10 Empowering Reasons to use Corepoint Integration Engine

As a manager of an interface team or as an interface developer, your chosen integration platform needs to make your work easier and more efficient. An integration engine needs to energize your work while enabling you to confidently work on any interfacing project. Corepoint Integration Engine delivers ability, but also much more.

At Corepoint Health, we have taken our years of experience performing essential healthcare interfacing work and built it into Corepoint Integration Engine, with the following goals:

- **Deliver** a feeling of empowerment in the integration work you do
- **Simplify** the complexities of health care interfacing with the ability to handle any issue that comes your way
- **Enable** high-performing data flow between various systems with the confidence that the patient data will always be there when needed

Corepoint Integration Engine dramatically enhances your confidence to implement and manage interfaces, in large part, due to the 10 features described in detail in this white paper:

- Ease of Use
- Consolidated Monitoring, Alerting and Logging
- Implementation Confidence
- Dedicated Team of Support Personnel
- Adaptability to Standards
- Configurable User Profiles
- Interface Version Control Management
- Native High Availability
- Documentation
- Metrics Reporting
Ease of use

Corepoint Integration Engine was designed to be an intuitive software that does not require programming. The software allows the user to do complex things in a simple way. An analyst with workflow knowledge can build once-difficult interfaces with menu-driven icons. The included HL7 wizard analyzes messages for the user, and building parsing rules can take minutes instead of hours.

The end result is a skillset that is less restrictive and resource intensive. The customer is not required to have knowledge of a legacy programming language for a legacy engine. Training and involving more people becomes an easier process. Valuable programmers can be used for other tasks, and users can take control of the interfacing, the data and the timeframe.

- Configure interfaces in less time
- Reduce syntax errors and bad programming
- Increase patient safety with pre-defined parameters
- Leverage existing interfaces with a copy-and-paste approach
- Test new interfaces as they are developed

Menu-driven icons

The engine Navigator is centered on a collection of intuitively organized icons that allow the user to quickly and easily build interfaces. The ribbon bar is located at the top of the Navigator window. It allows you to:

- Access product documentation, samples, support and other helpful information
- Create new objects and edit them
- Filter the list of objects or go directly to an object type without filtering
- Change the layout of the Navigator windows
The icons are organized in a tabular format based on the activity being performed. In Figure 2, the icons in the Map tab are used to map data in the messages as they are passed through the engine. Hovering over the icon gives greater detail to the function of the icon, such as formatting a date.

Wizards are also included, and they can be used to combine multiple steps into a single task. The wizard walks the user through common activities, such as connecting to a database.

**HL7 message analysis**

Corepoint Integration Engine includes built-in HL7 message analysis. This feature highlights HL7 differences so they are easily recognized. The HL7 message differences are shown between a base message and a comparison message. A run log summarizes which messages have HL7 differences. The messages with HL7 differences are marked with a red unequal sign. Messages with binary differences or non-HL7 messages are marked with a blue unequal sign.
Users can further examine the messages with HL7 differences. For example, in Figure 5, the ZND segment, shown in red, was deleted and there is some difference in the PV1 segment, shown in bold.

If the user wants to see only what has changed, the above example can be filtered to show the fields with message differences. The green highlighted text in Figure 6 indicates that those fields were added.

A tabular summary is available as well, based on user preference.
Data flow interface design

The Data Flow pane allows the user to conceptually define an interface based upon the flow of data. Connection icons are dragged onto a canvas to represent inbound and outbound connections. Then, simply by clicking and drawing arrows between connections, the user can define the data flow between the connections and build complete interfaces.

As the user defines the movement of message data between connections, the Data Flow diagram automatically creates the necessary Action List and subscription pool components, based upon your answers to the Data Flow wizard.

Not only is this powerful graphical environment an ideal tool for creating new interfaces, it also is a tool for viewing and modifying your existing interfaces.

From many years of hands-on interfacing experience, Corepoint Integration Engine was designed to balance robust processing with intuitive features. Corepoint Integration Engine empowers you to take control of your time and projects with greater confidence and capabilities.
Consolidated monitoring, alerting and logging

The function of the integration engine is no longer a black box with the Corepoint Integration Engine Web-based interface. Consolidated monitoring, alerting and logging give the user control over the interface environment with the ability to choose how they want to view the interface monitoring screen.

Configurable alerting and monitoring lets the customer be proactive, rather than reactive. As a result, the integration engine team is made aware of an issue before they are notified from a department. Within the monitor, messages can be tracked with the log files. Research and review tools on interface log files let the user quickly find a message, find a block of messages, or even find all of the messages per patient, per day.

Multiple users can view the monitoring tool with varying access rights. This allows departments to also view their interfaces, if desired.

- Don’t be caught off-guard with unexpected phone calls
- View connections anywhere with Mobile Monitor
- Create alerts for common problems
- Advanced filtering quickly finds individual records
- Resolve issues quickly with direct and concise access
Monitoring

Corepoint Integration Engine allows you to remotely control and monitor the engine from a secure web-based application. Because Corepoint Integration Engine Administration is viewed in a Web browser, the tools needed to manage Corepoint Integration Engine and your HL7 connections are readily available from any machine with safe access to the service allowing for maximum flexibility.

View interface connections through your smartphone or tablet using our mobile monitor.

Corepoint Integration Engine provides comprehensive, real-time information about each connection. Some of the information displayed includes:

- The state of the connection, illustrated by color and state column
- How long it has been since a message has been sent or received, presented as idle time
- How behind the connection is in processing data, reported as queue depths
- Message delivery age, delivery rate and time to clear up any backlog
- Active and resolved and inactive alerts

Corepoint Integration Engine provides various views of the high-level connection information including:

- A connection chart, with all connection details in a chart format and indications of the connection state by colored icons
- A connection grid showing connection states and alerts
- Various graphical formats

These views can be grouped by organizational structure, applications and connection groups. These views can also be left ungrouped to dynamically meet the data viewing needs at any given moment.
The connection chart shows details about the connection. The example in Figure 12 is grouped by organizational structure to show an enterprise view of the system.

A message routing graph can be viewed for a quick update of the status for the entire system. The user can also drill down on each icon to view details.

For a more compact view, the connection grid can show status at a glance in a single screen.

For an analysis of the performance of connections, the graphs in Figure 14 visualize the connection processing for the selected information.
Alerting

Corepoint Integration Engine’s proactive alerting provides immediate notification of any interfacing event that needs attention. This allows staff to quickly respond to the issue without constantly manually monitoring the system. Ultimately, issues are resolved quickly—often before there is a business or clinical impact.

The alerting functionality in Corepoint Integration Engine includes:

- Alert Configuration
- Managing Service Levels
- Alert Logging

Alerts are easily configured allowing complete control over which events will trigger an alert. Configuration can be by connection, by day and by time. The alert sub-system can proactively monitor any number of events that include the following unique attributes:

- Day and time of the event
- Type of event (message, connection or queue depth)
- Severity of event required to create an alert
- Escalation of alerts in the case of unresolved issues
- E-mail notification in the case of an alert

In the Figure 15, the alert configuration has different alerts for different days as shown on the calendars, and different times of a weekday as shown on the time bar.

Alerts are automatically displayed in the secure web-based Corepoint Integration Engine Management Administration application sorted by all alerts, active alerts or resolved alerts. Additionally, the system can be configured to send an e-mail to the appropriate staff when an alert occurs.

Corepoint Integration Engine Administration is also used to record resolution to alerts. The resolution information is recorded in the alert log to be reviewed in the future.
REASON 2

All alerts are logged in Corepoint Integration Engine providing a history of alerts and resolution taken to resolve the alerts. Corepoint Integration Engine creates a log that contains each alert and resolution for each connection.

Logging

Corepoint Integration Engine archives every message that flows through each connection.

- Archiving Logs
- Message Trends
- Message Resends
- Log Maintenance

The logs are automatically sorted per-connection and per-day for easy review. Within Corepoint Integration Engine Administration the logs can be sorted by date, connection and log type. Message data, history and metadata can be viewed for each message. In addition to the connection log file, messages that fail processing are stored in a special file for review, repair and resubmission.

In addition to message logging, Corepoint Integration Engine logs connection state changes per your specification during connection creation which can be viewed on the History tab. This allows you to maintain the desirable amount of history for each connection.
Corepoint Integration Engine allows you to view the logged information in a graphical format to see messaging trends for a connection. The graphical format allows you to compare messages received, messages sent, messages delivered, messages errored, queue depths, delivered age, inbound processing time and outbound wait time for the connection.

A quick glance at a connection graph in Figure 18 reveals that the delivered age of messages spiked three times but the connection quickly caught up each time and the queue depth was relatively low the entire time so this connection is keeping up in spite of message number peaks.

Message, error and alert logs are viewed easily in Corepoint Integration Engine Administration.

Corepoint Integration Engine provides the ability to:
- Track a message throughout the system
- Know exactly when a message was sent, received and acknowledged, and what information the message contained
- Easily troubleshoot communication problems
- View message body in text, tree or hex form
- View a graphical representation of data flow for a connection
- Edit messages for testing and resubmission purposes
- View a date range of logs for message load trending

Corepoint Integration Engine provides the ability to resend messages. The value of this very powerful feature is quickly realized when messages must be reprocessed due to an ancillary system issue or reload.

Corepoint Integration Engine automatically deletes log files that reach a configured age.

Corepoint Integration Engine also provides the ability to configure what time the log file maintenance is performed, a different number of purge days for different types of logs and the logging verbosity. Additionally, the number of days logs are kept can be configured separately by connection.
Implementation confidence

Corepoint Integration Engine allows connections to be implemented in the shortest timeframe possible.

One of the key components in building connections quickly is the ability to test throughout the product. The user can quickly determine where they need to focus efforts without end-to-end testing at every iterative modification. The testing tools compare inbound and outbound messages and highlight differences in the development cycle. This allows for validating the accuracy of interface development well in advance of connecting two systems together in a test interface.

Another key component of implementation is efficient message throughput time. Action List Profiling allows the user to quickly see which lines in the Action List are consuming the majority of processing time.

With health IT professionals working on multiple projects, connecting interfaces quickly frees up time for other critical activities. There is no waiting on vendors to set up connections. Corepoint Integration Engine allows the user to repurpose data for a new partner, dramatically reducing the time that would be required if a completely new connection was required. With the end-to-end testing and ease-of-use features previously mentioned, the customer is in control to meet and exceed the deadlines for interface implementation.

- Robust testing capabilities
- Avoid the redundant steps of compiling, loading, running, and searching logs
- Build interfaces correctly, the first time
- Save interfacing logic for future use
- Test-as-you-develop approach eliminates coding errors
Testing with sample messages

Recognizing and learning about vendor differences in the formatting of interface messages starts with examining the message specifications and sample messages. The set of sample messages should be large enough to represent all types of messages you may be receiving. Using Corepoint Integration Engine, you can compare the specifications from the healthcare vendor’s messages with your message format to uncover any differences.

Figure 18 illustrates how Corepoint Integration Engine compares a vendor’s specification for a result message to the HL7 2.2 standard definition of an ORU message. The differences between the message data and the published standard formats are outlined in red.

As shown in Figure 19, the standard contains the following segments that the vendor’s specification does not: PV1, NTE-1 and DSC. The vendor’s specification contains a custom ZPS segment after the order detail group.

Figure 20 illustrates running sample messages against a message format using Corepoint Integration Engine. In this example, sample messages from the vendor, whose specification is shown in Figure 18, are run against the 2.2 standard definition of an ORU message to gain further confirmation of message differences.
Testing with a custom derivative

Using the vendor specification or sample messages, the user can create a custom derivative that is representative of the vendor’s message format. This allows Corepoint Integration Engine to receive and send messages in the application’s desired format.

Figure 21 shows the same specification and a representative derivative created in Corepoint Integration Engine.

Once you have the derivative created to match the specifications, you can use Corepoint Integration Engine to check the conformance of the sample messages against the derivative and make modifications until all messages pass conformance.

Figure 22 shows the conformance checking options and results available in Corepoint Integration Engine.
Action list profiling

The Action List Profiler is a line-by-line run-time analysis tool that tells what percentage of total execution time was consumed by each line in an Action List. A component of the Action List Tester, the Action List Profiler is useful for determining which lines in the Action List are consuming the majority of processing time when processing a set of test messages. The user can define a set of results as a baseline and compare performance against that baseline after modifying the Action List.

The Action List Profiler also reports the performance of child Action Lists, reflecting the execution time of a child Action List as a percentage of total execution time. By drilling down further into the child Action List, the user can examine each line in the Action List in the same manner as in the parent Action List.

Profile results can be viewed in aggregate category view, showing numeric values for each performance category.

Profile results can also be shown in a tree view that shows a line-by-line numerical value indicating the percentage of total execution time that each line in the Action List represents.

Further customization options include profiling individual messages or profiling the entire message set, as well as including or excluding the one-time performance costs associated with such actions as loading an object or loading a type tree for the first time.

When testing an Action List, you can gain additional insight into Action List execution using the Action List Coverage pane. Using color coded text and graphical icons, the Action List Coverage pane shows whether or not every part of an Action List is being executed. This would be useful, for example, in verifying that conditional branches are functioning as designed.
Dedicated team of support personnel

Corepoint Health has a dedicated team of support personnel to ensure the customer is not alone in solving their interfacing challenges. Corepoint Health works closely with customers to fix issues to resolve any unique requirements they have for their particular install. The competent and supportive customer service team is easily accessible.

Corepoint Health is exclusively focused on healthcare integration software. The team at Corepoint Health has a long history of healthcare experience, giving the customer confidence in the company and its expertise.

Corepoint Health has solid references and is committed to taking care of its customers so it can continue to lead the industry, as evident by its #1 KLAS® ranking for interface engines in 2009, 2010, 2011, 2012 and 2013. The Corepoint Health team listens closely when supporting the customer. Many customer comments have influenced product features released in Corepoint Integration Engine. Corepoint Integration Engine is more than the product; it is the team behind the product.

- Customer service is easily accessible when it matters most, by phone or email, 24/7
- For five consecutive years, our customers have voted Corepoint Health Integration Engine #1 in KLAS®
- We listen to our customers - translating many suggestions into new product features
- Exclusively focused on healthcare, our expertise provides confidence in our company, our product and our support
Adaptability to standards

Corepoint Integration Engine is flexible in implementing methodology to accommodate a variety of workflow requirements without placing a burden on the customer. The goal of Corepoint Integration Engine is that there will be no limitation or omission that will keep the customer from doing their job. Things that allow for flexibility include derivatives, gears and healthcare standards support. Everything the customer needs to build interfaces is included without relying on a programming language.

Should you want to expand the product, it is ready for that as well. Message Manager or ItemInvoke custom objects are just a few tools to expand upon the product. Corepoint Integration Engine is not a stale product. New versions are continually released to keep up with market needs, such as EHR certification, HIE requirements and evolving healthcare standards.

- Product features and functions keep customers ahead of industry standards
- Corepoint Health leads the industry in all healthcare standards
- The first integration engine to gain ONC certification as an EHR Module
- Seamlessly work with HL7, CDA, CCD, CCR, X12, NCPDP, DICOM, XML, and others

Flexibility of standards

Corepoint Integration Engine can communicate with any HL7-enabled application. The message format differences between “standard HL7” and the way a given vendor implements HL7 are handled effectively in Corepoint Integration Engine. When you exchange messages with a variety of healthcare applications, Corepoint Integration Engine allows you to create and manage the differently defined HL7 messages.
Corepoint Integration Engine stores the vendor’s or the provider’s unique definitions of the HL7 messages they use. Corepoint Integration Engine’s HL7 repository contains all HL7-defined messages, segments, fields and data types. Users quickly leverage these base definitions to create a modified message where only the changes made to HL7 by a given vendor are entered.

Corepoint Integration Engine also supports XML, which is the basis for the HL7 v3 standard. Users have the ability to import custom XSD files and use the message structure defined in the XSD to parse and encode the XML instance. Corepoint Integration Engine comes preconfigured with all the HL7 Version 3 schemas, including CDA. CDA, CCD and CCR are all based on XML and readily supported.

In addition to support for HL7 and XML, Corepoint Integration Engine supports a variety of other standards including X12, DICOM, NCPDP and ELINCS.
Configurable user profiles

User roles allow the customer to filter activities that users of Corepoint Integration Engine are allowed to perform. The administrator can determine what user rights each person should have including access to start/stop interfaces or permission to view log files. The user profiles enable the administrator to assign access rights to users or groups so they can monitor connections pertinent only to their workflows. This empowers the departments to be proactive with monitoring the connections to their own equipment. This also aids the health IT professional to share first-level support responsibility with members of the department.

- Customize and designate user privileges, including access to start/stop and view message log files
- Assign access rights to individuals or a group of user profiles to monitor designated interface connections
- Audit logs provide added security by tracking who has signed into the system and when
**User roles overview**

With user roles in place, a set of connections might look like Figure 29. In this example, some of the permissions granted for users of the two connected systems overlap and some do not.

User roles allow designated individuals to view and troubleshoot certain portions of the interface without giving them the ability to modify the engine configuration. This relieves the interface team from the responsibility of being the first point of contact when interface issues arise, or allows the interface team to designate maintenance and troubleshooting responsibilities among its members.

User roles are particularly useful if a problem occurs in the workflow or with interface data for a connected system. A person with the appropriate user role can investigate the issue rather than needing to immediately contact the interface team. Examples of issues include:

- A user cannot find a patient in the database. What happened to the message?
- An order comes in and contains incorrect information, such as the wrong ordering physician, a bad priority or a bad date or time. Did something happen to the message in the engine, or did a user simply enter the data incorrectly?
- A system is not receiving data. Is the connection still up and running?
Example scenario

Two systems are exchanging messages when the lab technician notices that the LIS system is no longer receiving data.

As shown in Figure 30, without user roles in place, the first course of action would be for the lab technician to contact the interface team to check on the issue. The interface team would go into Corepoint Integration Engine to see what the problem is. Noticing that the connection is down, they would restart the connection and then notify the lab technician that the issue was resolved. A minor issue, with a simple solution, required interface team involvement to get data flowing again.

With user roles in place, the interface team can create unique permissions for the lab technician, granting full access to view his particular connection’s status, history and alerts. Additionally, the technician is also granted the ability to stop and start his connection. The technician is given no other permissions in Corepoint Integration Engine.

In this scenario, upon discovering the issue, the lab technician logs into Corepoint Integration Engine to check the status of the connection. Seeing that the connection is down, the technician restarts the connection, logs out of Corepoint Integration Engine and the data is now flowing to the LIS. The issue was resolved without needing to contact the interface team.
Interface version control management

Corepoint Integration Engine has Configuration Management that allows healthcare IT teams easily manage the transitions between development, test and production. This is an embedded configuration management and version control that keeps track of interface history, compares current interfaces to past interfaces and provides a mechanism to roll back to previous versions of the engine in the event of an emergency.

The interface allows the user to see which version of configured interfaces are running at any point in time. It also allows continuous development testing of current interfaces without impacting production environments with the added security of being able to return to known, tested configurations. Multiple configuration workspaces provide a platform for parallel interface development among team members. Should peer analysts access and modify the same interface object in support of their interface project, the engine will work with both users to automatically resolve conflicts as the workspace is committed to the repository.

- Return to a known, quality configuration when problems occur
- Schedule your “Go Live” in advance
- View who made interface changes and when they were made
- Multiple users can simultaneously make changes to the same interface
- Safely store known, working configurations for rollback
Object faces

The Object Faces feature defines a set of six “object face” environments: Default/Production, Test, QA, Developer 1, Developer 2 and Developer 3. These environments allow specific values to be assigned to supported configuration objects for each object face.

Using the Object Faces pane, you choose the faces that your development process requires and assign values to any supported configuration object such as connections, data points and subscription pools.

When it is time to change environments, the user activates the specific object face. All associated configuration objects readily update their values based upon the preconfiguration. This eliminates the need to manually reconfigure connections, data points or subscription pools when switching between development stages.

Because Action Lists are object face aware, the user can build simple and complex conditional statements that perform actions based upon which face is active.
Configuration object differences

Corepoint Integration Engine includes tools for quickly comparing two configuration objects including Action Lists, connections, data points, subscription pools, message associations, database associations, correlation associations, namespaces and Action List sets. When comparing configuration objects they are placed side-by-side and the differences are indicated by color-coding.

This feature can be used to:

- Troubleshoot interface behavior changes
- Determine if you want to overwrite an existing object with an import file
- Evaluate if you want to set a different active version for the running engine
- Find differences between versions of an object in the repository and workspace when using Configuration Management

The comparison highlights the differences and includes navigation buttons to quickly move through the differences.
Updating workspaces and merging action lists

Because changes made in separate workspaces are isolated from one another, there is a need to update the current workspace with repository changes. Corepoint Integration Engine provides an automated method to keep workspaces up-to-date while providing maximum user-control. During a workspace update, the engine determines what objects have changed since the last workspace update. It then updates the workspace with the changes. If there is an object in conflict, meaning it has been modified in both the workspace and the repository, it creates copies of the relevant versions in the workspace to be used for comparison.

As shown in Figure 33, because the dietary subscription pool was modified in the repository and workspace, the update process copied the base, repository and workspace versions in the workspace for easy comparison and resolution.

Additional support is available for Action Lists that conflict. The program merges the modifications from the workspace and repository copies of the Action List together and updates the copy of the Action List in the workspace with the new merged copy. If a line is modified in both versions, the workspace version of the line is used. The program creates copies of the relevant versions, just like for other objects, to be used to compare and resolve any lines in the Action List that were modified in both versions.
Native high availability

Although there are various definitions of high availability, with Corepoint Integration Engine Assured Availability™ (A2), a backup server and engine monitors the health and heartbeat of a primary engine processing all interfaces. In the event of server or engine failure the backup engine immediately takes over and restores interface processing, providing a reasonable level of continuous operation given a downtime event.

- Message persistency
- Configuration persistency
- Automated monitoring of I/O status and network
- Seamless integration with native alerts
  
  Since high availability is built natively into Corepoint Integration Engine:
- There is intimate knowledge of the health of the primary and backup server, both in active and passive states, allowing for synchronous replication of all the messages and configuration information providing for graceful failover and failback
- The synchronization of all configuration files and connection profiles is handled by Corepoint Integration Engine, making configuration of high availability less labor intensive and more robust
- It ensures that each message is processed only once and in the correct order—every time, in real time.
A2 in action

To enable A2, a primary and a backup server are required, each with a static IP address. The backup server should have similar hardware configuration as the primary server and only used for failover.

An additional static IP address (Service IP address) is required to be assigned at runtime. The Service IP address is the address that external systems will use to connect to Corepoint Integration Engine. The Service IP is assigned to the server that is acting as the active server.

With the A2 solution, the messages being communicated are written to the data persistency queues on the primary and backup servers. Once the message is enqueued on both servers, the acknowledgement is sent. The primary server processes the message, updating the backup server at each stage of message processing. During processing, the primary server copies over configuration and connection changes as soon as they are saved. Consequently, the backup server has the latest configuration and connection information.

Failover

If the backup server is unable to contact the primary server through normal means, it attempts to locate the heartbeat of the primary server. If the backup server does not find the heartbeat, it attempts to contact the Reference IP address(es) to ensure that the problem is not with the backup server.

If it is successful in contacting the Reference IPs, it assumes that the primary server has failed and initiates a failover alert. Automatically, the backup server takes over the service IP address, processes all incomplete messages and accepts and communicates new messages. Healthcare operations and patient care delivery continue seamlessly.
Return to normal operation

A manual process for failback is used to prevent confusion while the primary server is being restored. When the manual failback is initiated, all message data is synchronized. Once complete, the primary server picks up the Service IP address, starts the Corepoint Integration Engine service and automatically starts connections and begins processing. The backup returns to the passive state.

A2 effectively delivers high availability for any clinical data exchanges that are connected through Corepoint Integration Engine. It is an effective solution from multiple points of view including cost, productivity, robustness and manageability.
Documentation

Corepoint Integration Engine provides documentation that allows the customer to more effectively use the product. The product documentation exhaustively details all the features in the engine.

Categorized cookbooks are included that detail common workflow issues to guide you through the implementation of the solution. A context-sensitive help always takes the user directly to the documentation related to the object being used at that time.

Corepoint Integration Engine is also a self-documenting interface engine. Graphical representations of the connections are automatically created so that the user never has to write documentation for their interfaces again. Using this print feature to illustrate the interfacing leads to improved communication between the interface team and application analysts.

- Cookbook examples show detailed solutions
- Context-sensitive help provides pertinent results
- Self-documenting features aid in future projects
Self-documenting graphing feature

The graphing window in Figure 38 provides a view of the interface configuration in a graphical format. It allows the user to quickly create a visual representation of the interface. This graphing utility acts as a self-documenting feature and can be used to print and archive documentation for each interface. It also allows a user to locate information without having to drill down manually through a list of objects or an Action List. You can double-click any object in the graph to go directly to the Editor window for that object.

Additional available resources

Corepoint Health provides a wealth of additional resources on the Web to provide information for healthcare integration and HL7.

These resources can be accessed by visiting the links below:

- **HL7 Resources**  Provides high-level descriptions of HL7 topics.
  www.corepointhealth.com/resource-center

- **Corepoint Health IT Blog**  Provides practical information on healthcare IT and interfacing.
  www.corepointhealth.com/geni

- **HL7 Standards**  This site which is sponsored by Corepoint Health is designed to serve as a healthcare integration resource center with an emphasis on HL7 although it does contain links to other key healthcare integration websites related to other healthcare standards, organizations and commentary. It contains an HL7 blog which highlights key HL7 and integration news.
  www.hl7standards.com

- **Corepoint Health User Community**  This site provides access to postings of other Corepoint Integration Engine users as well as tips and best use case suggestions posted by Corepoint Health staff. Access is for Corepoint Health customers only, however.
Metrics reporting

The Metrics tab provides the ability to view historical connection and system information to gain a clear understanding of how the engine and environment are performing compared to past performance. Using this information you can quickly determine what changes have occurred that might be affecting current performance such as an increase in message volume or an increase in number of connections. It also provides the ability to look at trending to proactively handle future machine and staffing needs.

For example, if you compare message volume with queue depths or CPU usage, you may observe that queue depths have increased 10% over the last two months but message volume has increased 50%. This tells you that the engine is handling the additional message loads. If, however, you also notice that your CPU usage has increased to 90% during that same time you should consider updating hardware to meet the demands of the additional message volume.

For example, if you compare message volume with queue depths or CPU usage, you may observe that queue depths have increased 10% over the last two months but message volume has increased 50%. This tells you that the engine is handling the additional message loads. If, however, you also notice that your CPU usage has increased to 90% during that same time you should consider updating hardware to meet the demands of the additional message volume.

- Know how your server is scaling with your application
- Run historic metric reports and statistical queries
- Retrieve vital information for any length of time: 2 hours, 2 weeks, 2 months, or 12 months
- Use trends and analytical graphs to prepare for future growth

<table>
<thead>
<tr>
<th>DEFAULT CHARTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Counts</td>
</tr>
<tr>
<td>Queue Depths</td>
</tr>
<tr>
<td>CPU Usage</td>
</tr>
<tr>
<td>Memory Usage</td>
</tr>
<tr>
<td>ADT and ORM counts</td>
</tr>
<tr>
<td>Disk Space Free</td>
</tr>
<tr>
<td>Disk Queue Length</td>
</tr>
<tr>
<td>Msg Cnts and CPU Usage</td>
</tr>
<tr>
<td>Msg Cnts and Disk Queues</td>
</tr>
<tr>
<td>NISvc CPU vs Memory</td>
</tr>
</tbody>
</table>
**Viewing historical system and connection statistics**

By default, four charts are displayed on the Metrics tab. You can, however, zoom in to display a single chart, change the time range of the displayed data and drill-down to see chart details. In addition to the pre-defined charts, you can also build your own custom charts to track specific connection or system information that is especially useful for your environment.

Additionally, the Connection Line Chart displays connection statistics up to 24 hours old, rather than only displaying statistics gathered since the user logged into Administration. The connection line chart also has the ability to zoom in to a smaller time range within the last 24 hours and can drill down on group summaries in the chart.

Since historical statistical summaries are kept indefinitely by default, the space requirements to store historical statistics can become quite large over time. Corepoint Integration Engine provides a graphical way to move the data storage to a location that allows for growth without affecting critical processes.
About Corepoint Health

Corepoint Health has the healthcare IT experience and strength to deliver a dramatically simplified approach to internal and external data integration and health information exchange for hospitals, radiology centers, laboratories, and clinics.

Our next generation software solutions are transformational and will streamline your IT environment, provide a fast track to achieving your interoperability goals, and create operational leverage within your organization. Corepoint Health’s solutions achieve a needed balance of being both intuitive and sophisticated while delivering solid functionality and performance.

To learn more, please go to www.corepointhealth.com or email info@corepointhealth.com to request a tailored demonstration. Call us at 214-618-7000.