360 User Calendar Portlet	

Apply **Cancel** **Next >>**

A B C D E F G H I J K L M N O P Q R S T U V

4. Mark the **Add** check box next to your custom portlet and click **Apply**.

Civic Platform displays the main link details page.

Console Editor: Use the buttons and drop-down menus to customize your console.

Media Type : html **Edit Layout** **Add Portlet** Add Reference Add Page **Edit Properties**

Page :

Layout : V360 One Column Skin : V360 Default Skin

Apply **Cancel**

Accela Custom Portlet

Title* : My Custom Portlet

URL* : http://www.accela.com

Apply **Cancel**

5. Complete these fields:

- Title** The title of the custom portlet. It displays as the portlet name when users open the portlet.
- URL** The URL to the external application you want to run in the custom portlet. The URL must start with `http://` or `https://`.

6. Click **Apply** button to save the custom portlet settings.

Opening Your Custom Portlet

Click on the main link to which you added your custom portlet.

Civic Platform opens your custom portlet and runs the external application in it.

My Inspection Task

Tom Garrick
Inspector
ID: 10162
Department: Building
District: East Bay, CA

February Summary
Inspections: 5
Done: 4
To-do: 1

Today's Job

10-00000-00002
InspReqOnly
Contact: Alfredo Castro
Tel: (510) 373-3863
Address: 1214 Oak St, Alameda, CA

114CC-00000-00245
199 Final Structural
Contact: Evan Tisa
Tel: (510) 373-3853
Address: 400 Park St, Alameda, CA

Rejected List

114CC-00000-00235
226 Meter Connection
Contact: Ben Bush
Tel: (510) 764-0296
Address: 2201 SOUTH Shore Cr, Alameda, CA

114CC-00000-00231
190 Misc Structural
Contact: Warner Abbot
Tel: (510) 373-3802
Address: 920 Park St, Alameda, CA

114CC-00000-0023W
199 Final Structural
Contact: Adela Baker
Tel: (510) 373-3833
Address: 2263 Santa Clara Ave, Alameda, CA

Map | Satellite | Hybrid

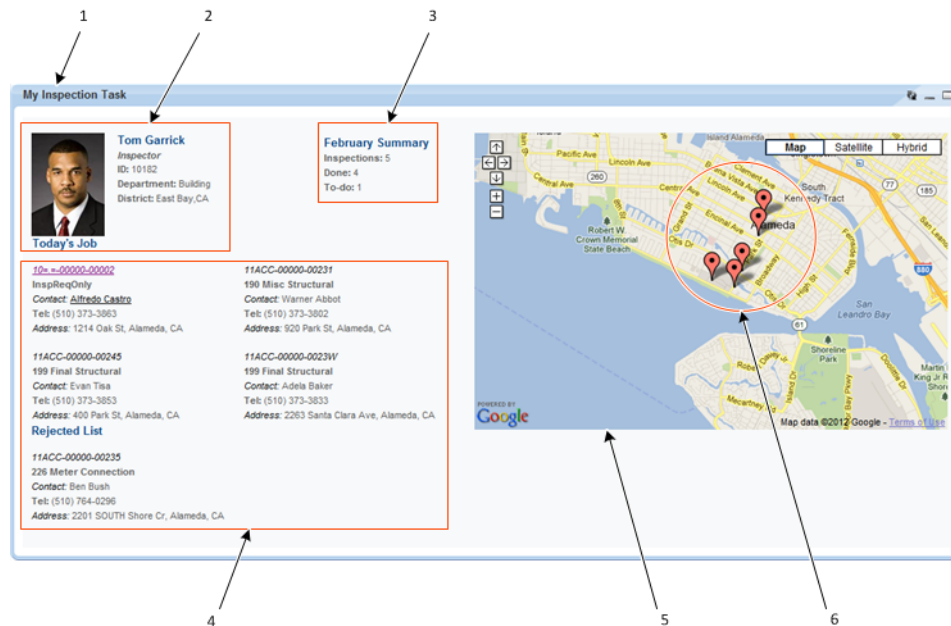
Map data ©2012 Google - Terms of Use

APPENDIX A

SAMPLE CUSTOM PORTLET

Accela provides a sample custom portlet that you can repurpose for your own use ([Figure 1: Sample Custom Portlet](#)). The sample portlet provides general inspector information, and provides summary, details, and geographic location information for inspections in the inspector's work queue. The summary information indicates the number of completed and remaining inspections, and the details provide a link to detailed inspection information for active inspections.

Figure 1: Sample Custom Portlet

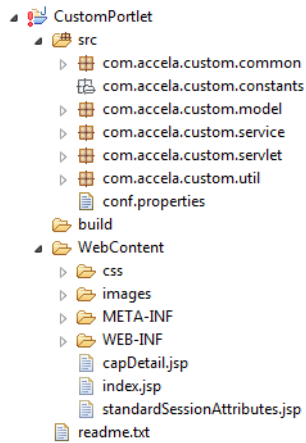


[Table 1: Custom Portlet Components](#) provides additional information on the components listed in [Figure 1: Sample Custom Portlet](#), and a cross-reference to the sections in the sample application code responsible for the functionality provided by those components.

[Figure 2: Project File for Sample Custom Portlet](#) shows the structure of the sample custom portlet project.

Table 1: Custom Portlet Components

Item	Function	Java Class/Function/Parameter
1	The custom portlet in which the external application runs	The title of the custom portlet specified when adding the custom portlet to the console (Adding Custom Portlets to Your Console on page 14).
2	The following information related to the current inspector: <ul style="list-style-type: none"> • personal photo • name • title • user ID • department • district 	The index.jsp file contains this information.
3	The following summary information related to the inspector's current work queue: <ul style="list-style-type: none"> • month of summary • total number of inspections • completed inspections • open inspections 	The index.jsp file contains this information.
4	The following detailed information related to the inspectors's monthly inspection assignments: <ul style="list-style-type: none"> • record ID of inspection. For open inspections, this is a live link to additional record details. • record type of inspection • contact name for inspection. For open inspections, this is a live link to contact information. • telephone number of contact • address of contact 	The index.jsp file associates with MyInspection.java and uses the GovXMLClient to get data for the first record from the Civic Platform server. The index.jsp file contains all other record information.
5	A map of the area containing inspections in the inspector's work queue.	The index.jsp file contains information for connecting to maps.google.com.
6	The location on the map of the inspections in the inspector's work queue.	The locations are all hard coded. Use Javascript to call the APIs in "maps.google.com" to add the balloons in the index.jsp.

Figure 2: Project File for Sample Custom Portlet

The `conf.properties` file ([Figure 3: Configuration File](#)) specifies connection information to Civic Platform and other operational parameters for the sample application as follows:

appServerUrl	Specifies the URL for GovXML client requests (requestGovXML). It is the standard URL for accessing GovXML Servlet in the custom application.
v360Url	Specifies the URL of the V360 for routing the GovXML requests
agencyCode	Agency code for V360 for routing the GovXML requests
moduleCode	Module code for V360 for routing the GovXML requests
sessionKeysUrl	Gets all available session keys in AA and returns JSON format string.
sessionValuesUrl	Gets all available session value in AA and returns JSON format string.

Note: The sample application includes script code that appends session identifiers to the URL for getting the session keys and session values.

Figure 3: Configuration File shows the configuration file used by the sample application.

Figure 3: Configuration File



```

conf.properties
BuildConfig...
R.java
RecordList.java
CAP.java
readme.txt
» 24

#http://{IP}:{port}/wireless/GovXMLServlet
appServerUrl=http://accela.sanramon.com:3080/wireless/GovXMLServlet

#https://{IP}:{port}
v360Url=https://accela.sanramon.com:5443

agencyCode=FLAGSTAFF

#AA module
moduleCode=Building

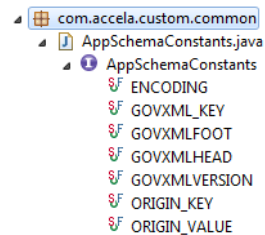
#https://{IP}:{port}/portlets/customPortletHelper.do?action=getAvailableSessionKeys
sessionKeysUrl=https://accela.sanramon.com:5443/portlets/customPortletHelper.do?action=getAvailableSessionKeys

#https://{IP}:{port}/portlets/customPortletHelper.do?action=getSessionValues
sessionValuesUrl=https://accela.sanramon.com:5443/portlets/customPortletHelper.do?action=getSessionValues

```

Figure 4: Constants shows the functions in that define constants for the sample application.

Figure 4: Constants



```

com.accela.custom.common
├── AppSchemaConstants.java
│   └── AppSchemaConstants
│       ├── ENCODING
│       ├── GOVXML_KEY
│       ├── GOVXMLFOOT
│       ├── GOVXMLHEAD
│       ├── GOVXMLVERSION
│       ├── ORIGIN_KEY
│       └── ORIGIN_VALUE

```

ENCODING	Defines the GovXML request encoding format
GOVXMLHEAD	Defines the GovXML request head and namespace
GOVXMLFOOT	Defines the GovXML request foot
GOVXMLVERSION	Defines the GovXML request version
GOVXML_KEY	Defines the GovXML request xml key
ORIGIN_KEY	Defines the GovXML request origin key
ORIGIN_VALUE	Defines the GovXML request origin key value

The CAP class ([Figure 5: CAP Class](#)) and CAPDetails class ([Figure 5: CAP Class](#)) store Civic Platform record data for the sample application.

Figure 5: CAP Class

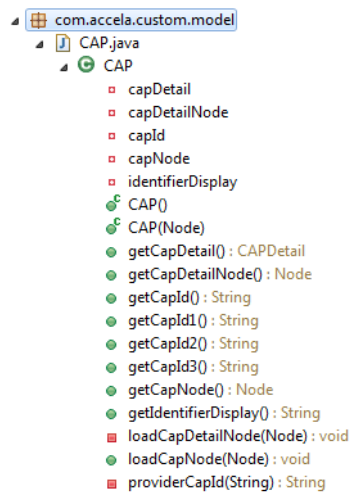
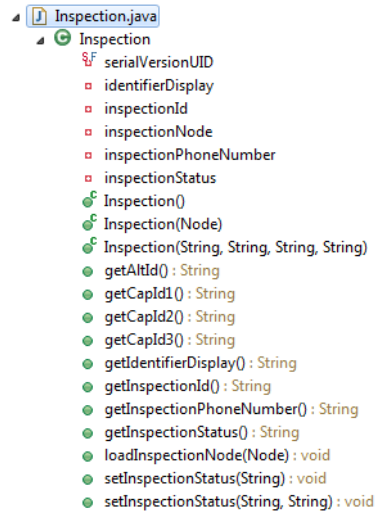


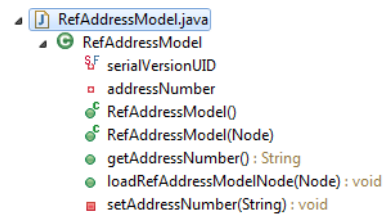
Figure 6: CAPDetail Class



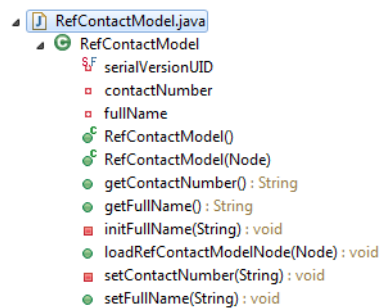
The Inspection class ([Figure 7: Inspection Class](#)) stores Civic Platform inspection data for the sample application.

Figure 7: Inspection Class

The RefAddressModel class ([Figure 8: RefAddressModel Class](#)) stores Civic Platform reference address data for the sample application.

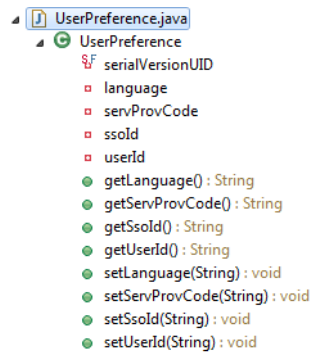
Figure 8: RefAddressModel Class

The RefContactModel class ([Figure 9: RefContactModel Class](#)) stores Civic Platform reference contact data for the sample application.

Figure 9: RefContactModel Class

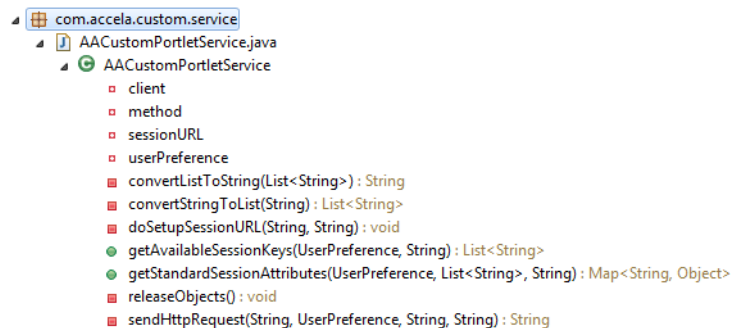
The UserPreference class ([Figure 10: UserPreference Class](#)) stores Civic Platform user preference data for the sample application.

Figure 10: UserPreference Class



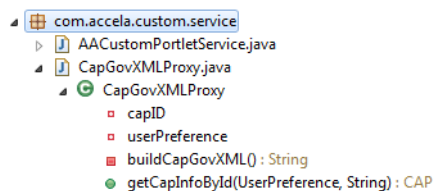
The AACustomPortletService class (Figure 11: AACustomPortletService Class) stores Civic Platform session values for the sample application.

Figure 11: AACustomPortletService Class



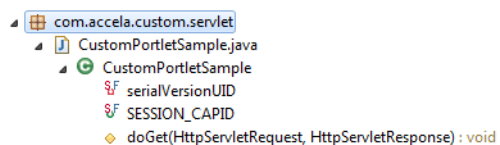
The CapGovXMLProxy class (Figure 12: CapGovXMLProxy Class) gets record data from the GovXML interface.

Figure 12: CapGovXMLProxy Class



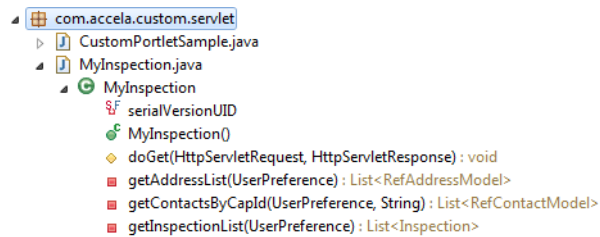
The CustomPortletSample class (Figure 13: CustomPortletSample Class) gets record data from the Civic Platform HTTP servlet.

Figure 13: CustomPortletSample Class



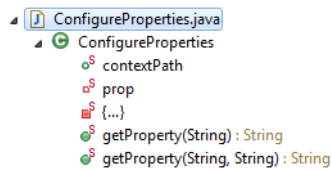
The MyInspection class ([Figure 14: MyInspection class](#)) gets inspection data from the Civic Platform HTTP servlet.

Figure 14: MyInspection class



The ConfigureProperties class ([Figure 15: ConfigureProperties class](#)) reads configuration property values from the configuration file ([Figure 3: Configuration File on page 20](#)), so the values can be used by other classes.

Figure 15: ConfigureProperties class



The custom utilities package ([Figure 16: Custom Utilities Package](#)) contains common utility class files for the sample application.

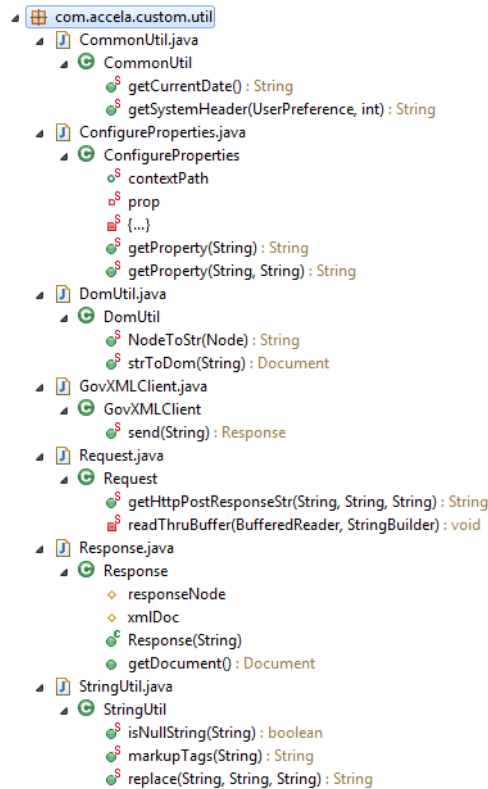
Figure 16: Custom Utilities Package**Figure 17: Servlet Configuration File** shows the servlet configuration file provided by the sample application.

Figure 17: Servlet Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.
  <display-name>CustomPortlet</display-name>

  <servlet>
    <description></description>
    <display-name>MyInspection</display-name>
    <servlet-name>MyInspection</servlet-name>
    <servlet-class>com.accela.custom.servlet.MyInspection</servlet-class>
  </servlet>
  <servlet>
    <description></description>
    <display-name>Custom Portlet Proxy</display-name>
    <servlet-name>customPortletSample</servlet-name>
    <servlet-class>com.accela.custom.servlet.CustomPortletSample</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>MyInspection</servlet-name>
    <url-pattern>/MyInspection</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>customPortletSample</servlet-name>
    <url-pattern>/customPortletSample</url-pattern>
  </servlet-mapping>

  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

This is standard servlet configuration. There are two servlets (MyInspection, and customPortletSample) in the custom portlet sample application. Below are the names and descriptions of the main elements.

servlet-name	Defines the servlet name.
servlet-class	Defines the servlet java class name.
url-pattern	Sets the url access pattern.