iLab Shared Architecture (ISA) Release Notes

Release 3.0.0

These are some of the new and updated features of the 3.0 release of the iLab Shared Architecture. Note updates are added at the end of the document

General

Te most important change from revision 2 to revision 3 of the iLab Shared Architecture is the support for international character sets.

UTF-16 in database and web pages for all fields except GUIDs and system defined ‘TYPES’ ProcessAgent,Ticket, ResourceMap, etc

Database neutral support, all database code now uses DBxxx classes and Types. The stored procedures have not been converted to another database so we are still dependent on sqlServer. Due to limitations in DBxxx for BLOBs a work around is used to support them.

Access to the actual database specific instances of the DBxxx classes is provided by the FactoryDB class. This class will be modified in the future to support additional database engines. See web.config connection string and database provider keys.

SelfRegistration is now handled by a common server control, additional fields have been added to the SelfRegistration form. When InstallDomainCredentials is called from the ServiceBroker after a successful install these additional fields from the ServiceBroker are transferred to the new service using Register with a request for a return Register call from the new service. These system information fields are no longer stored as ResourceMapping items, but are added to the ProcessAgent Table.

Added REDEEM_RESERVATION ticket type, this is used to request an existing reservation from the USS it is used in the auto-launch part of ssoAuth.

New in Release 3.0.2

The experiment/login records tab in the administrative section of the ServiceBroker has been changed to reports with some additional status reports.

Added iLab ServiceBroker URL to new registration emails.

iLabServiceBroker/ssoAuth.aspx provides an authorization process which routes the user to the experiment specified, if it can be resolved, or to pages where the remaining required information can be selected. Currently the only useful parameters are group name and client GUID, if a user name is specified it will be used to check if the current user is that user or if the login is for that user. Currently there is no support for third party authorization, so users must already be registered on the ServiceBroker and must have any required permissions and be in the correct groups to run the experiment. Depending on the arguments specified, the group memberships of the user, and clients permitted by those groups a user will be redirected to myGroups, myClients, the scheduling server or the client will be launched. Currently the launchClient page is not displayed if the user has launch permissions.

Any combination of the following arguments are allowed:

- **sso** – flags default.aspx to redirect the request to the ssoAuth.aspx
- **cid** – client GUID
- **grp** – the effective group for the experiment, the group must exist on the ServiceBroker
- **usr** – the user's name, currently this must be for an account on the service broker, this is not useful at this time since you will still be directed to a login form.
- **auth** – the third party authentication service. Currently this is not supported, so all authentication is performed by the ServiceBroker.
- **key** – an optional shared secret that could be used to authenticate the authentication authority

ssoAuth.aspx is not called directly by using a URL, but by passing arguments to the iLab ServiceBroker's default.aspx page. This is required by legacy Lab servers that depend on the ServiceBroker to maintain session state. An example URL:

http://yourServiceBroker/iLabServiceBroker/default.aspx?
sso=t&grp=math_class&cid=2341109D-E16E-48B1-B0F9-202726DF1DF2

It is recommended that you specify both group & client for the best performance.

Lab-Side Scheduling

Added an email address field to the experiment information page. If specified an email will be sent to the address for each experiment reservation confirmed on the LSS.

New in Release 3.0.3

Support for LabVIEW 2009. The interactive LabServer with LabVIEW now supports multiple versions of LabVIEW, but you should expect to only use a single version for all Labs on a specific LabServer. Versions 8.2, 8.6 and 2009 are shipped and built into the code. You may change the default version to use in the LabServer Web.config file or on the Lab Application page, the current default is 2009. Due to the LabVIEW DCOM limitations only one version of LabVIEW maybe supported on a lab server's machine.

New in Release 3.0.4

iLabServiceBroker

Dynamic Loader Script:

Dynamic replacement of tokens in loader scripts is now supported using the static method **`iLabs.Core.iLabParser.Parse(string, Hashtable) which`** returns a new string with all tokens in the original string replaced by the values of the tokens in the hashtable. Tokens use the format **${*key*}** keys are not escaped, values are any string, by default the following keys are parsed on the loader script:

ServiceBroker:
- ${sb:agentGuid }
- ${sb:agentName}
- ${sb:codebase}
- ${sb:serviceUrl}

LabServer:
- ${ls:agentGuid }
- ${ls:agentName}
- ${ls:codebase}
- ${ls:serviceUrl}

Operation Coupon:
- ${op:couponId}
- ${op:issuer}
- ${op:passkey}

These default keys are provided by an instance of the iLabs.Core.iLabProperties class to which may be added additional key/value pairs. Convenience methods of the iLabProperties class include adding Coupons, ProcessAgents and ProcessAgentInfos with a string prefix. Currently the use of the parser & defaults is restricted to the parsing of the loader script and the default keys, by modifying the myClients.aspx.cs file additional

keys may be added. There is no reason the parser could not be used in other areas of the ISA.

One consequence of the use of the dynamic parsing is that arguments that were appended by default to redirect URLs have been removed and must now be specified as part of the loader scripts. For both batch & interactive clients these included the operation coupon specification which was also used for session and experiment ticket redemption. This means most launch scripts should add: “?coupon_id=${op:couponId}&passkey=${op:passkey}&issuer_guid=${op:issuer}”.

While this lengthens the loader script it allows for different parameter names when passing the arguments and provides a mechanism to pass arguments to applets and Java applications that are launched. Note: due to user requests these arguments are currently automaticly added to redirect scripts if they are not present, in the future the automatic feature will be retired. **Note: as of Release 4.2.0 this class was changed from a static class and a Parse method with an option for adding the default parameters was added, the existing two argument Parse method does not add the arguments.**

Group Management

The group management page has been modified to use check boxes to select users and groups. Multiple users may be selected and multiple target groups, the selected action is performed for all selected users on copy they are added to all target groups, on move they are copied and then removed from the original groups, on remove they are removed from the group that they are selected in.

Session History

To support requests for better tracking use of the serviceBroker the user session information tracking has been modified. The original User_session table is used in much the same way as before. A record is created at the start of a session and it is used to store the current group, client & platform specific session key, this information is still used for session resumption, but as this limited group & client information to only what was assigned when the session was closed an additional table was added to record every change in a user session. Report formatting based on these two tables is still under development.

SSO

Additional work has been done on the ssoAuth page it no longer is responsible for launching applications. If all criteria are supplied & resolve to one authorized experiment a redirect to myClient.aspx with an autoLaunch argument is issued. The client is then launched. For criteria that do not resolve into a single solution ssoAuth redirects to the correct page for further refinement. Once a general SSO infrastructure is in place a variation of the ssoAuth page will replace the login target for failed authorization checks.

Release 3.0.5

Is a bug-fix update to release 3.0.4 no new features were added

Release 3.0.6

Is a bug-fix update to release 3.0.5 no new features were added

Release 3.5.1

Release 3.5.1 is a documentation update to release 3.5.0 most changes are to this file and a minor change to the database conversion scripts.

Due to the number of changes in this release there has been a minor revision version change. Database scripts are provided to update existing 3.0.6 databases where needed.

Several new features, additional Web Service methods, a code tree reorganization and internal changes require that the minor release version be changed. I had considered making a jump in the major version but as the Web Service data types have not changed 3.0.6 version systems will continue to work with the existing code. You will need to update your database stored procedures, one LSS table and add a few standard defined values, but the underlying tables will only require minor changes.

One change that might effect existing systems is the addition of three new ProcessAgentType AgentType's:

AgentType.GENERIC_PA	value = 000
AgentType.NOT_A_PA	value = 001
AgentType.AUTHORIZATION_SERVICE	value = 128

Currently there is no example AUTHORIZATION_SERVICE, but it is reserved for future use.

To migrate an existing iLab Domain from 3.03/4 or 3.06 databases the following database scripts need to be run to update tables and default values.

All: ..\DB_Scripts\ProcessAgent\update_306_to 350.sql

ISB: ..\DB_Scripts\ServiceBroker\ISB_303_306_Convert.sql
..\DB_Scripts\ServiceBroker\ISB_Convert_306_to_350.sql

LSS: ..\DB_Scripts\Scheduling\LSS_Convert 304 to 350.sql

In addition all of the *_Procedures sql files will need to be run for each database. The 3.5.0 codebase requires these changes

An example generic service is provided, this is a ProcessAgent with no additional methods. To derive a system from this code create an exported version of the iLabGeneric service tree, rename the WebService class (iLabGeneric) and add your web methods to the web service page. If your service is to implement any of the standard iLabServices add the bindings and update the Agent_Type value for the RegisterSelf

component on the selfRegistration page. By default there is not support for assigning relationships with additional services or administrative groups to the generic service.

For developers a major change is the reorganization of the code tree. A team at Birla Institute of Technology and Science (BITS), Pilani is working on a pure OpenSource platform implementation of the iLab SA. To support the addition of this code when it is complete the SourceForge SVN tree has been modified, the old top level directory iLabs is now iLab_SA, with subdirectories for Datatabase, Documentation, iLab_WSDL, dotNet and OpenSourceSystem. The database directory has also been split into mySQL and sqlServer trees. Currently there is no code in the mySQL tree but we hope to provide scripts that will work with the current dotNet code in the near future.

In addition there is a top level directory PartnerLabs where example labs from our partner institutions may be posted.

The Registration, Help and Report Bugs pages for all services have been modified by the addition of a Recaptcha (www.recaptcha.org) control which will force users to respond before the email request is sent. We are sorry to have to add this feature, but it seems that the sendmail features of the pages are being abused. The controls by default use a global Recaptcha key assigned to ilab.mit.edu. It is suggested that institutions generate their own keys. See the documentation.

Administrative methods for the management of clients, lab servers and groups have been modified to resolve issues with lab scheduling. Internal structures have been modified to remove lists of Lab Servers and ClientInfoItems from within the LabClient object and administrative pages have been modified to reflect these changes. To resolve issues with client configuration trashing existing reservations for scheduled experiments and scheduling service configuration, it is no longer possible to modify data which effects scheduling requirements and related services if there are groups assigned to the client. Prompts have been added to the different pages. You will have to use individual associate & disassociate buttons to make these changes on the service or client once it has been stored in the database. A side effect of this is that the initial registration of a service or client may be done on one page request.

When you remove a group from a scheduled experiment/client all future reservations for that client/group will be revoked on both the USS & LSS. An email will be sent by the ServiceBroker for each reservation revoked.

The ServiceBroker added a new WebService method ‘RequestAuthorization’ which provides for known Services or clients to request the creation of authorization tickets. Upon success the method returns an OperationCoupon which may be passed to other services in the domain, this provides the ability for third parties to schedule client reservations and in the future a framework for third party authentication.

Multiple changes in the scheduling servers now support the revoking of reservations and the reservation of blocks of time by LabServer managers from the LSS. In addition it is now possible to request scheduling authorization from the service broker & directly schedule by accessing the USS via two new web service methods.

In the LabServer_wLabVIEW support for LabVIEW 2010 has been added.

Release 4.0.1

The iLab 4.0 Release has a number of significant changes, some of these are visible to users and administrators others are internal to the code. These changes will require the creation of new databases. At his time there are not any scripts to convert 3.5.x databases to 4.0 databases. **See Release 4.3.0 notes.**

The user and group changes have the most impact. These changes are related to support of automatic login from multiple user domains or authentication authorities. Authorities may be LMS's or any other form of identity provider that is known to the ServiceBroker and can authenticate it's self to the service broker, currently authentication is limited to supporting the iLab ticket mechanism. An additional administrative page has been created to manage authorities, an authority may define a default group for all new users and if auto-registration is supported. The default authentication authority is the domain's ServiceBroker, Authority_ID = 0 is reserved for the ServiceBroker.

Prior to release 4.0 users and groups had a common base class and shared a primary key using the Agents table, both user name and group name had to be unique for an entire iLab domain . Splitting the groups and users dependency on a common key still requires that group names be unique within a domain, but user names may be the same as a group name and only need to be unique for a specific authentication authority. The Group/User changes have made the internal management of groups and users much more straight forward. An added benefit of these changes is that it now possible for a member of the group's 'TA Staff Group' to manage users without being logged in as a member of the Superuser Group.

Database changes include removing the Agents and Agent_Hierarchy tables, adding Group_Hierarchy, Authority and Metadata tables, modification of Groups and Users tables, and the creation, deletion and renaming of many stored procedures.

AgentType.AUTHORIZATION_SERVICE value has changed from 128 to 129 to reflect the proper BIT mapping.

To support authorization and authentication from authentication authorities and clients four Web Service methods are provided.

- RequestAuthorization -- provides for the creation of tickets that will allow clients and third parties to directly request authentication for access to experiment results, user scheduling for clients and access to iLab resources. This has been used for SMS based scheduling and an Android client.
- LaunchLabClient -- Is designed to be used for launching a client from an LMS, by use of a SCORM(or other LMS learning object) package. A single SCORM with authentication credentials will be produced by the ServiceBroker for each client, to be used by all authorized LMS's. The SCO uses the SCORM Run-Time

Environment to access individual user information and status. This is still under development.
- GetUserStatus -- An authority requests the 'completeness' status of a user. This will most likely only be supported for requests from a SCORM.
- ModifyUser -- An authority updates a user. This will most likely only be supported for requests from a SCORM.

Release 4.2.0

The iLabParser class was changed from a static class and a Parse method with an option for adding the default parameters was added, the existing two argument Parse method does not add the default arguments. There are also options for changing the token end delimiter.

The InteractiveLabServer libraries have been restructured to provide more generic ‘LabTask’ implementations, support has been improved for long running and reentrant tasks including the ability to load previously recorded records from the current experiment. File and FileWatcher data sources have been added.

The LabVIEW task classes have been modified to reflect the changes to the base task classes.

Release 4.3.0

Release 4.3.x will most likely be the last release that requires .Net Framework 2.2, and VisualStudio 2005 to compile. We have continued to require this run-time environment since most of the original WebLab clients depend on the Service Broker to maintain session state. This session state is used to associate client requests with a user and the current client/experiment. The trick that is used to persist session state, may not continue to work in later versions of the .Net Framework. In addition as we add support for third party authorization and LMS requests this hack complicates the process.

A branch has been created on the iLabProject Sourceforge repository for version 4.3.x code, the branch will retain the ability to be compiled using VisualStudio 2005. The trunk of the project will be converted to use a more recent version of VisualStudio, once converted the code and project files are not backward compatible.

4.3.0 New Features:

The Experiment Storage Service has internal modifications to the database that added some administrative information to the experiments table. A new web service method CreateExperiment() has been added to support the new fields (user_id, group_id & client_id). Additional criteria are supported for querying the ESS from the ServiceBroker, LabServer and client.

A binding for LabVIEW 2012 has been added. Please note there are still major issues with supporting VIserver via DCOM in operating systems after XP.

ServiceBroker Changes

Work continues on supporting third party authorities and seamless launching of experiments from these authorities. The LaunchLabClient web service method has been modified, the argument autostart has been removed, and on success it returns an IntTag with a status code and an error message or a URL to http://hostname/iLabServiceBroker/LaunchClient.aspx. A new ticket type (LAUNCH_CLIENT), which is only accessed by the service broker, provides all the information that is needed to actually launch the client once the user's browser is redirected to the LaunchClient page.

Java Applet and non-browser based clients should not depend on the service broker to maintain session state, but should include 'loader script' input parameters 'couponID' (long) and 'couponPassKey' (string) or equivalent parameters. The client must set these in the sbAuth SOAP header for all web service requests. Applet Lab Clients based on the WebLab 7.1 client support this feature, clients based on earlier versions of the WebLab client do not support SOAP headers. See WebLab 7.1 Graphical Client for an example. Launching a non-SOAP header client from the MyClients page will continue to work for this release

It is now possible to provide GroupRead permissions to another group using the service broker's grants page. This allows a user from a group with higher levels of access to launch and share an experiment with all the members of a different group. In the case of reentrant experiments this allows the GroupRead users to join an executing experiment.

Database Migration

Two scripts are provided to help migrate existing 3.5.x databases to 4.3.0. The ProcessAgent script checks for new agent and ticket types and adds them if necessary. The ServiceBroker script copies a subset of the 3.5.x data to a new 4.3.x database.

- sqlServer\DB_Scripts\ProcessAgentupdate_40x_to_43x.sql – For use on all processAgent databases that need to be migrated, these may be done in place. If you are migrating a ServiceBroker use the process below. This may be used on a 3.5.2 database.

- sqlServer\DB_Scripts\ServiceBroker\imigrate352_to_430.sql – Only for use on a backup copy of a 3.5.x ServiceBroker database and new SB database created using the current scripts. See details in the script.

  This SQL script is designed to migrate an iLab ServiceBroker 3.5.2 database to a 4.3.0 database. By default it will migrate all services, clients and users but not groups. Temporary tables are created for the ProcessAgent, Lab_Clients, and Users tables, by specifying WHERE clauses for the paWrk, clientWrk and userWrk table's select statements it is possible to prune old servers, clients and users. Pruning is a good idea especially for Users.

Note: for all database migrations all the 4.3.x procedure scripts will need to be run.

Release 4.3.1

Release 4.3.1 includes several bug fixes and cosmetic UI updates. No new features were added.

Release 4.3.3

Includes partial work on adding support for multiple lab sever resources in the Lab Scheduling Server, this will allow a single lab sever to support multiple ‘rigs’. This is not complete and has been hidden, to fully support it will require changes to the USS and LSS Web service methods.

A security update has disabled requesting a password reset for the Superuser account, the only way for the Superuser’s password to be changed is from the Manage Account page and the user must currently be in the Superuser Group. If the Superuser account is locked out and no other accounts are members of the Superuser Group the password must manually be reset in the database. The superuser user account may not be removed from the Superuser Group or added to any other groups.

Additional work has been done to support Launching a Lab from an LMS. This involves changes to the non-integrated web page ~/admin/clientMetadata.aspx which may only be accessed while a member of the Superuser Group, by directly entering the URL. This page currently is only used to create a fragment of metadata and an Authorize_Client ticket. The generated metadata may be given to an LMS and used to construct a Web

Service request to LaunchLabClient. See the main web service page http://hostname/iLabServiceBroker/iLabServiceBroker.asmx for further information. In addition http://hostname/iLabServiceBroker/resources/ExampleLanch_iLab.py is a Python script that calls the service, it includes a sample of the metadata. Note for this to work the Authority (LMS) must be registered with the ServiceBroker that generated the metadata and you must have a valid metadata fragment, the example will not work.

Note: the clientMetadata table has an additional field group_ID from the last release, since no one should have been using the table it would be best if you want to upgrade and keep your current database to “drop table clientMetadata” and re-create just that table from the database scripts for this release, you will also need to re-run the ServiceBrokerCoreProcedures.sql, which will not cause data loss.

Future Plans:

- Update iLab code to compile using VisualStudio 2012
- Provide Weblab and WebLabElvis LabServer code that supports ISA specification for field lengths and ticketing.
- Update WebLab clients to maintain session information vs. requiring ServiceBroker to maintain the session.
- Provide a Web Service based replacement for the VIserver interface to LabVIEW.