Business Link Kent Web Infrastructure

Implementors Manual

Doug Winter

8th April 2004

ICP Europe PLC

Revision: 1.1.1.1

Contents

1. Introduction

1.1. Audience

This document is designed for implementors of web sites for Business Link Kent and its partners. This document provides all the information required to use the suite of components that provide the web infrastructure for cooperating business support agencies in Kent.

Together these components provide for the implementor:

- Access to a centralised Contact Directory
- Access to a centralised Company Directory
- Simplified authentication against this contact directory
- Management and storage of user permissions
- Event logging and transmission to a central event storage and processing facility

This greatly simplifies the process of developing and deploying new web components, on any platform. A suite of COM components are provided to simplify implementation on Microsoft Windows platforms.

1.2. The Contact Directory - Aiakos

Managing user records is a problem central to the online experience for organisations with a significant online interaction with their customers and partners. Traditionally this would have been handled with a Customer Relationship Management system, however these are generally data-centric, and designed for call centres and marketing reports.

With the advent of the World Wide Web has come a requirement for customers to be able to access and manage their own data online. Similarly these data are often used to customise the online experience of the user, for authentication of the user for access to additional site services. Good websites, in fact, are very user-centric, and so may need information on users at any time.

An infrastructure that provides a user directory, plus all the attendant machinery for synchronisation, password recover etc. is therefore a major requirement.

However, another force is also at work. Organisations need to be able to deploy special-purpose websites, for particular marketing campaigns or run by specific business divisions. Restricting business divisions to use specific vendors or technologies leads to a lock-in that is often expensive and problematic.

Business divisions need to be able to select suppliers of online services based on their specific needs, and then provide them with the tools they need to ensure the services they deliver meet these diverse business needs.

Aiakos comprises a number of components that together solve this problem. At the core of Aiakos is an LDAP Directory that holds the user records. This LDAP Directory is accessible directly by vendor delivered sites, if necessary, providing support for any user-management requirements they may have.

Additionally, a registration and login interface is provided to be used by **all sites**. This means that:

1. The sites do not need to provide a login and registration interface of their own, reducing cost and complexity.
2. The single login and registration interface can be managed to ensure the user experience is appropriate. Changes to the data capture can be made in one place only.
3. User accounts are synchronised across all websites.
4. The single user repository can then be synchronised with a central internal CRM system for data-cleaning and data consistency across all communication.

1.3. Embrace

Embrace is the name given to Silver Bear's CRM system integration technology. This retrieves contact, company and event data from web sites and synchronises them with internal CRM system.

Data relating to companies and contacts may also be pushed out to the company and contact directory, reflecting changes in data initiated internally, or by telephone.

2. Architecture

2.1. Components

This manual covers the following components:

Zope/Plone LDAP User Folder Support
: These Open Source Zope Products by Jens Vagelpohl are available from http://www.dataflake.org/software. They provide complete LDAP support for Plone portals.

Aiakos Authentication Server
: Provides a central contact directory, with authentication information, registration login and distributed authentication. This automatically configures and installs LDAPUserFolder and CMFLDAP portal components in the portal.

Aiakos Authentication Client
: The libraries necessary to use the AS within a Microsoft IIS ASP site.

UNIX Embrace Client
: Communicates with the central Embrace system for the logging of events.

Windows Embrace Client
: Communicates with the central Embrace system for the logging of events.

2.2. Contact Directory Overview

Aiakos provides these interfaces to participating sites:

Simple Authentication Interface (SAI)
: Instead of providing registration and login screens, Participating Sites link to the SAI. The SAI provides all registration and login functionality.

  Once the user has successfully completed their registration activity, they are directed back to the site's provided Landing Page .

Library Interface
: A Library is provided for Windows systems that provides access to user data and permissions.

LDAP v3 Interface
: Participating Sites can also access the LDAP server directly, if required, to obtain advanced functionality not available through the provided components.

2.3. Embrace Overview

Installed along with the Integration Library is a MySQL database that holds event data. You should speak to Business Link Kent to agree which events that occur on your website are interesting to Business Link Kent.

Then in your website code, wherever these events occur, you call the event logging function to notify embrace that this event has occurred. This is logged in the attached database.

A seperate process, either a UNIX Daemon or a Windows Service depending on platform, will listen for connections from Business Link Kent and will transfer these events to them.

3. Contact Directory

Remote Zope/Plone portals that wish to use the central contact directory authentication should install the AiakosClient, LDAPUserFolder and CMFLDAP Zope products.

Because Plone transparently supports LDAP for all membership related activity, the AiakosClient product only:

1. Installs and configures the LDAPUserFolder and CMFLDAP products.
2. Replaces join_form and login_form with pages that redirect to the configured Aiakos Server
3. Provides a landing page that decodes the UAP and logs the user in on the portal.

Once the user is logged in, all portal features work as normal.

3.1. Collaborations Walkthrough

3.1.1. Simple Authentication

The Authentication Server (AS) provides a 'Simple Authentication Interface'. Using this interface, a Participating Site is able to authenticate a user against the Business Link Kent Contact Directory without accessing the directory directly.

The Authentication Server provides all login and registration screens. The Participating Site and Authentication Server **never** communicate directly during the process of Simple Authentication.

The Participating Site is still required to provide session maintenance, however it can trust the Authentication Server completely as to the identity of the user.

The way the components collaborate is illustrated below.

Step 1

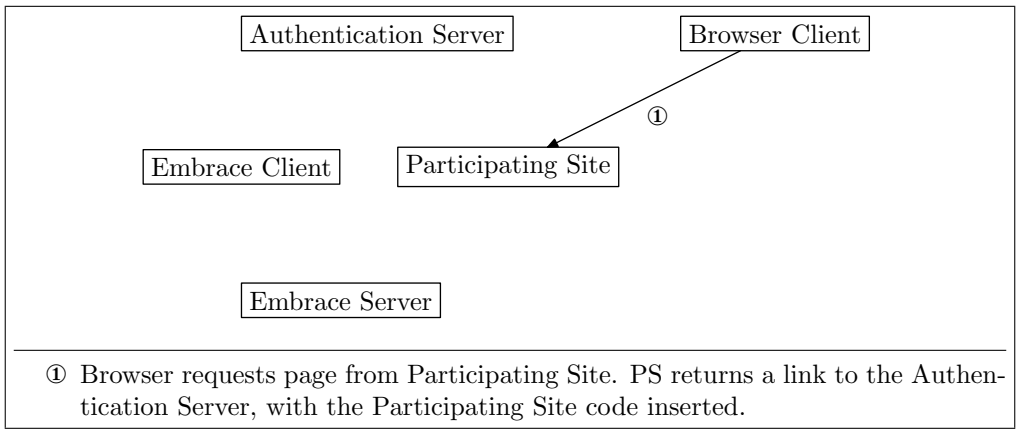
The user navigates to the Participating Site. Where the Participating Site would normally provide a set of login and registration screens, it instead provides a link to an interstitial page, explaining that the site uses the Kent Business Passport system. Marketing collatoral for these pages can be obtained from Business Link Kent.

The page contains a link to the passport login and registration system, with a site code inserted that refers to the participating site that is issuing the request.

For example, the link might resemble:

```
http://www.businesslinkkent.com/aiakos_login_director?id=medway
```

This is illustrated in Figure 3.1 on this page.



① Browser requests page from Participating Site. PS returns a link to the Authentication Server, with the Participating Site code inserted.

Figure 3.1: Simple Authentication Step 1

Step 2

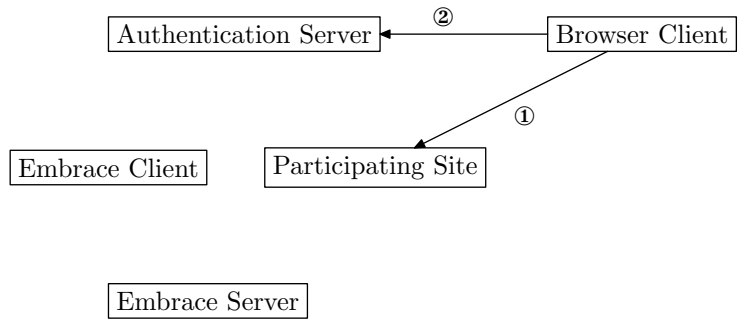
The Authentication Server presents all the user interface required for login and registration. This includes password recovery screens, as well as options for the user to change their preferences, password and so forth. This is illustrated in Figure 3.2 on the next page.

Once the user has navigated login and possibly registration, they are given an option to go back to the site from whence they came. This link is to the landing page specified by you when your site was registered with the authentication server.

Step 3

Embedded in the link back to the PS is an encrypted UAP that contains details on a successful login session, most notable the username . This is illustrated in Figure 3.3.

The UAP is encrypted using the Blowfish algorithm, which is a symmetric cipher. The key used to encrypt this UAP is unique to your site, but is shared with the Authentication Server. If the UAP can be decrypted with your key, then you can assume the UAP has been encrypted by the AS.



① Browser requests page from Participating Site. PS returns a link to the Authentication Server, with the Participating Site code inserted.
② Browser requests entry page on Authentication Server. This presents login and registration options. User completes login/register operation.

Figure 3.2: Simple Authentication Step 2

At this point you can store the user's username in the user's session, and you can trust this for later transactions with the user.

3.1.2. Permissions Management

The Authentication Server also stores permissions strings, that are accessible to the Participating Site. These strings are not interpreted by the authentication server in any manner, so the PS can use them as it wishes. Some examples are shown later. This is illustrated in Figure 3.4.

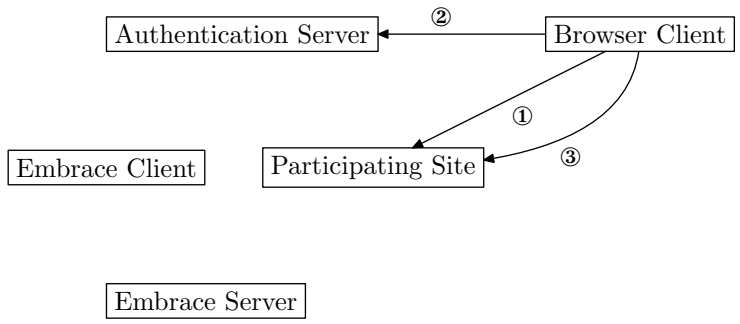
3.1.3. Event Recording

Step 1

Using the provided API, the Participating Site can record any interesting event. This is stored in a MySQL database that is supplied with the Business Link Kent software libraries. This is illustrated in Figure 3.5.

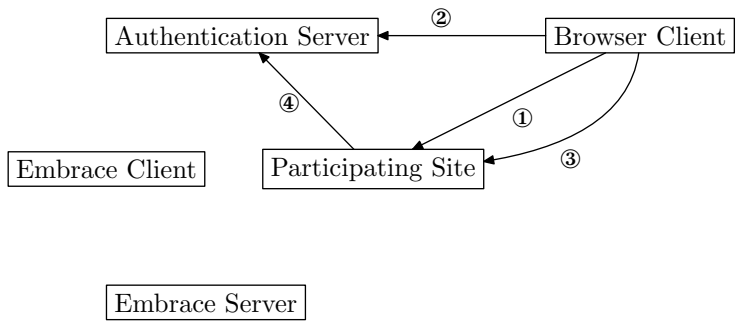
Step 2

Periodically the Embrace Server will connect to each Participating Site in turn, and request the list of new events. These are sent to the Embrace Server, acknowledged, and then removed from the database. See Figure 3.6.



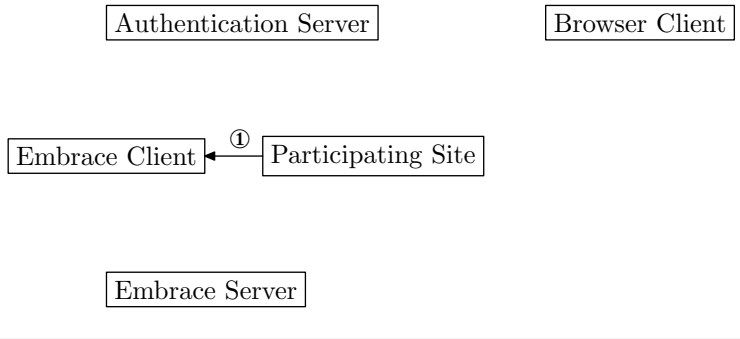
① Browser requests page from Participating Site. PS returns a link to the Authentication Server, with the Participating Site code inserted.
② Browser requests entry page on Authentication Server. This presents login and registration options. User completes login/register operation.
③ The Authentication Server provides a link back to the PS Landing Page. The user clicks the link, and is taken back to the PS, with a UAP in the URL.

Figure 3.3: Simple Authentication Step 3



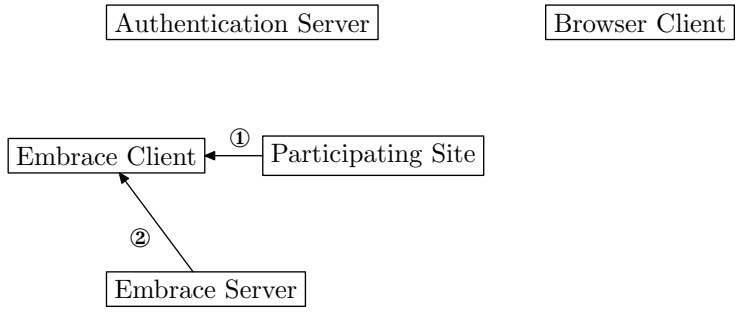
④ The participating site, now it has a valid username, may request that the AS set a permission string for the user for this site. Alternatively, it may request a permission string for the site, or enumerate all permissions.

Figure 3.4: Permissions Management



① The participating site informs the embrace client that an interesting event has occurred. This is stored in the local MySQL database.

Figure 3.5: Event Recording Step 1



① The participating site informs the embrace client that an interesting event has occurred. This is stored in the local MySQL database.
② Regularly, the Embrace Server requests all new events from the Embrace Client.

Figure 3.6: Event Recording Step 2

4. Business Link Kent Windows Package

If you are implementing on a Microsoft Windows platform you will have been provided with an installation package containing all components necessary for implementation.

4.1. Package Contents

The package contains a number of components written in the Python programming language, along with all required supporting software.

The list of third party software, with the original download locations, is provided here. Note that **all of this software is provided to you**. The download locations are provided for reference only.

Under the terms of the GNU General Public License, where it applies, source codes are available from wherever you obtained binaries.

4.1.1. Third Party Software

Python-2.3.3.exe
: Python 2.3.3 interpreter and standard library package.

  Distributed under the Python Software Foundation License

  Download from www.python.org

win32all-163.exe
: Python Windows extensions by Mark Hammond.

  Distributed under the Python Software Foundation License

  Download from Mark's site.

MySQL-python.exe-0.9.2.win32-py2.3.exe
: Python MySQL Libraries.

  Distributed under both the Python Software Foundation License and the GNU General Public License

  Download from Sourceforge

python-ldap-2.0.0pre14.win32-py2.3.exe
: Python LDAP Libraries. Linked against the OpenLDAP client libraries.

  Distributed under the GNU General Public License

  Binaries provided by Mauro Cicognini. Sources are available from Sourceforge.

pycrypto-1.9a6.win32-py2.3.exe
: Python Cryptography Toolkit.

  Distributed under the Python Software Foundation License

  Binaries provided by ICP Europe. Sources are available from Sourceforge

  Note that the only cipher used is Blowfish, which is patent- and copyright-free.

mysql-4.0.18-win.zip
: MySQL 4.0 database server.

  Distributed under the GNU General Public License

  Download from the MySQL site

4.1.2. Installation

On installation all of the above software is installed into a location you specify.

TO BE CONFIRMED

4.1.3. Configuration

On installation, you will be asked for all required configuration. This configuration is stored in the Windows Registry under the following keys:

- `\HKLM\Software\Python\PythonCore\2.3\PythonPath\icpembrace`

  The full path to the embrace installation. This is used by the Windows Service to locate it's coprocess daemon.

- `\HKLM\Software\icpeurope\embrace\DATABASE`

  The name of the mysql database to use.

- `\HKLM\Software\icpeurope\embrace\DATABASE_USERNAME`

  The username with which to connect to the database.

- `\HKLM\Software\icpeurope\embrace\DATABASE_PASSWORD`

  The password with which to connect to the database.

- `\HKLM\Software\icpeurope\embrace\HTTP\_PORT`

  The port on which to listen for HTTP connections from the Embrace Server.

4.1.4. Running the Package

For the full functionality of the package to be available, both the MySQL Database Server and the *EmbraceWindowsService* Windows Service must be running.

A. Windows Library API Reference

The API is provided by a collection of COM Components that can be accessed from within Active Server Pages.

A.1. Aiakos.UserAuthenticationPackage

A.1.1. Synopsis

```
set uap = Server.CreateObject("Aiakos.UserAuthenticationPackage")
encrypted = Request.QueryString("uap")
uap.decrypt "c09a1d3fc6d4e464", encrypted
username = uap.getProperty("username")
```

A.1.2. Description

UserAuthenticationPackage handles unpacking, decryption and parsing of the UAP sent by the Authentication Server.

When your site is registered with the Authentication Server you will be provided with a hexadecimal encryption key. This should be used to decrypt the UAP sent by the AS.

When the UAP is decrypted, a number of properties will be available, describing the authentication event encapsulated in the UAP.

Methods

decrypt(key, uap)
: If successful, the UAP object is initialised with the properties from the UAP. If unsuccessful, it throws an exception.

getProperty(name)
: Return the value of the listed property. This should only be called *after* `decrypt`.

Available properties are:

version The version of the UAP format.

system A unique identifier for the system that created this UAP.

audit_token
: A unique token for the authentication event.

timestamp
: The date and time that the event took place, in seconds since the start of the epoch.

username
: The name of the user who authenticated.

auto_login
: Whether an auto-login cookie should be generated.

event The name of the authentication event.

Generally you will only need to consult `username` and `auto_login`

getPermission(permission)
: With the UAP is also sent the block of permissions relevant to this site. This is a shortcut for using the Aiakos.Permissions functions, if you immediately need to check someone's permissions on login.

Provided with the name of a permission, this returns the permission string.

A.2. Aiakos.Permissions

A.2.1. Synopsis

```
set ap = Server.CreateObject("Aiakos.Permissions")
ap.setPermission "MySite", username, "MyService", "MyRole"
permission = ap.getPermission("MySite", username, "MyService)
```

A.2.2. Description

Your server may have any number of services, each of which may have any number of permission strings. These strings can be used in any way you wish. LDAP authentication information is stored in the registry and retrieved transparently by the COM object.

Note that you aren't restricted to accessing permissions on your own site - to allow cooperation between sites you have access to all permission values.

A.2.3. Methods

setPermission(site, username, service, value)
: Set the specified site and service combination to the specified value for the given username.

getPermission(site, username, service)
: Retrieve any previously set permission value for the given site, username, service combination.

A.2.4. Notes

Internally to the LDAP server, the permissions are stored as hash-delimited fields:

```
site#service#username
```

This means the hash character cannot be used in any of the fields.

A.3. Embrace.Adaptor

A.3.1. Synopsis

```
set ea = Server.CreateObject("EmbraceConnector.embraceadaptor")
id = ea.new_event('createcustomer', 'customer')
ea.add_event_param 'customerid', 'spreston'
ea.add_event_param 'title', 'MyTitle'
id = ea.record_event()
report = ea.events_status_report()
```

A.3.2. Description

EmbraceConnector allows your site to record events to be forwarded to Silverbear's EMBRACE system.

The events are recorded in a MySQL database and passed in batches following periodic HTTP requests made by the EMBRACE system.

Events are marked as acknowledged once successfully processed by EMBRACE.

The architecture supports ongoing creation of new event types as agreed with the EMBRACE system. Each event has an event type identifier and a corresponding data type identifier. A set of parameters in the form of name/value pairs encapsulates all the event information.

A.3.3. Methods

new_event(eventType, dataType)
: Initialise a new event to be recorded. This event is stored internally to the created `EmbraceConnector.embraceadaptor`

add_event_param(paramName, paramValue)
: Initialise an event parameter.

record_event()
: Writes the current event to the database.

event_status_report()
: Counts the number of events with each possible status from pending, ack and failed, and returns a short report as a string. The report resembles ”pending=2;ack=0;failed=1”

B. User Attributes

B.1. Notes

B.1.1. LDAP Assignments

Where appropriate, attributes have been selected from the standard LDAP schemas CORE, COSINE and inetOrgPerson. Any attributes specific to this application are prefixed with "aiakos". These attributes are located under the Enterprise Number assigned to Business Link Kent, 19589.

Within this namespace, entries are allocated as:

1.3.6.1.4.1.19589.1
: Aiakos LDAP Elements

1.3.6.1.4.1.19589.1.1
: Aiakos LDAP AttributeTypes

1.3.6.1.4.1.19589.1.2
: Aiakos LDAP ObjectClasses

B.1.2. Dates

The Datelike attributes are stored as integer seconds since the Epoch (00:00:00 UTC, January 1, 1970). This provides compatibility with the C standard library, and all derivative languages and libraries (i.e. most of them). The standard C function `strftime` or equivalent should be used to render dates, if required.

B.2. The Attributes

title
: The Personal Title attribute type specifies a personal title for a person. Examples of personal titles are "Ms", "Dr", "Prof" and "Rev".

  Attribute: personalTitle
  Schema: COSINE
  Syntax: caseIgnoreStringSyntax
  Equality: caseIgnoreMatch

firstname
: First name(s) for which the entity is known by.

  Attribute: givenName
  Schema: RFC2256
  Syntax: DirectoryString
  Equality: caseIgnoreMatch

surname
: Last (family) name(s) for which the entity is known by.

  Attribute: sn
  Schema: RFC2256
  Syntax: DirectoryString
  Equality: caseIgnoreMatch

email
: RFC822 Mailbox.

  Attribute: mail
  Schema: RFC1274
  Syntax: IA5String
  Equality: caseIgnoreIA5Match

id
: User Identifier. This is the unique username of the user, used to uniquely identify them in the Contact Directory.

  Attribute: uid
  Schema: RFC1274
  Syntax: DirectoryString
  Equality: caseIgnoreMatch

password
: The user's password.

  Attribute: userPassword
  Schema: RFC2256/RFC2307
  Syntax: OctetString
  Equality: octetStringMatch

position
: The user's position within their organisation, such as "Vice President".

  Attribute: title
  Schema: RFC2256
  Syntax: DirectoryString
  Equality: caseIgnoreMatch

company_name
: The organization this user is associated with.

  Attribute: organizationName, o
  Schema: RFC2256
  Syntax: DirectoryString
  Equality: caseIgnoreMatch

address(1,2,3)
: The three lines of the address. The standard LDAP Attribute postalAddress is not used, to map more closely onto the Business Link Kent internal CRM system.

  Attribute: aiakosAddressLine(1,2,3)
  Schema: Aiakos
  Syntax: DirectoryString
  Equality: caseIgnoreMatch

postcode
: The user's postcode at the address they have provided.

  Attribute: postalCode
  Schema: RFC2256
  Syntax: DirectoryString
  Equality: caseIgnoreMatch

telephone
: The user's telephone number at the address they have provided.

  Attribute: telephoneNumber
  Schema: RFC2256
  Syntax: telephoneNumberSyntax
  Equality: telephoneNumberMatch

salutation
: The preferred way in which this user likes to be addressed, i.e. 'Dear Sir' or 'Hi'.

  Attribute: aiakosSalutation
  Schema: Aiakos
  Syntax: DirectoryString
  Equality: caseIgnoreMatch

justification
: A text string describing why the user wishes to have a Kent Business Passport.

  Attribute: aiakosJustification
  Schema: Aiakos
  Syntax: DirectoryString
  Equality: caseIgnoreMatch

registration_date
: The date at which this user was registered. See 'Dates' above.

  Attribute: aiakosRegistrationDate
  Schema: Aiakos
  Syntax: integer
  Equality: integerMatch

email_format

Attribute: aiakosEmailFormat
Schema: Aiakos
Syntax:
Equality:

referring_site

The name of the site that originally referred this user when they registered.

Attribute: aiakosReferringSite
Schema: Aiakos
Syntax: IA5String
Equality: caseExactIA5Match

Index