



Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE9.1.3 First Prototype for CMS Integration and Feedback

Version: 1.2

Date: 30/08/2006 18:04:00

Responsible: EXITECH (ff@exitech.it) (revised and accepted by coordinator)

Project Number: IST-2-511299

Project Title: AXMEDIS

Deliverable Type: Public (only for this document, not the prototype code)

Visible to User Groups: Yes

Visible to Affiliated: Yes

Visible to Public: Yes

Deliverable Number: DE9.1.3

Contractual Date of Delivery: see annex I

Actual Date of Delivery: 30/08/2006

Title of Deliverable: First Prototype for CMS Integration and Feedback

Work-Package contributing to the Deliverable: WP 9.1.1, WP 9.1.2

Task contributing to the Deliverable: all of WP9

Nature of the Deliverable: Prototype

Author(s): EXITECH

Abstract:

This deliverable is related to the mock up of WP9.2 that has been improved in this period by producing a first real prototype integrated with CAMART, AXDB and therefore also with AXCS.

Keyword List:

Administrative data, AXCS, AXDB, CAMART

Table of Contents

1	EXECUTIVE SUMMARY AND REPORT SCOPE	3
2	INTRODUCTION	3
3	PROTOTYPE DESCRIPTION	3
3.1	CAMART FOR STATISTICS	3
3.1.1	Introduction	3
3.1.2	The prototype	7
3.2	CAMART AND ADMINISTRATIVE INFORMATION INTEGRATOR	12
3.2.1	Introduction	12
3.2.2	The prototype	13
4	BIBLIOGRAPHY	17

1 Executive Summary and Report Scope

This deliverable is the accompanying document for the Prototype having the name common to the deliverable.

2 Introduction

WP9.2 has been improved in this period by producing a first real prototype integrated with CAMART, Statistic engine for CAMART and Administrative Information Integrator. This integration processes involved also AXDB and AXCS.

This prototype is able to perform the following tasks:

- Getting logs coming from AXCS web service
- Organizing such logs in the internal database of the factory
- Generating internal XML format in polling mode (this has been done in order to demonstrate the new internal format
- Generating the XML format provided by one of the partners (ILABS in that case), and publishing at a predefined time frequency the resulting XML in an ftp directory
- Generating also in polling mode an XML format that is transformed according to the profiled XSLT.
- Generating top-bottom ten on demand for statistics.
- Authentication
- Development of a web service for gathering statistics instead of the GUI only
- The work to be performed yet is not functional but mainly related to refinement, usability and user interface. The work can be summarized in the following points:
 - Generate a user interface for creating user profiles (assign ftp, assign XSLT, etc to each factory user, getting users from the factory)
 - Produce the AII user interface meliorated in terms of usability and uniformity with respect to the other user interface present in the system (this should be discussed in order to have a common user interface in terms of graphics and CSS for all the web application in the AXMEDIS system)

3 Prototype Description

The first prototype of the DE 9.1.2 has been produced and presented. In this document a short presentation of the provided system will be given in order to evidence the work done.

3.1 CAMART for Statistics

3.1.1 Introduction

The role of Core Accounting manager and Reporting Tool (CAMART) for Statistics is strictly bound with database for logs (provided by AXCS) since it has to collect information from AXMEDIS Certifier and Supervisor about Action Log and store them into the AXMEDIS database, homogenizing fragmented information the different tools sent to the AXCS. This is very important as the AXCS or the SuperAXCS will not store forever data related to each user, but will use policies to remove detailed data, after a certain amount of time or after some triggered operations that have to be defined. **Only data for statistical purposes will be kept.**

The statistics are also accessible also by a Web Service Interface whose WSDL is reported below:

```
<?xml version="1.0" encoding="UTF-8"?><definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:y="http://new.webservice.namespace"
xmlns:cam="http://www.exitech.it/CamartStats.xsd" targetNamespace="http://new.webservice.namespace">
```

```

<types>
  <xs:schema targetNamespace="http://www.exitech.it/CamartStats.xsd"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
    <!-- version 1.3 (25-Jul-2006)
    JWSDP compatible -->
    <xs:element name="CamartRecord">
      <xs:annotation>
        <xs:documentation>Root element for a Camart
record</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="AXCSID" type="xs:string">
            <xs:annotation>
              <xs:documentation>log
ID</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="executionTimestamp"
type="xs:dateTime">
            <xs:annotation>
              <xs:documentation>timestamp of the execution of the Transaction</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="registrationTimestamp"
type="xs:dateTime">
            <xs:annotation>
              <xs:documentation>Timestamp of the registration of the Transaction in
AXCS</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="AXOID">
            <xs:annotation>
              <xs:documentation>ID of
the object inside the axmedis system</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
              <xs:restriction
base="xs:string">
                <xs:maxLength
value="40"/>
                <xs:whiteSpace
value="collapse"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="AXDID">
            <xs:annotation>
              <xs:documentation>Unique
AXMEDIS Object Distributor ID that has performed the transaction</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
              <xs:restriction
base="xs:string">
                <xs:maxLength
value="40"/>
                <xs:whiteSpace
value="collapse"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="AXCID" type="xs:string">
            <xs:annotation>
              <xs:documentation>Creator Id</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="operation"
type="xs:string">
            <xs:annotation>
              <xs:documentation>Use of
the content (reading, printing, aggregating, editing).</xs:documentation>

```

```

</xs:annotation>
</xs:element>
<xs:element name="location"
type="xs:string">
<xs:annotation>
<xs:documentation>Geographical area in which the object is used.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="ISRC" type="xs:string"
minOccurs="0">
<xs:annotation>
<xs:documentation>Factory Infomation: Unique standard ID for identifying the
piece</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="objectCode"
type="xs:string" minOccurs="0">
<xs:annotation>
<xs:documentation>Factory Information: this field fill be filled with the content of the
field "type" of the "DCMI" table if present </xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="CamartReport">
<xs:annotation>
<xs:documentation>Root element of internal Camart
report</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element ref="cam:CamartRecord"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="Statistics" minOccurs="0"
maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element
name="StatsItem" maxOccurs="unbounded">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="count" type="xs:int" use="required"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attributeGroup
ref="cam:StatsAttribGroup"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="CamartRequest">
<xs:annotation>
<xs:documentation>Root element of the request used
in WS</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element name="authorization">

```

```

name="user" type="xs:string" use="required"/>
type="xs:string" use="required"/>
type="xs:dateTime"/>
minOccurs="0"/>
minOccurs="0"/>
minOccurs="0"/>
minOccurs="0"/>
minOccurs="0"/>
maxOccurs="unbounded">
ref="cam:StatsAttribGroup"/>
value="TopChart"/>
value="BottomChart"/>
value="location"/>
value="operation"/>
</types>
<message name="CamartRequestMessage">
  <part name="parameters" element="cam:CamartRequest"/>
</message>
<message name="CamartReportMessage">
  <part name="parameters" element="cam:CamartReport"/>
</message>
<portType name="CamartStatsWS">
  <operation name="getStats">
    <input message="y:CamartRequestMessage"/>
    <output message="y:CamartReportMessage"/>
  </operation>
</portType>
<binding name="CamartStatsBinding" type="y:CamartStatsWS">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getStats">
    <input>
      <soap:body use="literal"/>
    </input>
  </operation>
</binding>
</service>
</schema>

```

```
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
</binding>
<service name="CamartStats">
    <port name="CamartStatsPort" binding="y:CamartStatsBinding">
        <soap:address location="http://localhost:8080/CamartStats/ws"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />
    </port>
</service>
</definitions>
```

3.1.2 The prototype

The system is divided mainly in two parts, that are the CAMART for statistical analysis, that in its first prototype is quite raw in the statistics provided and in the operators that can be applied.

The User interface is reported in the following. The first screen is a login screen:

The screenshot shows a web browser window titled "Camart Statistic Login - Mozilla Firefox". The address bar shows the URL "http://localhost:8080/CamartStats/". The browser's toolbar includes various icons for navigation and search. The main content area has a dark blue header with the "AXMEDIS" logo in yellow and the text "Camart Statistic Login" in white. Below the header, there is a light gray section containing a "Get Logs" button in a dark blue box. Underneath this button, there are two input fields: "User Name" and "Password:". Below the password field are two buttons: "Submit" and "Reset". At the bottom of the page, there is a dark blue footer with the text "(c) Exitech 2005" and "Camart" in yellow. The status bar at the very bottom of the browser window shows the word "Completato".

After the login

The screenshot shows a web browser window titled "Camart Statistic Service - Mozilla Firefox". The address bar shows the URL "http://localhost:8080/CamartStats/filter.jsp". The browser's toolbar includes buttons for back, forward, home, and search, along with a search bar. The page content features a dark blue header with the "AXMEDIS" logo and the text "Camart Statistic Service". Below the header, there is a "Get Logs" button. A "Start date:" label is followed by a text input field containing "2006-05-12 03:04:45". A "Filters" section contains five input fields labeled "AXOID:", "AXCID:", "AXDID:", "AXDOM:", and "AXWID:". Below these fields, there is a "Chart" section with a checkbox, a dropdown menu showing "Top", a text input field with "3", another dropdown menu showing "AXOID", and a link "add another chart". At the bottom of this section are "Submit" and "Reset" buttons. The footer of the page is dark blue and contains the text "(c) Exitech 2005" and "Camart". The browser's status bar at the bottom shows "Completato".

Where it can be evidenced that after providing a user name and password and a search criteria in the Date Field, the system will ask for the records contained in the AXCS and after getting such information the record are filtered according to some criteria based on “equal” operator applied to the fields AXOID, AXCID, AXDID, AXDOM, AXWID.

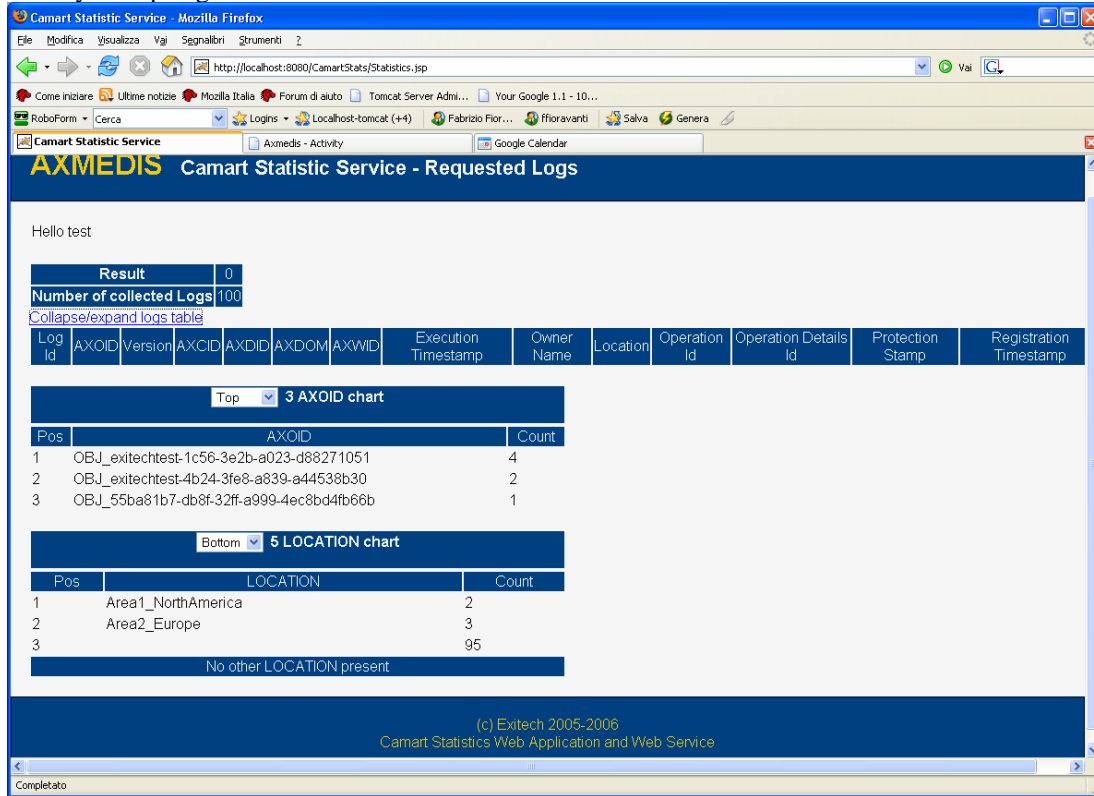
Moreover it is possible to add charts that are summary of the statistics in the form of Top and Bottom ten that can be provided for AXOID, AXCID, AXDID, location and operation.

By adding two charts we obtain:

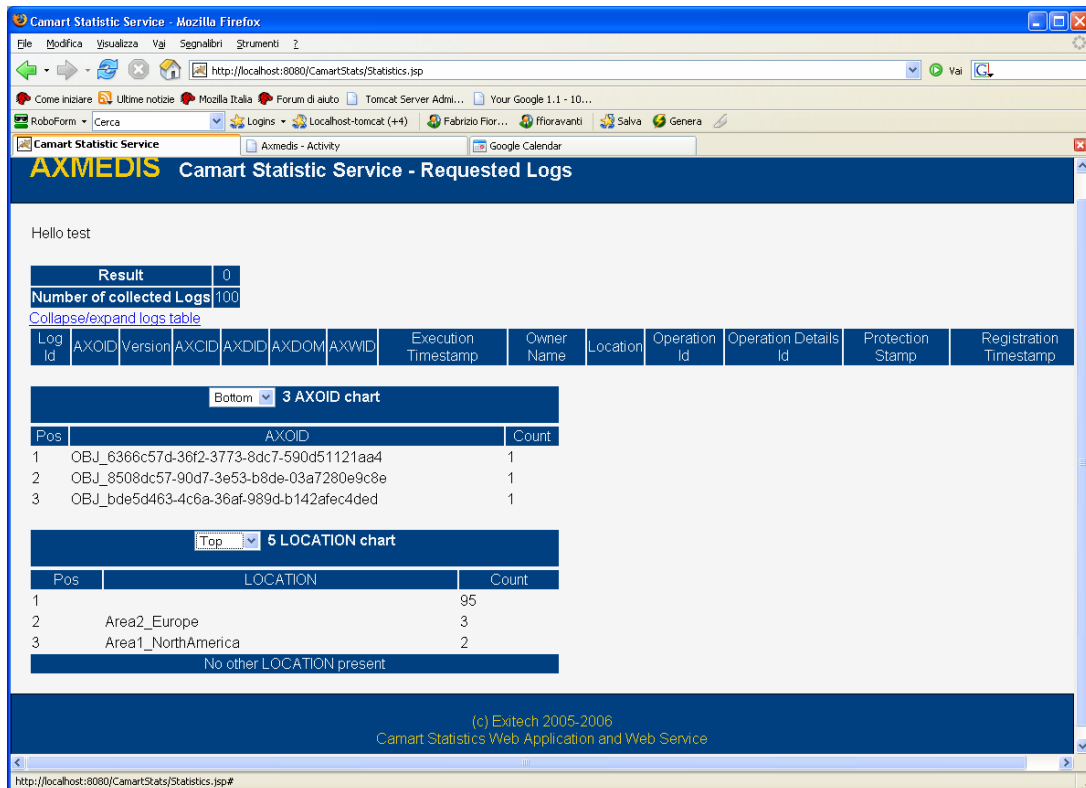
If a request without filtering is issued a web page like the following is obtained:

Log Id	AXOID	Version	AXCID
CSO_560ead4e-b2d8-339b-af26-940c49e27c3c	OBJ_fd7914a4-4476-3764-a70c-5adb0744cf23	1.0	BUS_d0719d28-e695-4db7-841c-f078ae7fdb6 DIS_
CSO_a22bcc04-ec8c-3115-bb70-ba488a5d3dc3	OBJ_887215ee-98f6-3eaf-972b-8f2d8a0613a1	7.0	BUS_d0719d28-e695-4db7-841c-f078ae7fdb6 DIS_
CSO_1efac220-5bac-3170-8d98-ed6d9e73b34a	OBJ_9ecc229e-a301-3e6a-a248-0bf23a7f9573	4.0	BUS_d0719d28-e695-4db7-841c-f078ae7fdb6 DIS_
CSO_eb9a2151-faab-3eeb-98f4-beb1153b6019	OBJ_14ac2d94-ae7c-3036-8e4a-b36d28e7eddb	4.0	BUS_d0719d28-e695-4db7-841c-f078ae7fdb6 DIS_
CSO_0bd70949-71c2-3e77-9cff-4095a2669d8e	OBJ_34c17c3f-e1eb-3b35-8c0c-70614dcd54ae	6.0	BUS_d0719d28-e695-4db7-841c-f078ae7fdb6 DIS_
CSO_87b076f0-05fb-3905-ba58-80a0e5ebc847	OBJ_74661a65-7bdc-3121-857a-d94df284356f	1.0	BUS_d0719d28-e695-4db7-841c-f078ae7fdb6 DIS_
CSO_f60f129c-5a84-3801-a501-b491a5b963c8	OBJ_b27a7f61-95cd-3042-b9b4-3574ec31df17	5.0	BUS_d0719d28-e695-4db7-841c-f078ae7fdb6 DIS_
CSO_b6cb2ca5-dd50-37df-a55d-f919a9ac4ff4	OBJ_4a1ce19f-2370-3edc-a9f0-f28db7cb4d35	4.0	BUS_d0719d28-e695-4db7-841c-f078ae7fdb6 DIS_
CSO_85a7894c-7a54-3789-b3f5-f44a307e5668	OBJ_f162f44-d840-3821-8163-b1c3e9b0deae	4.0	BUS_d0719d28-e695-4db7-841c-f078ae7fdb6 DIS_

And by collapsing the results we obtain:



The charts can be provided in the form of the opposite of the request in terms of bottom and top:



The filtering is now provided AXCS server side in order to minimize the load on the network, and the results can be filtered as in the example below, where a filter on both AXOID and AXCID are provided.

The screenshot shows the 'Camart Statistic Service' web application in a Mozilla Firefox browser. The page title is 'AXMEDIS Camart Statistic Service'. The URL is 'http://localhost:8080/CamartStats/filter.jsp#'. The page contains a 'Get Logs' section with a 'Start date' field set to '2006-05-12 03:04:45'. Below this is a 'Filters' section with input fields for 'AXOID', 'AXCID', 'AXDID', 'AXDOM', and 'AXWID'. The 'AXOID' field is populated with 'OBJ_exitechtest-1c56-3e2b-a023-d88271051'. The 'AXCID' field is populated with 'BUS_d0719d28-e695-4db7-841c-f078ae7fd6'. Below the filters is a 'Chart' section with a checked 'Chart' checkbox, a 'Top' dropdown, a '3' dropdown, an 'AXOID' dropdown, and a link 'add another chart'. There are 'Submit' and 'Reset' buttons at the bottom of the filter section. The footer of the page says '(c) Exitech 2005 Camart'.

The results obtained are:

The screenshot shows the 'Camart Statistic Service - Requested Logs' page. It displays a table of logs with columns: Log Id, AXOID, Version, AXCID, and DIS. There are four rows of log data. Below the table is a '3 LOCATION chart' section. The chart shows a table with columns: Pos, LOCATION, and Count. The data is as follows:

Pos	LOCATION	Count
1	Area1_NorthAmerica	2
2	Area2_Europe	1
3		1

The footer of the page says '(c) Exitech 2005-2006'.

3.2 CAMART and Administrative Information Integrator

3.2.1 Introduction

The role of Core Accounting manager and Reporting Tool (CAMART) is strictly bound with database for logs (provided by AXCS) since it has to collect information regarding the B2B. For example to have a table of Business counterparts that distribute, integrate, etc. AXMEDIS Objects. They should have a contract with the AXMEDIS Factory that host the AXMEDIS database (AXDB). This information is useful to interpreting the Action Log Information coming from the AXCS databases, when available in the AXCS.

The information will be collected on schedule and CAMART will act as a client of the AXCS Reporting Web Service.

AXMEDIS system has to be scalable and therefore we will have to deal with the fact that some installation can have AXDB, AXCS and other supporting tools on different machines, while others can be less distributed due to a lesser need for speed or storage capacity.

The core accounting manager is a sort of Client side of the bridge between the AXDB and the AXCS databases in order to allow AXCS to be independent by the database. The server side in the AXCS is the Web Service: AXMEDIS Reporting Web Service. The CAMART can be interpreted as a part of the AXMEDIS Database Interface, since is the part of the system that allows writing data related to Action-Logs into the DB.

Administrative Information integrator is a critical part of the AXMEDIS system since it is the real bridge between the AXMEDIS world and the world of company's CMS and CRM for taking in account administrative and legal aspects (such reclaim for payment not done and so on).

This component has also a double face since it can operate in a dual manner: used for polling information from AXMEDIS system when needed by distributor for example, or user for pushing information in the CMS as soon as they are available for example in the case of collecting societies.

The operating mode is determined by accounting people during the installation/configuration of the system when it will be established whose fields have to be exported from the DB to the CMS and the frequency of exporting. When a frequency is set, the Administrative Information Integrator will work in push mode, pushing information in the CMS import area, otherwise it operates in polling mode by starting the update in the CMS by a link to a web page.

The principal specification arises, as always, from requirements. Administrative Information Integrator has to:

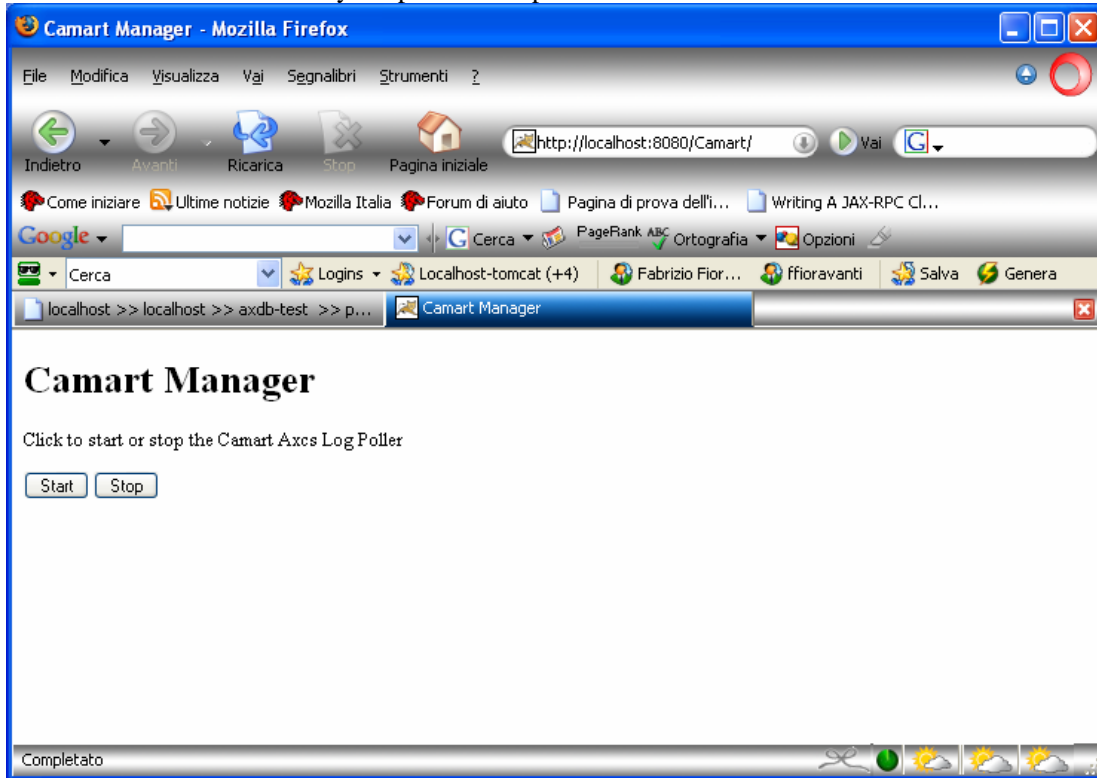
- ***interface with different CMS technologies***: this means that administrative information integrator has to export its reports in a portable format such as an XML that will be defined in the detailed specification. The generated XML can then be parsed and transformed by the way of standard mechanisms such as XSL transformation;
- ***store administrative information into the Content Provider database***: this operation will be possible in two ways, or by preparing a file to be put in the import area on the content provider database or by interfacing with the services offered by the CMS of the Content provider by standard mechanisms such as web services or other remote interface available to the Administrative Information Integrator;
- ***communicate with the AXDB to get administrative information related to a specific Content Provider***: this is the minimum security requirement that have to be established in order to be sure to distribute information to entitled persons only.
- ***guarantee privacy of sensitive data via protection mechanisms***: apart from what has already been stated, if a network connection is necessary to transfer data and the CMS of the Content Provider permits a connection on a secure channel, then the data will be sent encrypted.

3.2.2 The prototype

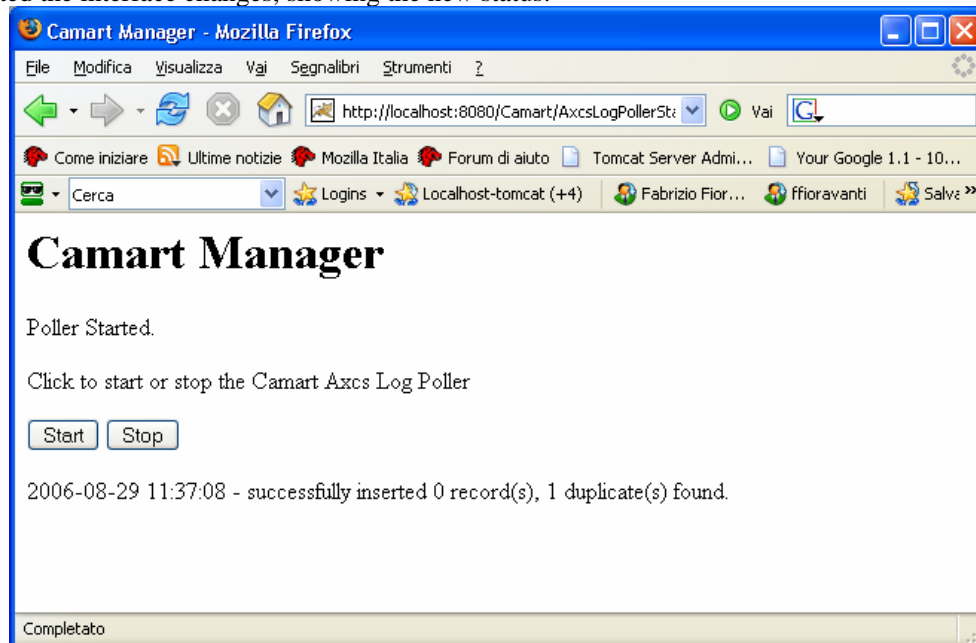
Regarding the Administrative Information Integrator (AII), the interface is quite raw, but the functionalities offered are quite stable apart some minor refinement already specified above.

This module is comprised of a daemon that collects info from AXCS and stores logs locally on AXDB for future reuse from AII.

The interface of this daemon is very simple and is reported below:

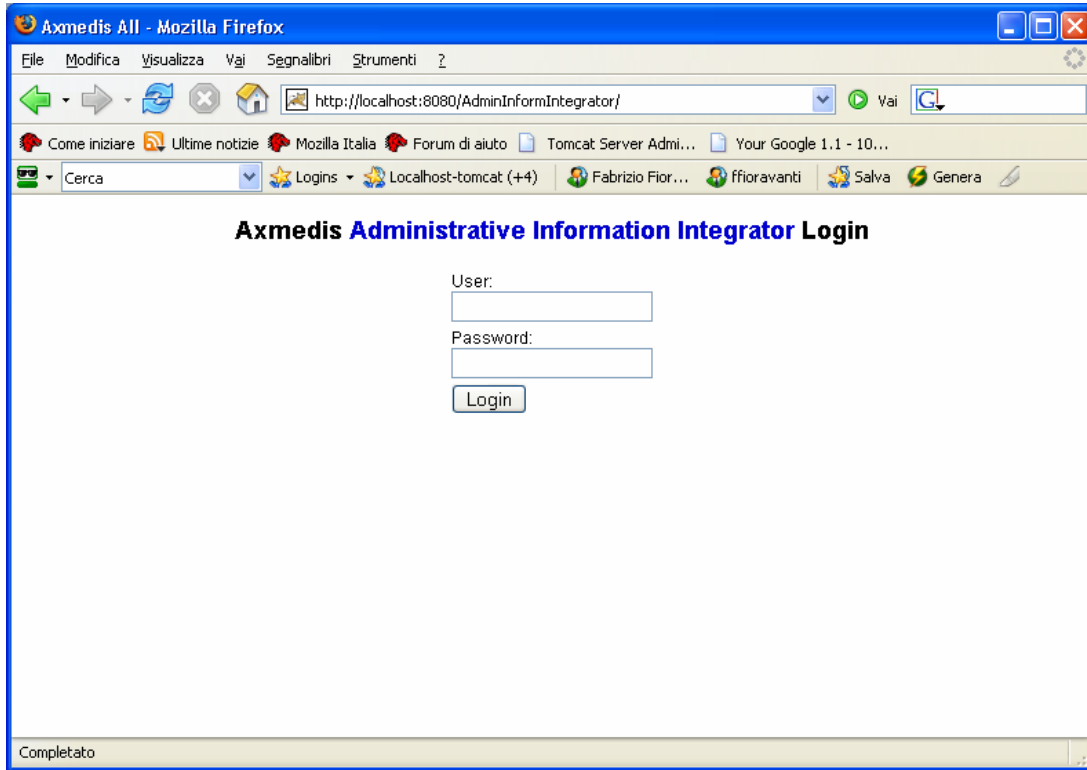


Once started the interface changes, showing the new status:



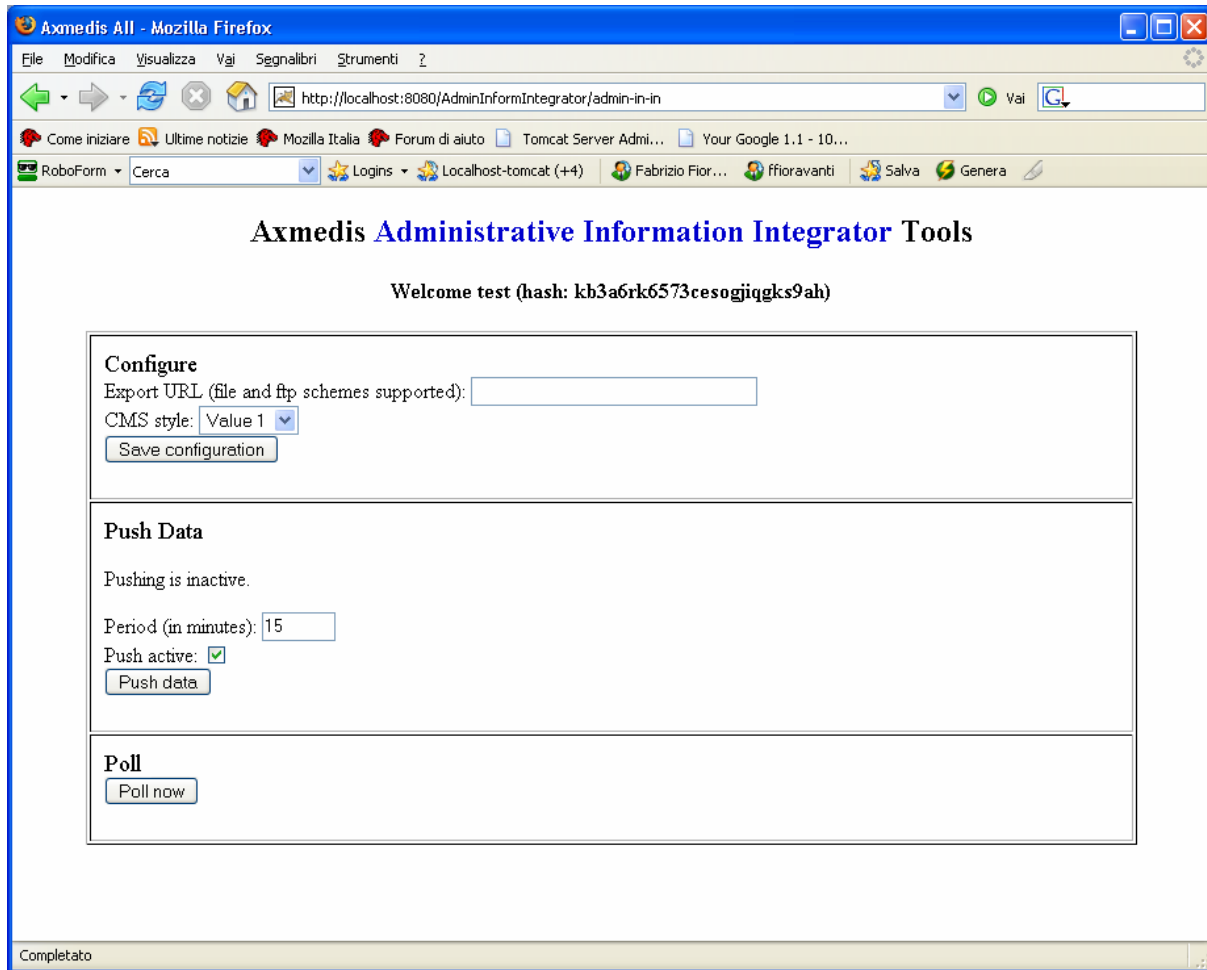
This thread runs in background and collects locally the remote logs provided by AXCS, showing also information on the logs inserted in the last poll with the amount of logs that were already present.

Regarding the Administrative Information Integrator seen from the user perspective, the authentication screen is the first page that appears:



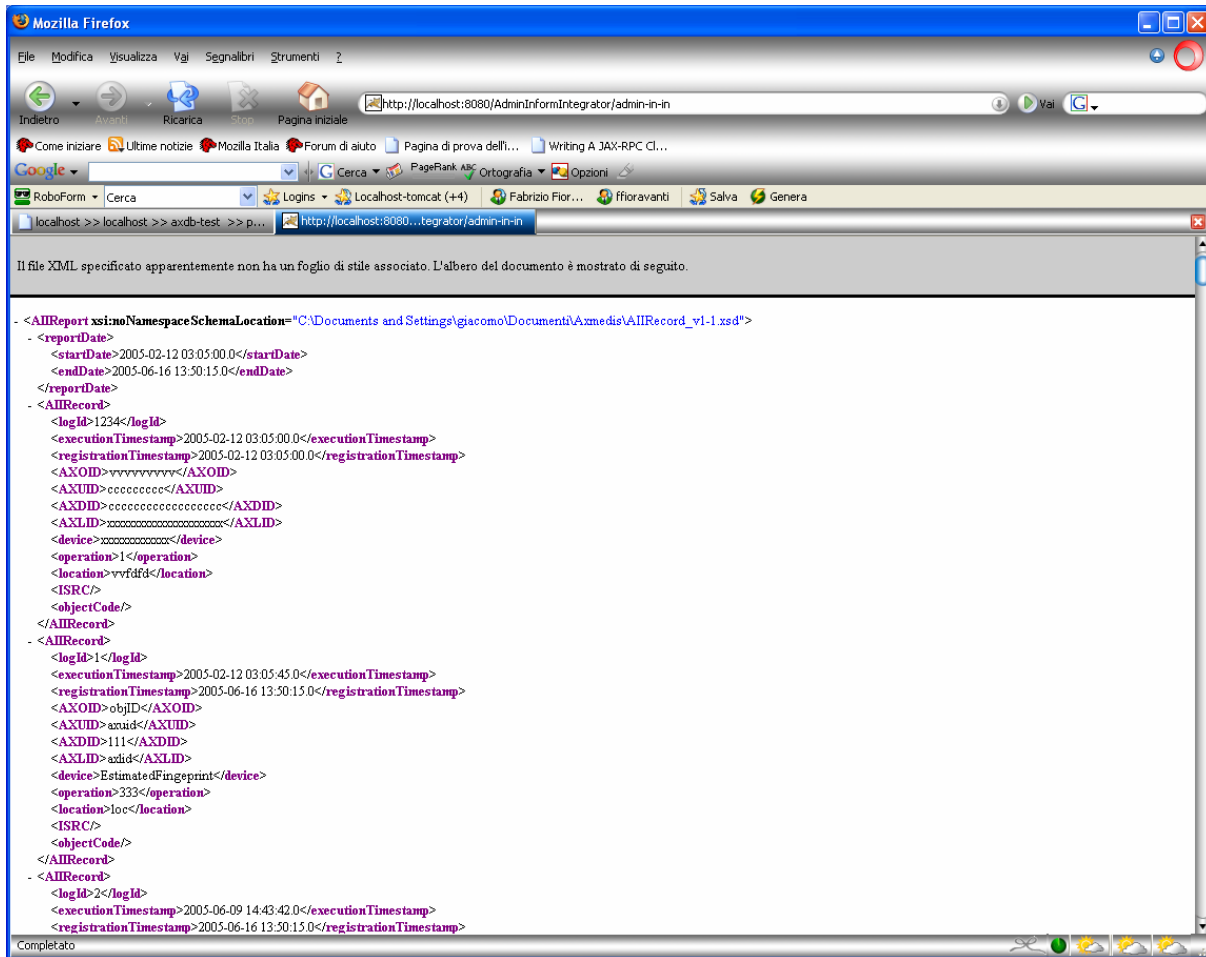
and the User after a login with authentication and authorization phase is bring to the following User Interface that allows to:

- Configure the AII for the logged user, giving him the chance to select a CMS style and the FTP URL where the logs will be transferred after each data pushing
- Configure the time intervals in minutes between two pushing of data and enabling or disabling pushing
- Poll raw data or formatted data in the case the user has already saved a configuration

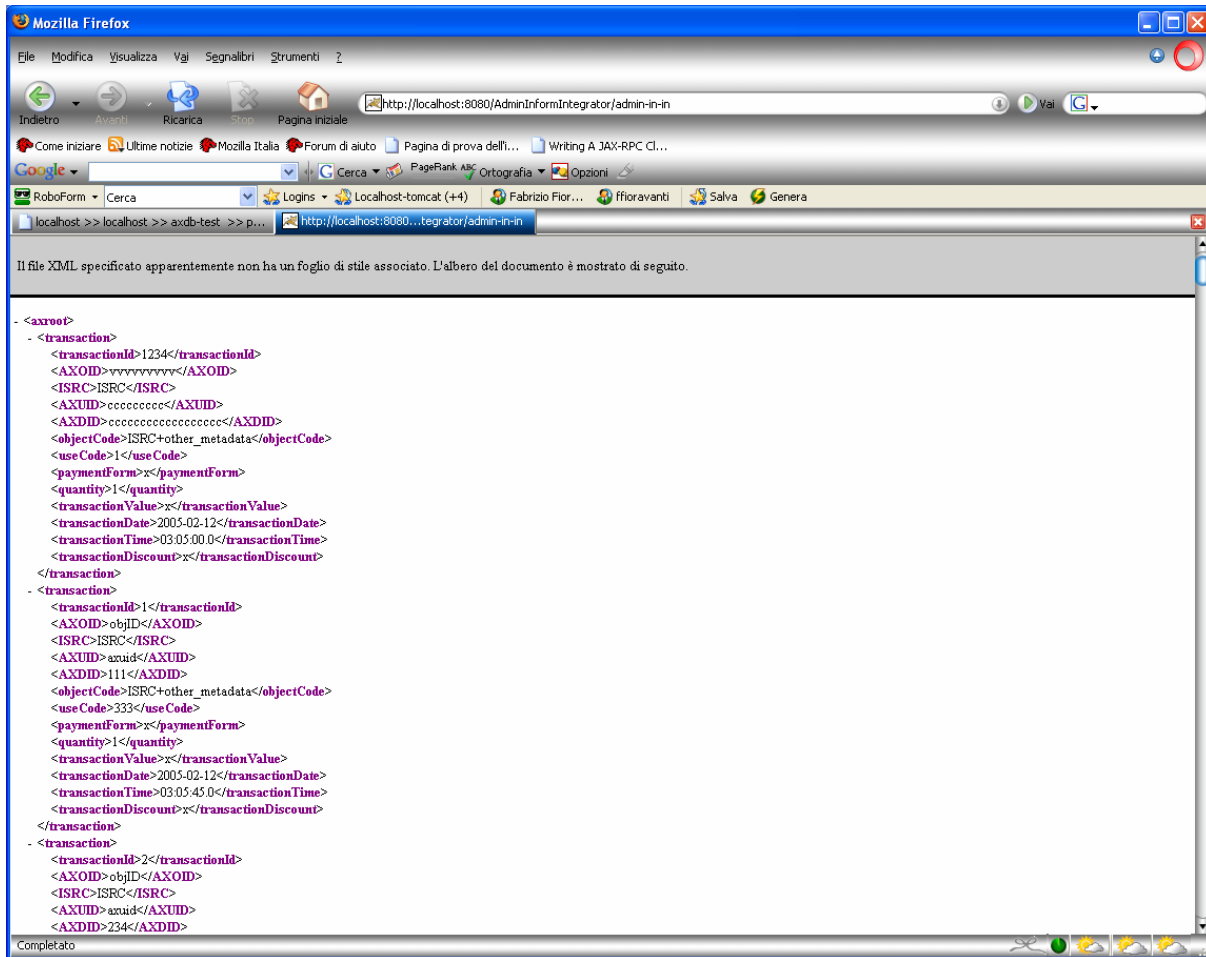


The system will push data in the configured ftp URL at the predefined interval in the format specified in DE 3.1.2.2.15 after getting them from the remote AXCS. In the case of polling the data in XML (formatted or unformatted) will be presented on the screen allowing the user to manually save the file for his own purposes and allowing him to use it immediately. The system tracks the time at which a log has been requested in order to avoid multiple gathering of the same data.

If the user has not chosen the configuration the general format is displayed, if the Poll now button is pressed:



While after the configuration the format defined for CMS1 will appear:



4 Bibliography

DE 9.1.1 Specification of CMS integration and feedback

DE 9.1.2 Mock-up for CMS Integration and Feedback

DE3-1-2-2-15 Specification of AXMEDIS Accounting and Reporting

Video Demonstration: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2131