2AB, Inc. July 2005

# Harmony

The 2AB Newsletter

## New Releases of orb2 for Java, C and C++

### June 2005 was a busy month at 2AB!

Special points of interest:

- Identity Manager for use with CSIv2
- Secure Naming Service
- Trader and Naming Service Management Consoles
- Role-Based access control
- Wide range of operating platforms

2AB continues to demonstrate their commitment to trusted middleware by adding new integration, security and management features in our CORBA products. orb2 for Java version 4.2 and orb2 for C/C++ version 3.7 were released in June.

These releases enhance the security and management features of orb2 by providing:

- New security options for protecting orb2's Naming Service using CSIv2.
- New Naming Service Console for remote administration of OMG / JNDI naming services.
- New Trader Service Management Console for remote browsing, administration and monitoring of orb2 Trader services.
- An Identity Manager Administrative tool and Security Center service to provide user, group and role-based access control for secured Naming Services and Java CORBA® servers.
- Support for new addressing and naming formats enable enhanced security for services when accessed via Internet applications and/or VPNs.


The Naming Console leverages new role-based identity management capabilities of orb2 for protecting access to secured naming services. The Trader Console will also leverage this in future releases.

2AB is committed to continually enhancing support for programming models, operating platforms and wire protocols while maintaining a commitment to interoperability standards.

In this issue:



## Developer's Corner

### Distributed JAAS: Using jLock with RMI


The 2AB whitepaper "Using jLock's Java Authentication and Authorization Service (JAAS) for Application-Level Security" provides a detailed overview of how to use jLock's scalable JAAS implementation within a single Java program. Many applications, however, use distributed technologies such as Java's Remote Method Invocation (RMI), Java Messaging Service (JMS) or CORBA to access business logic in remote servers. The server process may need to restrict access or filter information based upon the identity of an authenticated user. This requires that the server process have access to the credentials of the user authenticated by the client process. It is rare for a distribution technology to support the transparent delegation of user identity and/or credentials to the server process. This is a task that typically must be managed by the business application. jLock extends the JAAS permission paradigm to distributed environments and allows applications to transmit the authenticated identity information securely (even if the environment does not support encrypted data transmission).

*(Continued on page 3)*

# Sneak Preview

## orb2 Manageability & Analysis tools

orb2 development is focused on two major initiatives.

- Manageability—Tools for analysis of running CORBA environments
- Security—Integrated fine-grain access control and application-level security features

In this section we provide a sneak preview of how the Management consoles for the Trader and Naming Services are being enhanced to allow runtime tracing and response time monitoring of CORBA services.

Services are enabled for monitoring with the command-line argument (-ORBmonitor true) or by programmatically setting the corresponding orb property (Figure 2). Once a service has been enabled for monitoring, the Naming or Trader Console can be used to activate/deactivate the collection of server information. The console is also used to set client-side polling options (Figure 1). A server that is being pinged will show the status on the monitor view as shown in Figure 3 below. If tracing is also activated, information regarding the operations that have been invoked since the server-side tracing was activated is displayed. A menu option is available to reset statistics so that snapshots of server performance can be viewed.

We have shown the Trader Console here, but the features are also incorporated in the Naming Console for users who prefer a Naming Service to a Trader for location services.

orb2 for Java version 5 and orb2 for C/C++ version 4 are the release levels slated to include monitoring support in their services. The new monitor view in the Trader and Naming console will begin shipping with both orbs as soon as orb2 for Java version 5 is released.

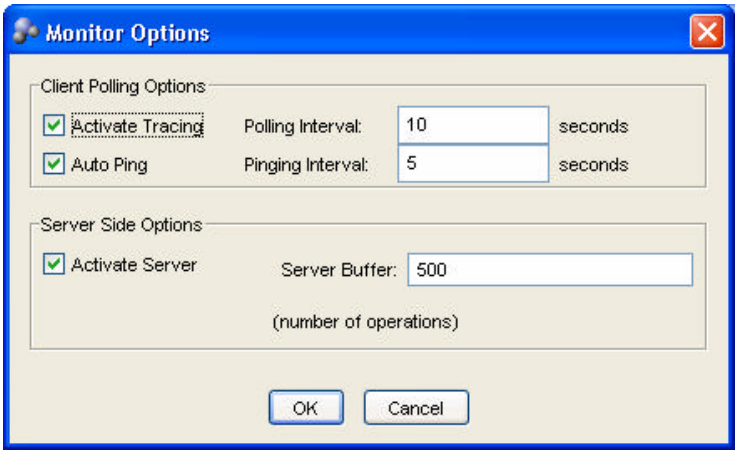


**Figure 1:**
**Monitoring Options include auto ping & operation tracing**

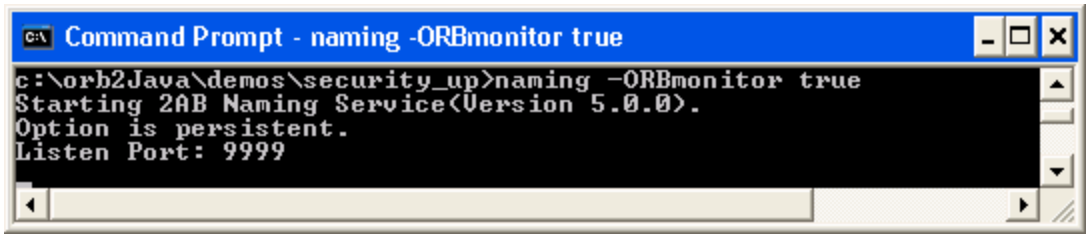


**Figure 2: Starting the Naming Service enabled for monitoring**

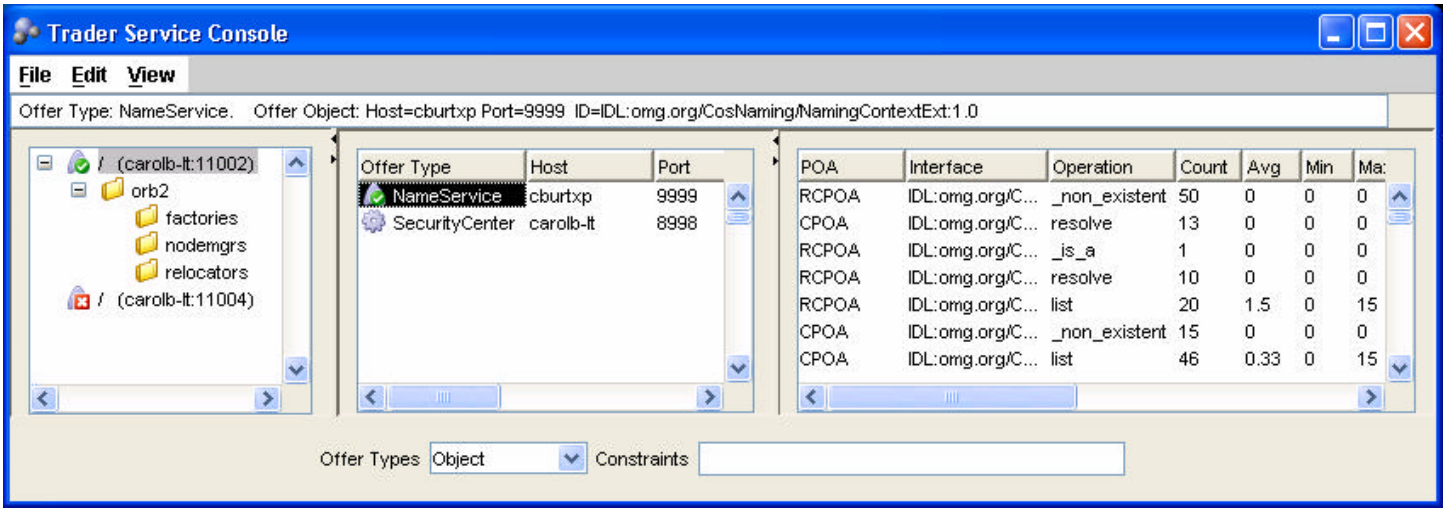


**Figure 3: The monitor "view" of the Trader Console displays server status and statistics on invoked operations**

jLock™ SCALABLE JAAS

*(JAAS Continued from page 1)*

2AB's jLock implementation of JAAS allows the client to obtain an identity token that can be passed to the server and used to re-establish the login context of the authenticated user. This ensures that permissions are consistent across the distributed environment. The identity token is encrypted and can be safely passed between processes (using RMI, JMS, SOAP, CORBA, FTP or any other messaging infrastructure) without the overhead of encrypting the entire message payload if desired. An iLock token can only be obtained after a user has authenticated and has a short lifespan (a few minutes). Passing an identity token to a server is more secure and more efficient than re-authenticating on the server.

We demonstrate this technique with a simple human resources RMI server that supports an Interface that returns a Compensation object. One of the attributes of the Compensation object is "salary." The client authenticates the user using the JAAS API (for a complete explanation of JAAS authentication, see the white paper referenced above). The two JAAS methods an application invokes to use a JAAS authenticator are shown in bold font in Figure 1.

The client then obtains an identity token and passes that token to the RMI server when requesting compensation information for an employee (identified by "empl_id"). See Figure 2.

In this example, the RMI server uses a filtering technique. The requested Compensation object is returned but filters out the salary if the user that is logged in is not authorized to see that information. The other attributes of compensation (i.e. the job code of the employee) are available to the user. The implementation of the remote object is shown in Figures 3 and 4. First, the token is used to set the context for the permission check to that of the remote user.

The server then makes the check to determine if access to Salary is allowed for the user. If the user is allowed to see Salary information, then the salary amount will remain in the Compensation object that is being returned to the client. If not, then salary will be filtered by placing a "Not Authorized" message in the salary attribute.

jLock allows the JAAS APIs to be useful for both local and remote Java environments. If your environment also spans other programming languages, talk to 2AB about companion products that allow your application security to remain consistent even when your application spans services written in different programming languages.

Our goal is to make application security simple to understand and easy to implement.

For more information, please visit our Web site at http://www.2ab.comor e-mail info@2ab.com.

```
LoginContext lc = null;
try {
    lc = new LoginContext("JaasDemo",
                new DialogCallbackHandlerUP());
} catch (LoginException le) {
    …
} catch (SecurityException se) {
        …
}
try {
    lc.login();
} catch (LoginException le) {
    System.out.println("\nAuthentication failed:");
    …
}
// Authentication Succeeded!
System.out.println("\nHello iLock World!\n");
```

**Figure 1: JAAS RMI Client Authenticating User**

```
byte[] token = com.twoab.jaas.Context.getToken();

try {
  Compensation co =
      hr.getCompensation(empl_id, token);
  …
```

**Figure 2: RMI Client Code to Create Token and Pass to Remote Object in RMI Server**

```
public class CompensationRmiImpl
            extends PortableRemoteObject
            implements CompensationRmiInterface {

  public Compensation
        getCompensation(int id, byte[] token) {
  ...
  try {
    ctx = new com.twoab.jaas.Context(token);
  } catch (com.twoab.jaas.ContextException ex) {
    …
  }
```

**Figure 3: RMI Server Code in Bold sets local Context to the Remote Authenticated User**

```
… (look up jobcode & salary in database)
Compensation co = new Compensation(id, jobcode, salary);
// determine if salary should be filtered...
try {
      ResourcePermission p =
          new ResourcePermission("Salary");
      AccessController.checkPermission(ctx, p);
  }
  catch (com.twoab.jaas.AccessControlException ace) {
      co.setSalary("User is Not Authorized");
  }
  …
  return co;
}
```

**Figure 4: Code to Check Permission for Salary and Filter Salary Attribute**

# Standards Update: The Future of CORBA®

## CORBA/i and CORBA/e

In December of 2004 members of the OMG Board of Directors who are CORBA vendors met face-to-face in Boston to discuss a common problem. Simply stated—no orb vendor currently has a simple answer to the question: "What version of the CORBA specification does your product support?"

So how did this happen? Unfortunately, as features were added to the CORBA standard over the last few years to support emerging and/or niche markets, there has not been a complementary effort to profile or package the standard. Full compliance to the 3.0 specification mandates a one-size-fits-all, solves-any-problem-including-the-ones-you-don't-have middleware offering. This is clearly not what the customers are demanding! All of the vendors in attendance (2AB, Borland, IONA, HP, Prism Technologies and Objective Interface Systems) expressed a desire to work together to create CORBA standards with compliance points that are focused on customer needs.

At a time when CORBA is being criticized as being "too complex," it is time to step up and decide how to leverage the strength of CORBA in the domains where it has proven to be the superior solution. Two market segments were selected as the focus of the initial effort to re-package and simplify the standards with the expectation that other focus areas would follow. These segments were selected because of the tremendous success that CORBA has in the markets they represent.

They are:

- Technology Integration
- Embedded Systems

The next generation of standards for these focused environments are being branded as

- CORBA/i
- CORBA/e

It has been agreed that these specifications will also interoperate so that enterprises who wish to manage embedded solutions with enterprise tools will be guaranteed the multi-vendor solution will work. These vendors continue to work between meetings, and have had face-to-face meets at the 2005 OMG meetings plus participated in a working session July 11th in Washington DC.

Drafts of the specifications are available to OMG members. CORBA/i is smsc/05-06-08, CORBA/e is smsc/05-05-01 and a matrix that compares these with CORBA3 is smsc/05-07-01.

The CORBA/i specification will include the core CORBA datatypes, APIs, the IIOP Protocol for Interoperability, the Naming Service and key standards for Integration with other key distribution technologies such as Java RMI and Web Services.

2AB plans to deliver compliant releases of orb2 with tools that enhance the manageability of the environment and simplify the development environment. We will include some of those tools in our "sneak preview" section of this newsletter.



1700 Highway 31
Calera, AL 35040

Phone: 205-621-7455
Fax: 205-621-7455
E-mail: info@2ab.com
Support: support@2ab.com

*2AB is a provider of Trusted Solutions for Distributed Business. 2AB offers the middleware, identity and access management tools needed by application developers to integrate systems and protect sensitive information from unauthorized access. 2AB is a business partner of IBM, Intel, HP, Sun, Microsoft and Stratus. The orb2 and iLock product suites are available for Java, Eiffel, C and C++ on AIX, HP-UX (PA-RISC and ia64), OpenVMS (ALPHA and ia64), Linux, Solaris, VOS, Windows and z/OS. Founded in 1997, 2AB is privately held. For more information, please see http://www.2ab.com.*