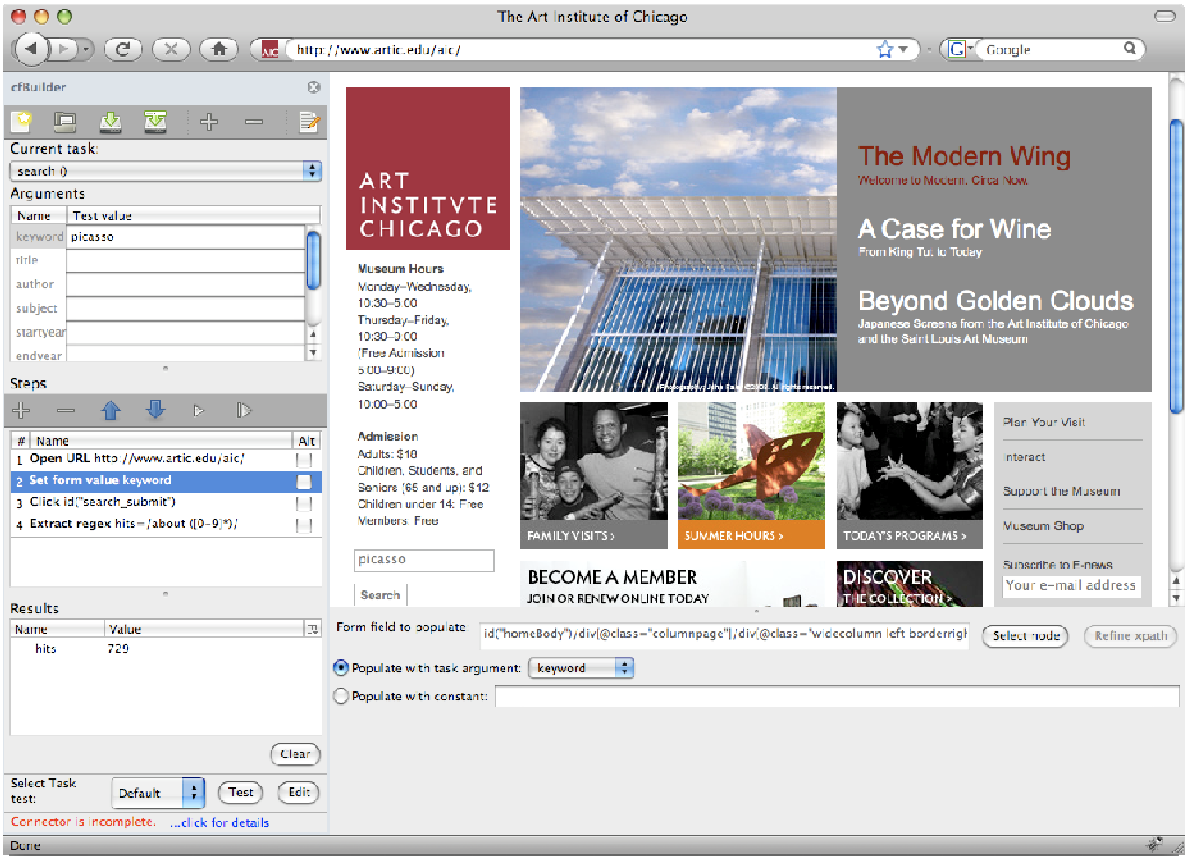
Index Data's MasterKey Connect
Product Description

MasterKey™ Connect™ is an innovative technology that makes it easy to automate access to services on the web. It allows non-programmers to create 'connectors' which act as glue between interfaces designed for people and for software applications. Connectors are a crucial part of many sophisticated services on the Internet, including metasearching, single-signon authentication systems, e-commerce solutions, and much more. Traditionally, creating connectors has been a tedious task, requiring days of work by skilled programmers. With MasterKey Connect, connectors can be created in moments by almost anyone. For any service that requires software access to data or systems behind human-facing interfaces, this technology can dramatically reduce costs and increase productivity.



The Connector Builder in action

The Challenge

Making software systems talk to each other is a constant challenge. Index Data's MasterKey Connect technology makes it easy to build and maintain Connectors to web-based interfaces – often the quickest way to access legacy systems. Our Connectors can be used to support a wide variety of applications, from metasearching to financial transactions. Index Data provides a full range of development and support services to solve your integration needs.

Today a vast number of situations require communication between one computer system and another, especially portals that gather information from multiple sources to present a unified response to an end-user. Familiar examples include metasearch systems in libraries, online booksellers, or travel agents, but could also encompass data mining or harvesting, or even automated software testing. In some cases, interoperability is simplified by the availability of APIs or protocols – either proprietary or industry-standard. In a great many situations, however, no formalized interfaces are available at all; the only access to systems is through a web-oriented interface that has been designed for use by humans, not by machines.

In these situations, software developers must resort to a technique sometimes described as 'screen scraping'. In a nutshell, this approach involves manually writing software that emulates the behavior of a browser interacting with a website, which is a demanding proposition. Creating logic on top of a user interface is a time-consuming process. As websites become ever more complex, programmers must deal with ever more complex interface elements. And, since user interfaces frequently change without warning, maintenance of software that interacts with those interfaces becomes a constant challenge.

Our Solution

Our solution to this problem has been to construct a technology that allows you to build even quite complex connectors by interacting with a website; not like a programmer reverse-engineering the behavior of a browser, but like a user interacting with that website through a browser. Our connector authoring environment (called the Connector Builder) is a browser plugin – a control panel that sits next to the website you are building a connector for, and allows you to construct the connector by essentially walking through the same steps that you as a user would take to obtain the desired results from the website. Because programming is not required, the task of creating connectors is accessible to a much broader range of people. And, because of the many functions that guide and help the author through the process, connectors can be created much more quickly – typically in minutes, as opposed to hours or even days using conventional, brute-force approaches to connector creation.

In the Connector Builder, a connector is viewed as a collection of tasks, or functions that must be supported in order to implement the desired functionality. For instance, to support broadcast searching across information system, typically three tasks are required: Searching, Parsing Results, and Paging through result pages (a fourth optional task is required to handle authentication against access-controlled resources). While these tasks may be manifested very differently between different websites (depending on the type of input forms, the way results are presented, etc.), the tasks themselves are generally always the same. The task of the connector author, then, is to create (at least) one instance of each task for a given website – to describe the

steps necessary to execute a function and to decode the results.

The set of tasks needed depends upon the application. Searching is one possible use of the technology, but it can be used for almost any application that would benefit from layering existing web-based user interfaces behind software-friendly interfaces. The set of tasks that make up a connector is governed by a so-called Connector Template, which can be created or adapted to any specific purpose.

In the Driver's Seat

The way you instantiate these different tasks (in the case of searching – start search, parse results, go to next page) is different for every website that has a different interface. Logically, a task is instantiated by putting together a sequence of steps that correspond to the activities a user would perform when interacting with the website. Some of the steps are high-level 'instructions', such as 'open a page with a given URL', 'populate an input field in a form', and 'click on a link'. By sequencing these simple steps, you create a kind of script that tells the system just how to perform a given task against a website. The steps are typically configured simply by pointing and clicking: i.e. you tell a 'click' step what you want to click on simply by pointing at that element on the screen and clicking. As you are adding steps to a task, you can test individual steps and the whole task, simply by clicking on a 'play' button. This way, it is possible to string together quite complicated sequences of interactions with a website.

Making Sense of It

Interaction with a website isn't very useful if you can't extract meaning from the results that are displayed to the user. To that end, MasterKey Connect has an array of supporting tools built-in. Simple data elements --i.e. result counts, status indicators, etc. -- can generally be retrieved in a simple way. For example, there is a step that will extract a simple data element from an area of the screen (an HTML element), and, if necessary, perform a simple transformation to remove unwanted noise (like display labels, etc.) Other steps can then be added, such as removing unwanted whitespaces, normalizing dates, URLs and more.

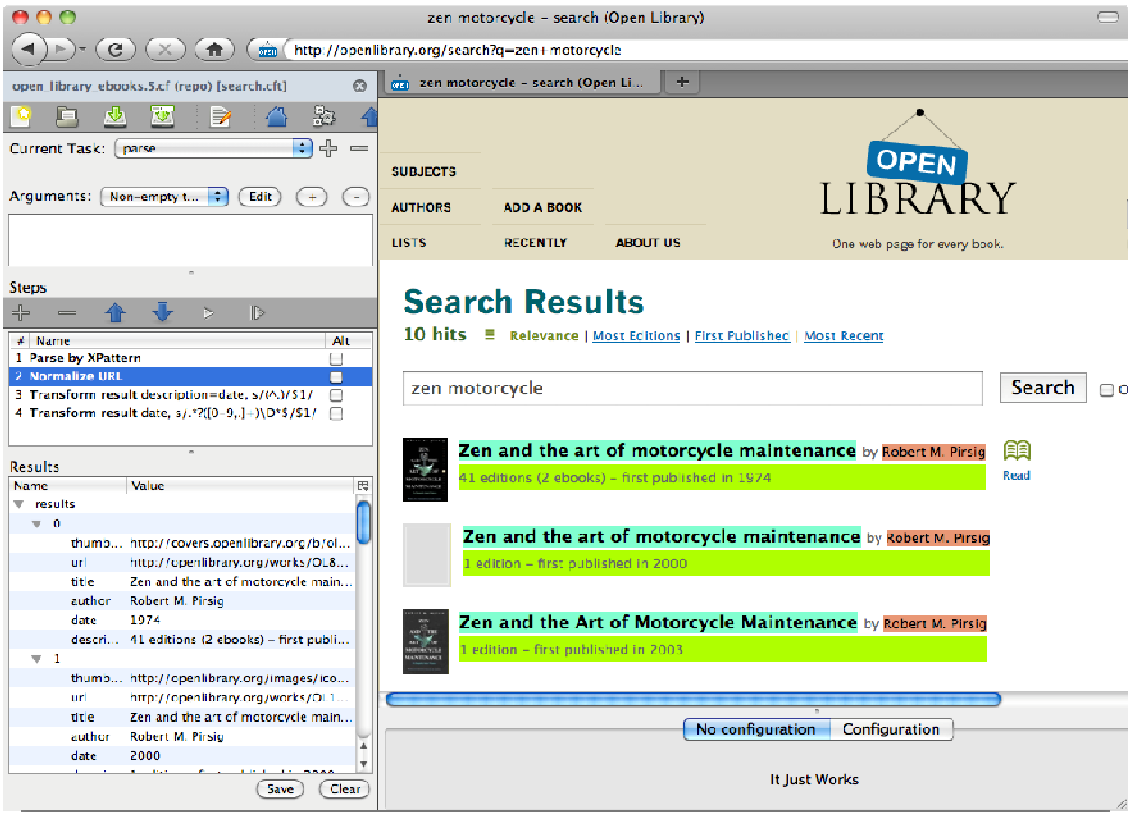
A particular challenge is parsing out more complex data, such as a list of result 'records', each of which consists of multiple 'fields', such as titles, dates, price, author names, descriptions, etc. This can be one of the more time consuming aspects of building gateways to websites. With MasterKey Connect it is remarkably easy.

The conventional way to interpret such data is to write complex sets of patterns to recognize individual parts of the HTML text that is returned by the server. This is a tedious programming task that can be quite error prone. In MasterKey Connect, we take a higher-level approach (although the lower-levels are still available for those who need them for the truly difficult cases). Rather than dealing with the HTML as an unstructured stream of text, we look directly at the logical structure that the browser produces as it displays the page. We have created a purpose-specific pattern language that allows us to explain the relationships between the different elements that make up a complex record, as well as mappings to logical data elements. This language serves two functions: it allows us to detect repeating, higher-order 'records', and to break these records up into their constituent parts.

While a connector author can individually script these patterns, a faster and easier way is to use the included authoring tool. With this tool, you simply identify the individual parts of a sample record, and the software automatically generates a pattern. Since it is possible that display records are different (i.e. they may contain optional or repeatable elements), the software allows the author to fine-tune the pattern to take such differences into account.

After the pattern has been created, it can be tested right in the browser. The Connector Builder will highlight the records it has matched and display the extracted data in a separate output pane, so that the connector author can ensure that the data that is extracted is what he or she is looking for.

Parsing complex result data



The Connectors Themselves

When the connector is complete, it can be augmented with simple descriptive metadata (the name and URL of the website it connects to, who created it, etc.) and saved or pushed onto a production server. Physically, the connector is a compact XML file, which contains the steps that make up each of the tasks supported by the connector. Rather than reams of code in a low-level language, each step simply contains its required parameters, such as an identification of the page element it operates on, a URL to open, etc. The resulting file is easy to share with others, or to aggregate into collections for applications that need access to many target websites.

Bringing it All Together

The connectors would not be very useful if all you could do were to look at them in the Connector Builder. To actually use a connector after it has been created by the Connector Builder, it is fed to the Connector Engine. This is a piece of software that, given a connector, simply allows software code to interact with the website by invoking the individual tasks supported by that connector. In effect, the connector is turned into a gateway to that website, allowing software to deal with it in a simple, straightforward way. The engine exposes this functionality through a simple API, on top of which more sophisticated software can be built, such as standards-compliant interfaces to allow for new kinds of systems integration, testing tools, XML-based web services, and much more. Index Data will work with partners and customers to implement the kinds of connector templates and interfaces needed to meet the requirements of any given application.

When a Connector is up-loaded into the production environment, it can be accessed by your application through a web-service API, or directly integrated into a meta-search (Discovery) environment through industry standard SRU/ Z39.50 protocols. Still, that is only part of what is required to support a complete information system.

Monitoring and Maintenance

All production Connectors are maintained in a Connector Repository, an environment used to manage the life cycle of a Connector from creation through release, and through all phases of monitoring and maintenance. The Repository also gathers statistical analysis of production use and automated test results in one place, to make it easy to find and repair broken Connectors.

The Connector Repository includes a full web-based administration interface, and exposes a simple webservice API, so it can be integrated into your specific application as tightly as desired.

Access Control

Sometimes it is necessary to access resources that are restricted to authorized users. Our technology allows the authentication function to be parameterized, so that different authentication tokens can be used for different users or groups of users, if necessary. Because the system is based on mainstream browser technology, it can deal with virtually any authentication scheme in use on the web.

At times, especially in Discovery applications, end-users need direct access to a resource returned via MasterKey Connect – most commonly to download full-text content. Managing authentication in such a complex environment can be somewhat challenging, since part of the interaction is between the resource and an automated software system, while another part of it occurs directly between the end-user and the resource itself.

To address this scenario, MasterKey Connect includes a Context Proxy that facilitates direct, seamless access between your end-users and the full-text content they seek. The Context Proxy immediately directs your user to the full-text link in the target resource, without requiring them to login separately for every resource, and without the use of other proxy mechanisms.

Doing Business with Index Data

At Index Data, we realize that everyone brings a unique set of requirements and preferences. For this reason, we customize solutions rather than shoehorning our customers into a set approach. Our technology is entirely based around a service-based architecture (HTTP webservices), and so components can be maintained and operated independently by us, by yourself, or by a third party.

It is fully possible to start out having Index Data host and support all components, and then to take over hosting of components along the way if needed (e.g. to achieve better integration with local systems, or to meet specific formal SLA parameters).

When we provide hosting, our developers can directly monitor the performance of the software, and deal with problems as quickly as possible. We can scale up our hosting environment as needed with a day's notice. In addition to technical support, we can provide full training for your technical staff including both software developers, and those who will be creating Connectors. And, we offer backstop support services for those 'difficult' Connectors that require a special effort to make the most of a given information resource. If you prefer, you can out-task the entire Connector creation process to us and have your staff focus on work more specific to your business.

We offer a range of different models for working with partners who wish to make use of the MasterKey suite of metasearch components, tools, and middleware to which the MasterKey Connect technology belongs. This includes Pazpar2, a high-performance Discovery engine which implements broadcast searching combined with local indexing, result set merging, ranking, sorting, and facets, all behind an easy-to-integrate networked API.

A specialized shop, yet versatile within the field of networked information retrieval, we strive to be flexible both in terms of technical working relationships and business relationships. If you are interested MasterKey Connect or our broader suite of search and information retrieval tools, please do not hesitate to contact us.

Index Data – Information Toolmakers since 1994

January 2011

INDEX DATA LLC, 8 FANEUIL HALL MARKETPLACE, BOSTON, MA 02109 / WWW.INDEXDATA.COM / INFO@INDEXDATA.COM