



# Merged iLab Bootstrapping and Time Of Day Test Lab Configuration

Author: Kirky DeLong, Sardis Harward and Antonia Harward

Last Modified: Summer 2009

Version: 9

<i>Overview</i> .....	5
<b>Service Broker Requirements</b> .....	<b>5</b>
iLab Process Agents .....	5
Grants and Qualifiers .....	6
Resource Mapping .....	6
<b>Installing iLab Servers</b> .....	<b>8</b>
Requirements .....	8
iLab Installation Directory .....	8
.NET 2.0 Framework .....	9
What is a GUID .....	9
<b>Building from the SDK</b> .....	<b>10</b>
<b>Opening the Visual Studio Project</b> .....	<b>10</b>
<b>Installing from the Pre-built Binaries</b> .....	<b>11</b>
<b>General Installation Process</b> .....	<b>11</b>
<b>iLabs Installation Check List</b> .....	<b>12</b>
<b>iLab Service Broker</b> .....	<b>13</b>
Service Broker Web Site .....	13
Create the Database .....	15
Grant Access to the Database .....	15
Service Broker Database Scripts .....	16
Web.Config .....	19
Update the Database Name .....	20
Update Email Addresses .....	20
Update Default Passkey .....	20
Web.Config Parameters to be aware of: .....	21
Testing Your Service Broker Installation .....	21
Self-Registration Fields .....	22
iLab Service Broker Self Registration .....	22
<b>Error Messages</b> .....	<b>24</b>
<b>Experiment Storage Service (ESS)</b> .....	<b>24</b>
ESS Web Site .....	24
Create the Database .....	25
Grant Access to the Database .....	25
ESS Database Scripts .....	25
Web.Config .....	27
Update the Database Name .....	28
Update Default Passkey .....	28
Web.Config Parameters to be aware of: .....	28
Test the Experiment Storage Service: .....	28
Self-Registration Fields .....	28

ESS Self Registration.....	29
<b>Userside Scheduling Services (USS) .....</b>	<b>30</b>
USS Web Site .....	30
Create the Database.....	30
Grant Access to the Database .....	31
USS Database Scripts .....	31
Web.Config.....	33
Update the Database Name.....	33
Update Default Passkey .....	34
Web.Config Parameters to be aware of: .....	34
Test the Scheduling Services: .....	34
USS Self Registration .....	35
<b>Labside Scheduling Service.....</b>	<b>35</b>
LSS Web Site .....	35
Create the Database.....	36
Grant Access to the Database .....	36
LSS Database Scripts.....	36
Web.Config.....	38
Update the Database Name.....	39
Update Default Passkey .....	39
Web.Config Parameters to be aware of: .....	39
Test the Scheduling Services: .....	39
LSS Self Registration.....	40
<b>Time Of Day Lab Server .....</b>	<b>40</b>
TimeOfDay Web Site .....	40
Create the Database.....	41
Grant Access to the Database .....	41
TOD Database Scripts.....	41
Web.Config.....	43
Update the Database Name.....	44
Update Default Passkey .....	44
Web.Config Parameters to be aware of: .....	44
Test the Time Of Day Server:.....	45
TOD Self Registration .....	45
<b>Bootstrapping.....</b>	<b>47</b>
<b>Users and Groups.....</b>	<b>47</b>
<b>Process Agents.....</b>	<b>49</b>
Service Broker .....	50
Experiment Storage Service (ESS).....	51
Userside Scheduling Server (USS).....	52
Labside Scheduling Server (LSS).....	54
Time Of Day Lab Server (LS) .....	55
<b>Time Of Day Lab Client.....</b>	<b>56</b>
Lab Client Record.....	56
Client Loader Script.....	56

Associate the Lab Server .....	57
Associate the ESS and USS .....	58
<b>Associate Groups.....</b>	<b>58</b>
<b>Create a Test User.....</b>	<b>58</b>
<b>Group Membership .....</b>	<b>59</b>
<b>Setup Time Of Day Experiment .....</b>	<b>60</b>
Manage USS .....	60
Manage LSS.....	60
<b>Running the Time Of Day Experiment.....</b>	<b>63</b>
<i>Setting Up Batched Experiments .....</i>	<i>65</i>
<b>Create a Batched Lab Server Record .....</b>	<b>65</b>
Create a Lab Client Record.....	66
Client Loader Script.....	66
Associate the Lab Server .....	67
Associate the ESS .....	68
<b>Associate Groups.....</b>	<b>68</b>
Check Grants.....	68
<b>MIT Certificate .....</b>	<b>69</b>
<b>Running the Experiment .....</b>	<b>70</b>
<b>Lab Client .....</b>	<b>70</b>
<i>Troubleshooting .....</i>	<i>71</i>
<b>Grants.....</b>	<b>71</b>

## Overview

This document describes the steps necessary to get an iLab Service Broker and Process Agents installed and functioning correctly in the single domain scenario. This document also contains instructions on how to install and configure the Time Of Day Interactive Experiment (to test your iLab Service Broker), and instructions for configuring a batched experiment.

### ***Service Broker Requirements***

You should have already installed and configured one or more computers to run the iLab Service Broker and Process Agents. This document will assume that you have at least one Service Broker installed and configured. Since the iLab architecture uses web services you can run all servers on a single machine or each service on a separate machine. See iLab document "iLabs Service Broker Machine Build Operating System and Database Install Guide" for instructions on installing and configuring the operating system and database. This document can be downloaded from the <http://icampus.mit.edu/iLabs/Architecture/Downloads/default.aspx> website.

- Windows 2003 Server Enterprise Edition
- SQL Server 2000 with Service Pack 3a
- Visual Studio.NET 2005 Professional and Library

It is also assumed that your computer(s) are connected to the Internet and have valid IP addresses and domain names. In order for the web services in iLab Architecture to communicate with each other, the service will need to communicate over the Internet. This document does not address network connectivity issues. Please contact your local systems administrator to make sure that your computer(s) are connected to the Internet and can communicate with each other correctly.

This document assumes that you are using **Windows 2003 Server Enterprise Edition** and **SQL Server 2000 Enterprise with Service Pack 3a**. There is a help document called "iLabs Service Broker XP notes.doc" with some configuration suggestions and notes about installing the iLab Service Broker under Windows XP with SQL Server Express. This document is not an install guide but a set of notes about the issues that can arise when installing under Windows XP with Microsoft SQL Express. We recommend and support the use of **Windows 2003 Server Enterprise Edition** and **SQL Server 2000 Enterprise with Service Pack 3a**.

### ***iLab Process Agents***

A process agent in the iLab architecture is a web service that implements the ProcessAgent interface. A process agent can be a web service or a web server. In addition to having the physical machines correctly installed and configured, you will need to have downloaded and installed the code or pre-built iLab services.

In the iLab architecture there are four types of services described; ISB, ESS, USS, LSS. This bootstrapping guide will assume that your iLab domain will have one Service Broker and at least one of each of the servers. Also included in this release is a test experiment

called the Interactive Time Of Day Experiment, designed to confirm that the Service Broker and its components are installed correctly. The Interactive Lab Server (ILS) with LabVIEW is not configured in this document but under a separate document called iLabs InteractiveLabServer with LabVIEW Setup and Configuration.

- iLab Service Broker (ISB)
- Experiment Storage Server (ESS)
- User-side Scheduling Server (USS)
- Lab-side Scheduling Server (LSS)
- Interactive Time of Day Experiment (TOD)
- Interactive Lab Server (ILS) with LabVIEW (this server is not configured in this document, but under a separate document called iLabs InteractiveLabServer with LabVIEW Setup and Configuration).

One important item to remember about Process Agents is that service (USS, ESS, and LSS) can have only one Administration group, but may have many Management groups.

### Grants and Qualifiers

Grants are the permissions given to an agent (either a group or user). Qualifiers are the resources (services or servers) that Agents are allowed to perform actions on.

Grants have three parts: Group, Function, and Qualifier.

- Groups
- Functions are the permissions or TicketType that a group has on a particular qualifier.
- Qualifiers are resources or resource mappings.

### Resource Mapping

- Resources such as scheduling and storage are associated with clients, process agents or groups.
- Resource Mappings are resource associations with one key to many values.
- Authorizations come from the Grant table not the Resource Mapping table.
- In the case of a single domain, resource mappings will be used to map client -> USS, client -> LSS and LS -> LSS. Mapping\_id's are unique. TicketTypes needed for each resource are stored in the grants information.

Map_ID	KeyType	KeyValue (DB)	ValueType	ValueValue (DB)
11	"client"	Client_object (client_id)	"ResourceType"	"ESS"
			"ProcessAgent"	ESS_Object (pa_id)
			"TicketType"	"Manage Experiment"
12	"client"	Client_object (client_id)	"ResourceType"	"USS"
			"ProcessAgent"	USS_Object (pa_id)

13	"LabServer"	LS_Object (pa_id)	"TicketType" "ResourceType" "ProcessAgent" "TicketType"	"Schedule session" "LSS" LSS_Object (pa_id) "Manage LS"
----	-------------	-------------------	--	--

## Installing iLab Servers

Currently, there is only one version of the iLab's distribution files that can be downloaded from <http://icampus.mit.edu/iLabs/architecture/> website:

The standard distribution contains both the SDK and a pre-built set of web sites **ilab\_Project\_Release\_2.zip**. If you want to use the pre-built services use the builds directory. If you intend to set up a development environment with full source code, use the code in the services directory for compiling. Note: Visual Studio 2005 is the recommended install, since you will need to build the local installation.

### Requirements

You should have the following software already installed and configured before you begin to set up the iLab Service Broker.

- Windows 2003 Server Enterprise
- SQL Server 2000 or 2005
- Microsoft .Net 2.0 Framework
- Visual Studio.NET 2005 Professional and Library (optional, but required for compiling the SDK version)

The installation and configuration of the server computers is described in the iLabs Service Broker OS and Database Install Guide.

### iLab Installation Directory

- Create a folder called **iLabs** (typically on the D drive)
- Unzip the iLabs Distribution archive **ilab\_Project\_Release\_2.zip** into this folder.
  - Right-click on the zip file
  - Select **Extract All...**
  - Click **Next>**
  - If you are using **ilab\_Project\_Release\_2.zip**, edit the directory to so that it simply reads "D:\iLabs" if you installed on the D drive. You can install the code in the directory of your choice. For notation purposes, this document will use `\*local_path*` to represent the directory on your local computer where the code is installed.
  - Click **Next>**
  - Click **Finish**

The structure of the new iLab directories includes the following (there may be other directories).

`\iLabs\Builds` - Precompiled versions.

`\iLabs\Database` - Database scripts and models.

`\iLabs\Documents` - Install documents.

`\iLabs\LabView` - LabView interface and support.

`\iLabs\Libraries` - ProcessAgent, ServiceBroker and Ticketing libraries.

`\iLabs\Project\iLabProject` - the Visual Studio solution file



\\iLabs\Services - The individual service's code

If you are using the pre-built version, you will need the Builds, Database and Documents directory. If you are using the SDK version, you will need all the directories except the Builds directory.

## **.NET 2.0 Framework**

The iLabs Architecture needs the .NET 2.0 framework to be installed. This is not installed when you installed Windows Server 2003.

Download the Microsoft .NET framework from <http://www.microsoft.com/downloads/> there should be a link to the .NET Framework Version 2.0 Redistributable Package (x86).

**This document assumes that you will be installing each service separately or on separate machines. This document is organized by service.**

**If you are installing all the services on a single machine, a more efficient way to set up the services is by activity; i.e., all the databases at once. This document is not configured to follow this method, but you can use the included checklist to install by activity.**

## **What is a GUID**

A Globally Unique Identifier or GUID is a special type of identifier used in software applications in order to provide a unique string for referencing. It is typically a string of letters and numbers separated by hyphens. iLab uses GUIDs to ensure the Process Agent IDs are unique. Since Process Agents such as the Service Broker, Experiment Storage Service, User-Side Scheduling and Lab-Side Scheduling exchange credentials, it is very important that their identifiers be unique. This ensures the correct service is accessed when needed.

There are several free, on-line, easy to use GUID generators available, or you can use the GUID generator provided with Visual Studio .NET. If you have installed Visual Studio.NET 2003, GuidGen.exe is available in the C:\Program Files\Microsoft Visual Studio.NET 2003\Common7\Tools directory. To create a GUID, select Registry Format, Click the Copy button, and paste the resulting GUID between the quote marks following the value field. Edit out the bracket and hyphen characters ({-}).

**BUG: The Service Broker GUID needs to be 35 characters or less due to some legacy code in the batched architecture. If you enter a GUID into the web.config file directly, please make sure that it is less than 35 characters long including dashes and underscores.**

## ***Building from the SDK***

Most developers prefer to run from their own compiled versions of the software. To do this you will need to use the SDK version of the iLab software located in the Services directory, if not, please use the pre-built version located in the Builds directory.

To build the files locally you must first create virtual directories for each of the services as defined in the solution file. The different site directories are located in the Services directory of the iLabs installation.

- ❑ http://localhost/ExperimentStorage
- ❑ http://localhost/iLabServiceBroker
- ❑ http://localhost/InteractiveTimeOfDay
- ❑ http://localhost/LabSideScheduling
- ❑ http://localhost/UserSideScheduling
- ❑ http://localhost/InteractiveLabServer
- ❑ http://localhost/ilab\_WSDLI

## ***Opening the Visual Studio Project***

Now, create the Visual Studio project.

- ❑ Start->Run->Microsoft Visual Studio.NET 2005- Microsoft Visual Studio.NET 2005
- ❑ Click on the **Projects** tab
- ❑ Click **Open Project**
- ❑ Browse to `\*local_path*\iLabs\projects\ iLabProject_wLabView.sln`
- ❑ Click **OK**
- ❑ Wait while Visual Studio loads the code. If you get an error message saying that VS cannot find the directory, browse to the location where the code in question currently resides.
- ❑ Do a Test Build: **Build->Build Solution**. If it builds without errors, your installation is a success. There may be a few warnings.

**Note:** You cannot develop in Visual Studio against a website that is running SSL. If you have problems, check to make sure that SSL is not turned on in web.config.

### Problems under XP

- Under the virtual directory properties, make sure that Default.aspx is in the documents section
- Make sure that the .NET framework is installed
- Run the .Net framework-iis installation aspnet\_regiis.exe from the command prompt (C:\WINDOWS\Microsoft.NET\Framework)

There is a help document titled "iLabs Service Broker XP notes.doc" with some configuration and notes about installing the iLab Service Broker under Windows XP with

SQL Server Express. This is not an install guide, but notes about the issues that can arise when installing under Windows XP with Microsoft SQL Express.

### ***Installing from the Pre-built Binaries***

If you do not want to build the project locally, please use the pre-built version of the code included in the distribution's Builds directory. To use the pre-built version, when creating the virtual directories, set the directory location to the Builds directory for each service rather than the services directory.

### ***General Installation Process***

A more detailed description of each installation process will follow. Each service installation will follow the same general process:

- ❑ Create a virtual directory
- ❑ Create the database (tables and stored procedures)
- ❑ Edit Web.config file as necessary
- ❑ Complete self-registration

After a service is installed the Service Broker needs to create a trusted relationship with the service. This is done in the bootstrapping section of this document.

**URGENT:** This document assumes that you have compiled the iLab architecture and will be using the Services directory for the IIS virtual directories and editing the web.config files. If you are using the pre-compiled version in the Builds directory you will need be aware and make the necessary changes so that the IIS virtual directories point to the correct prebuilt locations and that you edit the correct web.config files.

## ***iLabs Installation Check List***

### **iLab Service Broker**

- Create Virtual Web Site
- Create Database
- Database Permissions
- Database Process Agent scripts
- Database Ticketing scripts
- Database Interactive Service Broker scripts
- Edit Web.Config file
- Test service
- Register domain services

### **Experiment Storage Server**

- Create Virtual Web Site
- Create Database
- Database Permissions
- Database Process Agent scripts
- Database ESS scripts
- Edit Web.Config file
- Test service
- ESS self registration

### **UserSide Scheduling Service**

- Create Virtual Web Site
- Create Database
- Database Permissions
- Database Process Agent scripts
- Database USS scripts
- Edit Web.Config file
- Test service
- USS self registration

### **LabSide Scheduling Service**

- Create Virtual Web Site
- Create Database
- Database Permissions
- Database Process Agent scripts
- Database LSS scripts
- Edit Web.Config file
- Test service
- LSS self registration

### **Interactive Time Of Day Server**

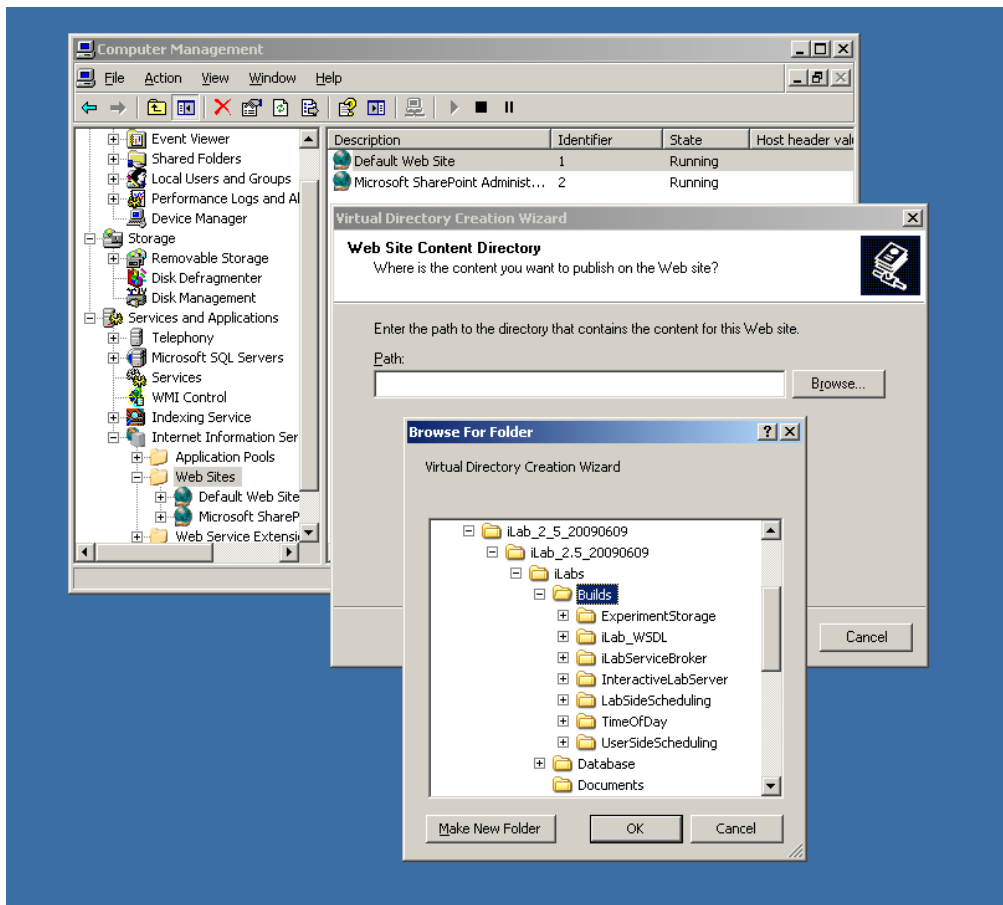
- Create Virtual Web Site
- Create Database
- Database Permissions
- Database Process Agent scripts
- Edit Web.Config file
- Test service
- ToD self registration

## iLab Service Broker

### Service Broker Web Site

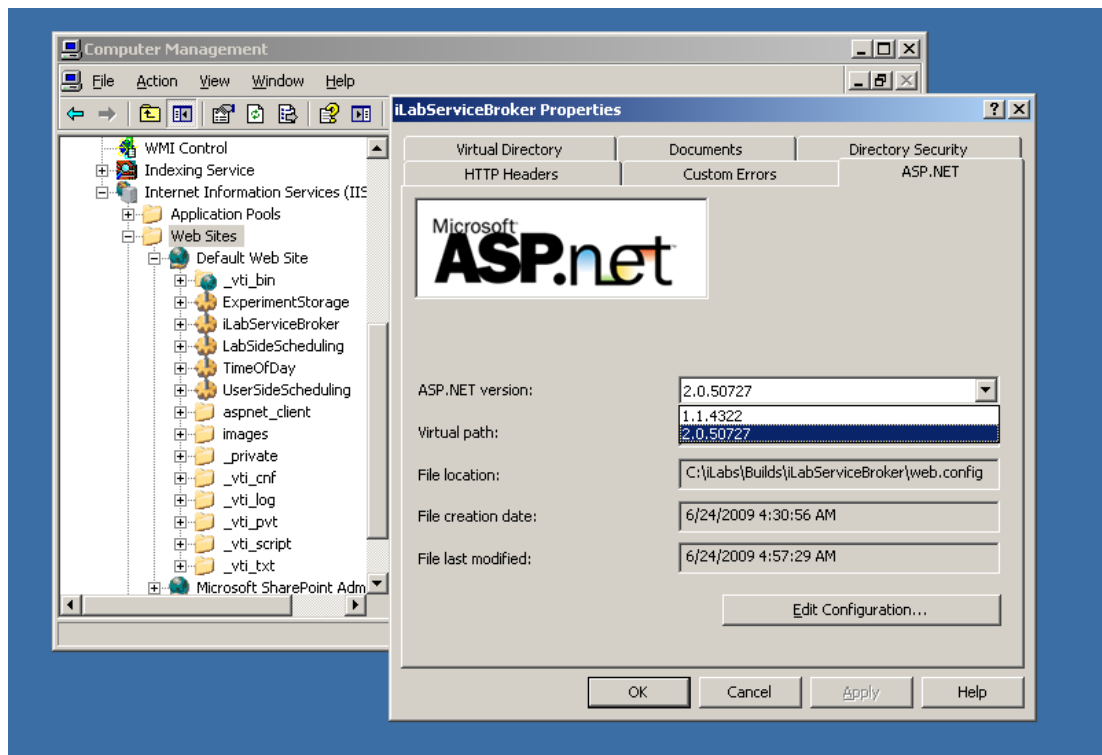
First, create a virtual directory in IIS.

- ❑ Right-Click on My Computer
- ❑ Click **Manage**
- ❑ Expand **Services and Applications**
- ❑ Expand **Internet Information Services**
- ❑ Expand **Web Sites**
- ❑ Expand **Default Web Site**
- ❑ Right-click on **Default Web Site**
- ❑ Select **New->Virtual Directory**
- ❑ Click **Next** to start the Virtual Directory Creation Wizard
- ❑ In the **Alias** box, type **iLabServiceBroker**
- ❑ Click **Next**
- ❑ Click the **Browse** button and navigate to the location where your Service Broker code is located. The path should resemble `\*local_path*\iLabs\Services\iLabServiceBroker` directory. In addition, there is a second set of precompiled builds located in `\*local_path*\iLabs\Builds\iLabServiceBroker`. Use the Service directory if you compiled the code yourself and the builds directory to use the pre-build versions.



For notational purposes we will use `\*local_path\` to represent the directory on your local computer where the code is unzipped.

- ❑ Click **Next**
- ❑ Access Permissions: make sure you check the **Run Scripts, Execute and Read** permission boxes
- ❑ Click **Next**
- ❑ Click **Finish**
- ❑ You should see the Service Broker site listed under the Default Web Site
- ❑ Right click on the **iLabServiceBroker** website and select **Properties**
- ❑ Select the **ASP.NET** tab
- ❑ Use the pull-down list for the ASP.NET version and select 2.0.xxx. Where xxx is the current release version.



- ❑ Select **OK** and close the properties window
- ❑ Close the Computer Management Window

This document assumes that you will be installing each service separately or on

separate machine. This document is organized by service.

If you are installing all the services on a single machine, a more efficient way to set up the services is by activity; i.e., all of the virtual directories at once. This document is not configured to follow this method, but you can use the included checklist to install by activity.

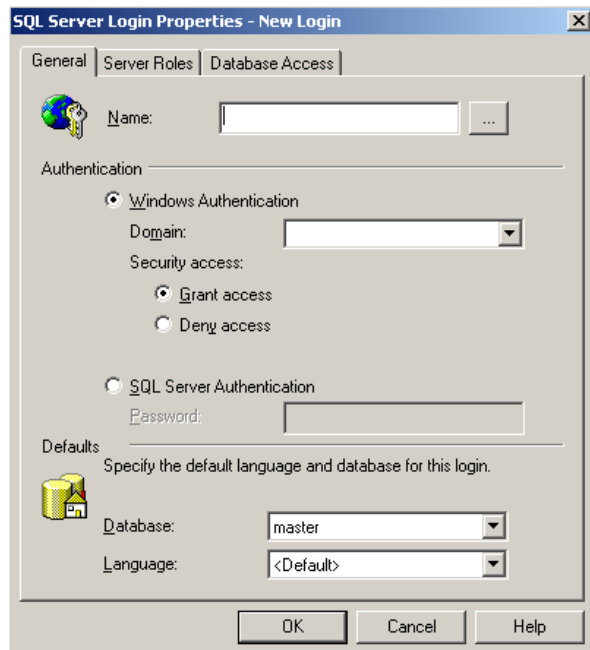
## Create the Database

First, create the database.

- ❑ Start-> Enterprise Manager
- ❑ Console Root->Microsoft SQL Servers->SQL Server Group->(local)->Databases
- ❑ Right-Click on Databases
- ❑ New Database...
- ❑ Name: **ilab\_ISB** (This is the default. You can use any name you would like but you must update the web.config file to use the different name)
- ❑ Click OK

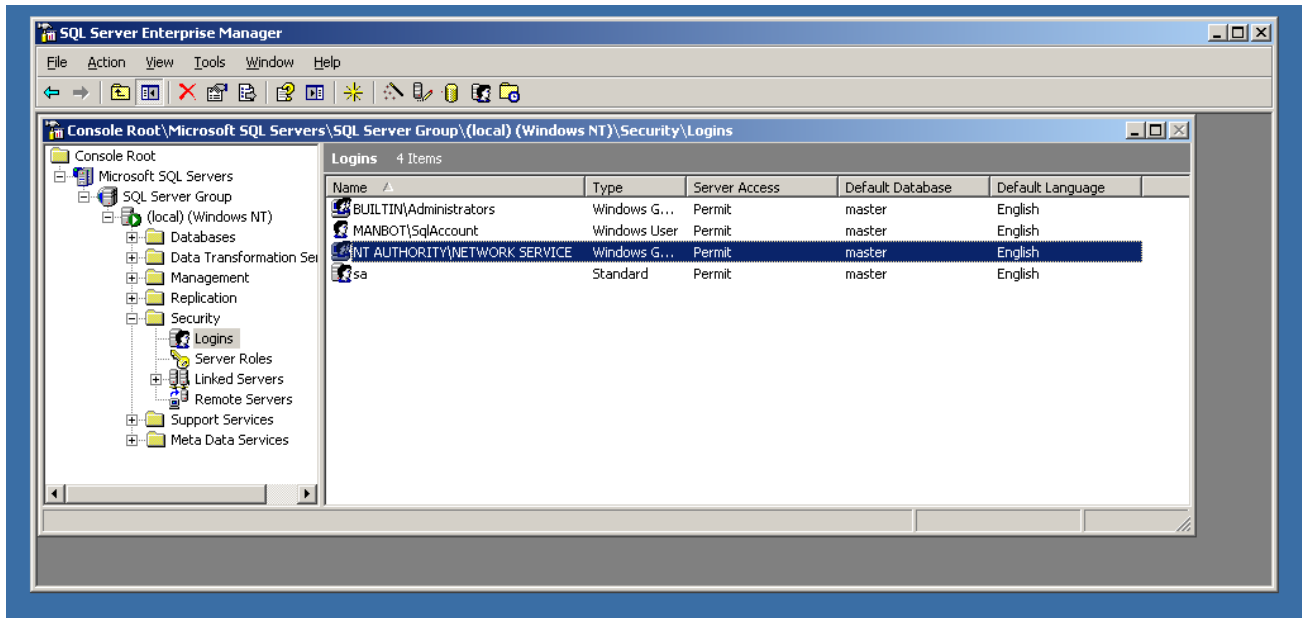
## Grant Access to the Database

- ❑ Expand the Security folder in Enterprise Manager.
- ❑ Right-click Logins, select New Login...



- ❑ In the Name box, type **NT AUTHORITY\NETWORK SERVICE**

- Note: if you already have a login for NT AUTHORITY\NETWORK SERVICE, double-click on it and continue with the next step. If you are installing under Windows XP, please check the Windows XP notes guide for the proper security name to use.



- ❑ Click the **Database Access** tab
- ❑ Check the box next to **ilab\_ISB** (or the database name you used above)
- ❑ In the box below, click the box next to **db\_owner**
- ❑ Click **OK**

### Service Broker Database Scripts

For each Process Agent created (ISB, USS, LSS, ESS, and TOD), you will need to run Process Agent scripts that are generic for all process agents, and specific scripts for the particular Process Agent you are creating. The generic process agent scripts are in a directory called ProcessAgent and specific database scripts will be in named directories, for example ServiceBroker.

This document contains two ways to run the database scripts for Microsoft SQL server; from within the SQL Query Analyzer or from the loader scripts that are in the DB\_Scripts directory. The preferred method of creating a new service database is to use **Method 2: Using the SQL loader scripts**. If you are just updating a stored procedure and not the database, it is best to use the SQL Query analyzer. We include Method1: Using the SQL Query Analyzer for your information.

NOTE: the SQL loader scripts do not work under Windows XP with SQL Express - if you are using Windows XP and SQL Express you will need to use the SQL Query Analyzer.

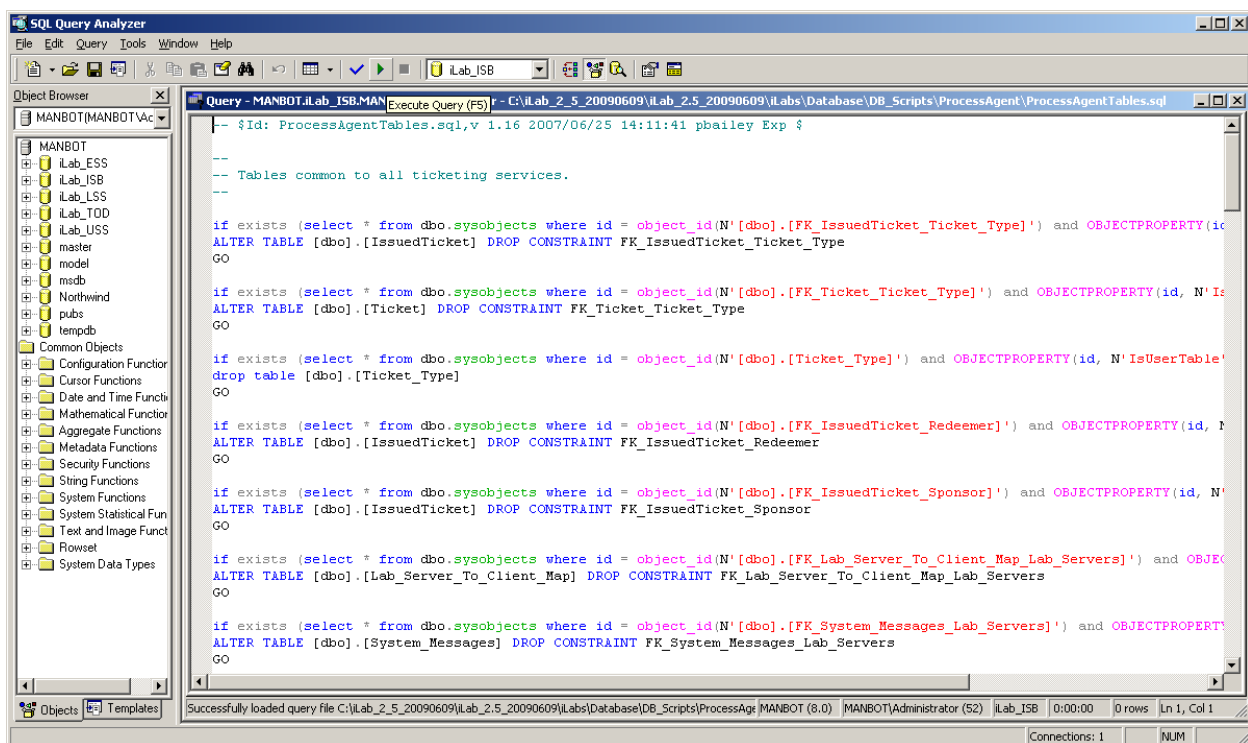
**Do NOT attempt both methods - choose one.**



## METHOD 1: Using the SQL Query Analyzer:

- ❑ Expand Databases
- ❑ Select the `ilab_ISB` database (or what you named it) in Enterprise Manager.
- ❑ On the menu bar, select **Tools -> SQL Query Analyzer**
- ❑ In Query Analyzer, click **File->Open...**
- ❑ Browse to database scripts directory typically `(\*local_path*\iLabs\ Database\DB_Scripts\ProcessAgent)`
- ❑ In the `ProcessAgent` directory select and open the `ProcessAgentTables.sql`

This directory contains the database scripts for all of the Process Agents. You will set up and run the `ProcessAgent` database scripts before running the service-specific database scripts. This is important because there is a key from the `iLab Service Broker` foreign to the `Process Agents` table.



- ❑ Again make sure the correct database, `ilab_ISB`, is the selected database in the dropdown menu at the top of the window. **This is very important because the scripts will be installed in the database specified by this pull-down menu.**
- ❑ Click the green triangle "play" button to run the script
- ❑ You should see a message: "The command(s) completed Successfully."
- ❑ Follow the same procedure, **File → Open**, to open and run the `ProcessAgentProcedures.sql` script.
- ❑ You should see the same message: "The command(s) completed Successfully."
- ❑ Open and run the `SetDefaultsPA.sql` script.

- ❑ You should see several messages: "1 row(s) affected." and "Checking Identity Information ....".

Next, set up and run the Interactive Service Broker-specific database scripts (typically located in `\*local_path*\iLabs\ Database\DB_Scripts\ServiceBroker`).

- ❑ In Query Analyzer, click **File->Open...**
- ❑ Browse to database scripts directory, typically (`\*local_path*\iLabs\ Database\DB_Scripts\ServiceBroker`)
- ❑ Open and run the `TIssuerTables.sql` script.
- ❑ Open and run the `TIssuerProcedures.sql` scripts.
- ❑ Open and run the `ServiceBrokerCoreTables.sql` script.
- ❑ Again you should see the message: "The command(s) completed Successfully."
- ❑ Open and run `ServiceBrokerCoreProcedures.sql`, as above
- ❑ Open and run `ServiceBrokerCoreDefaultValues.sql` as above
- ❑ You should see several messages: "1 row(s) affected." and "Checking Identity Information ....".
- ❑ Close SQL Query Manager and Enterprise Manager

#### METHOD 2: Using the SQL loader scripts:

- ❑ Open a Command Prompt window. Start -> Programs -> Accessories -> Command Prompt.
- ❑ A DOS window should open.
- ❑ Change the directory to the DBscripts directory.
- ❑ Type: `"cd \*local_path*\iLabs\ Database\DB_Scripts\"`
- ❑ Type: `"LoadISB ilab_ISB"` where `ilab_ISB` is the name of your database.
- ❑ You should see something similar to the following with your directory path and database name.

```
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>loadISB iLab_ISB

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo populating iLab_ISB
populating InteractiveSB InteractiveSB

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>call ISB_tables.bat iLab_ISB

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo InteractiveSB Tables:
InteractiveSB Tables: iLab_ISB

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d iLab_ISB
eSB -i .\Ticketing\ProcessAgentTables.sql -o isbBuild0.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d iLab_ISB
-i .\Ticketing\SetdefaultsTicketing.sql -o isbBuild1.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d iLab_ISB
-i .\Ticketing\TIssuerTables.sql -o isbBuild2.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d iLab_ISB
-i .\InteractiveSB\ServiceBrokerCoreTables.sql -o isbBuild3.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d iLab_ISB
-i .\InteractiveSB\ServiceBrokerCoreDefaultValues.sql -o isbBuild4.log
```

```

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>call ISB_procedures.ba
t ilab_ISB

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo InteractiveLS Pro
cedures: ilab_ISB
InteractiveLS Procedures: ilab_ISB

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d ilab_ISB
-i .\Ticketing\ProcessAgentProcedures.sql -o isbBuild1.log -r

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d ilab_ISB
-i .\Ticketing\TIssuerProcedures.sql -o isbBuild4.log -r

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d ilab_ISB
-i .\InteractiveSB\ServiceBrokerCoreProcedures.sql -o isbBuild6.log -r

```

## Web.Config

An overview of editing the web.config

Web.config is a text file containing configuration information in an XML format. It resides in the root of the Service Broker web site (typically in the `\*local_path*\iLabs\Services\iLabServiceBroker` directory). `Web.config.template` ships with this release.

If this is a new installation of the Service Broker, delete any existing copy of the `Web.Config` file. **Copy `web.config.template` and paste the file into the same directory. Rename the copy "web.config" and edit it using notepad.**

There are two ways to configure a service: editing the `web.config` file for that service directly using a program such as notepad, or using the self-registration webpage supplied as part of the installation package. Please note that not all fields can be configured using the self-registration webpages; there are certain fields that can only be edited in the `web.config` file. Also, changes made on the self-registration pages are not updated in the `web.config` file.

```

<?xml version="1.0"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <!-- This area contains the Services specific configuration properties YOU MUST SET THESE-->
  <appSettings>
    <!--
    Interactive ServiceBroker Web.Config -->

    <!--
    ProcessAgent required Keys -->
    <!-- THE FOLLOWING MUST BE SET FOR ALL SERVICES, used to define the service -->
    <!-- The following keys specify the Webservice required Values -->
    <!-- They are used by the SelfRegistration page to generate the ProcessAgent's
    Bootstrap ProcessAgent Record, which must be created before installing domain credentials -->

    <add key="serviceType" value="SERVICE BROKER"/>
    <!-- The name of the Service displayed for the user, please change -->
    <add key="serviceName" value="Default iLab ServiceBroker"/>
    <!-- This Service's GUID You must supply a valid GUID, see createGUID.exe, may be any string, 50 characters or less. --
  >
    <add key="serviceGUID" value=""/>
    <add key="codebaseURL" value="http://localhost/iLabServiceBroker"/>
    <add key="serviceURL" value="http://localhost/iLabServiceBroker/iLabServiceBroker.aspx"/>

```

```

<add key="defaultPasskey" value="Let me in!"/>

<!-- the database connection information
NOTE: You Must set the database catalog to the database name that you created as part of the instalation process -->
<add key="sqlConnection" value="data source=(local);initial catalog=iLab_ISB;Integrated Security=SSPI" />

<!-- Optional path and filename for application log. The directory must exist & be writeable to the IIS worker process
-->
<add key="logPath" value="" />

<!-- ServiceBroker Specific Keys -->
<!-- Cookie support for applets needed for .Net 2.0. This should only need to be changed if you host more than one SB
per machine -->
<add key="isbAuthCookieName" value="isbAuth" />

<!-- CHANGE THIS TO LOCAL ADDRESS: email address where help requests are sent -->
<add key="supportMailAddress" value="ilab-comment@mit.edu"/>
<!-- CHANGE THIS TO LOCAL ADDRESS: default email address where requests for membership in a group are sent.
If the group has a contact address mail will be sent to that address -->
<add key="registrationMailAddress" value="your_email@your_machine.edu"/>
<add key="genericFromMailAddress" value="your_email@your_machine.edu"/>

<!-- email address where bug reports are sent, this for the centralized bug tracking system at MIT.
Append a local mail address if you want to also recieve bug reports ( comma delimited ) -->
<add key="bugReportMailAddress" value="ilab-bug@mit.edu"/>
<add key="bugReportOptions" value="System-wide error,Login,Group Issues,Grant Issues,UserInterface,Business
logic"/>
<add key="helpOptions" value="General Help,Logging In,Running Lab,Page Specific Help"/>

```

.....

### Update the Database Name

Update the database connection variable in the web.config file to use the name you chose when you created the database. Note the default database name is set to ilab\_ISB. If you used a different database name, update the catalog value in the key. This name MUST match the name of your database exactly including capitalization.

```

<add key="sqlConnection"
value="data source=(local);initial catalog=iilab_ISB;Integrated Security = SSPI" />

```

### Update Email Addresses

Do not forget to update the email address keys or else users will encounter errors when registering and sending bugs.

```

<add key="supportMailAddress" value=" your_email@your_machine.edu " />
<add key="registrationMailAddress" value=" your_email@your_machine.edu " />
<add key="genericFromMailAddress" value="your_email@your_machine.edu"/>

```

### Update Default Passkey

The default Pass Key is a simple string used by process agents to exchange credential information the first time only. This can be any string you would like. We recommend that you not change this.

```

<add key="defaultPasskey" value="" /> Set to "Let me in!" by default

```

## Web.Config Parameters to be aware of:

There are several other parameters that you should be aware of in web.config, for example, bugreportMailAddress which is the email address that bug reports will be sent to.

serviceType	SERVICE BROKER (do not change this)
serviceName	A default name - should be changed in selfRegister (later)
serviceGuid	Must be created in selfRegister (later)
codebaseURL	SB application URL, change to DNS name in selfRegister (later)
serviceURL	SB's service URL, change to DNS name in selfRegister (later)
defaultPassKey	Initial Passkey for credential exchange
sqlConnection	Database name, default ilab_ISB
bugReportingMailAddress	Enter a local email address or leave ilab-bug@mit.edu you may add comma delimited additional addresses
supportMailAddress	Enter a local email address
registrationMailAddress	Enter a local email address
genericFromMailAddress	Enter a local email address

**NOTE: In the following steps it is best to use the fully qualified URLs in this document rather than localhost. localhost is good for testing but not for web service registrations. Make sure to use fully qualified URLs for registration. It is also important that your registration URLs be consistent. Do not mix the use of fully qualified URLs, localhost and IP addresses within a single service setup. Mixing registration URLs will cause problems with web service communications.**

## Testing Your Service Broker Installation

- ❑ Open Internet Explorer
- ❑ Browse to <http://localhost/iLabServiceBroker/iLabServiceBroker.asmx>
- ❑ The iLab Service Broker service description page should be displayed
- ❑ Browse to <http://localhost/iLabServiceBroker>
- ❑ Log in as follows:
  - Username: **SuperUser**
  - Password: **ilab**

Using the <http://localhost> is a good way to test your service installation, but you should also test with the fully qualified URLs in this document. The iLab architecture is designed to use fully qualified URLs, and unexpected things can happen when using localhost.

- ❑ Browse to <http://machinename.edu/iLabServiceBroker/iLabServiceBroker.asmx>

- ❑ The iLab Service Broker service description page should be displayed
- ❑ Browse to [http:// machinename.edu /iLabServiceBroker](http://machinename.edu/iLabServiceBroker)
- ❑ Log in as follows:
  - Username: **SuperUser**
  - Password: **ilab**

### **Self-Registration Fields**

Most of the setup information like the serviceName, serviceGUID and codebaseURL will now be entered in the Service Broker self-registration page when you first login to the Service Broker. This is the preferred method, but you can also enter the information into the web.config file and it will be loaded into the self-registration page for you to check.

**NOTE: Currently, changes made in the self-registration page are not updated in the web.config file.**

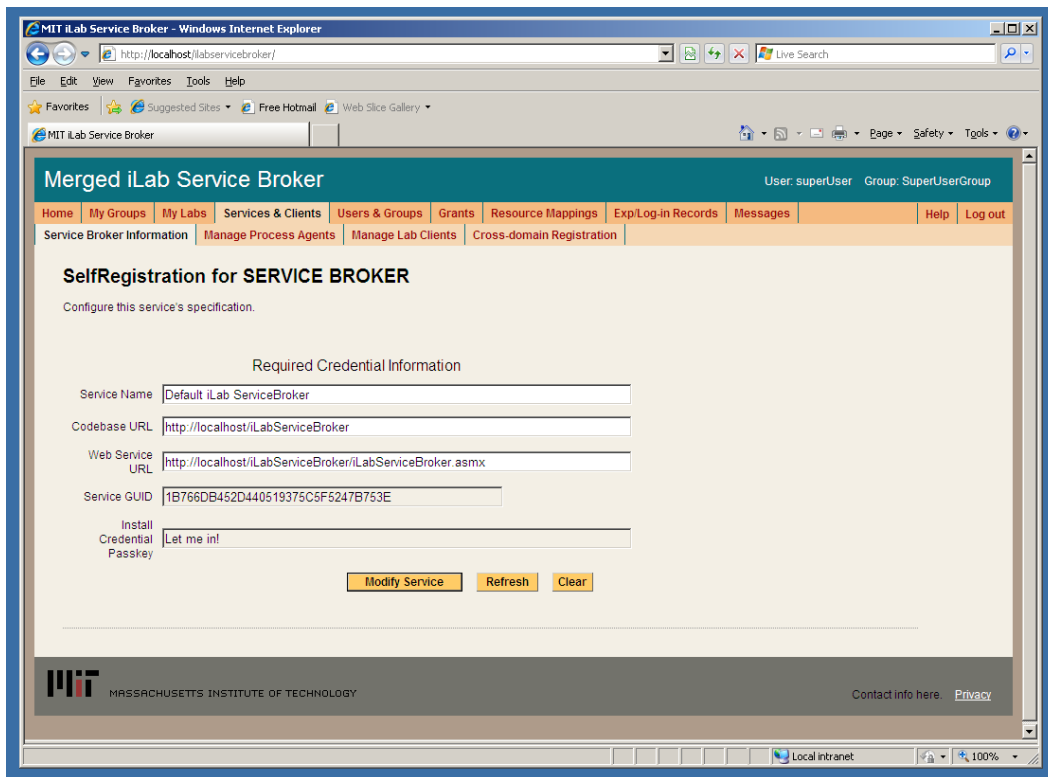
Each Service Broker and Process Agent must have a unique GUID (Globally Unique Identifier) which will identify this particular Service Broker to all lab servers and Process Agents. This GUID will be supplied to the Lab Server(s) administrator so that the Lab Server can be configured to communicate with this Service Broker. We use a GUID produced on the Service Broker's machine with all the '-', '{', and '}' characters stripped out. You can generate your own GUID and enter it in the web.config, or use the "Generate GUID" button on the Service Broker's self registration page and then remove the hyphens (this is the preferred method).

**BUG: The Service Broker GUID needs to be 35 characters or less due to some legacy code in the batched architecture. If you enter a GUID into the web.config file directly, please make sure that it is less than 35 characters long including dashes and hyphens.**

### ***iLab Service Broker Self Registration***

In order for process agents to exchange credentials, their information needs to be entered into the Process Agents table. There is a self-registration web service included with each process agent. You cannot register a process agent until after you have created and edited the corresponding web.config file.

- ❑ Open Internet Explorer
- ❑ Browse to <http://machinename.edu/iLabServiceBroker/>
- ❑ Login as SuperUser
- ❑ Click on the "Services & Clients" tab
- ❑ Click on the "Service Broker Information" tab in the sub-menu



- ❑ Modify Service Name, CodeBase & Webservice URLs as needed. Please make sure that the Service Name is unique. **Be sure to use fully qualified URLs.**
  - Codebase URL: `http://machinename.edu/iLabServiceBroker`
  - Web Service URL:  
`http://machinename.edu/iLabServiceBroker/iLabServiceBroker.asmx`
- ❑ If necessary, use the "Create GUID" button to generate a new GUID and remove the hyphens.

**BUG: The Service Broker GUID needs to be 35 characters or less due to some legacy code in the batched architecture. If you used the "Create GUID" button to generate the GUID, please remove the hyphens "-" from the GUID before saving.**

- ❑ Click the **Save Changes** button

Please be aware that there are certain fields that can not be configured from the Service Broker self-registration page. For example, the default passkey for registering process agents such as the ESS can only be edited in the web.config file. Similarly, configuring email addresses for bugs and notices can also only be edited in the web.config file. Please see the section on the web.config for more information.

## **Error Messages**

When testing your installation of the Service Broker you may see one of the following error messages:

- ❑ *" Cannot open database requested in login 'ilab\_ISB'. Login fails. Login failed for user 'NT AUTHORITY\NETWORK SERVICE'."*
- ❑ Edit the web.config and enter the correct database name.
- ❑ "The page cannot be displayed. You have attempted to execute a CGI, ISAPI, or other executable program from a directory that does not allow programs to be executed." The Virtual Directory in IIS does not have the correct permissions. Remove the virtual directory and recreate it with the correct permissions.
- ❑ In the System News and Messages section of the login page you may see one of the following messages:
  - An Administrator needs to edit web.config and add a value for registrationMailAddress. This is the default email address where requests for membership in a group are sent.
  - An Administrator needs to add a value for supportMailAddress. This is the email address where help requests are sent.

To remove error or warning messages from the System News and Message area,

- ❑ Login as SuperUser
- ❑ Select the Messages tab from the top menu
- ❑ Click on the system bullet in the message type area
- ❑ Select a message to delete and click remove.

## **Experiment Storage Service (ESS)**

### **ESS Web Site**

First, create a virtual directory in IIS.

- ❑ Right-Click on My Computer (renamed to your computer name)
- ❑ Click **Manage**
- ❑ Expand **Services and Applications**
- ❑ Expand **Internet Information Services**
- ❑ Expand **Web Sites**
- ❑ Expand **Default Web Site**
- ❑ Right-click on **Default Web Site**
- ❑ Select **New->Virtual Directory**
- ❑ Click **Next** to enter the Virtual Directory Creation Wizard
- ❑ In the **Alias** box, type **ExperimentStorage**
- ❑ Click **Next**
- ❑ Enter **\\*local\_path\*\iLabs\Services\ExperimentStorage** in the **Directory** box (or click the **Browse** button, and navigate to the location where the code is located.)



If you are using the pre-built code, please use the "Builds" directory, \\\*local\_path\*\iLabs\Builds\iLabServiceBroker.

- ❑ Click Next
- ❑ Access Permissions: make sure you check the Run Scripts, Execute and Read permission boxes
- ❑ Click Next
- ❑ Click Finish
- ❑ You should see the ExperimentStorage site listed under the Default Web Site
- ❑ Right click on the ExperimentStorage website and select Properties
- ❑ Select the ASP.NET tab
- ❑ Use the pull-down list for the ASP.NET version and select 2.0.xxx. Where xxx is the current release version.
- ❑ Select OK and close the properties window
- ❑ Close the Computer Management Window

### **Create the Database**

First, create the database.

- ❑ Start-> Enterprise Manager
- ❑ Console Root->Microsoft SQL Servers->SQL Server Group->(local)->Databases
- ❑ Right-Click on Databases
- ❑ New Database...
- ❑ Name: **ilab\_ESS** (You can use any name you would like but you must update the web.config file if you change the default name)
- ❑ Click OK

### **Grant Access to the Database**

- ❑ Expand the Security folder in Enterprise Manager.
- ❑ Double-click on the NT AUTHORITY\NETWORK SERVICE under Logins.
- ❑ Click the Database Access tab
- ❑ Check the box next to **ilab\_ESS** (or the database name you used above)
- ❑ In the box below, click the box next to **db\_owner**
- ❑ Click OK

### **ESS Database Scripts**

This document contains two ways to run the database scripts for Microsoft SQL server; from within the SQL Query Analyzer or from the loader scripts that are in the DB\_Scripts directory. The preferred method of creating a new service database is to use Method 2: Using the SQL loader scripts. If you are just updating a stored procedure and not the database, it is best to use the SQL Query analyzer. We include Method1: Using the SQL Query Analyzer for your information.

NOTE: the SQL loader scripts do not work under Windows XP with SQL Express - if you are using Windows XP and SQL Express you will need to use the SQL Query Analyzer.

**Do NOT attempt both methods - choose one.**

## METHOD 1: Using the SQL Query Analyzer:

- ❑ Start-> **Enterprise Manager**
- ❑ Console Root->Microsoft SQL Servers->SQL Server Group->(local)->**Databases**
- ❑ Expand Databases
- ❑ Select the **ilab\_ESS** database (or what you named it) in Enterprise Manager.
- ❑ On the menu bar, select **Tools -> SQL Query Analyzer**
- ❑ In Query Analyzer, click **File -> Open...**
- ❑ Browse to `*local_path*\iLabs\ Database\DB_Scripts\ProcessAgent`

Every service must have its own Process Agent information. These tables should exist before the service-specific tables are added to the database.

- ❑ In the ProcessAgent directory select and open the **ProcessAgentTables.sql**
- ❑ Make sure the correct database is the selected database in the dropdown menu at the top of the window- **This is very important**, because the scripts will be installed in the database specified by this pull-down menu.
- ❑ Click the green triangle "play" button to run the script
- ❑ You should see a message: "The command(s) completed Successfully."
- ❑ Follow the same procedure, **File→Open**, to open and run the **ProcessAgentProcedures.sql** script.
- ❑ You should see the same message: "The command(s) completed Successfully."
- ❑ Open and run the **SetDefaultsPA.sql** script.

Next, set up and run the ESS database script.

- ❑ In Query Analyzer, click **File -> Open...**
- ❑ Browse to database scripts directory typically `*local_path*\iLabs\ Database\DB_Scripts\ESS`
- ❑ In the ESS directory select and open the **Ess\_Tables.sql**
- ❑ Make sure the correct database is the selected database in the dropdown at the top, then click the green triangle "play" button to run the script
- ❑ You should see the message: "The command(s) completed Successfully."
- ❑ Follow the same procedure to open and run the **Ess\_Procedures.sql** script.
- ❑ Again, you should see the message: "The command(s) completed Successfully."
- ❑ Close SQL Query Manager and Enterprise Manager

## METHOD 2: Using the SQL loader scripts:

- ❑ Open a Command Prompt window. Start -> Programs -> Accessories -> Command Prompt.
- ❑ A DOS window should open.
- ❑ Change the directory to the DBscripts directory.
- ❑ Type: `"cd *local_path*\iLabs\ Database\DB_Scripts\"`
- ❑ Type: `"LoadESS ilab_ESS"` where ilab\_ESS is the name of your database.

- You should see something similar to the following with your directory path and database name.

```
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>loadESS ilab_ESS

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo populating ESS ilab_ESS
populating ESS ESSdb

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>call ESS_tables.bat ilab_ESS

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo creating ESS Tables: ilab_ESS
creating ESS Tables: ilab_ESS

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d ilab_ESS -i .
\Ticketing\ProcessAgentTables.sql -o essBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d ilab_ESS -i .
\Ticketing\SetdefaultsTicketing.sql -o essBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d ilab_ESS -i .
\ESS\Ess_tables.sql -o essBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>call ESS_procedures.bat ilab_ESS

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo Creating StoredProcedures ESS: ESSdb
Creating StoredProcedures ESS: ilab_ESS

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d ilab_ESS -i .
\Ticketing\ProcessAgentProcedures.sql -o essBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d ilab_ESS -i .
\ESS\ESS_Procedures.sql -o essBuild.log
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>
```

## Web.Config

An overview of editing the web.config.

Web.config is a text file containing configuration information in an XML format, residing in the root of the ESS Services web site (typically in the `*local_path*\iLabs\Services\ExperimentStorage` directory). `Web.config.template` ships with this release. If a `web.config` already exists, please delete it. Copy `web.config.template` and paste the file into the same directory. Rename the copy "web.config" and edit it using notepad. Please be sure to use only fully qualified URLs

**NOTE: Currently, changes made in the self-registration page are not updated in the web.config file.**

## Update the Database Name

Update the database connection variable in the web.config file to use the name you chose when you created the database. Note the default database name is set to ilab\_ESS. If you used a different database name, update the catalog value in the key.

```
<add key="sqlConnection"
value="data source=(local);initial catalog=ilab_ess;Integrated Security = SSPI" />
```

## Update Default Passkey

The default Pass Key is a simple string used by process agents to exchange credential information the first time only. This can be any string you would like. We recommend that you not change this.

```
<add key="defaultPasskey" value="ESS" />
```

## Web.Config Parameters to be aware of:

serviceType	EXPERIMENT STORAGE SERVER - Do not Change
serviceName	ESS unique name for this service
serviceGuid	ESS GUID - Set during self Registration (later)
codebaseURL	ESS application URL - Set during self Registration (later)
serviceURL	ESS's service URL - Set during self Registration (later)
defaultPassKey	Initial Passkey for credential exchange
sqlConnection	Database Connection string;

## Test the Experiment Storage Service:

- ❑ Open Internet Explorer
- ❑ Browse to <http://localhost/ExperimentStorage/ExperimentStorage.asmx>

Using the <http://localhost> is a good way to test your service installation, but you should also test with the fully qualified URLs in this document. The iLab architecture is designed to use fully qualified URLs, and unexpected things can happen when using localhost.

## Self-Registration Fields

Most of the setup information like the serviceName, serviceGUID and codebaseURL will now be entered in the self-registration page. This is the preferred method, but you can also enter the information into the web.config file and it will be loaded into the self-registration page for you to check.

```
<?xml version="1.0"?>
<!--
```

```
Note: As an alternative to hand editing this file you can use the
web admin tool to configure settings for your application. Use
the Website->Asp.Net Configuration option in Visual Studio.
A full list of settings and comments can be found in
```

```

machine.config.comments usually located in
\Windows\Microsoft.Net\Framework\v2.x\Config
-->
<configuration>
<configSections>
</configSections>
<appSettings>
<!-- Experiment Storage Service Web.Config -->

<!-- ProcessAgent required Keys -->
<add key="serviceType" value="EXPERIMENT STORAGE SERVER"/>
<add key="serviceName" value="Default ESS"/>
<add key="serviceGUID" value=""/>
<add key="codebaseURL" value="http://localhost/ExperimentStorage"/>
<add key="serviceURL" value="http://localhost/ExperimentStorage/ExperimentStorage.asmx"/>
<add key="defaultPasskey" value="ESS"/>
<add key="sqlConnection" value="data source=(local);initial catalog=iLab_ESS;Integrated Security=SSPI"/>

<!-- Optional path and filename for application log. The directory must exist & be writeable to the IIS worker process
-->
<add key="logPath" value=""/>

```

Each Process Agent must have a unique GUID (Globally Unique Identifier) which will identify this particular Process Agent to all lab servers. This GUID will be supplied to the Lab Server(s) administrator so that the Lab Server can be configured to communicate with this Service Broker. We use a GUID produced on with all the '{', and '}' characters stripped out. You can generate your own GUID and enter it in the web.config or use the "Generate GUID" button on the Service Broker's self registration page (this is the preferred method).

**BUG: The Service Broker and process agent GUIDs needs to be 35 characters or less due to some legacy code in the batched architecture. If you used the "Create GUID" button to generate the GUID, please remove the hyphens "-" from the GUID before saving.**

### **ESS Self Registration**

In order for process agents to exchange credentials, their information needs to be entered into the Process Agents table. There is a self-registration web service included with each process agent. You cannot register a process agent until after you have created and edited the corresponding web.config file.

**KNOWN BUG: Currently, there is no CSS associated with the Experiment Storage self-registration page.**

- ❑ Open Internet Explorer
- ❑ Browse to <http://machinename.edu/ExperimentStorage/selfRegistration.aspx>

- ❑ Modify ESS Service Name, CodeBase & Webservice URLs as needed. Please make sure that the Service Name is unique.
  - Codebase URL: http://machinename.edu/ExperimentStorage
  - Web Service URL: http://machinename.edu/ExperimentStorage/ExperimentStorage.asmx
- ❑ If necessary, use the "Create GUID" button to generate a new GUID. Please remove the '-' from the GUID.
- ❑ Click the **Save Changes** button

## ***Userside Scheduling Services (USS)***

### ***USS Web Site***

First, create a virtual directory in IIS.

- ❑ Right-Click on **My Computer** (renamed to your computer name)
- ❑ Click **Manage**
- ❑ Expand **Services and Applications**
- ❑ Expand **Internet Information Services**
- ❑ Expand **Web Sites**
- ❑ Expand **Default Web Site**
- ❑ Right-click on **Default Web Site**
- ❑ Select **New->Virtual Directory**
- ❑ Click **Next** to enter the Virtual Directory Creation Wizard
- ❑ In the **Alias** box, type **UserSideScheduling**
- ❑ Click **Next**
- ❑ Enter `\*local_path*\iLabs\Services\UserSideScheduling` in the **Directory** box (or click the **Browse** button, and navigate to the location where your Service Broker code is). If you are using the pre-built code, please use the builds directory.
- ❑ Click **Next**
- ❑ **Access Permissions:** make sure you check the **Run Scripts**, **Execute** and **Read** permission boxes
- ❑ Click **Next**
- ❑ Click **Finish**
- ❑ You should see the **UserSideScheduling** site listed under the **Default Web Site**
- ❑ Right click on the **UserSideScheduling** website and select **Properties**
- ❑ Select the **ASP.NET** tab
- ❑ Use the pull-down list for the **ASP.NET** version and select 2.0.xxx. Where xxx is the current release version.
- ❑ Select **OK** and close the properties window
- ❑ Close the **Computer Management Window**

### ***Create the Database***

First, create the database.

- ❑ **Start-> Enterprise Manager**

- ❑ Console Root->Microsoft SQL Servers->SQL Server Group->(local)->Databases
- ❑ Right-Click on Databases
  
- ❑ New Database...
- ❑ Name: **ilab\_USS** (You can use any name you would like but you must update the web.config file if you change this name)
- ❑ Click OK

### **Grant Access to the Database**

- ❑ Expand the **Security** folder in Enterprise Manager.
- ❑ Double-click on the **NT AUTHORITY\NETWORK SERVICE** under Logins
- ❑ Click the **Database Access** tab
- ❑ Check the box next to **ilab\_USS** (or the database name you used above)
- ❑ In the box below, click the box next to **db\_owner**
- ❑ Click OK

### **USS Database Scripts**

This document contains two ways to run the database scripts for Microsoft SQL server; from within the SQL Query Analyzer or from the loader scripts that are in the DB\_Scripts directory. The preferred method of creating a new service database is to use Method 2: Using the SQL loader scripts. If you are just updating a stored procedure and not the database, it is best way is to use the SQL Query analyzer. We include Method1: Using the SQL Query Analyzer for your information.

NOTE: the SQL loader scripts do not work under Windows XP with SQL Express - if you are using Windows XP and SQL Express you will need to use the SQL Query Analyzer.

**Do NOT attempt both methods - choose one.**

METHOD 1: Using the SQL Query Analyzer:

- ❑ Start-> **Enterprise Manager**
- ❑ Console Root->Microsoft SQL Servers->SQL Server Group->(local)->Databases
- ❑ Expand Databases
- ❑ Select the **ilab\_USS** database (or what you named it) in Enterprise Manager.
- ❑ On the menu bar, select **Tools -> SQL Query Analyzer**
- ❑ In Query Analyzer, click **File -> Open...**
- ❑ Browse to **\\*local\_path\*\iLabs\Database\DB\_Scripts\ProcessAgent**

Every service must have its own Process Agent information. These tables should exist before the service-specific tables are added to each database.

- ❑ In the ProcessAgent directory select and open the **ProcessAgentTables.sql**
- ❑ Make sure the correct database is the selected database in the dropdown menu at the top - **\*This is very important,** because the scripts will be installed in the database specified by this pull-down menu.

- ❑ Click the green triangle “play” button to run the script
- ❑ You should see a message: “The command(s) completed Successfully.”
- ❑ Follow the same procedure, **File→Open**, to open and run the **ProcessAgentProcedures.sql** script.
- ❑ You should see the same message: “The command(s) completed Successfully.”
- ❑ Open and run the **SetDefaultsPA.sql** script.

Next, set up and run the USS database script.

- ❑ In Query Analyzer, Click **File -> Open...**
- ❑ Browse to database scripts directory typically `\*local_path*\iLabs\ Database\DB_Scripts\Scheduling`
- ❑ In the Scheduling directory select and open the **USS\_Tables.sql**
- ❑ Make sure the correct database is the selected database in the dropdown at the top, then click the green triangle “play” button to run the script
- ❑ You should see the message: “The command(s) completed Successfully.”
- ❑ Follow the same procedure to open and run the **USS\_Procedures.sql** script.
- ❑ Again you should see the message: “The command(s) completed Successfully.”
- ❑ Close SQL Query Manager and Enterprise Manager

#### METHOD 2: Using the SQL loader scripts:

- ❑ Open a Command Prompt window. Start -> Programs -> Accessories -> Command Prompt.
- ❑ A DOS window should open.
- ❑ Change the directory to the DBscripts directory.
- ❑ Type: `“cd \*local_path*\iLabs\ Database\DB_Scripts\”`
- ❑ Type: `“LoadUSS ilab_USS”` where `ilab_USS` is the name of your database.
- ❑ You should see something similar to the following with your directory path and database name.

```
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>loadUSS Ilab_USS

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo loading USS: USSd
b
loading USS: Ilab_USS

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>call USS_tables.bat US
Sdb

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo USS Tables: Ilab_USS

USS Tables: Ilab_USS

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d Ilab_USS -i .
\Ticketing\ProcessAgentTables.sql -o ussBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d Ilab_USS -i .
\Ticketing\SetdefaultsTicketing.sql -o ussBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d Ilab_USS -i .
\Scheduling\USS_Tables.sql -o ussBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>call USS_procedures.ba
```



```

t Ilab_USS

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo USS Procedures: U
SSdb
USS Procedures: Ilab_USS

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d Ilab_USS -i .
\Ticketing\ProcessAgentProcedures.sql -o lssBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d Ilab_USS -i .
\Scheduling\USS_Procedures.sql -o lssBuild.log
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>

```

## Web.Config

An overview of editing the web.config

Web.config is a text file containing configuration information in an XML format, residing in the root of the USS Services web site (typically in the `\*local_path*\iLabs\Services\UserSideScheduling` directory). Web.config.template ships with this release. If the web.config file already exists, delete it. Copy web.config.template and paste the file

**NOTE: Currently, changes made in the self-registration page are not updated in the web.config file.**

into the same directory. Rename the copy "web.config" and edit it using notepad.

```

<?xml version="1.0"?>
<configuration>
<appSettings>
<!--
LabSideScheduling Web.Config -->

<!--
ProcessAgent required Keys -->
<!-- THE FOLLOWING MUST BE SET FOR ALL SERVICES, used to define the service -->
<!-- The following keys specify the Webservice required Values -->
<!-- They are used by the SelfRegistration page to generate the ProcessAgent's
Bootstrap ProcessAgent Record, which must be created before installing domain credentials -->
<!-- The Service Type - Do Not Change -->
<add key="serviceType" value="SCHEDULING SERVER"/>
<!-- The name of the Service displayed for the user, please change -->
<add key="serviceName" value="Default UserScheduling Server"/>
<!-- This Service's GUID You must supply a valid GUID, see createGUID.exe, may be any string, 50 characters or less. -->
<add key="serviceGUID" value=""/>
<add key="codebaseURL" value="http://localhost/UserSideScheduling"/>
<add key="serviceURL" value="http://localhost/UserSideScheduling/UserScheduling.asmx"/>
<add key="defaultPasskey" value="USS"/>

<!-- the database connection information
NOTE: You Must set the database catalog to the database name that you created as part of the instalation process -->
<add key="sqlConnection" value="data source=(local);initial catalog=iLab_USS;Integrated Security=SSPI"/>
.....

```

### Update the Database Name

Update the database connection variable in the web.config file to use the name you chose when you created the database. Note the default database name is set to ilab\_USS. If you used a different database name, update the catalog value in the key.

```
<add key="sqlConnection"
value="data source=(local);initial catalog=i1lab_USS;Integrated Security = SSPI" />
```

### Update Default Passkey

The default Pass Key is a simple string used by process agents to exchange credential information the first time only. This can be any string you would like. We recommend that you not change this.

```
<add key="defaultPasskey" value="USS" />
```

### Web.Config Parameters to be aware of:

ServiceType	SCHEDULING SERVER - Do not Change
service Name	USS unique name - Change in selfRegister (later)
serviceGuid	USS GUID - Set in selfRegister (later)
codebaseURL	USS's applicaiton URL - Change in selfRegister (later)
serviceURL	USS's service URL- Change in selfRegister (later)
DefaultPassKey	Initial Passkey for credential exchange
sqlConnection	Database name - default ilab_USS

### Test the Scheduling Services:

- ❑ Open Internet Explorer
- ❑ Browse to <http://localhost/UserSideScheduling/UserScheduling.aspx>

Using the <http://localhost> is a good way to test your service installation, but you should also test with the fully qualified URLs in this document. The iLab architecture is designed to use fully qualified URLs, and unexpected things can happen when using localhost.

### Self Registration

Most of the setup information like the serviceName, serviceGUID and codebaseURL will now be entered in the self-registration page. This is the preferred method, but you can also enter the information into the web.config file and it will be loaded into the self-registration page for you to check. Make sure to use fully qualified URLs in the self-registration page.

Each Process Agent must have a unique GUID (Globally Unique Identifier) which will identify this particular Process Agent to all lab servers. This GUID will be supplied to the Lab Server(s) administrator so that the Lab Server can be configured to communicate with this Service Broker. We use a GUID produced on with all the '{', and '}' characters stripped out. You can generate your own GUID and enter it in the web.config or use the "Generate GUID" button on the Service Broker's self registration page (this is the preferred method).

## **USS Self Registration**

In order for process agents to exchange credentials, their information needs to be entered into the Process Agents table. There is a self-registration web service included with each process agent. You cannot register a process agent until after you have created and edited the corresponding web.config file.

- ❑ Open Internet Explorer
- ❑ Browse to <http://machinename.edu/UserSideScheduling/selfRegistration.aspx>
- ❑ Modify the USS Service Name, CodeBase & Webservice URLs as needed. Please make sure that the Service Name is unique. **Make sure to use fully qualified URLs.**
  - Codebase URL: <http://machinename.edu/UserSideScheduling>
  - Web Service URL:  
<http://machinename.edu/UserSideScheduling/UserScheduling.asmx>
- ❑ If necessary, use the "Create GUID" button to generate a new GUID
- ❑ Click the **Save Changes** button

## **Labside Scheduling Service**

### **LSS Web Site**

First, create a virtual directory in IIS.

- ❑ Right-Click on **My Computer** (renamed to your computer name)
- ❑ Click **Manage**
- ❑ Expand **Services and Applications**
- ❑ Expand **Internet Information Services**
- ❑ Expand **Web Sites**
- ❑ Expand **Default Web Site**
- ❑ Right-click on **Default Web Site**
- ❑ Select **New->Virtual Directory**
- ❑ Click **Next** to enter the Virtual Directory Creation Wizard
- ❑ In the **Alias** box, type **LabSideScheduling**
- ❑ Click **Next**
- ❑ Enter **\\\*local\_path\*\iLabs\Services\LabSideScheduling** in the **Directory** box (or click the **Browse** button, and navigate to the location where your Service Broker code is). If you used the pre-built code please use the **builds** directory.
- ❑ Click **Next**
- ❑ **Access Permissions:** make sure you check the **Run Scripts**, **Execute** and **Read** permission boxes
- ❑ Click **Next**
- ❑ Click **Finish**
- ❑ You should see the **LabSideScheduling** site listed under the **Default Web Site**
- ❑ Right click on the **LabSideScheduling** website and select **Properties**
- ❑ Select the **ASP.NET** tab

- ❑ Use the pull-down list for the **ASP.NET** version and select 2.0.xxx. Where xxx is the current release version.
- ❑ Select OK and close the properties window
- ❑ Close the Computer Management Window

### **Create the Database**

First, create the database.

- ❑ Start-> **Enterprise Manager**
- ❑ Console Root->Microsoft SQL Servers->SQL Server Group->(local)->**Databases**
- ❑ Right-Click on **Databases**
- ❑ **New Database...**
- ❑ Name: **ilab\_LSS** (You can use any name you would like but you must update the web.config file if you change this name)
- ❑ Click **OK**

### **Grant Access to the Database**

- ❑ Expand the **Security** folder in Enterprise Manager.
- ❑ Double-click on the **NT AUTHORITY\NETWORK SERVICE** under Logins
- ❑ Click the **Database Access** tab
- ❑ Check the box next to **ilab\_LSS** (or the database name you used above)
- ❑ In the box below, click the box next to **db\_owner**
- ❑ Click **OK**

### **LSS Database Scripts**

This document contains two ways to run the database scripts for Microsoft SQL server; from within the SQL Query Analyzer or from the loader scripts that are in the DB\_Scripts directory. The preferred method of creating a new service database is to use **Method 2: Using the SQL loader scripts**. If you are just updating a stored procedure and not the database, it is best way to use the SQL Query analyzer. We include Method1: Using the SQL Query Analyzer for your information.

NOTE: the SQL loader scripts do not work under Windows XP with SQL Express - if you are using Windows XP and SQL Express you will need to use the SQL Query Analyzer.

**Do NOT attempt both methods - choose one.**

METHOD 1: Using the SQL Query Analyzer:

- ❑ Start-> **Enterprise Manager**
- ❑ Console Root->Microsoft SQL Servers->SQL Server Group->(local)->**Databases**
- ❑ Expand **Databases**
- ❑ Select the **ilab\_LSS** database (or what you named it) in Enterprise Manager.
- ❑ On the menu bar, select **Tools -> SQL Query Analyzer**
- ❑ In Query Analyzer, click **File -> Open...**
- ❑ Browse to database scripts directory typically `\*local_path*\iLabs\ Database\DB_Scripts\ProcessAgent`

Every service must have its own Process Agent information. These tables should exist before the service-specific tables are added to each database.

- In the ProcessAgent directory select and open the **ProcessAgentTables.sql**
- Make sure the correct database is the selected database in the dropdown menu at the top - **\*This is very important**, because the scripts will be installed in the database specified by this pull-down menu.
- Click the green triangle "play" button to run the script
- You should see a message: "The command(s) completed Successfully."
- Follow the same procedure, **File→Open**, to open and run the **ProcessAgentProcedures.sql** script.
- You should see the same message: "The command(s) completed Successfully."
- Open and run the **SetDefaultsPA.sql** script.

Next, set up and run the LSS database scripts, typically located in `\*local_path*\iLabs\ Database\DB_Scripts\Scheduling`.

- In Query Analyzer, click **File -> Open...**
- In the Scheduling directory select and open the **LSS\_Tables.sql**
- Make sure the correct database is the selected database in the dropdown at the top, then click the green triangle "play" button to run the script
- You should see the message: "The command(s) completed Successfully."
- Follow the same procedure to open and run the **LSS\_Procedures.sql** script.
- Again, you should see the message: "The command(s) completed Successfully."
- Close SQL Query Manager and Enterprise Manager

#### METHOD 2: Using the SQL loader scripts:

- Open a Command Prompt window. Start -> Programs -> Accessories -> Command Prompt.
- A DOS window should open.
- Change the directory to the DBscripts directory.
- Type: `"cd :*\iLabs\ Database\DB_Scripts\"`
- Type: `"LoadLSS ilab_LSS"` where ilab\_LSS is the name of your database.
- You should see something similar to the following with your directory path and database name.

```
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>loadLSS LSSdb
```

```
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo populating LSS LS
Sdb
populating LSS LSSdb
```

```
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>call LSS_tables.bat LS
Sdb
```

```
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo LSS Tables: LSSdb
LSS Tables: LSSdb
```

```
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d LSSdb -i .
```

```

\Ticketing\ProcessAgentTables.sql -o lssBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d LSSdb -i .
\Ticketing\SetdefaultsTicketing.sql -o lssBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d LSSdb -i .
\Scheduling\LSSCreateTableScript.sql -o lssBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>call LSS_procedures.ba
t LSSdb

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo LSS Procedures: L
SSdb
LSS Procedures: LSSdb

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d LSSdb -i .
\Ticketing\ProcessAgentProcedures.sql -o lssBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d LSSdb -i .
\Scheduling\LSSstoreprocedure.sql -o lssBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>

```

## Web.Config

An overview of editing the web.config

Web.config is a text file containing configuration information in XML format, residing in the root of the LSS Services web site (typically in the `\*local_path*\iLabs\Services\LabSideScheduling` directory). `Web.config.template` ships with this release. If a `web.config` file already exists, please delete it. Copy `web.config.template` and paste the file into the same directory. Rename the copy "web.config" and edit it using

**NOTE: Currently, changes made in the self-registration page are not updated in the web.config file.**

notepad.

```

<?xml version="1.0"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <appSettings>
    <!--
    LabSideScheduling Web.Config -->
    <!--
    ProcessAgent required Keys -->
    <!-- THE FOLLOWING MUST BE SET FOR ALL SERVICES, used to define the service -->
    <!-- The following keys specify the Webservice required Values -->
    <!-- They are used by the SelfRegistration page to generate the ProcessAgent's
    Bootstrap ProcessAgent Record, which must be created before installing domain credentials -->

    <add key="serviceType" value="LAB SCHEDULING SERVER"/>
    <!-- The name of the Service displayed for the user, please change -->
    <add key="serviceName" value="Default LabScheduling Server"/>
    <!-- This Service's GUID You must supply a valid GUID, see createGUID.exe, may be any string, 50 characters or less. -->
    <add key="serviceGUID" value=""/>
    <add key="codebaseURL" value="http://localhost/LabSideScheduling"/>
    <add key="serviceURL" value="http://localhost/LabSideScheduling/LabScheduling.aspx"/>
    <add key="defaultPasskey" value="LSS"/>

    <!-- the database connection information
    NOTE: You Must set the database catalog to the database name that you created as part of the installation process -->

```

```
<add key="sqlConnection" value="data source=(local);initial catalog=iLab_LSS;Integrated Security=SSPI"/>
```

### Update the Database Name

Update the database connection variable in the web.config file to use the name you chose when you created the database. Note the default database name is set to ilab\_LSS. If you used a different database name, update the catalog value in the key.

```
<add key="sqlConnection"
value="data source=(local);initial catalog=ilab_LSS;Integrated Security = SSPI" />
```

### Update Default Passkey

The default Pass Key is a simple string used by process agents to exchange credential information the first time only. This can be any string you would like. We recommend that you not change this.

```
<add key="defaultPasskey" value="LSS" />
```

### Web.Config Parameters to be aware of:

ServiceType	LAB SCHEDULING SERVER -Do Not Change
serviceName	LSS unique name - change in selfRegister (later)
serviceGuid	This is the LSS's Guid - change in selfRegister (later)
codebaseURL	LSS application URL - change in selfRegister (later)
serviceURL	LSS's service URL
DefaultPassKey	Initial Passkey for credential exchange
sqlConnection	Database name - default ilab_LSS

### Test the Scheduling Services:

- ❑ Open Internet Explorer
- ❑ Browse to <http://localhost/LabSideScheduling/LabScheduling.asmx>

Using the <http://localhost> is a good way to test your service installation, but you should also test with the fully qualified URLs in this document. The iLab architecture is designed to use fully qualified URLs, and unexpected things can happen when using localhost.

### Self Registration

Most of the setup information like the serviceName, serviceGUID and codebaseURL will now be entered in the self-registration page. This is the preferred method, but you can also enter the information into the web.config file and it will be loaded into the self-registration page for you to check. Use fully qualified URLs.

Each Process Agent must have a unique GUID (Globally Unique Identifier) which will identify this particular Process Agent to all lab servers. This GUID will be supplied to the Lab Server(s) administrator so that the Lab Server can be configured to communicate

with this Service Broker. We use a GUID produced on with all the '{', and '}' characters stripped out. You can generate your own GUID and enter it in the web.config use the "Generate GUID" button on the Service Broker's self registration page (this is the preferred method).

## ***LSS Self Registration***

In order for process agents to exchange credentials, their information needs to be entered into the Process Agents table. There is a self-registration web service included with each process agent. You should not self register a process agent until after you have created and edited the corresponding web.config file.

- ❑ Open Internet Explorer
- ❑ Browse to <http://localhost/LabSideScheduling/selfRegistration.aspx>
- ❑ Modify Service Name, CodeBase & Webservice URLs as needed. Please make sure that the Service Name is unique. **Be sure to use fully qualified URLs.**
  - Codebase URL: <http://machinename.edu/LabSideScheduling>
  - Web Service URL:  
<http://machinename.edu/LabSideScheduling/LabScheduling.asmx>
- ❑ If necessary, use the "Create GUID" button to generate a new GUID
- ❑ Click the **Save Changes** button

## ***Time Of Day Lab Server***

### ***TimeOfDay Web Site***

First, create a virtual directory in IIS.

- ❑ Right-Click on **My Computer** (renamed to your computer name)
- ❑ Click **Manage**
- ❑ Expand **Services and Applications**
- ❑ Expand **Internet Information Services**
- ❑ Expand **Web Sites**
- ❑ Expand **Default Web Site**
- ❑ Right-click on **Default Web Site**
- ❑ Select **New->Virtual Directory**
- ❑ Click **Next** to enter the Virtual Directory Creation Wizard
- ❑ In the **Alias** box, type **TimeOfDay**
- ❑ Click **Next**
- ❑ Enter `\*local_path*\iLabs\Services\TimeOfDay` in the **Directory** box (or click the **Browse** button, and navigate to the location where your Service Broker code is). If you used the pre-built code, please use the **builds** directory.
- ❑ Click **Next**
- ❑ **Access Permissions:** make sure you check the **Run Scripts**, **Execute** and **Read** permission boxes
- ❑ Click **Next**



- ❑ Click **Finish**
- ❑ You should see the **TimeOfDay** site listed under the Default Web Site
- ❑ Right click on the **TimeOfDay** website and select **Properties**
- ❑ Select the **ASP.NET** tab
- ❑ Use the pull-down list for the **ASP.NET** version and select 2.0.xxx. Where xxx is the current release version.
- ❑ Select OK and close the properties window
- ❑ Close the Computer Management Window

### **Create the Database**

First, create the database.

- ❑ Start-> **Enterprise Manager**
- ❑ Console Root->Microsoft SQL Servers->SQL Server Group->(local)->**Databases**
- ❑ Right-Click on **Databases**
- ❑ **New Database...**
- ❑ Name: **ilab\_TOD** (You can use any name you would like but you must update the web.config file if you change this name)
- ❑ Click **OK**

### **Grant Access to the Database**

- ❑ Expand the **Security** folder in Enterprise Manager.
- ❑ Double-click on the **NT AUTHORITY\NETWORK SERVICE** under Logins
- ❑ Click the **Database Access** tab
- ❑ Check the box next to **ilab\_TOD** (or the database name you used above)
- ❑ In the box below, click the box next to **db\_owner**
- ❑ Click **OK**

### **TOD Database Scripts**

This document contains two ways to run the database scripts for Microsoft SQL server; from within the SQL Query Analyzer or from the loader scripts that are in the DB\_Scripts directory. The preferred method of creating a new service database is to use **Method 2: Using the SQL loader scripts**. If you are just updating a stored procedure and not the database, it is best to use the SQL Query analyzer. We include Method1: Using the SQL Query Analyzer for your information.

NOTE: the SQL loader scripts do not work under Windows XP with SQL Express - if you are using Windows XP and SQL Express you will need to use the SQL Query Analyzer.

**Do NOT attempt both methods - choose one.**

METHOD1: Using the SQL Query Analyzer:

- ❑ Start-> **Enterprise Manager**
- ❑ Console Root->Microsoft SQL Servers->SQL Server Group->(local)->**Databases**
- ❑ Expand **Databases**
- ❑ Select the **ilab\_TOD** database (or what you named it) in Enterprise Manager.

- ❑ On the menu bar, Select Tools -> SQL Query Analyzer
- ❑ In Query Analyzer, Click File->Open...
- ❑ Browse to database scripts directory typically (\\*local\_path\*\iLabs\ Database\DB\_Scripts\ProcessAgent)

Every service must have its own Process Agent information. These tables should exist before the service-specific tables are added to each database.

- ❑ In the ProcessAgent directory select and open the ProcessAgentTables.sql
- ❑ Make sure the correct database is the selected database in the dropdown at the top - **\*This is very important,** because the scripts will be installed in the database specified by this pull-down menu.
- ❑ Click the green triangle "play" button to run the script
- ❑ You should see a message: "The command(s) completed Successfully."
- ❑ Follow the same procedure, **File→Open**, to open and run the ProcessAgentProcedures.sql script.
- ❑ You should see the same message: "The command(s) completed Successfully."
- ❑ Open and run the SetDefaultsPA.sql script.
- ❑ Close SQL Query Manager and Enterprise Manager

For the Time Of Day Server, there are no process agent-specific database scripts that need to be run.

#### METHOD 2: Using the SQL loader scripts:

- ❑ Open a Command Prompt window. Start -> Programs -> Accessories -> Command Prompt.
- ❑ A DOS window should open.
- ❑ Change the directory to the DBscripts directory.
- ❑ Type: "D:" to change the disk if you need to
- ❑ Type: "cd :\*\\*local\_path\*\iLabs\ Database\DB\_Scripts\"
- ❑ Type: "LoadTOD ilab\_TOD" where ilab\_TOD is the name of your database.
- ❑ You should see something similar to the following with your directory path and database name.

```
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>loadTOD Ilab_TOD

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo populating TOD TO
Ddb
populating TOD Ilab_TOD

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>call TOD_tables.bat TO
Ddb

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo populating TOD TO
Ddb
populating TOD Ilab_TOD

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d Ilab_TOD -i .
\Ticketing\ProcessAgentTables.sql -o TODBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d Ilab_TOD -i .
\Ticketing\SetdefaultsTicketing.sql -o TODBuild.log
```

```

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>call TOD_procedures.bat Ilab_TOD

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo TOD Procedures: TODdb
TOD Procedures: Ilab_TOD

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d Ilab_TOD -i .\Ticketing\ProcessAgentProcedures.sql -o TODBuild.log
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>loadTOD Ilab_TOD

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo populating TOD TODdb
populating TOD Ilab_TOD

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>call TOD_tables.bat TODdb

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo populating TOD TODdb
populating TOD Ilab_TOD

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d Ilab_TOD -i .\Ticketing\ProcessAgentTables.sql -o TODBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d Ilab_TOD -i .\Ticketing\SetdefaultsTicketing.sql -o TODBuild.log

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>call TOD_procedures.bat Ilab_TOD

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>echo TOD Procedures: TODdb
TOD Procedures: Ilab_TOD

D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>isqlw -E -d Ilab_TOD -i .\Ticketing\ProcessAgentProcedures.sql -o TODBuild.log
D:\iLabsInteractive\iLabs_NET20\iLabs\Database\DB_Scripts>

```

## Web.Config

An overview of editing the web.config

Web.config is a text file containing configuration information in XML format, residing in the root of the InteractiveTimeOfDay web site (typically in the %local\_path%\iLabs\Services\InteractiveTimeOfDay directory). Web.config.template ships with this release. If a web.config already exists, please delete it. Copy web.config.template and paste the file into the same directory. Rename the copy "web.config" and edit it using

**NOTE: Currently, changes made in the self-registration page are not updated in the web.config file.**

notepad. Use only fully qualified URLs in the web.config file.

```
<?xml version="1.0"?>
<!--
```

Note: As an alternative to hand editing this file you can use the web admin tool to configure settings for your application. Use the Website->Asp.Net Configuration option in Visual Studio.

A full list of settings and comments can be found in machine.config.comments usually located in \Windows\Microsoft.Net\Framework\v2.x\Config

```
-->
<configuration>
  <appSettings>
    <!--
      Interactive TimeOfDay Web.Config -->

    <!--
      ProcessAgent required Keys -->
    <!-- THE FOLLOWING MUST BE SET FOR ALL SERVICES, used to define the service -->
    <!-- The following keys specify the WebService required Values -->
    <!-- They are used by the SelfRegistration page to generate the ProcessAgent's
      Bootstrap ProcessAgent Record, which must be created before installing domain credentials -->

    <add key="serviceType" value="LAB SERVER"/>
    <!-- The name of the Service displayed for the user, please change -->
    <add key="serviceName" value="Default TimeOfDay Server"/>
    <!-- This Service's GUID You must supply a valid GUID, see createGUID.exe, may be any string, 50 characters or less. -->
    <add key="serviceGUID" value=""/>
    <add key="codebaseURL" value="http://localhost/TimeOfDay"/>
    <add key="serviceURL" value="http://localhost/TimeOfDay/TimeOfDayWebService.asmx"/>
    <add key="defaultPasskey" value="TOD"/>

    <!-- the database connection information
      NOTE: You Must set the database catalog to the database name that you created as part of the installation process -->
    <add key="sqlConnection" value="data source=(local);initial catalog=iLab_TOD;Integrated Security=SSPI"/>
.....
```

### Update the Database Name

Update the database connection variable in the web.config file to use the name you chose when you created the database. Note the default database name is set to iLab\_TOD. If you used a different database name, update the catalog value in the key.

```
<add key="sqlConnection"
value="data source=(local);initial catalog=iLab_TOD;Integrated Security = SSPI" />
```

Note: there is an mistake in the web.config file in the release ... check the database name in the web.config file and make sure that the database name you used to create the database matches exactly including capitalization.

### Update Default Passkey

The default Pass Key is a simple string used by process agents to exchange credential information the first time only. This can be any string you would like. We recommend that you not change this.

```
<add key="defaultPasskey" value="TOD" />
```

### Web.Config Parameters to be aware of:

serviceType	LAB SERVER - Do Not Change
serviceName	Lab Server unique name - Change in selfRegistration
serviceGuid	This is the LSS's Guid name - Set in selfRegistration
codebaseURL	Lab Server URL - Change in selfRegistration
serviceURL	Lab Server's service URL - Change in selfRegistration

DefaultPassKey  
sqlConnection

Initial Passkey for credential exchange  
Database name - default ilab\_TOD

### **Test the Time Of Day Server:**

- ❑ Open Internet Explorer
- ❑ Browse to <http://localhost/TimeOfDay/TimeOfDayWebService.asmx>

Using the <http://localhost> is a good way to test your service installation, but you should also test with the fully qualified URLs in this document. The iLab architecture is designed to use fully qualified URLs, and unexpected things can happen when using localhost.

### Self Registration

Most of the setup information like the serviceName, serviceGUID and codebaseURL will now be entered in the web service self-registration page. This is the preferred method, but you can also enter the information into the web.config file and it will be loaded into the self-registration page for you to check.

Each Process Agent must have a unique GUID (Globally Unique Identifier) which will identify this particular Process Agent to all lab servers. This GUID will be supplied to the Lab Server(s) administrator so that the Lab Server can be configured to communicate with this Service Broker. We use a GUID produced on with all the '{', and '}' characters stripped out. You can generate your own GUID and enter it in the web.config or use the "Generate GUID" button on the Service Broker's self registration page (this is the preferred method).

### **TOD Self Registration**

In order for process agents to exchange credentials, their information needs to be entered into the Process Agents table. There is a self-registration webservice included with each process agent. You cannot register a process agent until after you have created and edited the corresponding web.config file.

- ❑ Open Internet Explorer
- ❑ Browse to <http://localhost/TimeOfDay/selfRegistration.aspx>
- ❑ Modify Service Name, CodeBase & Webservice URLs as needed. Please make sure that the Service Name is unique. **Be sure to use fully qualified URLs.**
  - Codebase URL: <http://machinename.edu/TimeOfDay>
  - Web Service URL: <http://machinename.edu/TimeOfDay/TimeOfDayWebService.asmx>
- ❑ If necessary, use the "Create GUID" button to generate a new GUID
- ❑ Click the **Save Changes** button

**KNOWN BUG:** Currently, there is no CSS associated with the Time Of Day Interactive Experiment self-registration page.

## Bootstrapping

Bootstrapping is the process of enabling the various iLab services to recognize each other and exchange credentials.

This initial bootstrapping process will assume that you are logged into the iLab Service Broker as a SuperUser and that all servers are in the same domain. To manage the USS and LSS directly, you will need to be logged in as the correct management group - not as SuperUser

This document will set up and configure the different services for the Service Broker, and configure the Time Of Day (TOD) test lab.

**NOTE: This document assumes that you will replace your computer's name and fully qualified URL where ever you see "machinename.edu".**

**NOTE: Before beginning the bootstrapping process, any and all current security patches and internet browser updates must be downloaded and installed.**

## Users and Groups

At the time of installation, the database script creates one default user and group called "SuperUser". This user is the Service Broker's administrator and has full permissions. Any user added to this group will also have full administration permissions on the Service Broker.

In addition to the SuperUser group, you will need administrator groups for Process Agent services such as the USS and LSS, and a management group for each user group. Members of the administration groups will have SuperUser permissions on the specified Process Agent; typically these are the lab or department administrators. Members of the management groups will have control over a particular group of students; typically these are the class TAs.

Create an administrator group for administering the Userside Scheduling Server (USS) called USS\_admin

- ❑ Navigate to <http://machinename.edu/ilabservicebroker>
- ❑ Log in as SuperUser
- ❑ Click on the *Users & Groups* tab on the main menu of the Service Broker.
- ❑ In the sub-menu, select the *Administer Groups* tab.
- ❑ Click on the *Add Service Admin Group* button. A popup window should appear.
- ❑ Enter the Group Name "USS\_admin".

- ❑ Enter a Description, "**Administration group for the USS**".
- ❑ Enter the administrator's contact email address.
- ❑ This service does not need a request group, so do not click on Need Request group.
- ❑ This is a Administer group, so you do not need to add or change the Direct Group Membership area.
- ❑ Click on the ***Save Changes*** button. The new group should be listed on the Administer Groups page.

Create an Administrator group for administering the Labside Scheduling Server (LSS) called LSS\_admin

- ❑ On the Administer Groups page, click on the ***Add Service Admin Group*** button. A popup window should appear.
- ❑ Enter the Group Name "**LSS\_admin**".
- ❑ Enter a Description, "**Administration group for the LSS**".
- ❑ Enter the administrator's contact email address.
- ❑ Click on the ***Save Changes*** button. The new LSS\_admin group should be listed on the Administer Groups page.

Create a management group for managing the Lab Server (LS) configurations on the Labside Scheduling Server (LSS). In our case we will be managing the Time Of Day Lab Server configuration on the LSS, but this group will manage all the Lab Servers using this LSS.

- ❑ On the Administer Groups page, click on the ***Add Service Admin Group*** button. A popup window should appear.
- ❑ Enter the Group Name "**Manage\_LS\_on\_LSS**".
- ❑ Enter a Description, "**Managers group for managing the Lab Servers using the LSS**".
- ❑ Enter the administrator's contact email address.
- ❑ Click on the ***Save Changes*** button. The new group should be listed on the Administer Groups page.

Create a management group for managing the Time Of Day users group on the Userside Scheduling Server (USS)

- ❑ On the Administer Groups page, click on the ***Add Service Admin Group*** button. A popup window should appear.
- ❑ Enter the Group Name "**Manage\_Users\_on\_USS**".
- ❑ Enter a Description, "**Managers group for managing the Time of Day lab users on the USS**".
- ❑ Enter the administrator's contact email address.
- ❑ Click on the ***Save Changes*** button. The new group should be listed on the Administer Groups page.

Create a group for users of the Time Of Day Experiment



- ❑ On the Administer Groups page, click on the **Add Group** button. A popup window should appear.
- ❑ Enter the Group Name **"Experiment\_Group"**.
- ❑ Enter a Description, **"User Experiment Group for Time Of Day Interactive Experiment"**.
- ❑ Enter the administrator's contact email address.
- ❑ The Time Of Day Interactive Lab service does not need any additional groups, so do not click on Need Request group or Need course Staff Group.
- ❑ Click on the **Save Changes** button. The new **Experiment\_Group** group should be listed on the Administer Groups page.
- ❑ Next, we need to give this group permission to create and store experiments. Click on the **Edit** button for the **Experiment\_Group**. A popup window will appear.

- ❑ In the **Experiment Grants** area under **Experiment\_Group** click on the **R**, **W** and **C** boxes.

## Process Agents

Process Agents are services (web servers or web service calls) that that the Service Broker can use. Setting up a new Process Agent is a two step process. We have tried to automate this process as much as possible to eliminate the need for human data entry. For each Process Agent (except the Service Broker) you will need to contact the Process Agent administrator via email and get the web services URL and an initial passkey. With these two items (web services URL and passkey) you should be able to install domain credentials.

## Service Broker

The Service Broker Information page can be found on the Service Broker Information tab in the Service & Clients sub-menu. This page provides the self-Registration function for the iLab Service Broker. Initially, this information is retrieved from the Service Broker's web.config file. On this page, you can create a new GUID for the Service Broker, if needed, and edit the defaults for serverName, serviceURL, applicationURL.

If you have already modified the defaults in the Service Broker's web.config file, you can use this page to make sure the information is correct. Remember all URLs must be fully qualified including protocol and actual host DNS name or IP address. These variables that used for all communication between this service and all other services.

This information is used to create process agent credentials and can not be changed once an external process agent is registered. Note: If you are developing and all services will reside on the development machine you may leave the machine name as localhost but this is not recommended. We recommend using full qualified URLs at all times.

If you have not already done so, register the Service Broker.

Add Service Broker's Information to the Process Agents table.

- ❑ Click on the *Service Broker Information* tab.
- ❑ Modify Service Name, CodeBase & Webservice URLs as necessary. Please be sure that the Service Name is unique. Be sure to use fully qualified URLs.
  - Codebase URL: http://machinename.edu/iLabServiceBroker
  - Web Service URL:  
http://machinename.edu/iLabServiceBroker/iLabServiceBroker.asmx
- ❑ If necessary, use the "Create GUID" button to generate a new GUID and remove the hyphens.

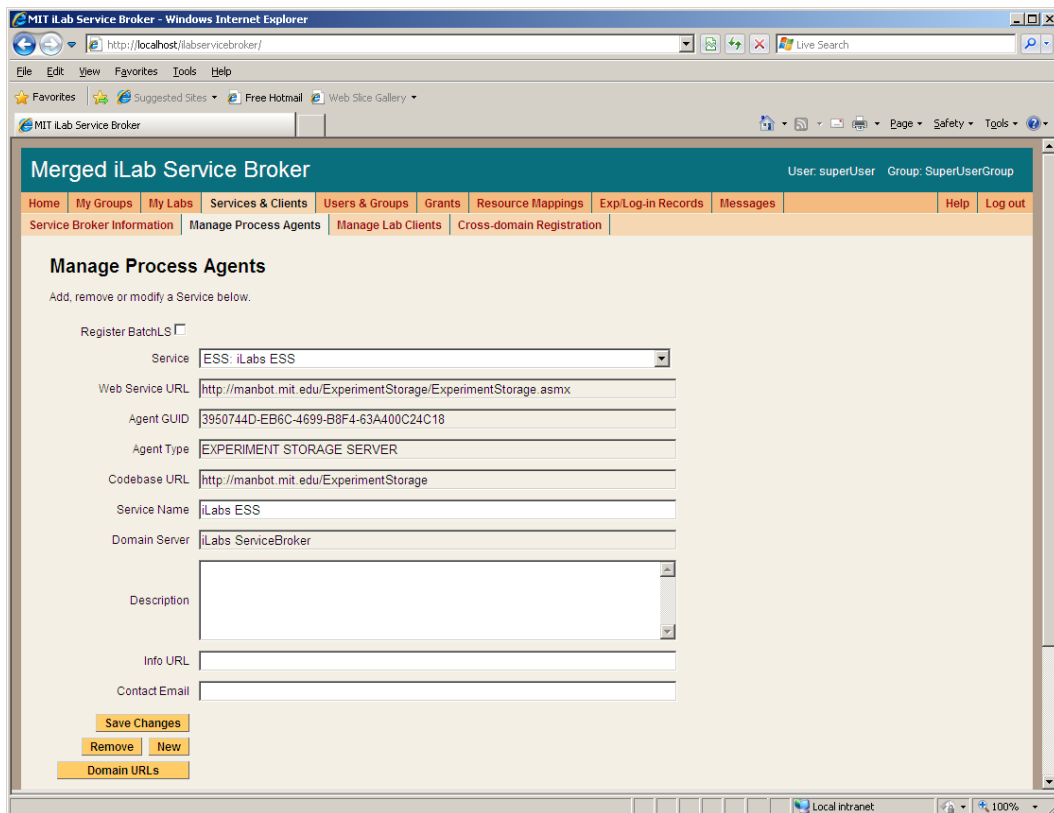
**BUG: The Service Broker GUID needs to be 35 characters or less due to some legacy code in the batched architecture. If you used the "Create GUID" button to generate the GUID, please remove the hyphens "-" from the GUID before saving.**

- ❑ Click on the *Save Changes* button to add the Service Brokers credentials to the Process Agents table.

Next, complete the Service Broker information in the Process Agent's table

- ❑ Click on the *Services & Clients* tab
- ❑ Click on the *Manage Process Agents* tab
- ❑ Use the *Services* pull-down menu to select the Service Broker.
- ❑ Enter your University Name and Service Broker name into the *description* field, if it is not already entered.

- ❑ Enter a URL for service information in the info\_URL box, if available
- ❑ Enter a contact email into the correct box.



- ❑ Click *Save Changes*. You should see a green box at the top of the page stating that the record has been modified.

For the Service Broker, there are no additional Domain URLs that need to be entered. Changes made from the Manage Process Agents page will **NOT** be reflected in the web.config file.

## Experiment Storage Service (ESS)

To add the Experiment Storage Service to the Service Broker, you will need the web service URL and an initial passkey for the ESS. This should be exchanged via email if you are not the owner of these services. For this document, we will assume that you are the administrator for all services and that you will be using the default initial passkey values.

Add the ESS to the Service Broker and exchange credentials.

- ❑ Click on the *Manage Process Agents* page.
- ❑ Enter ESS info (web services URL & passkey) into the correct fields. Default Web Services URL:  
http://machinename.edu/ExperimentStorage/ExperimentStorage.asmx  
Default Passkey: ESS
- ❑ Click on the *Install domain credentials* button.

- ❑ A register web service call is made using the web services URL and initial passkey to get the Process Agent's information and credentials.
- ❑ You should see a green message bar at the top of the page stating that a "Relationship has been created and saved."

Complete the information record for the ESS.

- ❑ You can use the **description area** to make notes about the process agent as needed. Make any other necessary changes.
- ❑ For the ESS, you do not need to select an Admin Group pull down list.
- ❑ Click *Save Changes*. You should see a green box at the top of the page stating that the record has been successfully updated.

The ESS does not need to have any Domain URLs entered.

### ***Userside Scheduling Server (USS)***

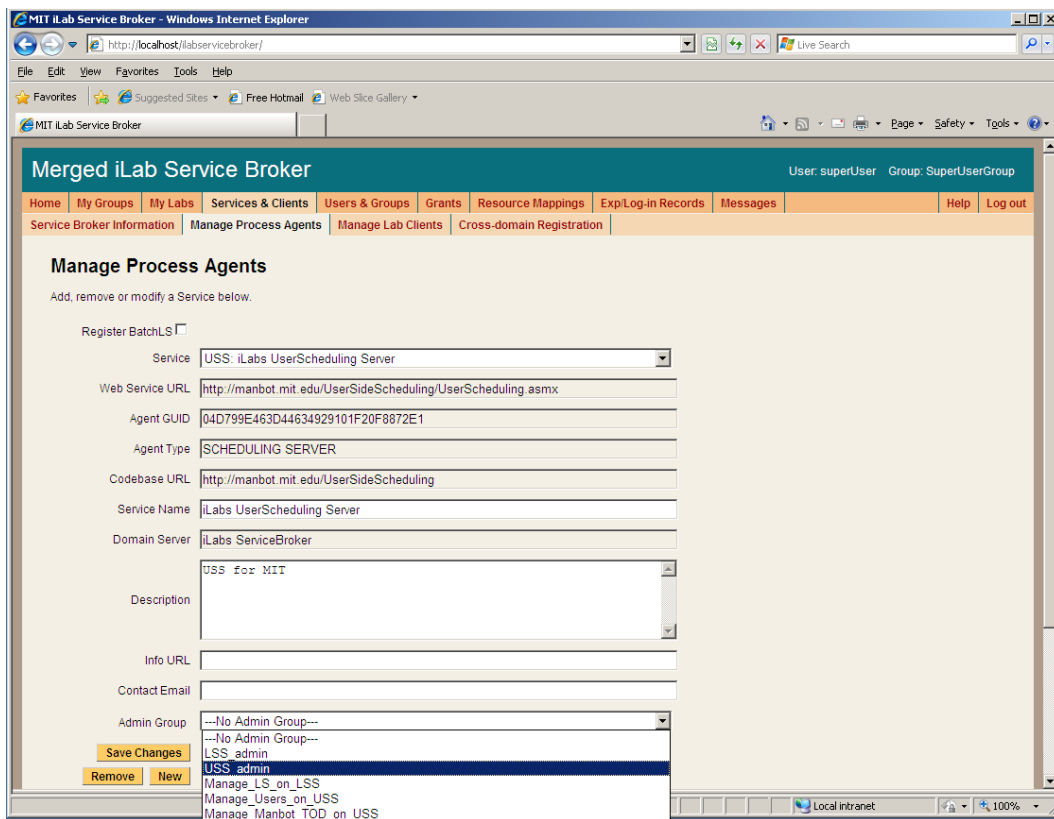
To add the User-side Scheduling Service to the Service Broker, you will need the web service URL and an initial passkey for the USS. This should be exchanged via email if you are not the owner of these services. For this document, we will assume that you are the administrator for all services and that you will be using the default initial passkey values.

Add the USS to the Service Broker and exchange credentials.

- ❑ Click on the *Manage Process Agents* tab
- ❑ Enter USS info (web services URL & passkey) into the correct fields. Default Web Services URL:  
http://machinename.edu/UserSideScheduling/UserScheduling.aspx
- ❑ Default Passkey: USS
- ❑ Click on the *Install domain credentials* button.
- ❑ A register web service call is made using the web services URL and initial passkey, to get the process Agents information and credentials.
- ❑ You should see a green message bar at the top of the page stating that a "Relationship has been created and saved."

Associate the USS Administration Group with the USS.

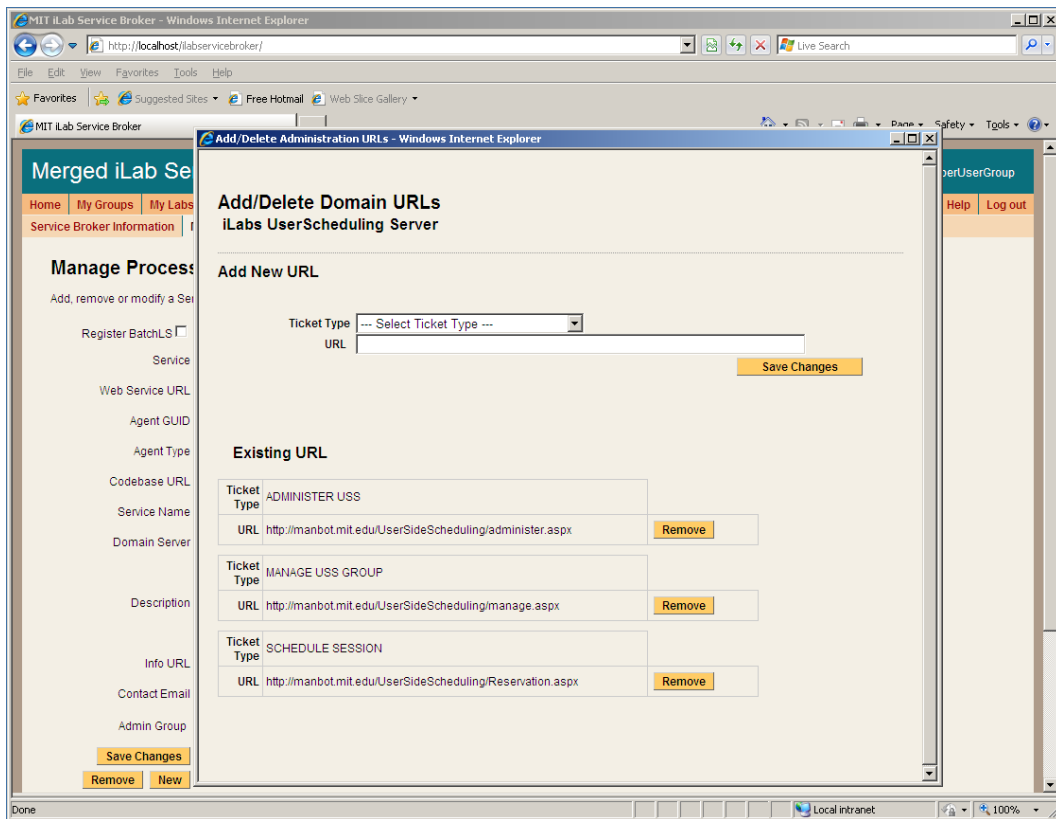
- ❑ Still on the "Manage Process Agents" page, select the "USS\_admin" group from the **Admin Group** pull down list. (This is the administrator group that will manage the USS that was created earlier.)



- ❑ You can use the **description** area to make notes about the process agent as needed. Enter “USS for your\_university name” into the description area.
- ❑ Click **Save Changes**. You should see a green message bar at the top of the page stating that the record has been successfully updated.

Domain URLs are needed for administering and managing the USS. These URLs are automatically generated and entered into the database. To check the domain URLs:

- ❑ Click on the *Domain URLs* button. The domain URLs will be displayed in the Existing URL area.
- ❑ For the USS, there should be three domain URLs. **Administer\_USS**, **Manage\_USS\_Group**, and **Schedule\_Session**



- ❑ Close the Domain URLs popup window.

## ***Labside Scheduling Server (LSS)***

To add the Labside Scheduling Service to the Service Broker, you will need the web service URL for the LSS and an initial passkey. This should be exchanged via email if you are not the owner of these services. For this document, we will assume that you are the administrator for all services and that you will be using the default initial passkey values.

Add the LSS to the Service Broker and exchange credentials.

- ❑ Click on the *Manage Process Agents* tab
- ❑ Enter LSS info (web services URL & passkey) into the correct fields.
- ❑ Default Web Services URL:  
http://machinename.edu/LabSideScheduling/LabScheduling.asmx
- ❑ Default Passkey: LSS
- ❑ Click on the *Install domain credentials* button.
- ❑ A register web service call is made using the web services URL and initial passkey, to get the process Agents information and credentials.
- ❑ You should see a green message bar at the top of the page stating that a "Relationship has been created and saved."

Associate the LSS Administration Group with the USS.

- ❑ Select the “LSS\_admin” group from the **Admin Group** pull down list. (This is the administration group that will manage the LSS.)
- ❑ You can use the description area to make notes about the process agent as necessary. Enter “LSS for your\_university name” into the description area.
- ❑ Click *Save Changes*. You should see a green message bar at the top of the page stating that the record has been successfully updated.

Domain URLs are needed for administering and managing the USS. These URLs are automatically generated and entered into the database. To check the domain URLs:

- ❑ Click on the *Domain URLs* button. The domain URLs will be displayed in the Existing URL area.
- ❑ For the USS, there should be two domain URLs: **Administer\_LSS** and **Manage\_Lab**.
- ❑ Close the Domain URLs popup window.

### **Time Of Day Lab Server (LS)**

To add the Lab Server to the Service Broker, you will need the web service URL and an initial passkey for the Lab Server. This should be exchanged via email if you are not the owner of these services. For this document, we will assume that you are the administrator for all services and that you will be using the default initial passkey values.

Add the ToD Lab Server to the Service Broker and exchange credentials.

- ❑ Click on the *Manage Process Agents* tab
- ❑ Enter LS info (web services URL & passkey) into the correct fields.
- ❑ Default Web Services URL:  
http://machinename.edu/TimeOfDay/TimeOfDayWebService.asmx
- ❑ Default Passkey: TOD
- ❑ Click on the *Install domain credentials* button.
- ❑ A register web service call is made using the web services URL and initial passkey, to get the process Agents information and credentials.
- ❑ You should see a green message bar at the top of the page stating that a “Relationship has been created and saved.”

Associate the LSS and management group with this Lab Server.

- ❑ Select the LSS to associate with this Lab Server from the pull-down list
- ❑ Select a Management group (Manage\_LS\_on\_LSS) from the pull-down list
- ❑ Click the *Associate* button

The Lab Server does not need an Admin Group nor does it have any Domain URLs entered.

## Time Of Day Lab Client

### Lab Client Record

Create a Time Of Day Lab Client record.

- ❑ Click the *Manage Lab Clients* tab.
- ❑ Lab Client Name: **Time Of Day Interactive Client**
- ❑ Click the **Make Guid** button to generate a new GUID for this client. GUIDs can not be changed once the client record is saved.
- ❑ Version: **1.5**
- ❑ Short Description: **Interactive Time Of Day Client**
- ❑ Long Description: **Interactive Time Of Day Client with elephant image.**
- ❑ Contact First Name: Your Name
- ❑ Contact Last Name: Your Name
- ❑ Contact Email: your@email\_address
- ❑ Click on the box next to Needs Scheduling
- ❑ Click on the box next to Needs ESS

MIT iLab Service Broker

### Manage Lab Clients

Add, remove or modify a lab client below.

Lab Client: Time of Day Interactive Client

Lab Client Name: Time of Day Interactive Client

Client GUID: 1FAA4DB4B49C4F018658504AD1844312

Version: 1.5

Short Description: Interactive Time of Day Client

Long Description: Interactive Time of Day Client with elephant image

Contact First Name: ContactFirstName

Contact Last Name: ContactLastName

Contact Email: contact\_name@AdminEmail.com

Documentation URL:

Notes:

Client Type: Interactive Redirect

Loader Script: http://manbot.mit.edu/TimeOfDay/Redirect.aspx

Needs ESS:  iLabs ESS [Dissociate](#)

Needs Scheduling:  iLabs UserScheduling Server [Dissociate](#)

Associated Lab Servers

- Name: iLabs TimeOfDay Server  
Desc: http://manbot.mit.edu/TimeOfDay/TimeOfDayWebService.a  
[Edit List](#)

Additional Lab Client Links

Name	URL	URL	Description
------	-----	-----	-------------

[Associate Groups](#) [Add/Edit Links](#)

[Save Changes](#) [Remove](#) [New](#)

- ❑ Click **Save Changes**
- ❑ You should see a green message bar at the top of the page stating that the Lab Client has been added."

### Client Loader Script

The Time Of Day Interactive Client is an HTML redirect client. Make sure to use fully qualified URLs.



Set up the Client loader scripts.

You should still be on the Manage Lab Clients page.

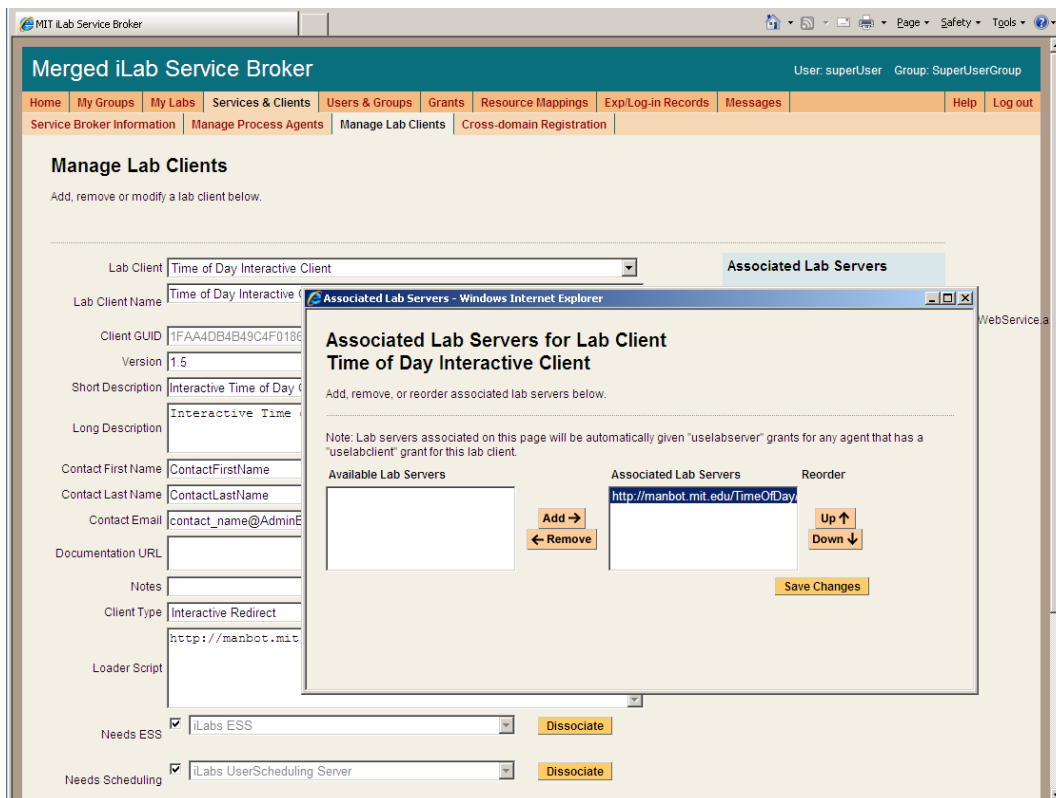
- ❑ On the Time Of Day Interactive Clients page, select “Interactive Redirect” from the Client Type pull-down list.
- ❑ Enter the redirect URL “http://machinename.edu/TimeOfDay/Redirect.aspx” into Loader Script area.
- ❑ Click **Save Changes**
- ❑ You should see a green message bar at the top of the page stating that the record has been modified.

## Associate the Lab Server

You should still be on the Manage Lab Clients page.

Associate the Time Of Day Lab Server with the Time Of Day Client.

- ❑ Click the **Edit List** button under **Associated Lab Servers** in the upper right column. A popup window will appear.
- ❑ In the popup window, notice the Time Of Day Server web service URL in the left hand box labeled Available Lab Server.
- ❑ Select the Time of Day Lab Server web service URL so it is highlighted, then select the **Add >>** button.



- ❑ Click the **Save Changes**.

- ❑ The Time Of Day Lab server should now appear in the blue box labeled Associated Lab Servers.

### ***Associate the ESS and USS***

The Time Of Day experiment uses both the USS and the ESS. Before the experiment can be run, the USS and the ESS must be associated with the client.

- ❑ From the **Manage Lab Clients** page, use the **Needs ESS** pull-down list to select which ESS this client will use.
  - ❑ Click the **Associate** button.
  - ❑ You should see a green message bar at the top of the page stating that the record has been successfully associated.
- 
- ❑ Use the **Needs Scheduling** pull-down list to select which USS this client will use.
  - ❑ Click the **Associate** button.
  - ❑ You should see a green message bar at the top of the page stating that record has been successfully associated.

### ***Associate Groups***

Associate the Time Of Day Interactive client with the user and management groups.

- ❑ Click the ***Associate groups*** button at the bottom of the Manage Lab clients page.
- ❑ The **Time Of Day interactive client** should already be selected.
- ❑ The associated services (USS, ESS, Lab Server) should contain the correct information.
- ❑ Select the group that will use this client. In this case, select the **Experiment\_Group** from the pull-down list.
- ❑ Use the pull-down list to select the management group for the **Experiment\_Group**. In this case, select the **Manage\_Users\_on\_ USS** group.
- ❑ Click the **Save Changes** button. The associated groups and manager should now be listed.
- ❑ A green bar should appear at the top of the window indicating a successful update.
- ❑ Close the popup window and return to the Manage Lab Clients Page.

### ***Create a Test User***

You will need a new user for testing.

Create a test user.

- ❑ Click the **Users & Groups** Tab.
- ❑ Click the **Manage Users** tab.

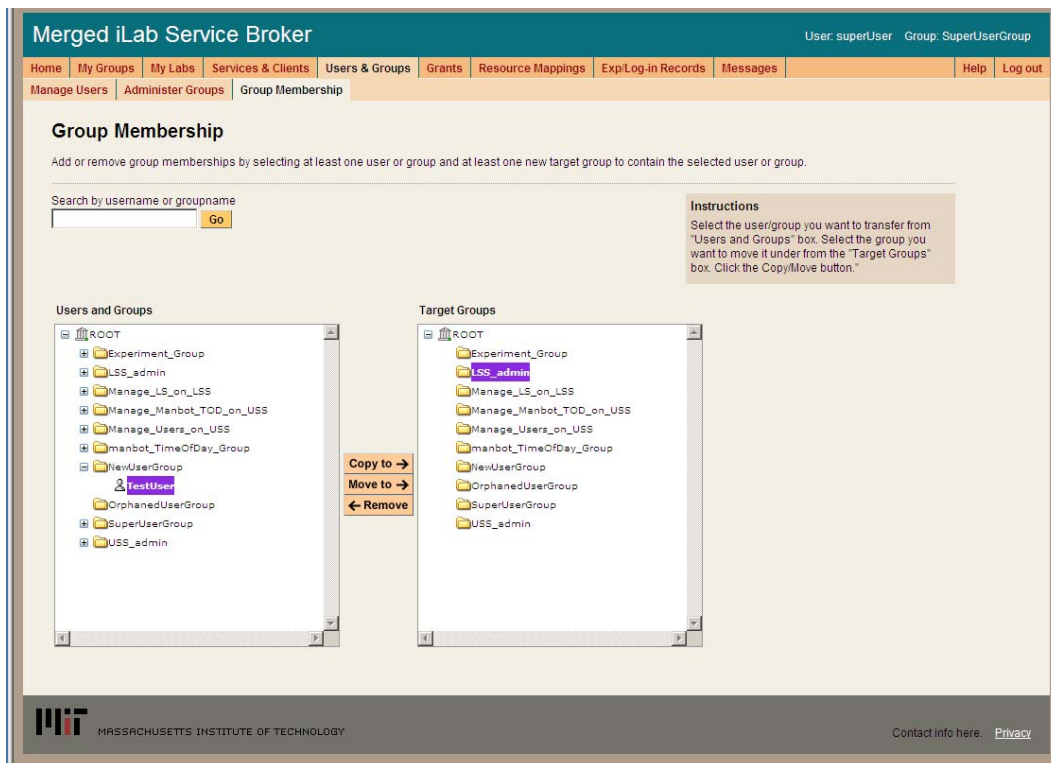
- ❑ Enter the user information for a test account. Use any username, information and password that you want.
- ❑ Select the **Save Change** button.
- ❑ A green bar at the top of the page should indicate that you have successfully created a new user.

## Group Membership

Now that there is a test user, it needs to be given the correct group memberships. For testing purposes, the user needs to be part of five groups.

Add the user to the LSS\_admin, Manage\_LS\_on\_LSS, USS\_admin, Manage\_Users\_on\_USS, and Experiment\_Group groups.

- ❑ Click the **Group Membership** Tab.
- ❑ In the left hand "Users and Groups" tree, click the + sign next to the **NewUserGroup**.
- ❑ You should see your test user listed.
- ❑ Click on your test username to highlight it.
- ❑ In the right side "Target Groups" tree, select **LSS\_admin** group



- ❑ Click on the **Move To ->** button
- ❑ In the left hand "Users and Groups" tree, the test user should now appear under the **LSS\_admin** Group.
- ❑ Click on the test username under the **LSS\_admin** group
- ❑ Click on the right side "Target Groups" tree, select **Manage\_LS\_on\_LSS** group
- ❑ This time select the **Copy To ->** button.

- ❑ **Repeat this process** and copy the test user to the Manage\_Users\_on\_USS group, the USS\_admin group, and the Experiment\_Group.
- ❑ You should now be in 5 groups.

## **Setup Time Of Day Experiment**

Log out of the SuperUser account and log in as your test user.

### **Manage USS**

Manage the TimeOfDay users on the USS

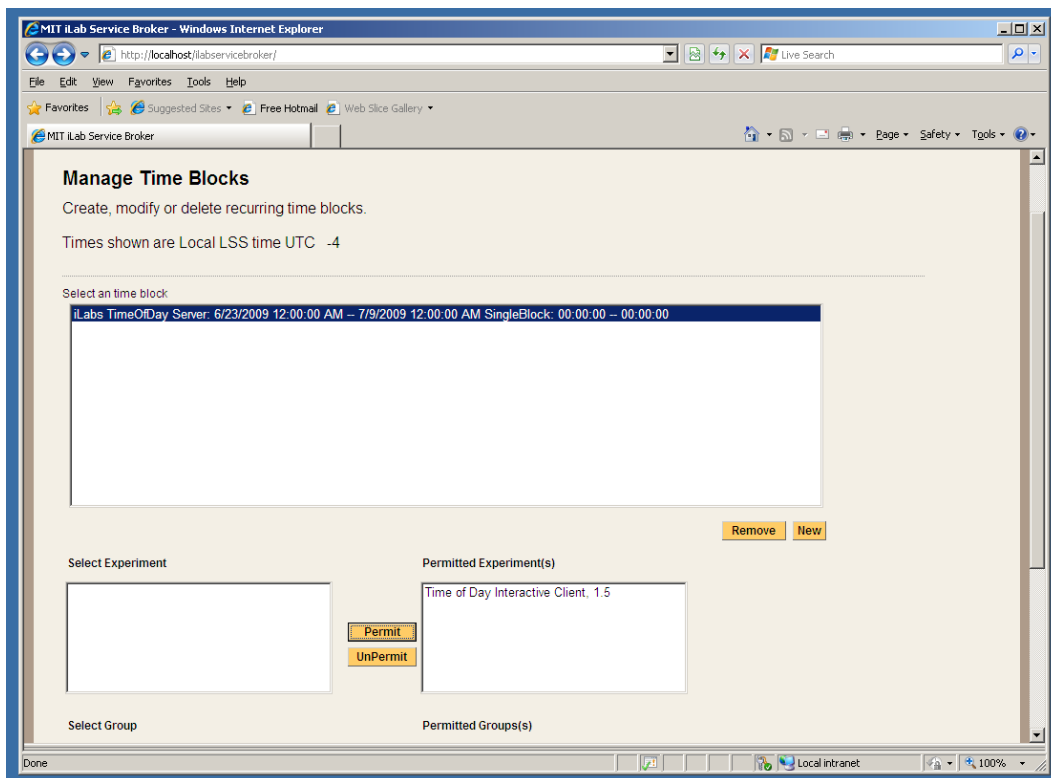
- ❑ Select **Manage\_Users\_on\_USS** as your group.
- ❑ In the Service Administration page, use the - **Select Process Agent** - pull down list to select your USS.
- ❑ Click the **USS Manage Group** button
- ❑ The User Side Scheduling Server should load.
- ❑ Click on the **Policy Management** tab.
- ❑ Use the pull down list to select the Experiment
- ❑ The group should automatically fill in.
- ❑ Enter the **Minimum Time Required: 1**
- ❑ Enter the **Maximum Reservation Time: 5**
- ❑ Click the **Save Changes** button
- ❑ Click the **Back to Service Broker** tab.
- ❑ Click on the **My Groups** tab.

### **Manage LSS**

Manage the TimeOfDay server on the LSS

- ❑ Select **Manage\_LS\_on\_LSS** as your group.
- ❑ In the Service Administration page, use the - **Select Process Agent** - pull down list to select your LSS.
- ❑ Click the **Manage Lab** button
- ❑ The Lab Side Scheduling Server should load.
- ❑ Click on the **Experiment** tab.
- ❑ Select the Lab Client "TimeOfDay Interactive Client, 1.5" from the list on the left.
- ❑ The client information should automatically populate the fields.
- ❑ Update the fields as follows:
  - Enter Minimum Time (min): 1
  - Enter Prepare Time (min): 0
  - Enter Recover Time (min): 1
  - Enter Early Arrive Time (min): 0
- ❑ Click the **Save Changes** button.
- ❑ Click on the **TimeBlocks** tab.

- ❑ Click **New**.
- ❑ Use the **Lab Server Resource** pull down list to select your Time of Day Server.
- ❑ In Recurrence Type: select **Single Block**
- ❑ In the Date Span field, use the calendar icons to set the **Begin Date** and **End Date**.
- ❑ Set the **Quantum** to 1
- ❑ Leave the Start Time set to 12:00 AM
- ❑ Change the End Time to 11:59 PM
- ❑ Click **Save**
  
- ❑ In the **TimeBlock** tab, select the time block you just created.
- ❑ In the left column of the **Select Experiment** block, select the TimeOfDay Interactive Client
- ❑ Click on the **Permit** button



- ❑ A green bar at the top of the page should state that permissions have been granted to the experiment.
- ❑ In the left column of the **Select Group** block, select the Experiment\_Group
- ❑ Click on the **Permit** button
- ❑ A green bar at the top of the page should state that permissions have been granted to the group.
- ❑ Click the **Back to Service Broker** tab.
- ❑ Click on the **My Groups** tab.

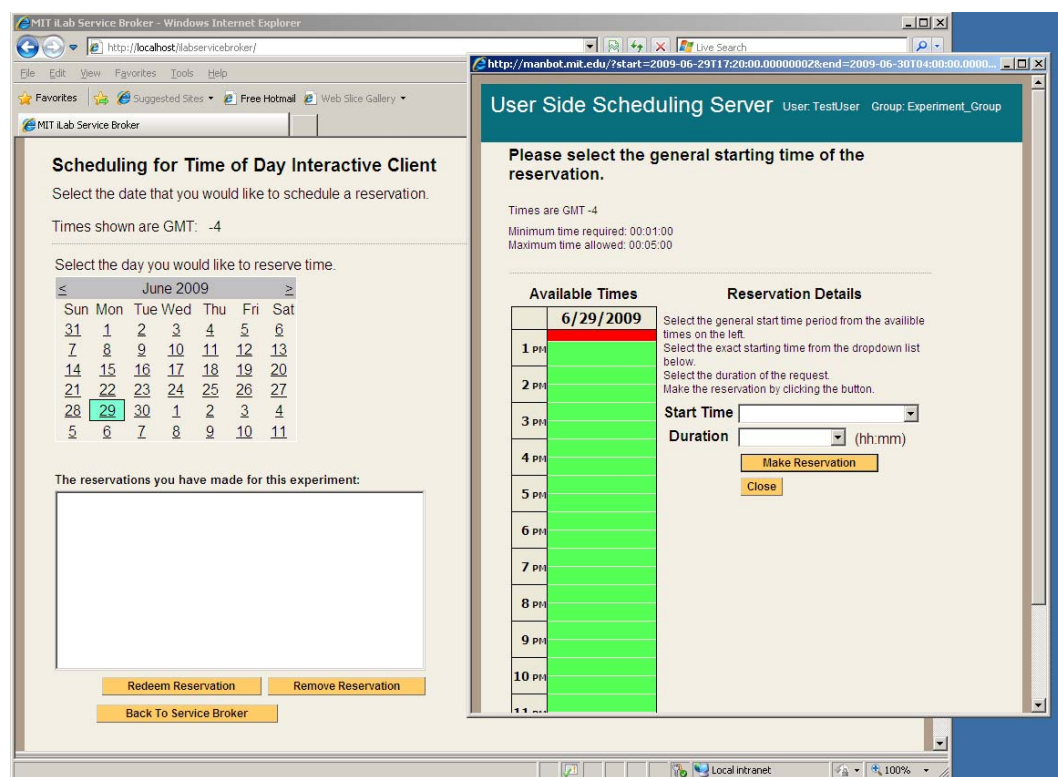
The following experiment is just for confirmation of a clean install, no administration or configuration is needed.

## Running the Time Of Day Experiment

Make sure that you have logged out of the SuperUser account and log in as your test user.

### Scheduling and Running the Time of Day Experiment

- ❑ Select **Experiment\_Group** as your effective group.
- ❑ From the **My Labs** page, click on the **Schedule/Redeem Session** button.
- ❑ In the scheduling window, the highlighted date should be set to today.
- ❑ Click on today's date in the calendar area - if your browser has popup blocker turned on, you will need to hold the ctrl-key down and click on the date.
- ❑ In the scheduling popup window, the available times are identified in green and the reserved times are red. Click on the first green area.

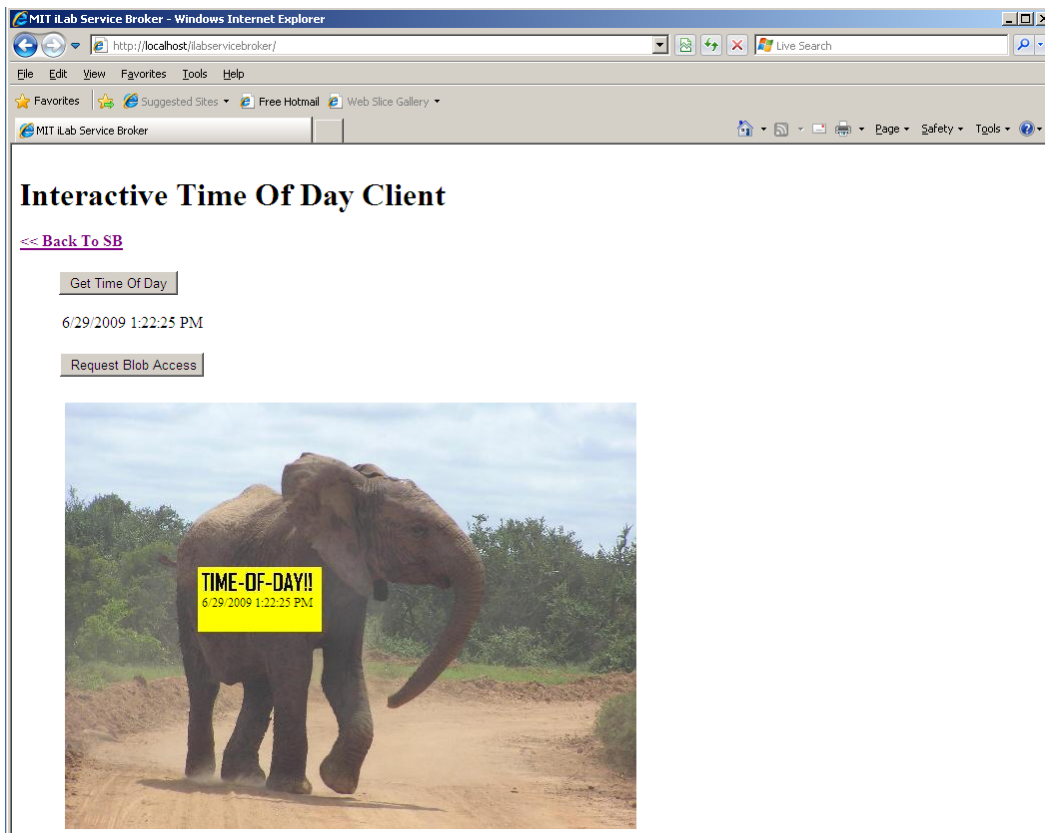


- ❑ In the Reservation Details area, use the **Start Time** pull down to select your start time and the **Duration** pull down to select your reservation length. In this case select 5 minutes for your duration.
- ❑ Click on the **Make Reservation** button.
- ❑ You will see a message in a green bar at the top of the window that says **"The reservation is confirmed successfully"**.
- ❑ Click on the **Close** button
- ❑ The reservation should appear in the reservation window.

**BUG: There is an occasional issue in the Firefox browser and the reservation does**

not automatically appear in the reservations box area. Use the “Back to Service Broker” button, then the Schedule/Redeem Session button. Your reservation should appear. This is an inconsistent bug and has not been tracked down yet.

- ❑ Click on the reservation to highlight it.
- ❑ Click on the **Redeem Reservation** button.
- ❑ Depending on the computers clock a small green bar may appear at the top of the page letting you know how long until the reservation is valid. If this happens, wait the allotted time and click the **Redeem Reservation** button again.
- ❑ On the **My Labs** page, click the **Launch Client** button.
- ❑ The Time Of Day Interactive Lab Client will load. (not very exciting)
- ❑ Click on the **Get Time of Day** Button.
- ❑ The date and time is returned and a blob is stored in the database with the Date and Time.
  - Click the **Request Blob Access** button to see the elephant image with the Date and Time stamp. (Cute Elephant!)



- ❑ Use the **Back To SB** link to return to the Service Broker.



## Setting Up Batched Experiments

In this example, we will be setting up the MIT Microelectronics Weblab experiment, but the process is the same when setting up any batched experiment. Unlike the interactive services, in batched experiment registration there is no automatic registration process.

Before beginning the registration process for a batched experiment, you will need to contact the experiment administrator and make arrangements to use the experiment. In this case, please email Jim Hardison ([weblab@mit.edu](mailto:weblab@mit.edu)). Please be sure to include the following information in your email:

- The name of your Service Broker
- Your Service Broker's GUID
- Your Service Broker's web service URL
- An incoming passcode (this can be a GUID or just a password string)
- Contact information

The Lab server administrator, in this case, Jim, will use this information to create a Service Broker account on the lab server for you to use. He will send you back an outgoing passcode that you will need to create the lab server entry on your service broker.

### Create a Batched Lab Server Record

- ❑ Log in to the Service Broker as **SuperUser**.
- ❑ Navigate to **Services & Clients** on the top navigation bar
- ❑ On the **Manage Lab Clients** page, click on the **New** button at the bottom of the page to clear the page form.
- ❑ Click on the **Register Batched LS** box at the top of the form. Notice that the **Agent type** has been entered as **BATCH LAB SERVER**
- ❑ Enter the following information into the web form:
  - Web Service URL: This is the location of the Lab Server's web service.  
Type **https://weblab2.mit.edu/services/WebLabService.aspx**
  - Agent GUID: **a7aa3c00d7404c63858ff4832b4217ed**
  - Incoming Passcode (this is the passcode you sent to Jim)
  - Outgoing passcode (this is the passcode that Jim sent to you)
  - Codebase URL: **https://weblab2.mit.edu**
  - Service Name: **Microelectronics WebLab 6.1**
  - Description: **The MIT microelectronics device characterization weblab Lab Server.**
  - Lab Info URL: **http://weblab.mit.edu**
  - Contact Email: **weblab@mit.edu**
- ❑ Click the **Register Lab Server** button
- ❑ You should see a green message bar at the top of the page that says: "**Lab Server Microelectronics WebLab 6.1 has created and saved**"

**Passcodes: Passcodes are the strings used by the batched Lab Server to**

communicate with the Service Broker. Passcodes are only used in Batched labs. Passcodes are not the same as passkeys specified in the web.config files.

### **Create a Lab Client Record**

- ❑ From the Servers & Clients menu, click the **Manage Lab Clients** Tab from the submenu.
- ❑ If needed, click on the **New** button at the bottom of the page to clear the page form.
- ❑ Enter the following information into the web form:
  - Lab Client Name: **MIT Microelectronics WebLab Client**
  - Click on the **Make GUID** button.
  - Version: **6.1**
  - Short Description: **MIT Microelectronics WebLab Graphical applet**
  - Long Description: **The standard MIT Microelectronics WebLab Graphical applet configured to interface to the WebLab 6.1 server.**
  - Contact First Name: **Jim**
  - Contact Last Name: **Hardison**
  - Contact Email: **weblab@mit.edu**
  - Documentation URL:  
**<http://weblab2.mit.edu/docs/weblab/v6.1/manual/index.htm>**
  - Client type: **Batched Applet**
  - Check the **Needs ESS** check box.
- ❑ Click **Save Changes**
- ❑ You should see a green message bar at the top of the page that says: **“Lab Client WebLab Client has been added.”**

### **Client Loader Script**

This is crucial part of the Lab Client Record, responsible for the following:

- Name and location of the Client (a Java applet compiled into a Jar archive)
  - The URL of the Service Broker’s web service. This web service contains “passthrough” methods. The Lab Client will invoke a passthrough method on the Service Broker, which will in turn invoke a corresponding method (one with exactly the same name) on the Lab Server.
  - The GUID (a.k.a. Lab Server ID) of the Lab Server that the Service Broker will communicate with when its passthrough methods are called.
- 
- ❑ Copy the loader script from the example below, and paste it into the Loader Script box on the **Manage Lab Clients** page
  - ❑ Replace the text that reads **[YOUR HOST NAME HERE]** with your host name (e.g. “**somemachine.myschool.edu**”).
  - ❑ Click the **Save Changes** button at the bottom of the screen when finished.
  - ❑ You should see a green message bar at the top of the page that says **“Lab Client WebLab Public Client has been modified”**

```

<APPLET
  height=1 width=1 archive="
http://weblab2.mit.edu/client/v6.1.3g/signed_Weblab_Graphical.jar"
  code="weblab.client.graphicalUI.GraphicalApplet"
>
<PARAM NAME="serviceURL" VALUE="http://[YOUR HOST NAME
HERE]/ServiceBroker/Services/ServiceBrokerService.asmx">
<PARAM NAME="labServerID" VALUE="a7aa3c00d7404c63858ff4832b4217ed">
<PARAM NAME="helpURL"
VALUE="http://weblab2.mit.edu/docs/weblab/v6.1/manual/index.htm">
</APPLET>

```

- ❑ Click Save Changes
- ❑ You should see a green message bar at the top of the page: "Lab Client WebLab Client has been updated."

Note: If you intend to log in from a browser on the Service Broker machine using "localhost" in the URL, you must specify localhost as the host in the serviceURL parameter. If you intend to log on from a remote machine, you must change the serviceURL parameter in the loader script to contain machine's external URL. Leaving "localhost" in the applet tag when logging in remotely using the real hostname is a common cause of failure. In that case, the WebLab client will return a "failed to load lab configuration" error message.

### **Associate the Lab Server**

You should still be on the Manage Lab Clients page.

- ❑ Associate the Microelectronics WebLab Batched Lab Server with the MIT Microelectronics WebLab Client.
- ❑ Click the *Edit List* button under *Associated Lab Servers* in the upper right column. A popup window will appear.
  - In the popup window, notice the Microelectronics WebLab web service URL in the left hand box labeled Available Lab Server.
  - Select the Microelectronics WebLab web service URL so it is highlighted, then select the Add >> button.
- ❑ Click the Save Changes.
- ❑ The MIT Microelectronics Lab server should now appear in the blue box labeled Associated Lab Servers.

## Associate the ESS

- ❑ The MIT Microelectronics Experiment uses only the Experiment Storage Server (ESS). Before the experiment can be run, the ESS must be associated with the client.
  - From the **Manage Clients** page, use the **Associate ESS** pull-down list to select which ESS this client will use.
  - Click on the **Associate** button.

## Associate Groups

- ❑ Associate the Microelectronics WebLab client with a user group.
  - Click on the **Associate groups** button at the bottom of the Manage Lab clients page.
  - The **Microelectronics WebLab** client should already be selected.
  - The associated services (USS and Lab Server) should contain the correct information.
  - Select the group that will use this client. In this case, select the **Experiment\_Group** from the pull-down list.
  - Batched experiments do not need a management group.
- ❑ Click on the **Save Change** button. The associated groups should now be listed.
- ❑ A green message bar should appear at the top of the page indicating a successful update.
- ❑ Close the popup window and return to the Manage Lab Clients Page.

## Check Grants

The authorization mechanism in iLabs requires that there be a grant to use each resource controlled by iLabs. For our sample lab to work, we will need our WebLab Group to have a “useLabServer” grant on the Microelectronics WebLab 6.1 Lab Server, and a “useLabClient” grant on the WebLab Public Client. The useLabServer grant was just created, when the Microelectronics WebLab 6.1 Lab Server was associated with the WebLab Group. We’ll see how that looks on the Grants page.

- ❑ On the top navigation bar, click **Grants**
- ❑ In the **Users and Groups** box on the left, click once on **Experiment Group**
- ❑ In the **Functions** box in the middle, click once on **useLabServer**
- ❑ Beneath the **Qualifiers** box on the right, there are three buttons. Click on **View Grant**
- ❑ In the **Qualifiers** box, the entry for Microelectronics WebLab 6.1 Server should now be red.
- ❑ Making sure that **Experiment Group** is still selected in the **Users and Groups** box, click once on **useLabClient** in the **Functions** box
- ❑ In the **Qualifiers** box, click once on **MIT Microelectronics WebLab Client**
- ❑ Click the **View Grant** button
- ❑ If you click the **View Grant** button with **Experiment Group** and **useLabClient** selected, the **MIT Microelectronics WebLab Client** should turn red, indicating that the grant already exists.
- ❑ If not, click the **Add Grant** button

- ❑ You should see a green message bar at the top of the page that says: “The grant {WebLab Group, useLabClient, WebLab Public Client} was successfully added.”
- ❑ If you click the **View Grant** button with **Experiment Group** and **useLabClient** selected, the **MIT Microelectronics WebLab Client** should turn red.

## **MIT Certificate**

The WebLab web service runs over a Secure Sockets (SSL) connection. Therefore, your Service Broker is required to have the **MIT Certification Authority** certificate installed in its **Trusted Root Certification Authority** store. Here is how to obtain and install the certificate:

- ❑ Go to <http://web.mit.edu/is/topics/certificates/>
- ❑ Under **Get Certificates Now**, click on **Get MIT CA (Certificate Authority)**
- ❑ Save the file **mitca.cer** to your hard drive
- ❑ Create a Certificate Microsoft Management Console (MMC) Snap-in. **Start->Run...mmc**
- ❑ **File->Add/Remove Snap-In**
- ❑ Click the **Add...** button
- ❑ Select **Certificates**
- ❑ Click the **Add...** button
- ❑ Select **Computer account**
- ❑ Click **Next>**
- ❑ Select **Local Computer**
- ❑ Click **Finish**
- ❑ Click **Close**
- ❑ Click **OK**
- ❑ Expand **Certificates (Local Computer)**
- ❑ Right-click **Trusted Root Certification Authorities**
- ❑ **All Tasks->Import...**
- ❑ **Next>**
- ❑ **Browse** to the location where you saved **mitca.cer**
- ❑ Select **mitca.cer**, and click **Open**
- ❑ **Next>**
- ❑ Make sure **Place all certificates in the following store** is selected, and that **Trusted Root Certification Authorities** appears in the box. If it does not, click the **Browse** button and correct it.
- ❑ **Next>**
- ❑ **Finish**
- ❑ **OK**
- ❑ The Certificates console is a handy thing to keep on your machine. To save it so that you can open it without having to add the Certificate snap-in again, click **File->Save As**, type **Certificates.msc** in the box, and click **Save**. It should now appear as an option in **Start->All Programs->Administrative Tools**
- ❑ Close the Certificate MMC Console.
- ❑ Reboot the machine

## Running the Experiment

- ❑ Log in to the Service Broker as the user you created
- ❑ Following login you should see the **My Labs** page, which should contain one Lab Client, the **WebLab Client** (Note: if you are a member of more than one group, you will have to select the WebLab Group link first before you get to this page)
- ❑ Click the **Launch Lab** button
  - Note: if you do not see the Launch Lab button, check the following:
    - User is in the WebLab group
    - WebLab group has useLabServer grant on the Microelectronics 6.1 Lab Server
    - Weblab Group has useLabClient grant on the WebLab Public Client
- ❑ A **Warning - Security** box should pop up. Click **Yes**
- ❑ A new screen - the Java Client - should load
  - If the screen does not load, check the Java Console Messages. In the system tray, at the lower right near the clock, look for the Java icon (a blue and red steaming coffee cup). Right-click on the icon, and select **Open Console**
  - Scroll down and look at the messages. If there is a problem, chances are that either the client classes didn't load properly into the virtual machine (in which case there will be a stack trace), or that there was something wrong with a parameter on one of the Web Service calls. In that case, look at the SOAP messages from the Web Service calls that have been passed from the client to the Service Broker, along with the responses. If you see the term "SOAP Fault" in a response, that should point to the problem.

## Lab Client

- ❑ If everything is working, you should see a rudimentary circuit diagram in the upper window. If things are **not** configured correctly, you will get a message that says **Failed to Retrieve Lab Configuration**. If that happens, check to make sure the following is done:
  - MIT Certificate is on your machine
  - Lab Server GUID is correct on the Lab Server record
  - Service Broker ID is in the correct place in web.config
  - Service Broker ID is correct and has been keyed into the WebLab Administrative Interface by the WebLab Administrator
  - Lab Client Loader script has the correct serviceURL
  - Lab Client Loader script has the correct Lab Server ID (should be the same as the Lab Server GUID in the Lab Server record)
- ❑ For information on how to submit experiments to Weblab, please see <http://weblab.mit.edu>

# Troubleshooting

## Grants

Currently, the bootstrapping process does not automatically give the managers of the LSS the correct permissions to manage the lab server on the LSS. This will be fixed in the next release.

- Give the Managers of the LSS permission to manage the LS.
  - Click on the *Grant* tab in the main menu.
  - In the **Users & Groups** tree on the left side, click on **Manage\_LS\_on\_LSS** group to highlight it.
  - Under **functions**, click on **Manage Lab**.
  - Under **Qualifiers**, select the resource mapping between the Interactive Lab Server and the LSS; Look for a resource mapping where the Process Agent: ILS and the ResourceType: LSS
  - Click the **Add Grant** button.
  - A green message bar should appear at the top of the screen indicating a successful update.
  
- Note: You may get this error message: "Unable to find script library '/aspnet\_client...WebUIValidation.js' If you see this, do the following:
  - Open a command prompt with Start->Run-CMD
  - Type CD\Windows\Microsoft.NET\Framework\v1.1.4322
  - Type aspnet\_regiis -c
  - You should see a message that says "Start copying the ASP.NET client side script files for this version (1.1.4322.0)...Finished copying the ASP.NET client side script files for this version (1.1.4322.0)"
  - Type **Exit** to close the command prompt
  - Click **OK** on the "Unable to find script library" error message