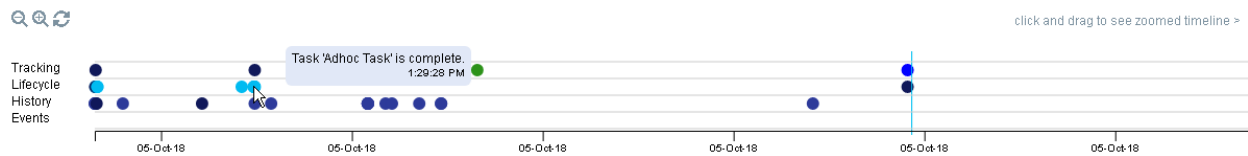




The Information Company™

AppWorks Extension

Case Event Timeline View



Introduction	3
Deployment	3
Configuration	3
About OpenText	6

Introduction

This extension provides a graphical timeline view of case events and related time-based entity information, that can be displayed in the Web Content panel of a layout. The Timeline view can display the following entity data in one or more 'channels', along with an indicator of the current date and time:

- History - all information from the History Building Block that is not shown in another channel
- Tracking - last modified date and deadlines
- Lifecycle - task actions and manual state transitions
- Discussion - topics and replies
- Custom Event - where there is a need to display related time-based entity data in a graphical view

Hovering over an event in the timeline shows a popup with additional details, and the view can be zoomed and scrolled in order to show events that are closely grouped together.

Because the extension runs inside a Web Content panel, it cannot directly interact with or respond to actions in the main AppWorks application. This means that if event information is updated by the user in some way the timeline cannot be auto-refreshed. To mitigate this limitation, the timeline view can be manually refreshed or configured to auto-refresh periodically.

Deployment

The extension is packaged in a web application archive that is best deployed to your AppWorks TomEE server in order to avoid any Cross-Origin Resource Security (CORS) issues. Please note that the WAR file may also contain other similar extensions.

To install it, download the 'extensions' WAR file from GitHub and unzip it into the TomEE web apps folder, for example:

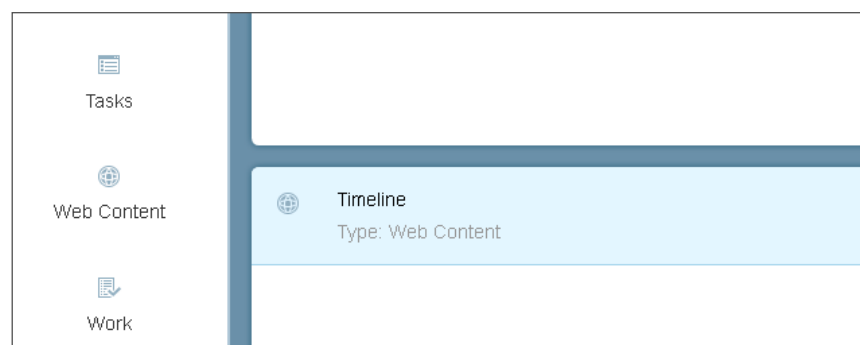
```
C:\Program Files\Apache Software Foundation\TomEE\webapps\extensions
```

Restart your TomEE server and check the deployment with this URL:

```
http://<tomee_server>:<port>/extensions/extensions.html
```

Configuration

The extension is accessed from an AppWorks Layout Building Block, by adding a Web Content panel.



In the Panel Properties, enter the URL to the extension, with the mandatory arguments for Base URL and Entity ID.

▼ Panel Properties

URL

http://nkproc16.otdemo.net:81/extensions/timeline.html?url={system.baseURL}&id={item.Identity.ItemId}

Ex: http://anyurl.com

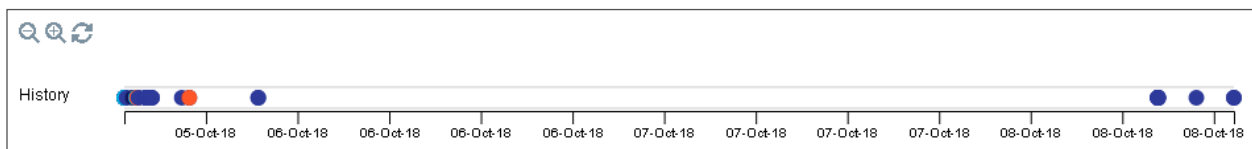
The syntax for the URL is as follows:

```
http://<tomee_server>:<port>/extensions/timeline.html?<one or more arguments>
```

The possible arguments are as follows (the first two are mandatory):

- url={system.baseURL} - passes the AppWorks URL for API service calls
- id={item.Identity.ItemId} - the ItemId of the entity
- include=<comma separated channels> - list of channels to show
- show_today - whether to display an indicator of the current date and time
- refresh=<interval in seconds> - how frequently to auto-refresh the data

With just the two mandatory arguments the timeline will only show events from the History Log, colour coded by type

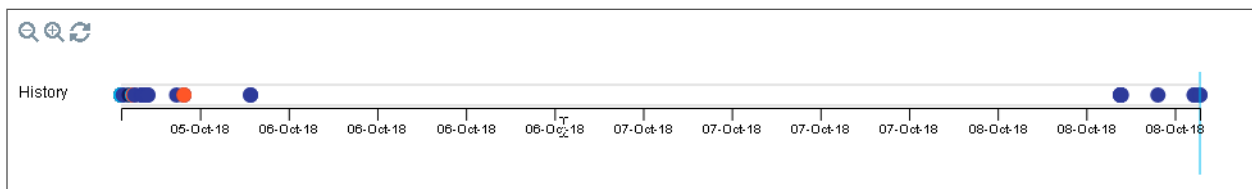


URL ...?url={system.baseURL}&id={item.Identity.ItemId}

Adding the refresh argument will cause this to automatically update (every 10 seconds in this example):

...?url={system.baseURL}&id={item.Identity.ItemId}&refresh=10

If you add the show_today argument then a line indicating the current date and time is included:

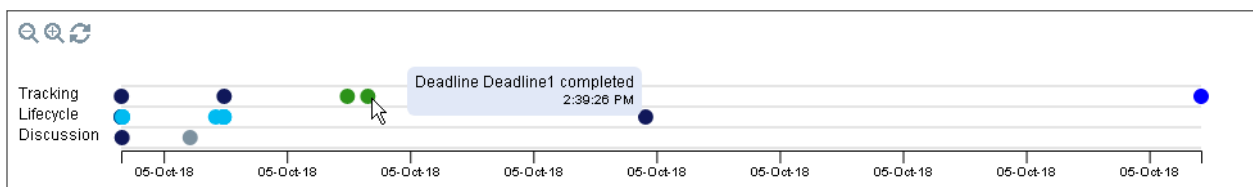


URL: ...?url={system.baseURL}&id={item.Identity.ItemId}&show_today

You can also break out specific sets of events into their own channel in the timeline by adding the `include` argument and specifying one or more channels to include. The possible channels are as follows:

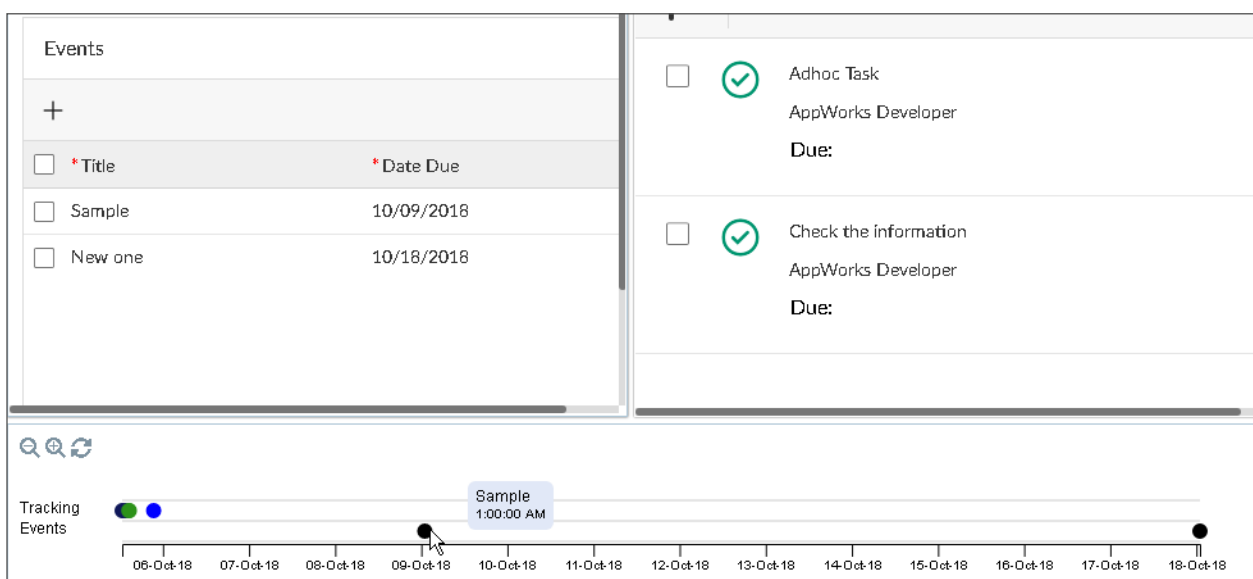
- `tracking` - shows deadline information and the last modified date
- `lifecycle` - shows manual Lifecycle events, such as task and user-driven state changes
- `discussion` - shows creation of topics and replies
- `history` - shows all events not included in any of the above channels
- `custom` - shows data from a related entity (requires additional settings defined below)

As an example, adding the tracking, lifecycle and discussion channels shows the following:



URL: `...&id={item.Identity.ItemId}&include=tracking,lifecycle,discussion`

Finally, when using the `custom` channel, you need to specify the related entity name, property to use as the date/time value, and a property to use as the display label in the timeline. As an example, when displaying the tracking channel and the `Title` & `DateDue` properties from the child entity `Events`, the display looks like this:



URL: `...&id={item.Identity.ItemId}&include=tracking,custom[Events:DateDue:Title]`

About OpenText

OpenText enables the digital world, creating a better way for organizations to work with information, on premises or in the cloud. For more information about OpenText (NASDAQ: OTEX, TSX: OTC) visit opentext.com.

Connect with us:

[OpenText CEO Mark Barrenechea's blog](#)

[Twitter](#) | [LinkedIn](#) | [Facebook](#)