

ACTIVE
Deliverable D5.4.2

Fully functional prototype
of
ACTIVE SDK

Editor:	Igor Dolinsek, HSL
Deliverable nature:	Prototype (P)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	31.5.2010
Actual delivery date:	14.06.2010
Suggested readers:	Developers of the ACTIVE knowledge workspace, technical partners, case study partners. Readers should be familiar with D5.1.1, D5.1.2, D5.3.2 and ACTIVE DOW for better background understanding of the topics covered in this document and related software deliverables.
Version:	0.2
Total number of pages:	12
Keywords:	ACTIVE knowledge workspace

Abstract

This document describes the fully functional prototype of the ACTIVE SDK (software development kit). With ACTIVE SDK it is possible to develop software which extends the ACTIVE knowledge workspace functionality. By using the ACTIVE SDK other programs can make use of workspace features. All ACTIVated applications are developed by using the ACTIVE SDK and also ACTIVE services communicate by using the interfaces published in the SDK. ACTIVE SDK consists of the documentation in HTML form, of the Web service interface definitions, ACTIVE event bus message schemas and ACTIVE client library for C#. This deliverable has the same structure and purpose as deliverable D5.4.1 [3]. The main difference is that D5.4.2 covers all the new workspace features which were developed since the demonstrator version release which was described in D5.4.1.

[End of abstract]

Disclaimer

All ACTIVE consortium parties have agreed to full publication of this document.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the ACTIVE consortium as a whole, nor a certain party of the ACTIVE consortium warrant that the information contained in this document is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.

[Full project title] ACTIVE – Enabling the Knowledge Powered Enterprise

[Short project title] ACTIVE

[Number and title of work-package] Work Package 5, Knowledge workspace

[Document title] Fully functional prototype of ACTIVE SDK

[Editor: Name, company] Igor Dolinsek, HSL

[Work-package leader: Name, company] Igor Dolinsek, HSL

[Estimation of PM spent on the Deliverable] 19

Copyright notice

© 2010 Participants in project ACTIVE

Executive summary

ACTIVE knowledge workspace provides its functionality to the users through its GUI components in the ACTIVE web portal and ACTIVE Taskbar. In addition, software programs can interface the ACTIVE knowledge workspace by using the APIs described in the ACTIVE SDK. Most of the Workspace features are accessed and can be controlled through the APIs.

ACTIVE SDK consists of the

- API documentation in HTML form
- Web service interface definitions
- ACTIVE event bus message definitions
- ACTIVE client library for C#.

By using the ACTIVE SDK other programs can make use of workspace features. All ACTIVated applications are developed by using the ACTIVE SDK and also ACTIVE services communicate by using the interfaces published in the SDK. With ACTIVE SDK it is possible to develop software which extends the ACTIVE knowledge workspace functionality.

Note that the purpose of this document is to provide the index of the various parts of the SDK and provides the instructions how to access them in the ACTIVE SVN repository. The SDK documentation is delivered in HTML format and is not included directly in this deliverable document.

This deliverable has the same structure and purpose as deliverable D5.4.1. The main difference is that D5.4.2 covers all the new workspace features which were developed since the demonstrator version release which was described in D5.4.1.

Note that this is software deliverable and is thus subject to the software deliverable review process, described on the ACTIVE project Wiki in the deliverable D13.1 under 'Software development guidelines'.

List of authors

Company	Author
HSL	Igor Dolinsek
JSI	Tadej Stajner
ISOCO	Carlos Ruiz Moreno
EMIC	Marcel Tilly

Table of Contents

Executive summary	3
List of authors	4
Table of Contents	5
List of figures	6
Abbreviations	7
Definitions	8
1 Introduction	9
2 Location of SDK components in repository	10
References	12

List of figures

Not applicable

Abbreviations

AKWS – ACTIVE knowledge workspace

API – applications programming interface

GUI – graphical user interface

HTML – hyper text mark-up language

IE – Internet Explorer

SDK – software development kit

RDOC – Ruby programs documentation format in HTML form

Definitions

Not applicable

1 Introduction

Programs can interface the ACTIVE knowledge workspace by using the APIs described in the ACTIVE SDK. Most of the Workspace features are accessed and can be controlled through the APIs.

ACTIVE SDK consists of the

- API documentation in HTML form
- Workspace Web service interface definitions
- ACTIVE event bus message definitions
- ACTIVE client library for C#.

API documentation in HTML form is composed of the chapters which describe the workspace services and their functionality and some chapters which describe particular application interfaces or the structure of Event bus messages. In addition some examples of the interfacing code with explanations are provided in the HTML documentation.

Web service interface definitions can be accessed online when the ACTIVE workspace is running. Programmers can generate the interfacing code for those WSDLs by using the appropriate tools of their choice.

Ruby interface methods of the infrastructure, context and metadata services are described in RDOC format in the HTML documentation, interface methods of the task service are available in automatically generated HTML pages and context mining and metadata recommender services are described in plain HTML format.

ACTIVE event bus message structures and topics are described in HTML documentation and for most PrimitiveEvents there are XML schemas available in the active_server.zip package. ACTIVE event bus messages can be processed by using the low level Apache ActiveMQ libraries for the respective platform and language. For Windows XP those are Apache.NMS.dll and Apache.NMS.ActiveMQ.dll. For convenience reasons they are included in the ACTIVE client package (AKWS_Client.zip).

ACTIVE client library for C# provides some convenience C# methods which are useful when developing ACTIVated applications in C#. They are documented in a separate section of the HTML documentation. They are provided in the ACTIVE client package (AKWS_Client.zip) in form of 2 DLLs: AKWSLib.dll and AKWSUtil.dll.

In order to experiment with the SDK it is necessary to install the ACTIVE knowledge software, provided by deliverable D5.3.2[2].

2 Location of SDK components in repository

SDK documentation in HTML form is located in the ACTIVE project subversion repository

<http://svn.active-project.eu/active/repos>

in

akws/bin/fully_functional

directory.

Documentation is available in a zip package and in the related “readme” file:

- active_sdk_doc.zip
- active_sdk_doc_readme.docx

Note that for convenience reasons the *active_sdk_doc.zip* is also contained in the ACTIVE server package *active_server.zip* in SVN, which is also publically downloadable from the project’s Web site [1].

Low level libraries for accessing the **ACTIVE event bus** are available from the Apache website (visit link related to the ActiveMQ software). For Windows XP platform those are Apache.NMS.dll and Apache.NMS.ActiveMQ.dll. For convenience reasons they are included in the ACTIVE client package (AKWS_Client.zip).

Schemas for most **PrimitiveEvent bus messages** are located in the server package (*events_demo/ActiveMQConsumerSource/*.xsd*).

ACTIVE client C# libraries AKWSLib.dll and AKWSUtil.dll are located in the ACTIVED client package.

For proper operation of the ACTIVE C# client libraries it is necessary that client configuration file, client registry key and certain folder structure are present on the respective system. These requirements are usually established on the system during AKWS Client package installation. However if ACTIVE Client is not deployed, developer must add configuration file, folder and one registry key manually. The following section describes how to manually perform these settings.

1. Create ACTIVE Client configuration file.

Create new text file and give it name *akws_desktop_config.cfg*. This file must contain all ACTIVE Client configuration parameters. Here is example of this file.

```
AKWS_SERVER activeserver
AKWS_PORT 3000
AKWS_WS_TIMEOUT 7
AKWS_MAXFILESIZE 5
AKWS_USER root
AKWS_PASSWORD root
AKWS_USERANDDOMAINNAME rootuser,rootuserdomain
AKWS_ACTIVEMQURI tcp://activeserver:61616
AKWS_SMW_SERVER activeserver
```

You can adjust values for these parameters to match you configuration.

2. Create ACTIVE Client registry key.

Position of various AKWS client-related directories and files on the client system is defined with the Windows registry key named ‘AKWS’. This registry key is located under

'HKEY_CURRENT_USER\Software\'. Create new string value named AKWS_DESKTOP_DIRECTORY under that key, and set its value to path to some folder on your system.

By default ACTIVE Client installation will set this value to 'C:\Program Files\ACTIVE\AKWS_Client\'. If this folder does not exist you have to create it.

3. Copy configuration file to ACTIVE Client folder

In the 'ACTIVE Client' file folder create a subfolder named 'config'. Then copy akws_desktop_config.cfg file you created in step 1 to this subfolder.

4. Create log, temp and data folders

In the 'ACTIVE Client' file folder create subfolders with names 'log', 'temp' and 'data'.

References

- [1] <http://www.active-project.eu> : Project external website
- [2] ACTIVE deliverable D5.3.2: Fully functional prototype of ACTIVE Knowledge Workspace
- [3] ACTIVE deliverable D5.4.1: Demonstrator version of ACTIVE Knowledge Workspace SDK