



MD Link Integration



© 2007 MDI Solutions Limited



	THE MD LINK INTEGRATION STRATEGY	3
	JAVA TECHNOLOGY FOR PORTABILITY, COMPATIBILITY AND SECURITY.....	3
	LEVERAGE XML TECHNOLOGY FOR INDUSTRY STANDARD DATA TRANSFORMATION	3
	EASY TO DESIGN PROCESS-ORIENTED INTERFACES	3
	MD LINK DEVELOPMENT CYCLE	4
	INTERFACES	4
	PROCESSES	4
	SERVICES	4
	TRANSACTIONS.....	4
	MD LINK DESIGN STUDIO.....	5
	EVENTS, TASKS AND RESOURCES	5
	LINKING TASKS TO CREATE PROCESSES	ERROR! BOOKMARK NOT DEFINED.
	DATA MAPPING.....	5
	DEBUGGING	5
	MD LINK SERVER CONSOLE.....	6
	PROCESS SERVICES.....	6
	PROCESS MONITOR.....	6
	TRANSACTION LOG	6
	INTERFACE ADAPTERS.....	7
	DATABASE MANAGEMENT SYSTEMS	7
	MESSAGING SYSTEMS	7
	XML DOCUMENTS.....	7
	FORMATTED TEXT FILES	7
	HEALTH LEVEL 7 PROTOCOL (HL7)	7
	REMOTE FILE TRANSFER	7
	WEB SERVICES.....	7
	HOST APPLICATION ACCESS	7
	ELECTRONIC MAIL.....	7
	WIRELESS TEXT MESSAGING	7
	FILE MONITORING	7
	CUSTOM SCRIPTING.....	7
	INTEGRATION SCENARIOS	8
	SCENARIO 1: INSURANCE CARRIER INTERFACE TO PORTAL.....	8
	SCENARIO 2: HEALTH LEVEL 7 (HL7) PATIENT ADMISSION INTERFACE TO NURSING	9
	SCENARIO 3: WEBSITE MONITOR AND ALERTS.....	10
	PRODUCT SUMMARY	11
	ABOUT MDI SOLUTIONS.....	11

Notice © Copyright 2005. MDI Solutions Inc. All rights reserved

Publication date: February 1, 2005. MDI Solutions does not warrant, guarantee, or make representations concerning the contents of this document. All information is provided "AS-IS," without express or implied warranties of any kind. MDI Solutions reserves the right to change the contents of this document and the features or functionalities of its products at any time without obligation to notify anyone of such changes.

The MD Link Integration Strategy

MD Link is a unique and innovative platform for integrating multiple heterogeneous systems. System integrators can use MD Link to create custom, process-oriented interfaces that can be run on almost any computing platform.

MD Link's integration strategy can be summed up as:

- Support the widest-possible range of data, application and network standards with Java technology
- Easy-to-use data mapping tools leverage transformational power of XML
- Process-oriented Interfaces support complex detailed business logic
- Plugin architecture supports custom development of visual Interfaces for legacy systems
- Design Studio requires no Server for testing and debugging
- Interface-based licensing reduces cost for small to medium implementations
- Processes can be distributed across the network, so that processing is performed closest to the user or data



Java technology for portability, compatibility and security

MD Link Studio and Server are based on the Java 2 Standard Edition and support almost any operating system, without the needs for complex and expensive Enterprise Servers.

Interface Adapters are Java plug-ins that support Visual Mapping of Event, Task and Resource input. Custom Plugins can implement any proprietary or open-source Java library.

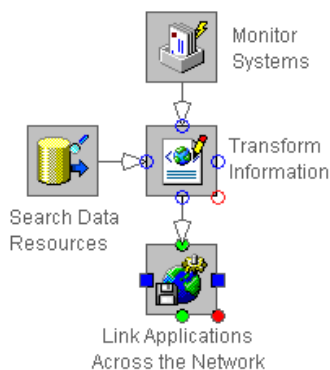
At run-time, these plug-ins generate XSLT templates for powerful transformation of content and structure of almost any data type.



Leverage XML technology for Industry Standard Data Transformation

XML is the industry standard language for data interchange. MD Link Adapters exchange data automatically as XML and use XSLT templates for content and structural transformation of any type of documents.

The visual Design Studio hides the complexity of XML, while supporting advanced XSLT template customization. A wide range of XPath functions support text, math, date and other specialized data conversion requirements.







Easy to Design Process-Oriented Interfaces

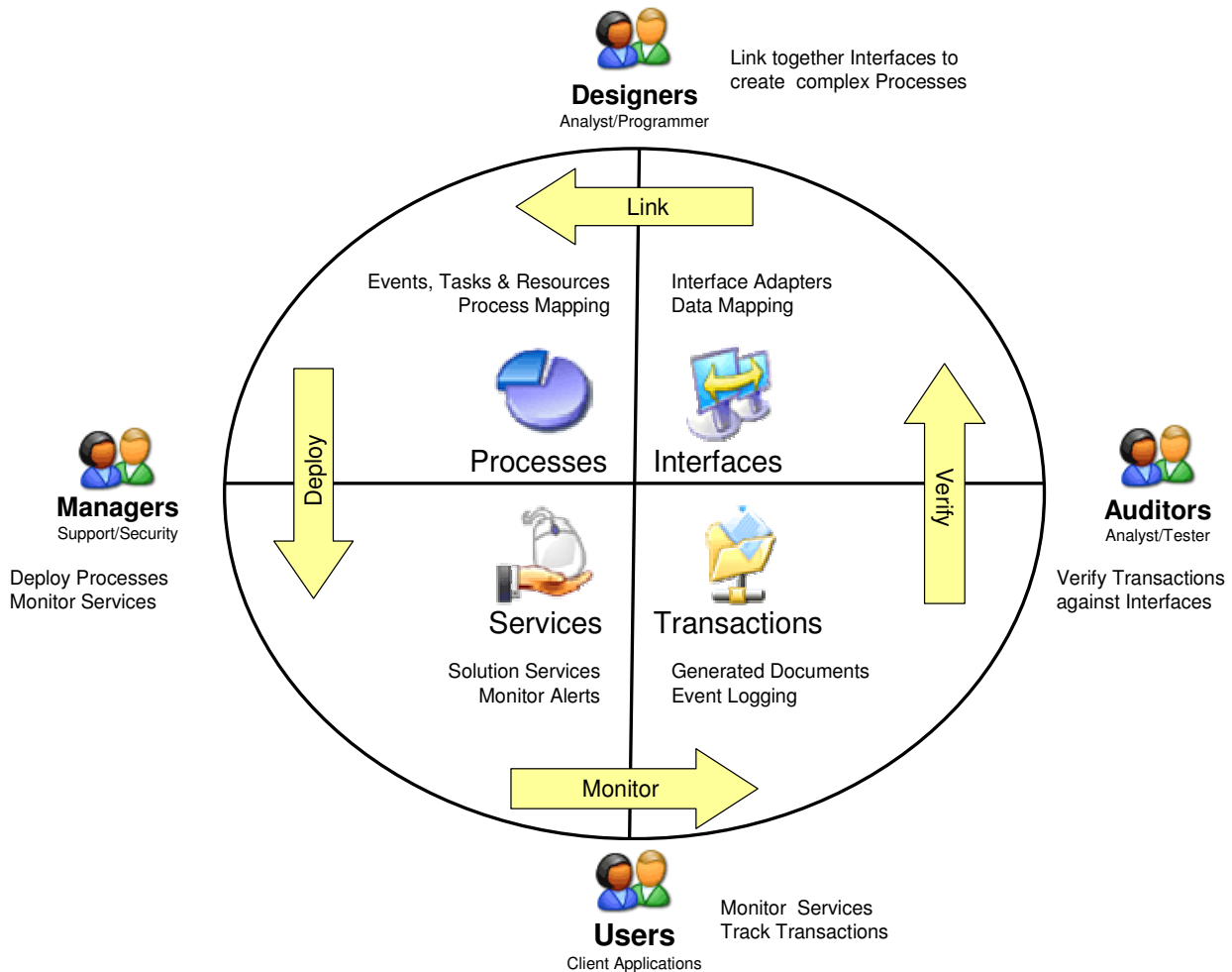
System-to- interfaces are usually more than simple point-to-point queues of data feeds. MD Link supports complex processes that extend the definition of the term "Interface" to mean "create your own process-oriented interface".

To create fully functional business interfaces, you need to create multi-tasking processes that have many steps. MD Link's Event-driven any -to-any Task Mapping model lets you create complex solutions to Interface problems.

MD Link Development Cycle

MD Link supports a role-based, cyclical development model. Designers create Interfaces and Processes. Managers deploy Processes and run Services. Users monitor Services and Transactions. Auditors verify transaction output against Interface definitions. If changes are required, the Designer can implement them.

- Interfaces**  Interfaces are the customized mapping of data, application and network resources. MD Link's growing list of Interface adapters provide visual customization and data mapping. (See Interface Adapters)
- Processes**  Processes link Event, Task and Resource Interfaces to create active solution Services. (See Design Studio)
- Services**  Services are processes that can be hosted on any server. Services monitor Interface status and Transactions activity. (See Server Console)
- Transactions**  Transactions are Service requests that can generate documents, messages and database updates. (See Server Console)



The MD Link Designer Studio lets Designers (Programmers and/or Analysts) create visual interface mapping solutions. The result is a Process that can run as a Monitored Service and generate Transaction Logs.

Events, Tasks and Resources

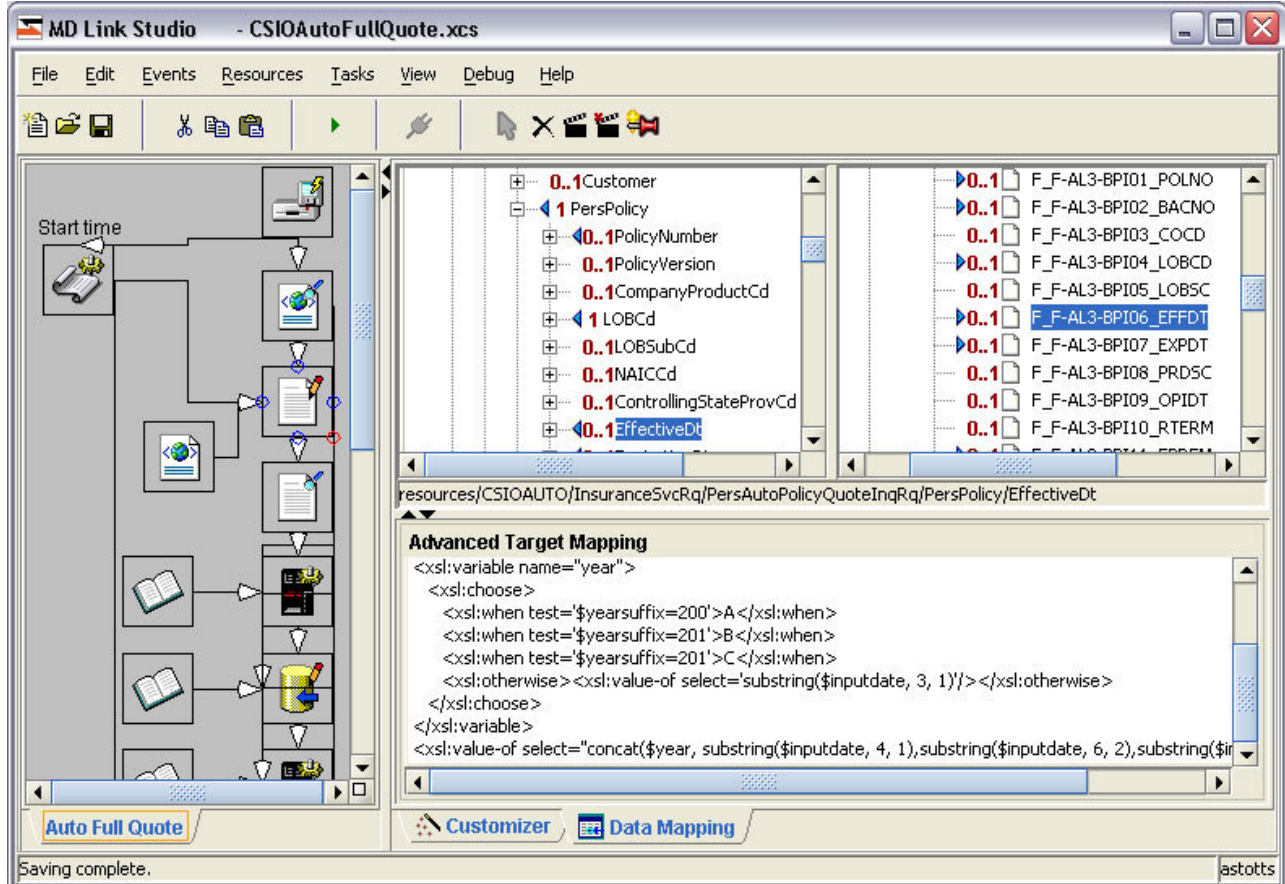
All types of Interface Adapters can be customized to point to data sources, schema definitions and metadata. **Events** monitor data resources and act as application "listeners" that invoke a **Process**. **Tasks** are specialized functions for accessing data, mapping and conversion, linked together to form complex transaction interfaces. **Resources** access static reference information and default values for inclusion into a Task. Error information and working data can be passed to an error-handling process, which can send email or text alerts. **Processes** always start with an **Event**, such as a new file in a folder or message in a queue. The incoming **Event** data record is passed to a series of linked **Tasks**, all of which can access diverse sources of information.

Data Mapping

All types of **Tasks** have a visual mapping screen for drag-and-drop mapping of source data to target data. Any **Task** can be linked to any other **Task** type, resulting in a common mapping view and unlimited combinations of Interfaces can be achieved. The Advanced Target Mapping window on the bottom-right, supports custom coded XSLT templates, as well as text, date and math functions.

Debugging

The Design Studio has a dedicated Console (Server not required) for running solutions and displaying **Log** records. Logging levels settings can filter out debug and warning messages, so you can track the output. The Studio Console lets designers check XML and XSLT templates during testing.



The screenshot shows the MD Link Studio interface for a project named "CSIOAutoFullQuote.xcs". The interface includes a menu bar (File, Edit, Events, Resources, Tasks, View, Debug, Help) and a toolbar with various icons. The main workspace is divided into several panes:

- Start time**: A vertical flow diagram showing a sequence of tasks and resources connected by arrows.
- Data Tree**: A hierarchical tree view showing source data elements such as "0..1 Customer", "1 PersPolicy", "0..1 PolicyNumber", "0..1 PolicyVersion", "0..1 CompanyProductCd", "1 LOBCd", "0..1 LOBSubCd", "0..1 NAICCd", "0..1 ControllingStateProvCd", and "0..1 EffectiveDt".
- Target List**: A list of target data elements including "F_F-AL3-BPI01_POLNO", "F_F-AL3-BPI02_BACNO", "F_F-AL3-BPI03_COCD", "F_F-AL3-BPI04_LOBCD", "F_F-AL3-BPI05_LOBSC", "F_F-AL3-BPI06_EFFDT", "F_F-AL3-BPI07_EXPD", "F_F-AL3-BPI08_PRDSC", "F_F-AL3-BPI09_OPIDT", and "F_F-AL3-BPI10_RTERM".
- Advanced Target Mapping**: An XSLT editor showing a template for mapping the "EffectiveDt" source to the "EFFDT" target. The XSLT code includes a variable for "year" and a choose statement based on the year suffix.

At the bottom of the window, there are buttons for "Auto Full Quote", "Customizer", and "Data Mapping". The status bar at the bottom left indicates "Saving complete." and the bottom right shows the user name "astotts".

Process Services

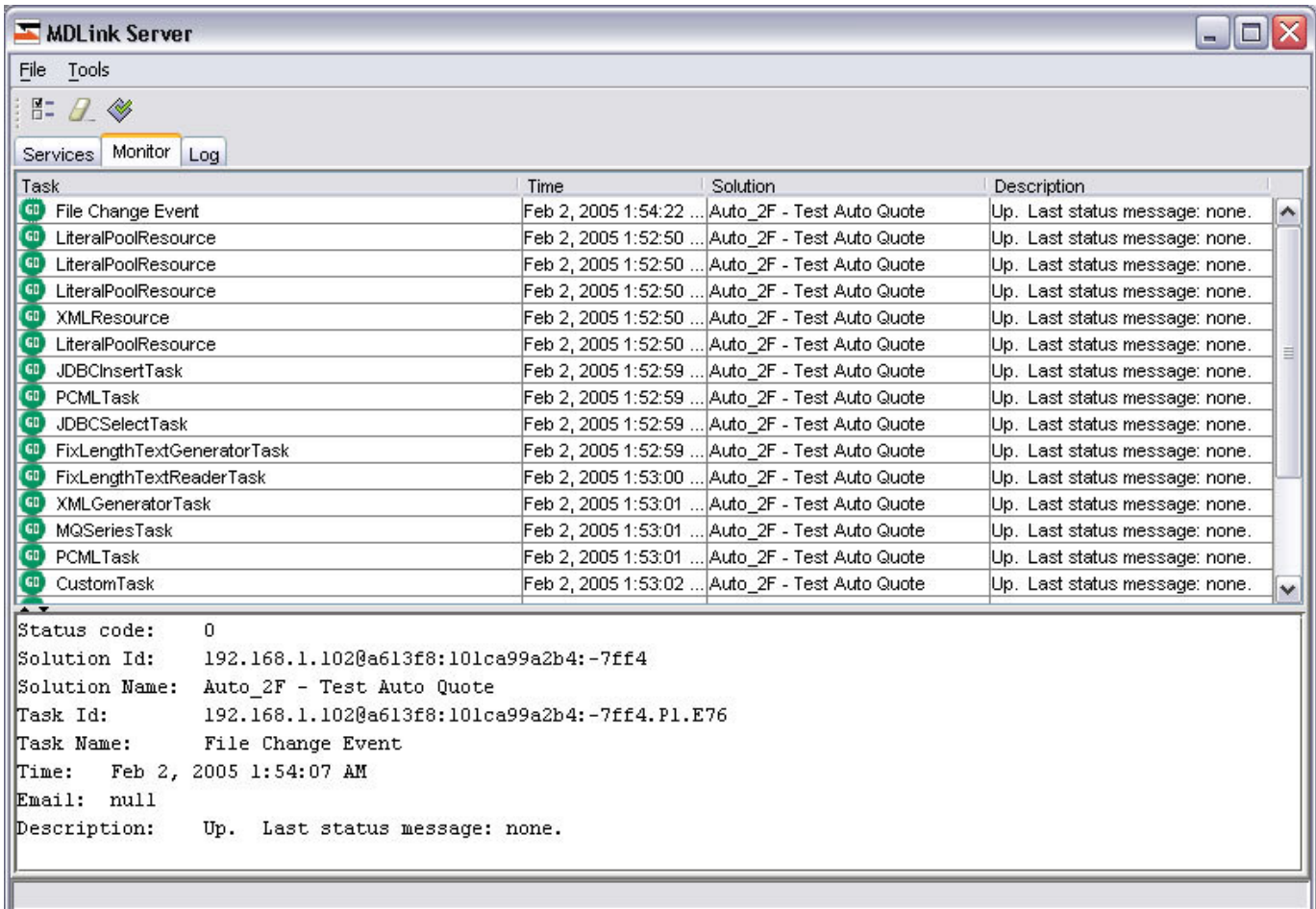
MD Link Processes are saved by the Design Studio as XML solution files containing all the Interface definitions. The MD Link Server can load these XML solution files and run the Process as an active solution Service. While it is running, it will accept data via Event, and then transform the data via XML templates.

Process Monitor

While the Process is running the Monitor displays the color-coded status of each Interface in a Process. Each Interface shows latest activity and will turn yellow for warnings, and red for errors, such as network trouble. Each interface can be paused and resumed by a Manager.

Transaction Log

As each Event and Task executes in sequence, Log records are generated that show informational, warning, error and debug information. The level of details can be set, depending on debugging and performance objectives.



The screenshot shows the MDLink Server Monitor window. The window title is "MDLink Server" and it has a menu bar with "File" and "Tools". Below the menu bar are icons for Services, Monitor, and Log. The Monitor tab is active, displaying a table of tasks. The table has four columns: Task, Time, Solution, and Description. The tasks listed are:

Task	Time	Solution	Description
File Change Event	Feb 2, 2005 1:54:22 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
LiteralPoolResource	Feb 2, 2005 1:52:50 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
LiteralPoolResource	Feb 2, 2005 1:52:50 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
LiteralPoolResource	Feb 2, 2005 1:52:50 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
XMLResource	Feb 2, 2005 1:52:50 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
LiteralPoolResource	Feb 2, 2005 1:52:50 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
JDBCInsertTask	Feb 2, 2005 1:52:59 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
PCMLTask	Feb 2, 2005 1:52:59 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
JDBCSelectTask	Feb 2, 2005 1:52:59 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
FixLengthTextGeneratorTask	Feb 2, 2005 1:52:59 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
FixLengthTextReaderTask	Feb 2, 2005 1:53:00 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
XMLGeneratorTask	Feb 2, 2005 1:53:01 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
MQSeriesTask	Feb 2, 2005 1:53:01 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
PCMLTask	Feb 2, 2005 1:53:01 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.
CustomTask	Feb 2, 2005 1:53:02 ...	Auto_2F - Test Auto Quote	Up. Last status message: none.

Below the table, the details for the selected task (File Change Event) are displayed:













```

Status code: 0
Solution Id: 192.168.1.102@a613f8:101ca99a2b4:-7ff4
Solution Name: Auto_2F - Test Auto Quote
Task Id: 192.168.1.102@a613f8:101ca99a2b4:-7ff4.P1.E76
Task Name: File Change Event
Time: Feb 2, 2005 1:54:07 AM
Email: null
Description: Up. Last status message: none.
  
```



Interface Adapters

Interface Adapters provide drag-and-drop access to a variety of data, application and communication protocols. The plug-in Java programs provide a graphical representation of each interface's unique properties and data structure. The input or output from any data source can be linked together, offering unlimited possibilities for creating customized integration solutions. Simply drop these plug-ins into the MD Link Studio and begin creating customized integration processes. Developers can create custom Interface Adapters using Java or JPython scripting languages.

 <p>HL7 Protocol</p>	<p>Health Level 7 Protocol (HL7)</p> <ul style="list-style-type: none"> • Parse & Generate HL7 2.X messages • Parse, Generate HL7 3.0 XML • Send and Receive HL7 over TCP/IP 	 <p>File Transfer</p>	<p>Remote File Transfer</p> <ul style="list-style-type: none"> • Send Data Files using File Transfer Protocol • Exchange Files between Remote Systems
 <p>Database</p>	<p>Database Management Systems</p> <ul style="list-style-type: none"> • Interface any Relational Databases • Structured Query Language (SQL) • Triggers and Stored Procedures 	 <p>Messaging</p>	<p>Messaging Systems</p> <ul style="list-style-type: none"> • Exchange Data with Message-Oriented Middleware (MOM) Applications • IBM Websphere MQ Series • Java Message Service (JMS)
 <p>XML</p>	<p>XML Documents</p> <ul style="list-style-type: none"> • Parse, Generate and Serialize XML • XML Schema Definition (XSD) • Document Type Declarations (DTD) 	 <p>Text Files</p>	<p>Formatted Text Files</p> <ul style="list-style-type: none"> • Design, Parse and Generate Text Files • Fixed or Delimited Fields • Hierarchical Record Matching
 <p>Web Services</p>	<p>Web Services</p> <ul style="list-style-type: none"> • Send Data to any Web Page • Receive Data from Java Servlets or ColdFusion Custom Tags (CFX) 	 <p>Host Access</p>	<p>Host Application Access</p> <ul style="list-style-type: none"> • Send and Receive Data with Host Systems • Interface with Legacy COBOL & RPG Programs
 <p>E-Mail</p>	<p>Electronic Mail</p> <ul style="list-style-type: none"> • Simple Mail Transfer Protocol (SMTP) • Post Office Protocol (POP) • Dynamic Message Content 	 <p>Mobile Access</p>	<p>Wireless Text Messaging</p> <ul style="list-style-type: none"> • Short Message Service (SMS) • Dynamic Message Text • Requires Simplewire Subscription
 <p>Monitor Files</p>	<p>File Monitoring</p> <ul style="list-style-type: none"> • Monitor Folders for New or Changed Files • Load Documents into Solution for Processing 	 <p>Custom Script</p>	<p>Custom Scripting</p> <ul style="list-style-type: none"> • Define Customized Tasks using Python Scripting Language • Utilize 3rd party Java Libraries for Extended Functionality

Integration Scenarios

Scenario 1: Insurance Carrier Interface to Portal

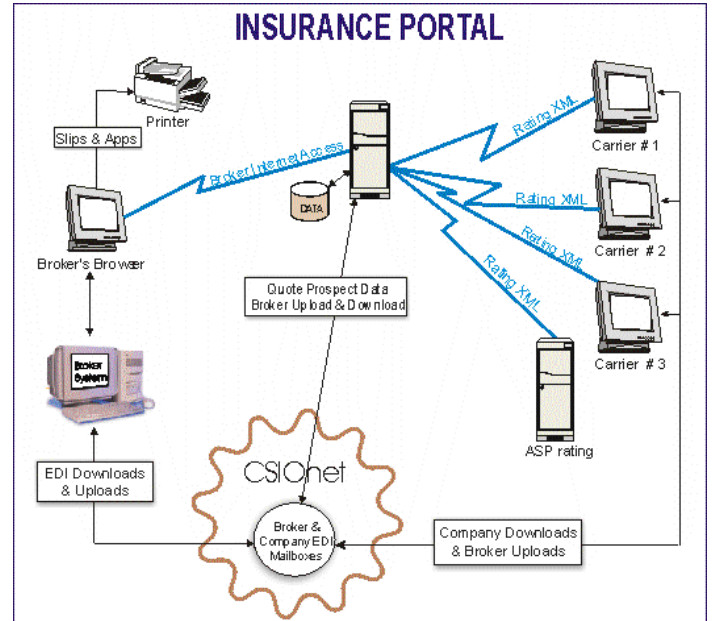
The Center for the Study of Insurance Operations (CSIO) is the Standards body for Canadian Insurance Industry and operator of the CSIO Insurance Portal. Members exchange broker transactions over CSIONet and must communicate with the network using MQSeries messaging. Brokers and Carriers must comply with new XML standard ...but legacy systems only understand legacy EDI formatted text files.

Problem: How to link the Carrier's EDI-based system (AL3) to the new CSIO Insurance Portal (XML) without changing the legacy AS/400 software?

Solution: Use MD Link Design Studio to create an Interface Process to integrate the Carrier's EDI-based rating system with the XML-based network of brokers.

For more information about the CSIO Insurance Portal, see:

<http://www.csioinsuranceportal.com>



The MD Link Studio below shows an Insurance Quotation process. The mapping detail shows the XML-to-EDI conversion.

- Receive MQ Message from CSIO Portal
- Parse Quote Request (XML)
- Write Legacy EDI Quote Format (AL3)
- Insert to Quote Database on DB/2
- Call Legacy RPG program on IBM AS/400
- Read Quote Response from DB2 Database on AS/400
- Parse Legacy EDI Response (AL3)
- Write CSIO Response (XML)
- Send MQ Message to Broker via CSIO Portal

```

Advanced Target Mapping
<xsl:variable name="year">
  <xsl:choose>
    <xsl:when test='$yearsuffix=200'>A</xsl:when>
    <xsl:when test='$yearsuffix=201'>B</xsl:when>
    <xsl:when test='$yearsuffix=201'>C</xsl:when>
    <xsl:otherwise><xsl:value-of select='substring($inputdate, 3, 1)'/></xsl:otherwise>
  </xsl:choose>
</xsl:variable>
<xsl:value-of select="concat($year, substring($inputdate, 4, 1), substring($inputdate, 6, 2), substring($inputdate, 8, 2))"/>
  
```

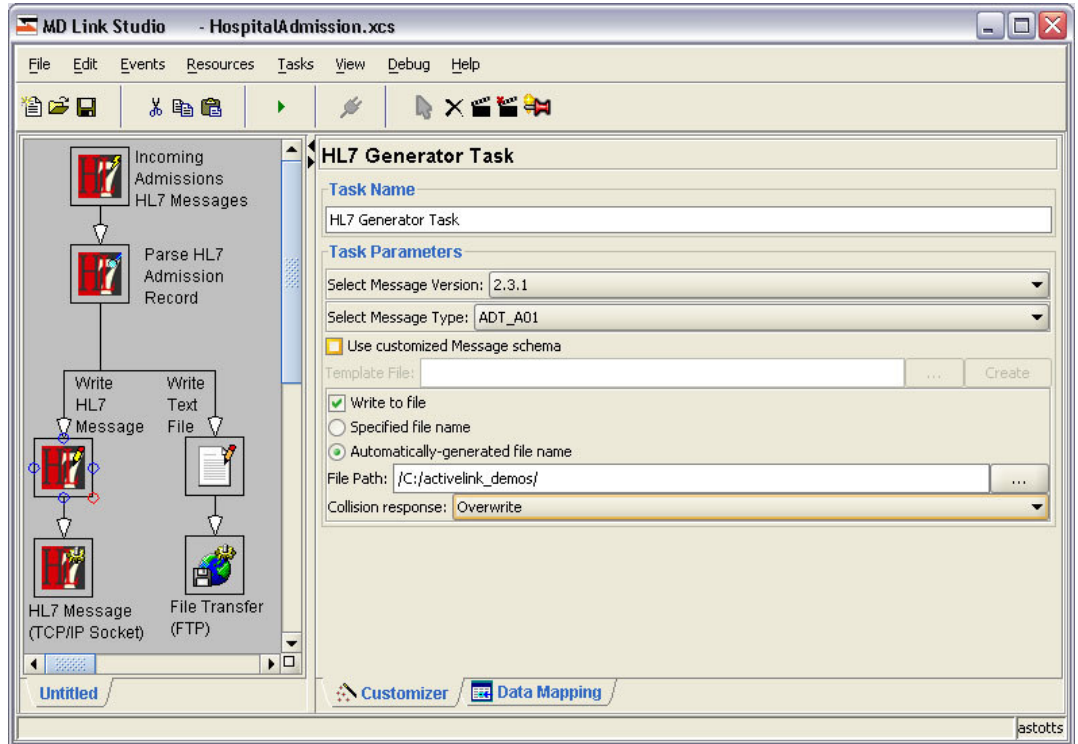
Scenario 2: Health Level 7 (HL7) Patient Admission Interface to Nursing

This scenario shows how to receive, parse, generate and send HL7 messages. An Admissions message queue is monitored using the HL7 Socket Event. The message is parsed, then a Nursing HL7 record is generated. Finally, an HL7 Socket message sends the converted Admission record to a Nursing System.

The HL7 Generator Task's Customizer shows:

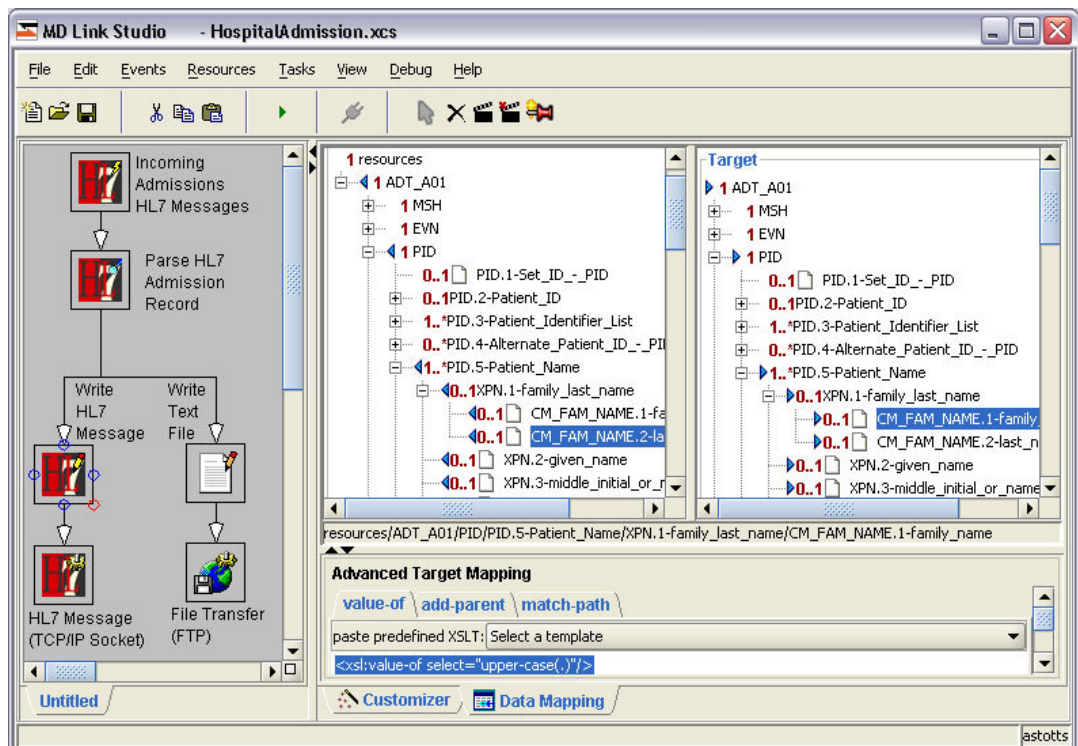
- HL7 Message Version
- HL7 Message Type
- Not using a Customized Message
- Writing to a local folder

The process also shows a proprietary format EDI text file written and sent to another system via File Transfer Protocol (FTP).



The HL7 Generator Task's Data Mapping window shows:

- Source is A01 (Admission)
- Target is also A01 (Nursing)
- The Advanced Target Mapping is converting last name to upper-case characters. This is an example of using XPath function.



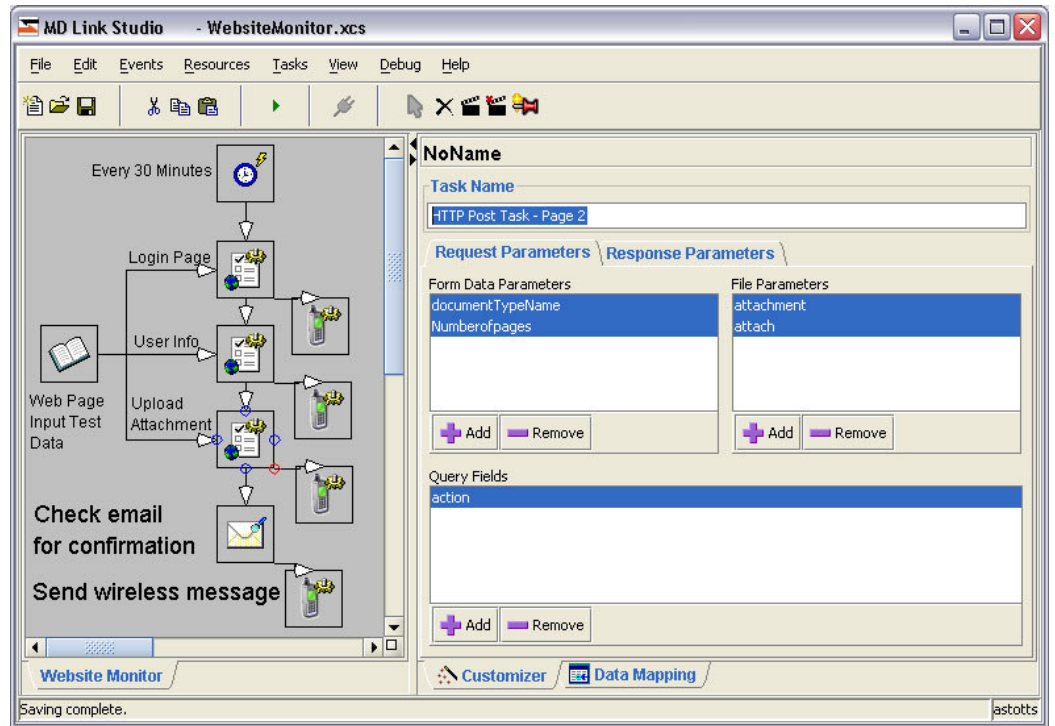
Scenario 3: Website Monitor and Alerts

This scenario shows how MD Link can interface with other systems via web pages, using the HTTP Post Task. In the scenario shown, a Timer Event triggers a form post and attachment upload to a website. After waiting for an email, the solution can send a text message if an error occurs on any Task.

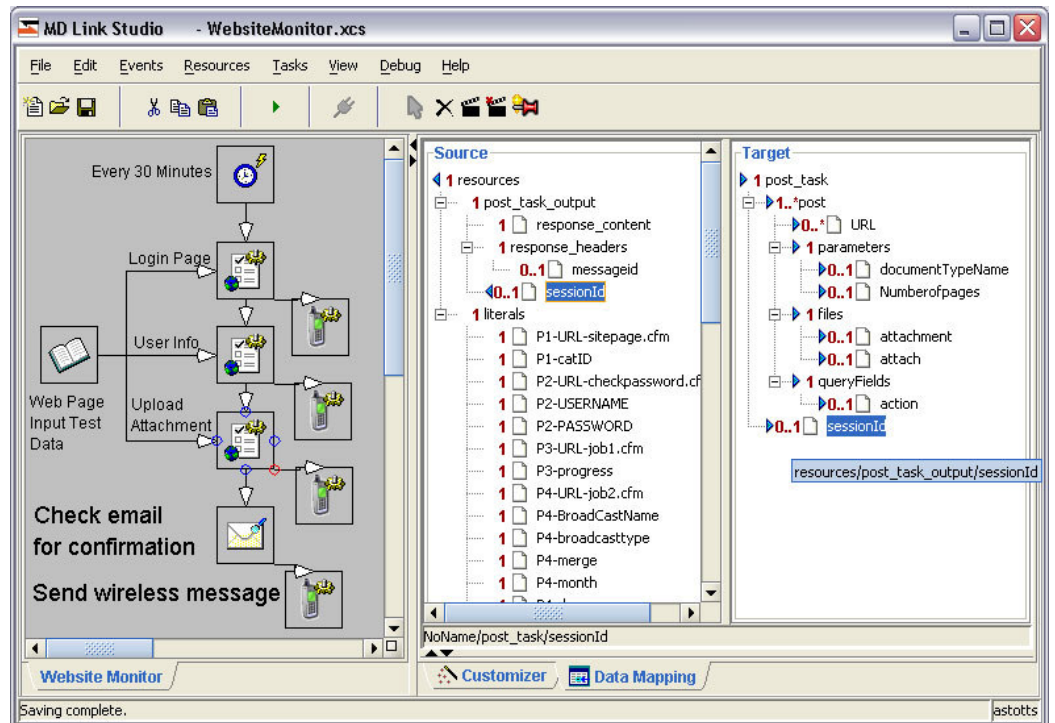
The "Upload Attachment" HTTP Post Task is highlighted.

This process illustrates:

- How to create a "Monitoring" process using Interfaces
- Interface Data can be passed from one Task to another
- How Text Messages can be sent on a Task failure (example: website is not responding)



- The Data Mapping panel show the session ID passed from the previous task
- Both Task and Resource (literal values) are on the Source (left) and the Target (right).





Product Summary

MD Link is unique and innovative software for integrating heterogeneous systems by creating Java-based software “services” that enable an enterprise’s data, application and network resources.

About MDI Solutions



MDI Solutions offers a complete range of outsourced data integration and interfacing services to the healthcare industry. Data integration and systems interfacing provide a strategic foundation of interconnectivity, as healthcare systems strive to make better use of information technology. A strong integration infrastructure, based on qualified, professional resources that develop, manage and continually support it, is an invaluable asset to all healthcare organizations.

MDI Solutions is committed to using leading-edge information technology for the ultimate purpose of improving patient care in our healthcare system. We offer outsourced services, develop and deploy data integration products that enable interoperability between medical systems.

Organizations large and small create and interact with billions of bytes of data on a daily basis. MDI Solutions respects the critical nature of healthcare data. Our services make patient information available to providers and decision-makers when needed, while maintaining the highest levels of data security and integrity.

Integration is defined as forming, coordinating, or blending into a unified whole. MDI Solutions specializes in seamlessly enabling disparate applications to communicate and work together through accepted, industry- standard mechanisms. Our technical staff is proficient with interfacing protocols such as HL7 and XML.

MDI Solutions Limited

Head Office

155 University Avenue
Suite 301
Toronto, Ontario
CANADA
M5H 3B7

Telephone: (416) 255-5113

Web: www.mdisolutions.com

EMail: info@mdisolutions.com

Java is a trademark of Sun Microsystems

Websphere is a trademark of IBM Business Machines

