

Developer Guide

Fabasoft app.telemetry 2012 SDK Samples

Valid from February 28, 2012

Fabasoft[®]

Copyright ©

app.telemetry GmbH, A-4020 Linz, 2012.

All rights reserved. All hardware and software names used are registered trade names and/or registered trademarks of the respective manufacturers.

These documents are highly confidential. No rights to our software or our professional services, or results of our professional services, or other protected rights can be based on the handing over and presentation of these documents.

Distribution, publication or duplication is not permitted.

Contents

1 Introduction	4
2 Installation	4
2.1 Prerequisites for Installation	4
2.2 Installation of Online Shop Services	4
2.2.1 Setup Java Stock Service	4
2.2.2 Setup C++ Repository Service	5
2.2.3 Setup ASP.NET Frontend Web Service	5
2.2.4 Application Registry Settings	6
2.2.5 File system permissions for ASP.NET web service:	6
3 Running the Online Shop Sample	6
4 Appendix	7
4.1 Configuration of IIS with .NET 2.0	7

1 Introduction

The Fabasoft app.telemetry 2012 SDK samples should help developers instrumenting their own application using the Software-Telemetry SDK.

The “Online Shop” SDK sample extends the SDK documentation with a simple reference implementation of an instrumented multi-tier application consisting of the following components:

- Online Shop front-end service written as ASP.NET application in C# (OnlineShopCS)
- Online Shop back-end services delivering the data for the front-end service
 - Stock service managing availability of shop articles realized as Java web service (OnlineShopStockJava)
 - Repository service delivering article data written as C++ application (OnlineShopRepositoryCPP)

2 Installation

The front-end service of the Online Shop is written as ASP.NET application hence the sample application can only be run on a Microsoft Windows Server.

2.1 Prerequisites for Installation

- Microsoft Windows Server 2003 with installed Internet Information Services (IIS)
- Microsoft .NET Framework Version 2.0 or higher

Note: If the application server component (IIS) is installed after the .NET Framework 2.0, the .NET application settings for the IIS are not updated automatically. You have to ensure that the IIS is capable of running .NET 2.0 applications (for details see appendix)
- SUN Java Runtime Environment (JRE) 6
- Installed Fabasoft app.telemetry Server (including Agent and WebAPI on Microsoft Windows systems)

2.2 Installation of Online Shop Services

- Copy the “Install” directory from the “...\Developer\Samples\OnlineShop” directory to the desired target directory (e.g. “C:\Program Files\OnlineShop-Sample”)
- The new directory structure should look the following way:
 - C:\Program Files\OnlineShop-Sample\Frontend (ASP.NET web service)
 - C:\Program Files\OnlineShop-Sample\Stock (Java stock service)
 - C:\Program Files\OnlineShop-Sample\Repository (C++ repository service)
 - On 64 bit Microsoft Windows systems use the 64 bit C++ repository service DLL form the subfolder “Repository_x64” instead.

2.2.1 Setup Java Stock Service

The Java stock service is a simple Java application consisting of some Java class files. The main class file is called “SimpleHttpStockServiceInstrumented.class” which has to be called to start the Java web service. The Java application is instrumented with the Fabasoft app.telemetry Software-Telemetry SDK so it requires the Java SDK library “softwaretelemetry.jar”.

To start the Java stock service, just call the script “StartStockService.cmd” either from the Microsoft Windows Explorer window or from the Command prompt (has to be called from inside the directory of the Java class files)

The service is started using the main class file and including the Java SDK library in the classpath.

```
java -cp softwaretelemetry.jar;. SimpleHttpStockServiceInstrumented

25.02.2010 13:37:47 SimpleHttpStockServiceInstrumented main
INFO: HTTP Server is listening on port: 8085

25.02.2010 13:37:47 SimpleHttpStockServiceInstrumented main
INFO: Waiting for incoming requests ...
```

The Java stock service starts a local HTTP server listening on port 8085 and waits for incoming calls from the frontend service.

2.2.2 Setup C++ Repository Service

The repository service is implemented as C++ web service DLL (IIS 6 Web Service Extension), so you have to create a virtual directory inside your IIS for request against the C++ repository:

C:\Program Files\OnlineShop-Sample\Repository\OnlineShopRepository.dll

C:\Program Files\OnlineShop-Sample\Repository_x64\OnlineShopRepository.dll (for 64 bit)

- 1.) Create a new IIS application pool for the Online Shop web services (this application pool can be used for both web services – and it is optional, you could also use the default application pool).
- 2.) Create a **new virtual directory** in the “Default Web Site” (or in your own desired web site) for the repository service with the following parameters:
 - New Virtual Directory:
 - Application name/Alias: “repository”
 - Local path: “C:\Program Files\OnlineShop-Sample\Repository” or (...Repository_x64)
 - Execute permissions: Read + Execute
 - Change some default properties:
 - Application pool: the one that you have just created
 - Directory security: Anonymous access (+ Integrated Windows authentication)
- 3.) Create a **new IIS Web Service Extension** for the repository service DLL of the appropriate architecture and allow access to the file/extension:
 - a. C:\Program Files\OnlineShop-Sample\Repository\OnlineShopRepository.dll
 - b. C:\Program Files\OnlineShop-Sample\Repository_x64\OnlineShopRepository.dll

2.2.3 Setup ASP.NET Frontend Web Service

The frontend of the web application is realized as ASP.NET web service implemented in C#, so you have to create a virtual directory inside your IIS for all requests from the browser to your Online Shop sample.

The Online Shop ASP.NET web service is based on the content (static content and dynamic ASP pages) from the directory:

C:\Program Files\OnlineShop-Sample\Frontend

- 1.) Create a **new virtual directory** in the “Default Web Site” (or in your own desired web site) for the Online Shop frontend service with the following parameters:

- New Virtual Directory:
 - Application name/Alias: “webshop”
 - Local path: “C:\Program Files\OnlineShop-Sample\Frontend”
 - Execute permissions: Read + Execute
- Change some default properties:
 - Application pool: the one that you have just created
 - Directory security: Anonymous access (+ Integrated Windows authentication)
 - Set ASP.Net Version for the Website “Online Shop” to 2.0.50727 or higher
 - **Note:** The “ASP.NET” configuration tab is located in the virtual directory properties dialog. If this dialog is missing, the .NET Framework might not be installed or the IIS ASP features might be missing – for details see appendix.

2.2.4 Application Registry Settings

If you do not already have any registry key for “Fabasoft app.telemetry” software, create a new one: `\HKEY_LOCAL_MACHINE\SOFTWARE\Fabasoft app.telemetry`

Important: Set registry-permissions for that key (`HKLM\SOFTWARE\Fabasoft app.telemetry`) to “Everyone”=“Full Control” (The application tries to create the sub key “OnlineShopSample”)

- 1.) Create a new sub key for the sample application:
`HKLM\SOFTWARE\Fabasoft app.telemetry\OnlineShopSample`
- 2.) Create a new String value inside that key:
`HKLM\SOFTWARE\Fabasoft app.telemetry\OnlineShopSample\RepositoryServiceUrl` and set the value pointing to the repository service (normally:
`http://localhost[:port]/repository/OnlineShopRepository.dll`)
Note: This registry key value is mandatory for the Online Shop – if it is missing the sample will not work.
- 3.) The other application parameters (RequestSize and WorkerThreads) can be set at runtime via the Online Shop settings page. These values will be persisted below the previously mentioned registry key: `HKLM\SOFTWARE\Fabasoft app.telemetry\OnlineShopSample.`
 - a. ... \OnlineShopSample\RequestSize
 - b. ... \OnlineShopSample\WorkerThreads

2.2.5 File system permissions for ASP.NET web service:

Important: Set the file system permissions for the ASP.NET temporary folder for the web service user to “Everyone”=“Full Control”:

“C:\Windows\Microsoft .NET\Framework\v<2.0.x>\Temporary ASP.NET files”

or on 64 bit Microsoft Windows systems:

“C:\Windows\Microsoft .NET\Framework64\v<2.0.x>\Temporary ASP.NET files”

3 Running the Online Shop Sample

- Start the Fabasoft app.telemetry services (agent, server, (WebAPI))
- Start Java “Stock Service”
- Start the IIS hosting the Online Shop web services
- Start your desired web browser and navigate the Online Shop web application page which is available on <http://localhost/webshop/WebShop.aspx> (if setup in default web site)

- or [http://localhost\[:port\]/webshop/WebShop.aspx](http://localhost[:port]/webshop/WebShop.aspx) (if using another web site with custom port)
- Setup the application parameters via the settings page of the web application
 - Worker Thread count < 10 will result in an additional delay of 10-x seconds
 - Characters per Request: a good starting value should be 50 characters per request.
 - These values will be persisted in the Microsoft Windows registry.
- Test the Online Shop application (and check the Java console output if requests arrive)
- Open another web browser window and navigate to the Fabasoft app.telemetry web client on <http://localhost/apptelemetry/index.html>
 - Switch to Edit view
 - Create a new Software-Telemetry log definition
 - Define application filters limiting the log pool to the Online Shop sample: Application="Fabasoft app.telemetry", Application ID="Demo"
 - Specify Online Shop log definition sample delivered with the sample: `OnlineShopSample_Definition.xml`.
 - Set the permanent Software-Telemetry log level to DETAIL or DEBUG to see details of the requests.
 - If running the Online Shop sample application in a distributed environment with some services running on another agent than the local one you have to follow these steps:
 - Install the Fabasoft app.telemetry agent also on these remote agents
 - Create infrastructure agent objects for all these remote agents
 - Switch to Telemetry view
 - Add your configured log pool for the Online Shop sample
 - Invoke some more requests within the Online Shop sample application frontend
 - Watch the incoming requests and analyze them
 - Change the application parameters of the Online Shop sample and watch the changing requests.

4 Appendix

4.1 Configuration of IIS with .NET 2.0

Check if the IIS is capable of running .NET 2.0 applications via the IIS Manager in the Computer Management and check if a "Web Service Extension" for "ASP.NET v<2.0.x>" or similar is available and is "Allowed".

1. *Web Site Properties*: If ASP.NET is correctly installed and configured any IIS Web Site (e.g. Default Web Site) must have the "ASP.NET" configuration tab in the properties dialog.
2. *Web Service Extensions*: If no "ASP.NET v<2.0.x>" web service extension is available, create a new web service extension allowing the file "`C:\Windows\Microsoft .NET\Framework[64]\v<2.0.x>\aspnet_isapi.dll`" to be run from inside the IIS.

If you don't have a .NET Framework already installed (check your installed programs list) you have to install a .NET Framework >= 2.0 and restart your computer afterwards.

Important: Set the file system permissions for the ASP.NET temporary folder:

"`C:\Windows\Microsoft .NET\Framework[64]\v<2.0.x>\Temporary ASP.NET files`" for the web service user to "Everyone"="Full Control".