Monitor example project

3 October 2019

Introduction

ExampleMonitor contains an extract of the Monitor application which is described below.

ExampleMonitor is a Visual Studio 10 solution which contains a web site called ExampleMonitor and a class library project called CascadesInterface2.

ExampleMonitor displays screens of applications, components and events logs. It has been modified to read from an XML file called Status.xml in the sub-folder Data. The path of the status file is specified in the **fileName** attribute in the CascadesInterface2 section in the web.config.

The CascadesInterface2 class library periodically reads the Status element from the file and applies the transforms to generate HTML. The HTML references cascading style sheet, Execute.css, in the NXAUTO sub-folder.

The Status.xml file is a snapshot of the status of applications, components and events that occurred in the components. An application contains one or more components. Components emit events when executing functions and when state changes occur. The StatusSchema.xsd is also supplied.

Transforms

The following xsl stylesheets are supplied:

- AbstractFormMainMenu.xsl
- AbstractFormOverview.xsl
- AbstractFormApplications.xsl
- AbstractFormComponents.xsl
- AbstractFormLogs.xsl
- Execute.xsl

Execute.xsl transforms the XML output of the AbstractForm stylesheets to HTML.

Configuration file

The **CascadesInterface2** section of the web.config file contains the transform names and paths in the **transforms** section. It does not contain a **services** section as described below because it reads from a file.

Figure 1 Maine menu

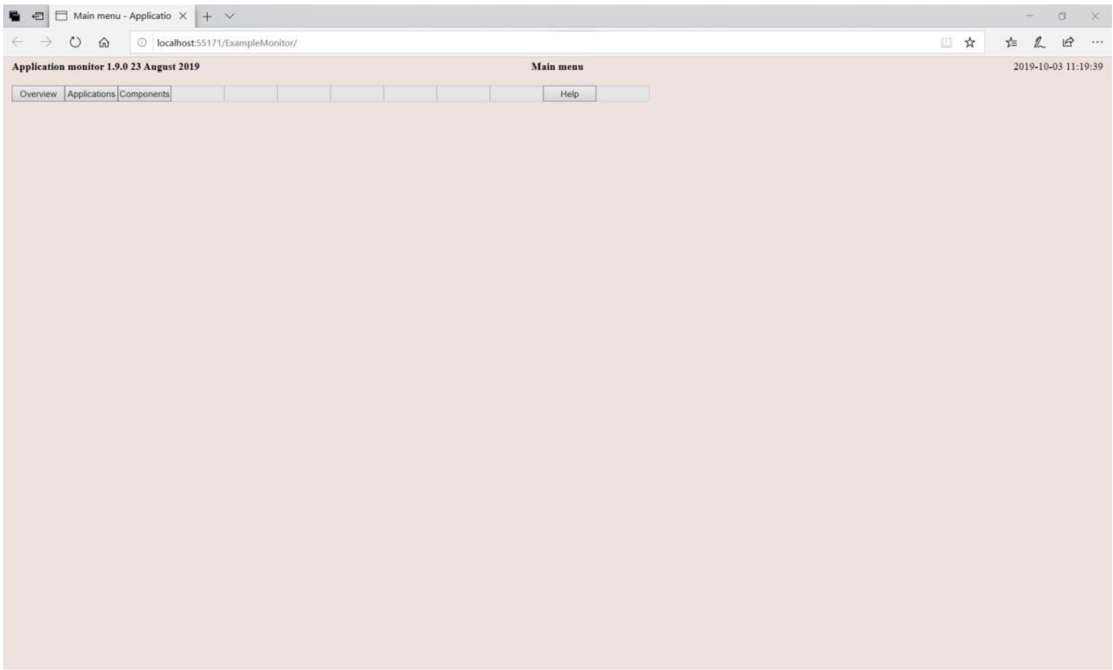
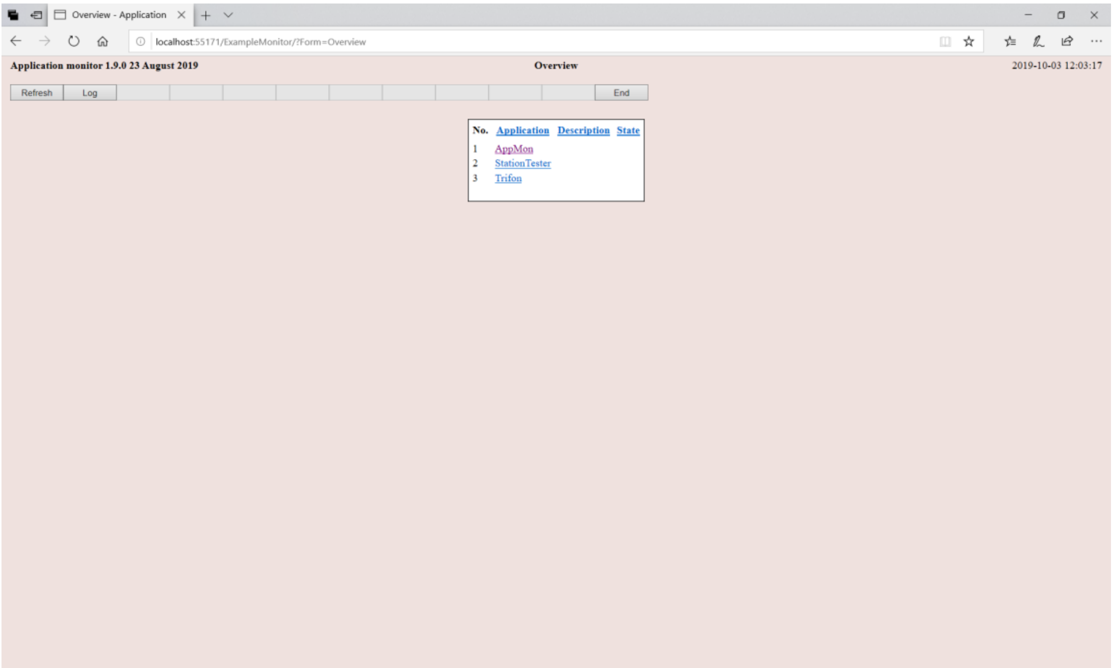
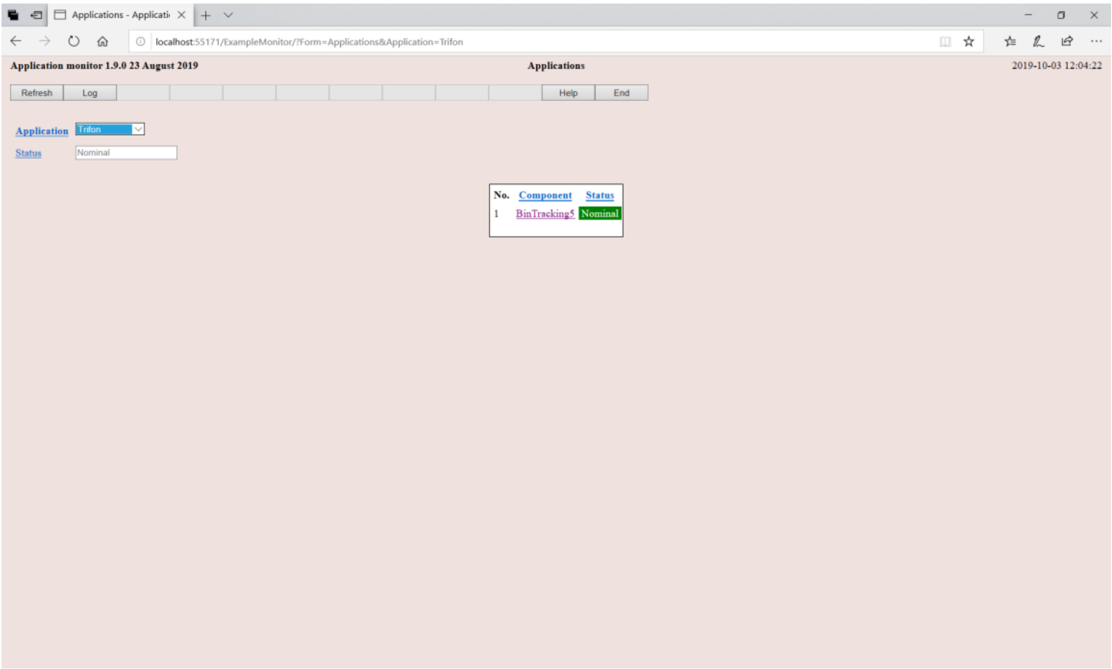


Figure 2 Overview. Three applications


Overview - Application
localhost:55171/ExampleMonitor/?Form=Overview
Application monitor 1.9.0 23 August 2019
Overview
2019-10-03 12:03:17
Refresh
Log
End

No.	Application	Description	State
1	AppMon		
2	StationTester		
3	Trifon		

Figure 3 Application Trifon contains one component, BinTracking5



localhost:55171/ExampleMonitor/?Form=Applications&Application=Trifon

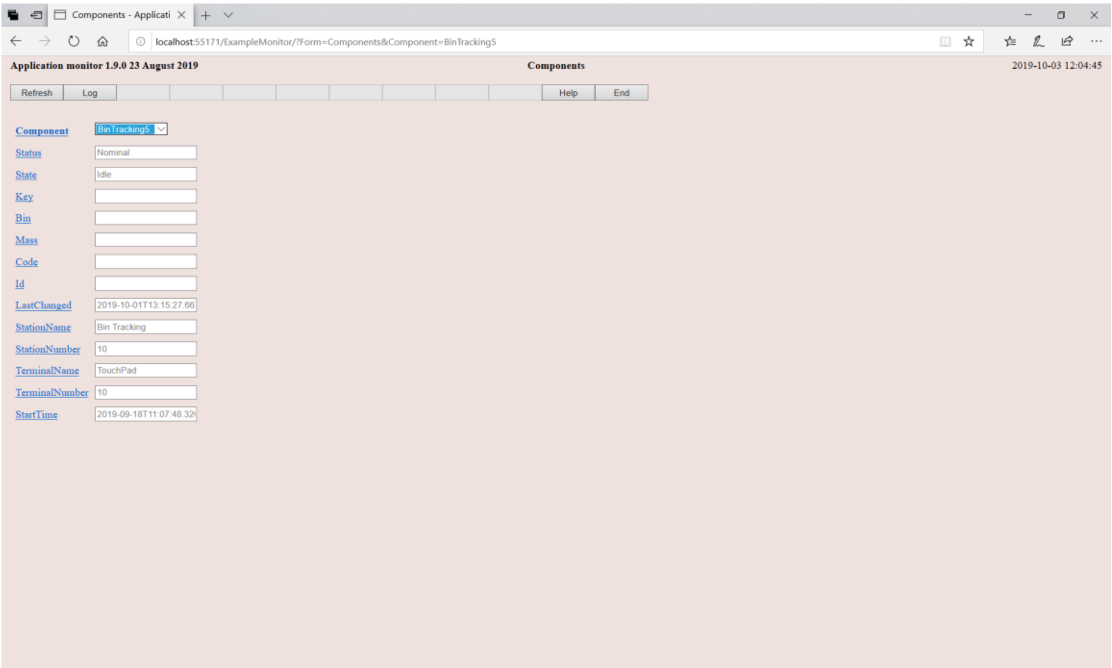
Application monitor 1.9.0 23 August 2019 **Applications** 2019-10-03 12:04:22

Application: Trifon

Status: Nominal

No.	Component	Status
1	BinTracking5	Nominal

Figure 4 Component BinTracking5 in application Trifon

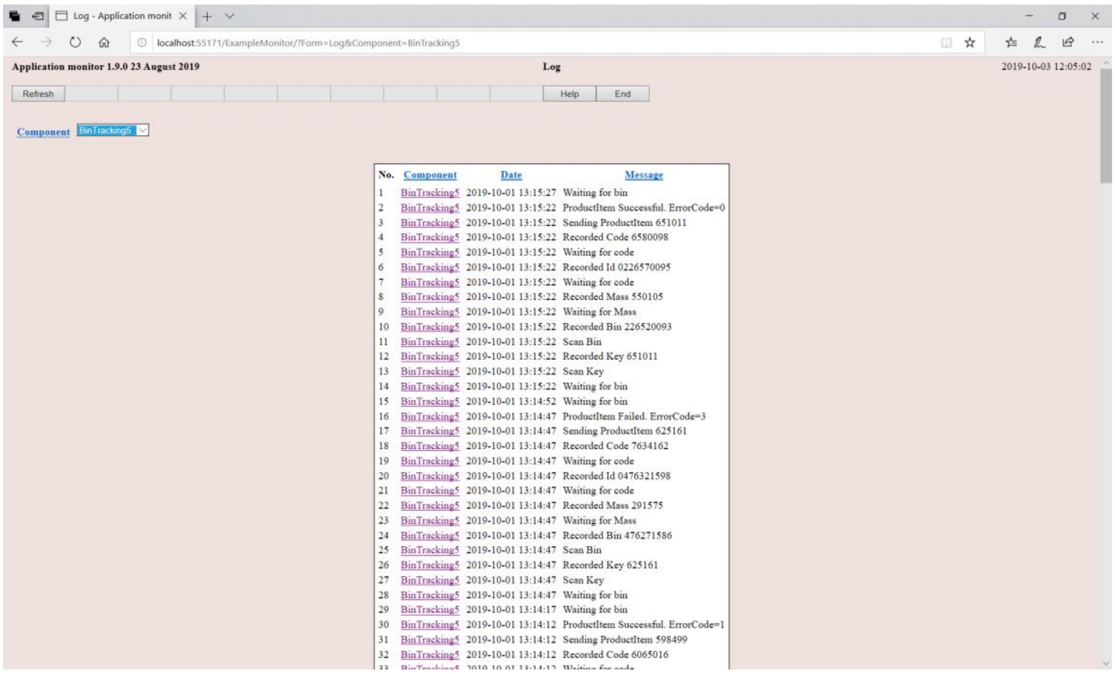


localhost:55171/ExampleMonitor/?Form=Components&Component=BinTracking5

Application monitor 1.9.0 23 August 2019 **Components** 2019-10-03 12:04:45

Component	BinTracking5
Status	Nominal
State	Idle
Key	
Bin	
Mass	
Code	
Id	
LastChanged	2019-10-01T13:15:27.66
StationName	Bin Tracking
StationNumber	10
TerminalName	TouchPad
TerminalNumber	10
StartTime	2019-09-18T11:07:48.32

Figure 4 Component BinTracking5 event log


Log - Application monit
localhost:55171/ExampleMonitor/?Form=Log&Component=BinTracking5
Refresh
Help
End

Application monitor 1.9.0 23 August 2019 **Log** 2019-10-03 12:05:02

Component: BinTracking5

No.	Component	Date	Message
1	BinTracking5	2019-10-01 13:15:27	Waiting for bin
2	BinTracking5	2019-10-01 13:15:22	ProductItem Successful. ErrorCode=0
3	BinTracking5	2019-10-01 13:15:22	Sending ProductItem 651011
4	BinTracking5	2019-10-01 13:15:22	Recorded Code 6580098
5	BinTracking5	2019-10-01 13:15:22	Waiting for code
6	BinTracking5	2019-10-01 13:15:22	Recorded Id 0226570095
7	BinTracking5	2019-10-01 13:15:22	Waiting for code
8	BinTracking5	2019-10-01 13:15:22	Recorded Mass 550105
9	BinTracking5	2019-10-01 13:15:22	Waiting for Mass
10	BinTracking5	2019-10-01 13:15:22	Recorded Bin 226520093
11	BinTracking5	2019-10-01 13:15:22	Scan Bin
12	BinTracking5	2019-10-01 13:15:22	Recorded Key 651011
13	BinTracking5	2019-10-01 13:15:22	Scan Key
14	BinTracking5	2019-10-01 13:15:22	Waiting for bin
15	BinTracking5	2019-10-01 13:14:52	Waiting for bin
16	BinTracking5	2019-10-01 13:14:47	ProductItem Failed. ErrorCode=3
17	BinTracking5	2019-10-01 13:14:47	Sending ProductItem 625161
18	BinTracking5	2019-10-01 13:14:47	Recorded Code 7634162
19	BinTracking5	2019-10-01 13:14:47	Waiting for code
20	BinTracking5	2019-10-01 13:14:47	Recorded Id 0476321598
21	BinTracking5	2019-10-01 13:14:47	Waiting for code
22	BinTracking5	2019-10-01 13:14:47	Recorded Mass 291575
23	BinTracking5	2019-10-01 13:14:47	Waiting for Mass
24	BinTracking5	2019-10-01 13:14:47	Recorded Bin 476271586
25	BinTracking5	2019-10-01 13:14:47	Scan Bin
26	BinTracking5	2019-10-01 13:14:47	Recorded Key 625161
27	BinTracking5	2019-10-01 13:14:47	Scan Key
28	BinTracking5	2019-10-01 13:14:47	Waiting for bin
29	BinTracking5	2019-10-01 13:14:17	Waiting for bin
30	BinTracking5	2019-10-01 13:14:12	ProductItem Successful. ErrorCode=1
31	BinTracking5	2019-10-01 13:14:12	Sending ProductItem 598499
32	BinTracking5	2019-10-01 13:14:12	Recorded Code 6065016
33	BinTracking5	2019-10-01 13:14:12	Waiting for code

Monitor application description

26 September 2012

Background

The Monitor software displays screens of applications, components and event logs in **Internet Explorer**. It is deployed under the default web site in IIS in a web site called **Monitor**. It is a web server that reads **Status** elements from a **SharedStateEndpoint**, transforms it to HTML and displays it in **IE.** The **Status** XML elements conform to the new **Stat**

usSchema.xsd. The **StatusSchema.xsd** was included in the Cascades deployment.

Start **IE** and enter the following URL http://localhost/Monitor/?Form=MainMenu

Monitor reads **XML Status** messages from a Cascades endpoint, in this case a **SharedStateEndpoint**. It can be any Cascades endpoint. The advantage of using a **SharedStateEndpoint** is that it will always receive a message even if the data had not changed because the row in the SQL Server database is always available.

Monitor was created as a website in the **Default Web Site** with **IIS.**

The **web.config** file contains a section called **CascadesInterface** which specifies the **<services>** section in the same format as all the Cascades configuration files. The **CascadesInterface** section name must be specified in the **<configSections>** section.

The **service** section must contain either an **inbound endpoint** or an **inbound channel**. If a message queue endpoint is used it must be specified as **inbound endpoint**. If a **SharedStateEndpoint** is used, it must be specified as **inbound channel**, because **SharedStateEndpoints** can only connect – they cannot accept incoming connections.

If it is configured to process commands, it must contain an **outbound channel**

The transforms

The **transforms** section must contain an entry to specify the XML transform to be used for every screen. The transform name must be the same as the name of the screen.

These transforms are used to transform the **Status** elements to **AbstractForm** elements. The **AbstractForm** element is a legacy of the *Execute!Now* and *Palette* web applications. There is no schema for it. Originally it was used to model a web form with a collection of controls and links.

In addition, the **transforms** section must contain an entry with the name *Brand.* The **Brand** transform is used to transform the **AbstractForm** element of each screen to HTML for display in Internet Explorer. It references a cascading style sheet *NXAUTO/nxauto.css* which is used to select styles, colours and positions of controls.

We currently have the following screens

Overview

MainMenu

Applications

Components

Log

We can add screens by adding transforms.

If the **Monitor** application is to process commands, the **transforms** section must also contain an entry with the name *Command.* The **Command** transform uses the name of the command (obtained from the button that was pressed) and the current **Status** element to generate a **Request** element. It does no validation. All validation is done in the **StatusStateMacine**.

Deployment

Create a web site *Monitor* under the default web site in IIS

Copy the contents of the **Monitor** folder in *Cascades deployment\Current\Application monitor* to the folder for the **Monitor** web site.

If a **SharedStateEndpoint** is used, the following changes must be made in SQL Server

Add a login for the associated IIS account. In my case it was **IIS APPPOOL\monitor**

Create a user in the **SharedState** database and associate it with the user account above. I used the same user account.

Grant **execute** permission on stored procedures **GetSharedState** and **SetSharedState**

References

1. *Title*, Author, Affiliation, Date

References

2. *Title*, Author, Affiliation, Date.