

Neuron 3.7.5.835

Runtime

FIX – Null Exception Thrown on Startup of Adapter Endpoints/Service Connectors – The master control proxy would be null when starting adapter endpoints that are set to an uninitialized state. For example, any Adapter Endpoint that is in Subscriber or Query mode by has the processing mode to None.

FIX – Null Exception Thrown on Shutdown of Adapter Endpoints/Service Connectors – This could occur if there are endpoints with a state of uninitialized. Since these were never started, their internal ID would be null.

MOD – Neuron ESB and Peregrine Connect libraries – these are now digitally signed.

FIX – Missing Database would stop ESB Service startup – If a database was not configured for a solution, the ESB Service would fail to start.

Business Processes

FIX – Parallel Process Step - If a process contains a parallel step and an object is added to context.Properties that is not a value, enum, or string type or it does not inherit from IEnumerable or ICollection, or it is not from the namespace System.Collections.Generic, the parallel step would fail while cloning the PipelineContext if any property of the object did not have a getter and setter, or if a getter or setter threw an exception.

FIX – Parallel Process Step - If a process contained a foreach loop with a parallel step inside, or a loop with a parallel step inside that used/added objects to the context.Properties dictionary, memory usage and the size of the context.Properties dictionary would rise exponentially on each execution of the parallel step.

This fix changes the behavior of the parallel step in regards to the context.Properties dictionary. Before, each branch of a parallel step would copy its own context.Properties to the processes' context.Properties at the end of the parallel step's execution for each time that same parallel step was executed in the Process. The parallel step generated unique entries in the context.Properties for each execution of that parallel step (per branch) within the same instance of a process (note: the parallel step must be in a loop to be executed more than once in the same instance of a process). Having unique entries would allow users to grab the context.Properties from each branch from each execution of the loop.

Now the behavior is that the context.Properties entry for each parallel branch will be overwritten on each iteration of the loop unless a unique key is given to the context.Properties entry on each iteration of the loop for each branch. Giving a unique key per branch per loop iteration, however, will likely lead to the same issue that this behavior change fixes so it is highly discouraged.

This means users should use the context.Properties entry that was created in a parallel branch before the next iteration of a loop if they do not give the context.Properties entry (that was created in a parallel

branch) a unique key on each iteration for each branch (which again is highly discouraged). It also means that after the end of the loop's execution, there will not be a unique entry in the context.Properties for each branch for each loop iteration, but rather just the last loop iteration's context.Properties from each parallel branch.

This is a reminder that the context.Properties key for each parallel branch after execution of the parallel step is of the form <Branch Name>.<context.Property Key from the branch>. Also, any context.Properties that exist before executing a parallel step are copied to each branch's context.Properties and will be copied back to the original context.Properties after the execution of the parallel step as <Branch Name>.<context.Property Key>.

Furthermore, each branch of each parallel step within a process should be uniquely named. Otherwise if, for example, there are 2 or more parallel steps within a process and for parallel step 1, branch 1 is named "branch" and parallel step 1 branch 2 is named "new branch", and for parallel step 2 branch 1 is named "branch" and parallel step 2 branch 2 is named "new branch", the resulting context.Properties of parallel step 1 could be overwritten by parallel step 2's resulting context.Properties if they have the same key.

FIX – Logging would fail - The log object was not being set for the PipelineInstance object for Parallel process steps. This meant that context.Instance.Trace* methods would not log anything when used in a parallel branch.

FIX – Parallel Process Step - The Parallel process step would throw an exception if an object was added to the context.Properties collection and that object did not have a parameter-less constructor or if that object contained a StringBuilder type field.

FIX – Looping and “Turn off Tracking” - Fixed issue with the "Turn Off Tracking" property on the looping process steps – For, For Each and While. The value of this property is supposed to be passed to all steps contained within the loop. It was discovered that the property value was not passed to steps that are further contained in a Decision, Parallel, Execute Process, Exception, Timeout, Transaction, Lock or Retry step (i.e. a Decision step inside a For Each step). This issue would cause the ExecutionPath message header to become extremely large which could potentially cause issues when publishing the message to RabbitMQ topics, resulting in an error like:

ERROR - Unexpected Shutdown event from RabbitMQ for 'Opera Publisher' Party on Topic 'GeneralLedger'. Initiating source = Peer; Reply Code = 501; Reason = FRAME_ERROR - type 2, all octets = <<>>: {frame_too_large,251620,131064}

Neuron 3.7.5.822

Adapters

FIX - Salesforce Adapter - When receiving events with the Streaming API function, “Unknown Client” errors could occasionally be received from Salesforce during the Bayou protocol handshake. The errors would be reported in the log files as this:

```

{
  "advice": {
    "interval": 0,
    "reconnect": "handshake"
  },
  "channel": "/meta/connect",
  "id": "350",
  "error": "403::Unknown client",
  "successful": false
}

```

The connection logic has been modified to recover from these errors.

FIX - Salesforce Adapter – When using the Streaming API function, if the attempt to retrieve an OAuth token failed, an error would be logged but the adapter endpoint would still be shown in a started state. This has been fixed so that the adapter endpoint would exit to a failed state.

Workflow

FIX - Workflows could get into a bad state when a user Aborts an unloaded workflow, then tries to perform another operation on the aborted workflow (e.g. cancelling, restarting, etc.). This would result in being unable to delete a workflow completely from the database, which in turn could affect correlation functionality.

FIX - Trying to perform an operation on a workflow in workflow tracking when the workflow endpoint and endpoint host were not running on the local machine (i.e. failover setup) would result in a Key Not Found exception popup in Neuron Explorer. This would cause the operation to not be performed.

FIX - Correlated Workflows would not work properly if “Enable tracking workflow activity execution” was disabled on the Workflow Endpoint’s settings.

FIX – Serialization Error in Workflow Tracking - If a workflow definition contained an execute process activity, and the process that the activity uses threw an error that caused the process to fail, the workflow would not be able to be opened in Workflow Tracking in Neuron Explorer due to a serialization exception.

Logging

FIX - If a user has a single instance adapter endpoint, when they failover a new log file would be created if the endpoint eventually failed over back to the original Neuron Instance. Rather, the log file the endpoint was using before failing over should be reused.

FIX - Endpoint Host failover could result in multiple Endpoint Manager logs being created rather than reusing an existing one.

FIX - If a user has a single instance adapter endpoint that will fail and shutdown on startup (i.e. an ODBC adapter set to stop polling on error with a bad stored proc name), it would keep trying to startup without failing over if the failover adapter instance had not attempted to start at least once. This would cause many entries in the Master Service log.

FIX – Log Format String Errors - There were some cases where the logging of an error would use a bad format string causing the logged error to be wrapped in a log4net error similar to "Exception during StringFormat: Input string was not in a correct format."

Database

NEW - Feature: Neuron Explorer will now apply any Patch sql scripts in the Databases screen when Test/Create is clicked.

Health Monitoring

FIX - Restarting endpoint hosts in Endpoint Health did not work properly if there was a failover server involved. This could result in endpoint hosts not failing over properly and endpoint hosts reporting as stopped when they are still running.

FIX – Timeouts and Lock ups – In some cases when calling functions on either an Endpoint Host or Endpoint, such as Restart, Stop or Clear Errors and Warnings, the UI may appear to lockup and the attempted action may timeout after 1 minute. This was due to the calls being synchronous in nature. These have been changed to Asynchronous calls.

FIX – Clear Errors and Warnings would fail – The Clear Panel toolbar button and the Clear Errors and Warnings shut cut menu would not work on either an Endpoint Host or any endpoints hosted by and Endpoint Host.

NEW - Added "Shutdown" Option to Endpoint Health – This new service state has been added for when a user stops an endpoint or endpoint host from Endpoint Health. This state means the endpoint, if in Single Instance mode, or endpoint host, if it has a failover server, will not monitor for failover until it is restarted from endpoint health or the ESB service is restarted.

FIX – Rabbit MQ Monitoring - The RabbitMQ/AMQP management client would send requests without authorization first, then only after a 400 status code was returned would it use the authorization. This didn't cause any error within Neuron but could be noticed in the RabbitMQ Server's management api logs if they were enabled.

Runtime

FIX - Endpoint Host – If the endpoint host experienced an unhandled exception, an error would be logged but the host would not shut down. Now, when an unhandled exception is thrown by the endpoint host, all endpoints will be shut down and then the endpoint host will shut down into a failed state.

FIX – System.Timeout Exceptions for Master Control and Management Service – Users could experience continuous System.Timeout exceptions after 1 minute from the Heartbeat calls and Report Client Statistics calls that each endpoint makes to the internal Management and Master Control Service. This could happen if the database became unavailable for a length of time exceeding 1 minute. The underlying cause is the database connectivity, but database errors by default would not be recorded for up to 11 or 12 minutes due to internal retries. During that time period all endpoint calls which happen approx every 15 seconds would fail after 1 minute with a System Timeout Exception. With a large

number of endpoints, this could cause a cascade effect causing TCP topics and other service calls to fail due to socket exhaustion. After several failed heartbeats, the default behavior of the endpoints and Endpoint Hosts is to shut down. Previously, the only way to recover once database connectivity is restored is to restart the Endpoint Host via Endpoint Health.

This behavior has been changed. The periodic calls into our Management and Master Control service are now datagram and async. If there is a database outage, the errors are now reported immediately. Also, if the database connectivity fails, endpoints and Endpoint Hosts will not stop but continue in a Started state and continue processing messages.

Warnings similar to the ones below would appear in the log files of the Endpoints and Endpoint Hosts as well as the Neuron ESB Event Log.

2024-01-02 12:38:49.742-06:00 [49] WARN - A problem occurred while reporting the client stats for the 'Publish.Compare.SalesReport' Party. Exception : System.TimeoutException: This request operation sent to net.tcp://localhost:50000/DEFAULT/ESBManagementService/ did not receive a reply within the configured timeout (00:01:00). The time allotted to this operation may have been a portion of a longer timeout. This may be because the service is still processing the operation or because the service was unable to send a reply message. Please consider increasing the operation timeout (by casting the channel/proxy to IContextChannel and setting the OperationTimeout property) and ensure that the service is able to connect to the client.

2023-12-19 07:16:23.959+01:00 [10] WARN - Failed to send status of 'Finance.Transport.Sap' endpoint to master control service: System.TimeoutException: The open operation did not complete within the allotted timeout of 00:01:00. The time allotted to this operation may have been a portion of a longer timeout. ---> System.TimeoutException: The socket transfer timed out after 00:00:59.9990073. You have exceeded the timeout set on your binding. The time allotted to this operation may have been a portion of a longer timeout. ---> System.Net.Sockets.SocketException: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond

2023-12-19 07:02:23.015+01:00 [58] WARN - Endpoint Host 'Neuron Default Host' heartbeat has failed. System.TimeoutException: This request operation sent to net.tcp:// localhost:50004/MasterControl/ did not receive a reply within the configured timeout (00:01:00). The time allotted to this operation may have been a portion of a longer timeout. This may be because the service is still processing the operation or because the service was unable to send a reply message. Please consider increasing the operation timeout (by casting the channel/proxy to IContextChannel and setting the OperationTimeout property) and ensure that the service is able to connect to the client.

2023-12-19 07:14:27.843+01:00 [871] WARN - Failed to send status of 'ManagementService' endpoint to master control service: System.TimeoutException: This request operation sent to net.tcp://localhost:50004/MasterControl/ did not receive a reply within the configured timeout (00:01:00). The time allotted to this operation may have been a portion of a longer timeout. This may be because the service is still processing the operation or because the service was unable to send a reply message. Please consider increasing the operation timeout (by casting the channel/proxy to IContextChannel and setting the OperationTimeout property) and ensure that the service is able to connect to the client.

2023-12-19 07:13:33.330+01:00 [1063] WARN - Send or Ping resulted in CommunicationException. The connection to the client endpoint is faulted or closed. This can result from the client failing to perform a clean disconnect or the server is in the process of removing the client from the pool. Party File.Subscriber.Party, Topic FileProcessing, Session 909a29a4-b2c3-494d-9e9e-40acbb514866, Url net.tcp://localhost:61030/ESBTcpPubSubClient/909a29a4-b2c3-494d-9e9e-40acbb514866, Exception : The socket connection was aborted. This could be caused by an error processing your message or a receive timeout being exceeded by the remote host, or an underlying network resource issue. Local socket timeout was '00:01:00'.

FIX – Single Instance restart loop on failed restart - If an endpoint was marked as single instance and it failed during startup (causing it to shutdown), the endpoint would keep restarting.

FIX – Single Instance restart loop on database failure - If an endpoint was marked as single instance and the database went down, the endpoint would keep restarting.

FIX – Single Instance Endpoints inconsistent between runtimes - The behavior of single instance endpoints was not consistent between ESBServices hosted and Endpoint Host hosted endpoints. Now they are the same.

FIX – Endpoint Host startup in cluster - In some cases, if multiple Neuron servers are present and used for clustering (setup for primary and failover for endpoint hosts) the failover server could startup the endpoint host during server startup even though it started on the primary servers.

Neuron 3.7.5.809

Adapters

FIX - AlphaTrust, Box, Dropbox, Dynamics CRM Web API, HubSpot, NetSuite Rest, Salesforce Rest and ServiceNow Adapters – The value for the send timeout property was ignored when calling the API's of the target systems, incorrectly resulting in a timeout of 60 seconds.

FIX - File Adapter - fixed inconsistency with file locking in publish mode. Binary files stayed locked until after the message is published while the lock on a non-binary files would be released before the message was published. Both file types will now retain their exclusive lock on a file until after it is published.

MOD - Rest Adapter - Added support for certificate authentication.

FIX – Rabbit MQ Adapter – If the Adapter loses connectivity to the Rabbit MQ server, but then later the connectivity is restored the adapter endpoint may still report a failed status until the endpoint is restarted.

FIX – SFTP Adapter – Timeouts could occur in Publish mode when attempting to download files when using against Maverick SSH SFTP servers. This was due to the lack of support the server had for opening multiple SFTP Sessions.

FIX – SFTP Adapter - If a network error occurred during the sending of a file, the error message may not be recorded, and it may look like the message was sent successfully.

FIX - Dynamics 365 Web API Adapter - fixed issue when generating schemas for certain categories. User would get an error similar to "Unable to find component
"/components/schemas/Microsoft.Dynamics.CRM.product".

Topics

FIX – Poison Message Exchange – When using the Non Blocking Poison Message Exchange, if a user makes a change to an endpoint associated with a Party, upon saving the change, the internal routing key used by the Poison Message Exchange Handler to route the failed message back to the source party would be removed. The result would be a lost message and no redelivery.

Services

FIX – Client and Service Connectors - If the endpoint loses connectivity to the Rabbit MQ server, but then later the connectivity is restored the adapter endpoint may still report a failed status until the endpoint is restarted.

Business Processes

FIX – Execute Process Step throws Delegate Exception - If a process utilized the ExecuteProcess step and the process being called by the ExecuteProcess step contained a step that can have child steps, an exception would be thrown when testing the process in Neuron Explorer. The Exception would cause the process to abort with message "Aborted: Neuron.Pipelines.PipelineException: Delegate to an instance method cannot have null 'this'. "

FIX - Flat File to XML Process Step - When a file that has both a row with column names and rows to ignore, the flat file wizard would not apply the logic for both when importing a file. Now, it is assumed that the lines to ignore come before the line containing the column names.

FIX - Flat File to XML Process Step - Fixed regression from changeset 10583. When selecting a file with column header names, switching between columns on the Set Column Details page would cause the header row to be displayed twice.

FIX - Flat File to XML Process Step - When importing a delimited file into the wizard, a max size was calculated and making every row the same size. This would unnecessarily add extra spaces to the last field in each row, causing issues when setting a quote character for the last field.

Workflow Tracking

FIX - The "View Message" context menu item was not displaying in workflow tracking on activities that use the Message argument.

Management Suite

FIX – Workflow Commands Fail - If a machine in a Neuron ESB Deployment Group was defined by including a fully qualified domain name, instead of a machine name, then workflow commands from the Peregrine Management Suite would not work as expected.

Neuron Explorer

FIX – Document Repository throws Object Reference Error - An 'Object reference' error could occur when attempting to add a record in Documents (XML Documents, Text Documents, and JSON Documents) and Data Contracts (XML Schemas, JSON Schemas) if a blank data mapper record is saved.

Runtime

FIX - Neuron Operation service, runtime API would throw error 'The reason phrase must not contain new-line characters', when attempting to return runtime instance list information.

Installation

FIX – Installation fails if .NET 4.8 not installed - If .Net Framework 4.8 was not installed on machine and the user runs the InstallNeuronESB.exe, the exe installs 4.8 framework and requests a reboot of the machine before the installer shows any user interface to configure the installation (expected behavior). Once the machine reboots, the Neuron installer would automatically go into the install phase without showing any of the user interface that allows configuration of the installation so Neuron attempts to install with default settings and no license key which would cause the installation to fail.

Scheduler

FIX – Scheduler Party Queue Not Found - If a user changes a Neuron ESB RabbitMQ based Topic, the related Scheduler Party Queue would be deleted, but not recreated. This would cause jobs for the scheduler to fail with the following error, if the Business Process scheduled published to a Topic.

RabbitMQ.Client.Exceptions.OperationInterruptedException: The AMQP operation was interrupted: AMQP close-reason, initiated by Peer, code=404, text='NOT_FOUND - no queue

Neuron 3.7.5.785

Data Mapper

FIX – Concurrency Issue – When two different messages are coming in at the same time and when the two Business Processes run the same Data Map, they could both return the same message output. This was due to a caching bug.

Adapters

NEW – REST Adapter – Publish Mode has been added to the existing REST Adapter. Now users can configure an Adapter Endpoint to poll an existing REST/Cloud API, passing in HTTP headers and Query Parameters while dynamically using static, constants or Environment Variables as their values during the polling.

MOD - Salesforce Adapter - Added support for API versions 58.0 & 59.0

MOD - Salesforce Adapter - Added .NET 5.0 requirement for RFC 6265 path handling when using the Streaming API. Pre-.Net 5.0, cookie paths did not conform to this standard. This was addressed by manually setting the path on each cookie used by CometD.

FIX – Marketo Adapter - While pushing account updates to Marketo, It would work well for some time. However, it would then start to fail with 'Access token expired or Access token invalid' unless the adapter endpoint was restarted. Also, we are not setting the HTTP Header, ContentType, to 'application/json' only if it's not already set, as some APIs need ContentType other than 'application/json'.

Business Processes

FIX – Design Time Testing of Parallel Step - The Parallel step would throw the following exception during design time testing when a parallel step immediately follows a parallel step:

"System.Reflection.TargetParameterCountException: Parameter count mismatch"

Also, during design time testing, the Messages collection created by the branches in the first parallel step would get discarded after the 2nd parallel step executed.

FIX – Design Time Testing of Parallel Step – Steps contained within a branch would not fire their Step Starting and Step Ending events. In some cases, the labels for an event in a branch could point to the incorrect process step.

Neuron 3.7.5.777

Business Processes

FIX – Process Testing at Design Time – The Publish Step could throw an error at design time when testing if the Source ID was the System Party. This could happen when using the feature that allows the user to attach to live endpoints for testing Business Processes. Also, the process being tested would not use the current set of Environment Variables being passed to it during the testing process.

Configuration

FIX – Configuration Monitoring – At runtime and when in the Neuron ESB Explorer, if changes were made to the underlying solution, they would be detected, and the internal configuration would be reloaded so those changes would be immediately available. However, both the runtime and the Neuron ESB Explorer only monitored changes to *.xml type files. Now the following file types are monitored:

- *.txt
- *.json
- *.adm
- *.gz
- *.xslt
- *.xsd
- *.xml

Neuron 3.7.5.773

Messaging

MOD – Creating Reply Messages – When using the Service Endpoint Process Step, reply messages would be created without setting the source id to the current subscriber.

FIX – Rabbit MQ NON Blocking Poison Message Handling – It was possible for the number of retries to exceed the Max Retry Cycles if processing was interrupted.

FIX – Rabbit MQ – When receiving a message, the subscriber channel will always stamp the DestID with the existing Party ID if the DestID value is blank.

FIX – Rabbit MQ NON Blocking Poison Message Handling – If an endpoint was restarted or the host was restarted after a message was redelivered, the retry counter would automatically be reset to Zero because it was being maintained in memory. That is no longer the case as the retry counter is now a persisted property with the message.

FIX - Rabbit MQ NON Blocking Poison Message Handling – There was a chance that a message could be Nack'd and returned to the queue even if submitted to the delayed exchange.

MOD – Rabbit MQ and Blank Source ID – if the source ID of the message being received from a Rabbit Topic is an empty string or null, we will now throw an exception.

MOD – Rabbit MQ and Resubmit of Unack'd Messages – In the course of Resubmitting unack'd messages, if the resubmit failed, we would just log the exception. Now we also send the original message to the Neuron Failed Audit database.

Auditing

FIX – Auditing without Body – If the Audit Message Body property was unchecked (set to False), and the Peregrine Management Suite was being used for reporting, the Message Body would still be audited and persisted.

Neuron 3.7.5.767

Workflow

FIX – Backup and Purge of Workflow Database taking too long - If a workflow tracking activity item had many rows and the purge workflow tracking job ran, deleting the record could take a long time and/or cause the SQL database to become unresponsive. Users must run the updated "UpdateWorkflowTrackingDeletePurgePending.sql" script manually to fix the issue.

Services

FIX – Azure Oauth Policy - When azure ad b2c is used the policy is now specified in the URL path instead of query string and policy includes the domain name/policy or user flow name. Also, the discovery URL format and token default scope format used to get the token for testing has changed recently in OAuth V2.0.

Neuron Explorer

FIX – Using Shortcut Link to open solution fails - Opening a solution in Neuron ESB Explorer would fail when using a shortcut link e.g. setting the target link like so: "C:\Program Files\Peregrine\Neuron ESB v3\DEFAULT\NeuronExplorer.exe" "C:\Users\Martin\Downloads\global_machine"

Adapters

FIX - SharePoint Adapter - Added option to not upload the ESB Message body as an attachment to a list item. Previously the adapter automatically added the ESB Message body as an attachment. This only affects Lists. Document and Forms libraries still automatically upload the ESB Message body.

Messaging

FIX – Poison Message Handling – Optimized the Non Blocking poison message handling so that retry count will survive an endpoint shutdown or host shutdown. Previously retry counts and collections were maintained in memory. Additional logging has been added. Redundant nack logging has been removed. There was also a chance, if an error was thrown in a Business Process and NON Blocking Poison message handling was enabled, the message could still have been Nack'd and returned to the underlying queue when it shouldn't have been.

Neuron 3.7.5.763

Adapters

FIX - SFTP Adapter - If the SFTP Server sent a disconnect event to the adapter during processing of files in Publish mode, it could appear as if the polling thread had stopped, but the heartbeat thread for the adapter would indicate that the adapter was still running. No other files would be processed in publish mode until the adapter was restarted. This was due to the disconnect event being sent was not being processed correctly.

Workflow

FIX – VB Code Activity and Arrays - If a user added an array to the variables tab in workflow designer and they had a VB Code Activity, the code activity would fail due to the presence of square brackets in the type name.

Business Processes

FIX – Lock Process Step - Fixed an issue that occurs when three or more instances of a process containing a Lock step are executed simultaneously. One instance will start executing the steps inside the Lock step immediately while the other instances of the process wait (expected behavior). However, once the first instance is complete, all other waiting instances would then continue executing the steps in the Lock step (unexpected behavior).

FIX – Copy and Paste incorrect Process IDs – Each Process Step within a Business Process is represented by a unique ID, during a copy and past operation, the new copied Process Step would did not have a new ID generated. This also occurred when copying a Business Process. This has been corrected.

FIX – Flat File Parser Step – Modified the internal caching logic to use the underlying ID of the Process Step, rather than the name of the Process Step.

Neuron Runtime

FIX – Redundant calls against Configuration Service – When retrieving neuron entities or documents, if not found in the configuration service, the call would be made an additional 5 times.

FIX – Patch Installation Script - Users would see that the assembly redirects for Newtonsoft.Json were not the correct version after running the patch. Also, some appSettings were not getting copied over to the new appSettings config file.

Security

FIX – Azure AD Client Oauth and Design Time Error – Several corrections have been added for handling the differences between Azure AD Oauth v1 and v2. Also, the UI has been augmented to provide better error handling and guidance.

Neuron 3.7.5.747

Messaging

FIX – Rabbit MQ Topic – If a message was published with the new Files header property populated, it would not get deserialized with the message upon receipt by the subscriber.

MOD – Rabbit MQ Topic and Client API – If hosting the Neuron ESB Client API in a .NET application and subscribing to messages over a Rabbit MQ Topic using a version of Client API that is older than the version on the Neuron ESB server, the hosted Subscriber would throw a KeyNotFoundException upon receipt of a message.

FIX – Subscriber Not Found Exception - If a subscriber utilizing a TCP topic is added to a running solution and that subscriber is used in an Endpoint, the subscriber would not receive messages published to the topic until the ESB Service is restarted.

Business Processes

MOD – Edit Message Dialog – The FaultType, From and FaultTo ESB Message properties have been added to the property grid.

FIX – Parallel Process Step - fixed issue with cloning context properties. The exception thrown was "System.MissingMethodException: No parameterless constructor defined for this object.". This bug was inadvertently introduced with build 3.7.5.716.

Data Mapping

FIX – Number to Double - Number to Double conversion would not provide decimal places.

Database

FIX – Deadlocks during backup - When running the PurgeAndBackupNeuronDatabase stored proc (called from the job created by CreateJob_PurgeArchiveNeuron.sql job creation script), if there was a lot of workflow data to be deleted, it could cause lock escalation to occur on the workflow tables leading to the database becoming unresponsive (causes workflows to not process).

Adapters

FIX – SAP Adapter and Connection management - When using transactions, the current connection will automatically be closed when a COMMIT or ROLLBACK message is sent to SAP.

FIX – ActiveMQ Adapter fails to reconnect – If an exception occurred on the connection to Active MQ, the adapter could fail to reconnect if the first attempt to reconnect encountered an exception.

Neuron Explorer

FIX – Schema Generation throws null exception - when creating a schema using the Meta Generation Wizard, if a user clicks on another schema object, a null exception may be raised when we check dependencies.

Neuron 3.7.5.738

Business Processes

FIX – HTTP Client Utility UI issue- The save and cancel buttons would be forced out of view when the swagger drop down was shown.

MOD – Thrown Exception from Code Step and Line Numbers – Testing a business process that throws an error within a C# step or C# class will now record the correct line number in the PipelineException class property and report it correctly in the UI.

Adapters

FIX – Not populating Metadata – If all database or ODBC adapters, if the adapter was placed in Query mode, the meta data for the adapter would not be returned in the header properties.

FIX – Metadata Generation Wizard – Connection string properties were not dynamic and displayed in the incorrect order.

Neuron 3.7.5.721

Business Processes

FIX – Adapter Endpoint process step - if a user renamed an adapter or service endpoint and then attempted to run a process that used either the Adapter Endpoint or Service Endpoint process step configured with the old name of the endpoint, it would throw a generic key not found error.

FIX – Service Endpoint Process step step - if a user renamed an adapter or service endpoint and then attempted to run a process that used either the Adapter Endpoint or Service Endpoint process step configured with the old name of the endpoint, it would throw a generic key not found error.

Adapters

FIX – ODBC Adapter SQL Injection - Updated SQL Injection code to improve handling of checking special keywords. Also to avoid checking keywords in case of parameters are used for batch size 1 query.

Neuron 3.7.5.716

Adapters

MOD – FTP/FTPS/SFTP – All Xceed libraries have been updated to the latest version (7.1) from 6.9.

Workflow

FIX – Correlation Set not set to null - If a user has a Publish Message activity in a workflow and the user opens the Correlation Set editor on the Workflow Designer for the activity, then saves an empty correlation set, the correlation set would no longer be set to null in the saved activity. This caused an issue with starting multiple workflows in certain cases (whether concurrent or not) due to the workflow endpoint thinking there was a correlated message that could possibly correlate to the endpoint.

FIX – Setting Workflow Variable to ESB Message raises error – If a user attempted to set a variable to type ESB Message an error would occur. Under the type dropdown for variables, two ESB Message types appear. When either is selected, an error would be shown.

Messaging

MOD – Rabbit MQ Client – Rabbit MQ client library has been updated from 6.4 to 6.5. Along with that System.Threading.Channels.dll was also upgraded to 7.x from 4.x. System.Memory redirects have been updated from 4.0.1.1 to 4.0.1.2.

MOD – Rabbit MQ and Time to Live – Time to Live is now supported for Quorum Queues IF the version of Rabbit MQ Server is 10 or higher.

MOD – Rabbit MQ and Erlang Installers – The Neuron ESB Installation has been updated to support Erlang version 26.0.2 (from 23.0.1) and Rabbit MQ version 3.12.2 (from 3.8.5).

FIX – Non-Blocking Poison Message Handling – Previously if an exception occurred, the resubmitted poison message was forwarded to the original topic, causing all eligible subscribers to receive the poison message for reprocessing, rather than the original Party that threw the exception.

NEW – Conditions on Subscriptions – Having conditions on a Party's subscription to a Topic or subtopic is now supported. This is evaluated on the subscriber side before processing.

Business Processes

FIX - Parallel Process Step - instances of complex objects saved as context properties were not deep copied when a copy of the context was created for each parallel branch. This affected the use of the Push and Pop process steps as well, if the first Push step was used before the parallel step.

FIX – Edit Test Message Dialog - Switching body formats using the radio buttons would not be reflected in the property grid. Also, in some cases when loading binary messages from external locations, the binary flag would not be propagated to the property grid. if using a binary message, if the user tested it again....it would still be the binary message, but the body type would be marked as xml.

FIX – Edit Test Message Dialog – When selecting the context menu “Load from Repository”, the contents of the body selected would only appear if the message window was set to a different body type than that selected.

FIX – Edit Test Message Dialog - When selecting the context menu “Load from Repository”, clicking ok for Import Document button would not import the document. Users would have to double click on the item for it to be selected and returned.

FIX – HTTP Client Utility - if a user opened the Http Client Utility designer to make changes, but did not change the Body Type, the designer would automatically change the Body Type to Raw when the user saved the changes.

Monitoring

FIX – Export to Excel for Workflow Tracking – Attempting to export the Workflow Tracking as an Excel document would throw an error.

Neuron Explorer

FIX – Unhandled exception when upgrading a solution - If a user opened a solution created in an earlier version of Neuron, opened a code step that was on an existing business process and then opened the references dialog box and then hit OK to close, an unhandled exception could occur. Usually this was due to redirects in the new Neuron Explorer.

FIX – Dependency check for Business Process throws Exception - when checking dependencies, if a workflow endpoint was not configured with a subscriber, an unhandled exception would be thrown.

FIX – UI Layout Issues – Fixed several UI layout issues in Neuron Explorer including issues when Windows Scaling is set to over 100 percent.

Data Mapping

FIX – Null value assignment throws exception – An exception would be thrown if a null value was assigned to any number field for a JSON target document.

FIX – DateTime Validation – When using simple date formats, an invalid date could be generated rather than an error.

FIX – Data Mapper Process Step fails to throw exception – if an error occurred during the generation of transform, the process step did not throw the error back to the user.

Neuron 3.7.5.687

Topics

MOD – Rabbit MQ Poison Message Handling - When using non blocking poison message handling, a message submitted to the delayed exchange for retries will now have the Destination ID set to the name of the Party submitting the message. This will ensure that when the message is redelivered by the exchange for retry, other eligible subscribers to the topic will discard the message if the Destination ID does not match their Party name. Previously, all subscribers to the topic would receive the failed message on retry, regardless of whether they were the party that generated the processing error that caused the retry.

Business Processes

FIX – Edit Message Test dialog – When selecting either the Load ESB Message, Load From Repository or Load from File option from the shortcut context menu, the message body was not getting set within the main message viewer.

FIX – Edit Message Test dialog – When trying to load an ESB Message that lacked the HTTP Content Type property, and Object reference not set to an instance of an object exception would be thrown.

FIX – Testing Business Process could crash Explorer – When a user tests a Business Process (not DEBUG mode) and sets breakpoints a crash of the Neuron ESB Explorer could occur if the user attempts to continue past the breakpoint by pressing either the F5 or F11 function key. This has been corrected. Pressing either key after the start of the test and when on a breakpoint will cause the test to continue to the next breakpoint.

Neuron 3.7.5.685

Business Processes

FIX - Neuron Explorer and Impartial Configuration after Save - After a user creates an initial solution and hits save for the first time, the Zone object and other dependencies are still not added to the active configuration currently loaded in the Neuron ESB Explorer. Hence if a user clicked the Attach to Endpoint toolbar button in the Process Designer, they would receive a null exception. Now we reload the solution after the initial save to ensure consistency.

FIX – Attach to Endpoint continues to poll status - After the Attach to Endpoint form was closed, timers would still be firing in memory that continually queried the control service. This has been corrected.

FIX – Process Test throw Unhandled Exception - If master control service threw an error due to an impartial configuration being loaded in the Neuron ESB Explorer, an unhandled exception would be thrown.

FIX – Message Viewer for Testing – Could throw unhandled exception if switching from binary to Text or anything to Binary for the body format.

FIX – Data Mapper Process Step – would fail during design testing due to incorrect URL for service

FIX – Debug Fails – Pressing the Debug option for testing would result in a failure message: an error occurred with your process

Neuron 3.7.5.674

Topics

MOD – Rabbit MQ Topic – When starting, Neuron will now log creation of Exchanges and Queues as informational logging entries rather than verbose logging entries.

MOD – Rabbit MQ Topic – If a solution is modified to work under the Management Suite, the Queue for the Scheduler may not be created as expected.

MOD – Rabbit MQ Topic – On startup or change, if a Party was removed from the subscription of the topic, we would attempt to remove the underlying Queue only if there were no pending messages in the queue. This could result in more messages being forwarded to the queue even though there was no corresponding Neuron Subscriber. This has been changed so that if we can't delete the Queue, we'll remove the routing keys on the Queue so that no other messages will be forwarded to it.

Business Processes

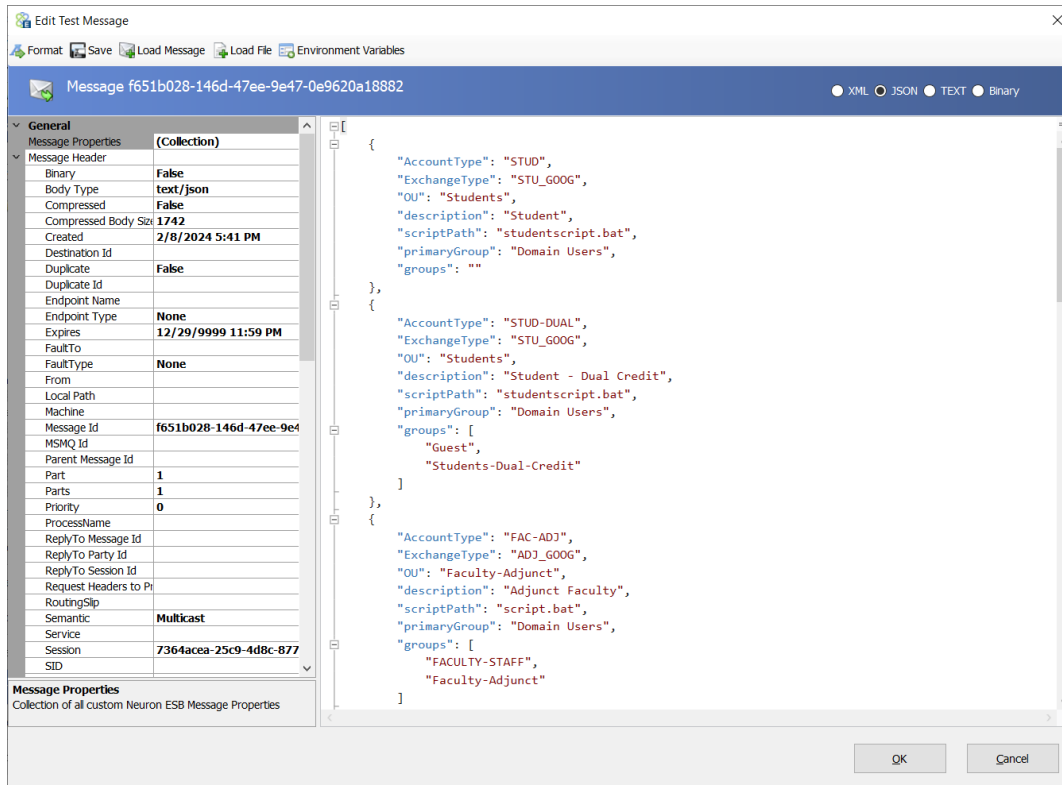
FIX – Data Mapper Process Step – This would always fail when in a Debug vs Test execution within the Business Process Designer if the default port of the mapping service was changed on startup.

FIX – Testing Process could throw null exception – When testing a Business Process in the Designer, if the sample message used a “System” Party, the following error may be thrown:

*Unable to compute subscribers. System.NullReferenceException: Object reference not set to an instance of an object.
at Neuron.Esb.Administration.ESBTopic.TopicMatch(String topic, String subscription)*

MOD – Business Process Testing Enhancements – Several enhancements were made:

1. Message Viewer for submitting Messages –
 - a. This has been refactored to use a property grid instead of a data grid for displaying and editing ESB Message properties. All properties can now be edited or added.
 - b. The message viewer now supports native JSON messages and formatting
 - c. Users can now load messages directly into the editor from the Repository by right clicking the message window and selecting “Load from Repository...” from the short cut menu
 - d. Custom Properties have been moved from the toolbar to the Property Grid for consistency.
 - e. Users can now save any message in the Message Viewer as a *.esbmsg file

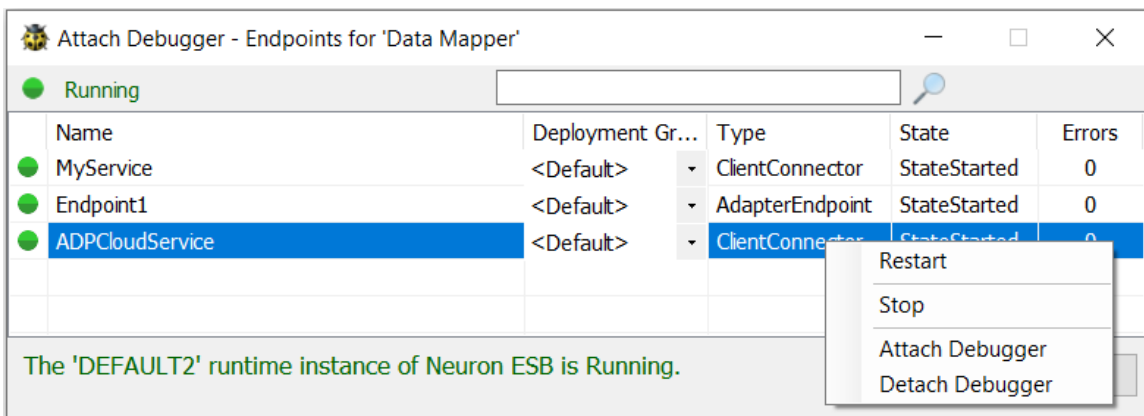


2. Process Testing Modifications

- Trace Window Modifications – Users can now see distinct icons that represent whether the trace line item in the window is Verbose, Informational, Warning or Error.
- There is now an ellipsis button at the end of each line item row in the Trace Window. Clicking on it will bring up the Message Viewer and the entire message present at that point in the process's execution.
- Users can now set Breakpoints on Process Steps. When set, and after hitting the TEST button, the execution process will now stop at the breakpoint. To continue to the test to the next breakpoint, users can hit the "Step Over" toolbar button. Using the Trace window, users can now see the message input into any process step as well as the message output by any process stop by clicking on the Ellipsis button.
- When a breakpoint is hit, users can now select the ellipsis button on the trace line item that reads "Step Output Message..." directly above the current process step where the breakpoint is. Users can now edit that message and hit OK to save the edit. Then, after hitting the "Step Over" toolbar button, the process will continue to execute using the newly modified message.

Type	Step	Text
→	Data Mapper - ...	Press the 'Step Over' toolbar button to start the test or continue...
→	Data Mapper - ...	Step input message...
→	Data Mapper - ...	Machine name resolved to 'DESKTOP-6V9K70R'
→	Data Mapper - ...	Execute - Data Mapper name 'DataMapper1' STATIC
→	Data Mapper - ...	Data Mapper Input Source Body:"
→	Data Mapper - ...	Retrieved the 'DataMapper1' Data Map to execute.
→	Data Mapper - ...	The Data Mapping Runtime URL is resolved to 'http://DESKTOP-6V9K70R:8500/v2/atlas/transform'
→	Data Mapper - ...	ERROR [\$/Order/@Id] \$Cannot read field '/Order/@Id' of document 'OrderXml-XSD', document is null
→	Data Mapper - ...	ERROR [\$/Order/LastName] \$Cannot read field '/Order/LastName' of document 'OrderXml-XSD', document is null
→	Data Mapper - ...	ERROR [\$/Order/Address] \$Cannot read field '/Order/Address' of document 'OrderXml-XSD', document is null

3. Testing Live Endpoints – This is a new testing feature. If the solution is running locally or remotely, users can click the new “Attach to Endpoint” toolbar button. This will display a new list form that displays all the Endpoints currently using the Business Process opened in the Designer. The window will display live status of the endpoints and allow users to restart or “Attach” to an endpoint. All operations are accessible via a short cut context menu. When a user attaches to an endpoint, for example a client connector i.e., webhook, if a message is submitted to the URL of that endpoint it will be redirected to the Process Designer in Test mode. From there, the user can hit the “Step Over” toolbar button to start the execution of the test. To test remotely, the current Neuron ESB Explorer MUST be connected to the remote solution using File -> Connect option. Users can also overwrite the Environment Variables that are used by the Designer during the live test by selecting a different Deployment group in the associated dropdown box in the list.



Adapters

FIX – Metadata Generation Wizard - The wizard would not detect that it was set for an ODBC adapter.

FIX – Metadata Generation Wizard – The Connection String button would be partially hidden by a scroll bar if the Scroll bar became active.

FIX – Adapter Endpoints fail to restart if initial Connect fails – On startup, if an Adapter Endpoint’s underlying Party threw an exception on Connect it would fail and be marked as Stopped in Endpoint Health. It would fail the startup process and not attempt to reconnect on its own.

Services

FIX – Service Endpoints fail to restart if initial Connect fails – On startup, if the Service or Client connector's underlying Party threw an exception on Connect it would fail and be marked as Stopped in Endpoint Health. It would fail the startup process and not attempt to reconnect on its own.

Neuron 3.7.5.662

FIX – Detect Duplicates Step – The function that was added in build 3.7.5.655 which removed any stale entries in the table could generate the following exception:

Exception Type: System.Data.SqlClient.SqlException

Exception Message: Subquery returned more than 1 value. This is not permitted when the subquery follows =, !=, <, <=, >, >= or when the subquery is used as an expression.

This has been fixed. If you are running build 3.7.5.655 or later, to apply the fix, directly edit the value in the DatabaseVersionNumber SQL Server table within the Neuron ESB Database. Change the value from 49 back to 48. Then either click the Test/Create button for the database in the Neuron ESB Explorer or manually run the 0049_UpdateTo3_7_5.sql SQL Script found in the following default location:

C:\Program Files\Neudesic\Neuron ESB v3\Sql

Neuron 3.7.5.661

Topics

FIX – Rabbit MQ Poison Message Handling – When the type is set to NON-Blocking, Neuron ESB uses the Delayed Exchange to provide this functionality. In previous releases, if an unhandled exception was caught, the message at the time of the exception would be published to the Poison Message Exchange. This meant that on redelivery, instead of receiving the original message, the user would get the message that may have been partially modified by their own process. Now, when an unhandled exception is caught, the original unmodified message is published to the Poison message Exchange for redelivery. This is currently how Blocking Poison Message handling works.

FIX – Rabbit MQ and Scheduler Party – When using the Scheduler to start a Business Process, if the Business Process published its message to a Rabbit MQ topic, the following error would be reported.

Publish process step 'Publish' failed. Party SchedulerParty cannot send the message to topic

Installation

MOD – Runtime Library Signing – We are now signing the ESBSvc.exe and NeuronEndpointHost.exe executables

NEW – Embedding MSI within Installer - The NeuronESB installer msi is now embedded into the exe. The Msi will be extracted to the current location of where the exe is executed from upon executing the exe. The Msi can be used for silent installs.

Scheduler

FIX – Starting the Scheduler Endpoint Host Fails – When starting the following error may be reported:

*Starting Peregrine Scheduler 'Peregrine Scheduler' generated the following error.
System.InvalidOperationException: Cannot run without an instance id.*

Workflow

FIX – Testing Workflow fails - When submitting a test message in the workflow designer to test a workflow definition, an exception may pop up on the second and any subsequent attempts to test stating that a dictionary key is null.

Adapters

FIX - SFTP Adapter throws NullReferenceException – When attempting to connect to Microsoft's Azure Blob storage account that's exposed as an SFTP server, the first connection to it works. However, subsequent connection attempts to run other operations will throw the following error:

NullReferenceException: Object reference not set to an instance of an object.

Services

FIX - Amazon OAuth provider - when using any value for the scope, you would receive an "Object Reference not set to an instance of an object" error.

Neuron 3.7.5.655

Licensing

FIX – Deactivate License throws Error - Deactivate License from Neuron ESB Explorer could throw an error saying the license key was null. However, the license would still normally be deactivated. This has been fixed.

FIX – Validate License Warnings on Startup – When starting the ESB Service, 2 warnings may be logged like those below indicating that the license key is invalid. They would indicate an incorrect key, but then the service would eventually validate and startup. This has been resolved. The warnings were like the following and could be found in the Windows Event Log or the Neuron ESB master log file:

Validate License File - Failed. 'The license key UNQJ0X0G07G1HJKM8P1G3W3ZNVQG XK7SIE is invalid. Please enter a valid License key.'

The Neuron ESB Online License 'C:\Program Files\Neudesic\Neuron ESB v3\DEFAULT\NeuronQImInfo.json' failed local validation. Attempting server side validation. The license key UNQJ0X0G07G1HJKM8P1G3W3ZNVQG XK7SIE is invalid. Please enter a valid License key.

Adapters

FIX - Metadata Generation wizard – Could throw an “Object reference not set to an instance of an object” error when the statement used for testing does not parse correctly against ODBC data source. It is fixed to show correct error and provide help with expected syntax.

FIX - Metadata Generation wizard – Could throw an “Index was out of range” error when the execute stored procedure statement didn’t have any parameters for the stored procedure. It is fixed to handle this variation of the execute stored procedure syntax.

FIX - Metadata Generation wizard – Would display a message that the connection string property has not been initialized when clicking on Test Connection or Next button without selecting an existing adapter endpoint. This is fixed to handle setting up the connection string for a new adapter endpoint.

FIX – Pool Size always set to 1 – The Pool Size property internally would always be resolved to 1 regardless of the value set.

Topic

FIX – Rabbit MQ Topic – if a Party subscribed to multiple sub topics and then later removed any of them, we did not detect the removal and sync it with the routing keys of the underlying queue. Hence it still could have received messages sent to the sub topic even though in the Neuron ESB Explorer it would show it was not subscribed to the sub topic.

FIX – Rabbit MQ Topic – if a Party is disabled, all subscriptions for the underlying queue will now be removed. Once removed, it means no messages will be routed to that Queue until the Party is Enabled. Once enabled, the routing keys will be restored allowing it to subscribe to messages again. This is consistent with our MSMQ transport implementation.

FIX – Rabbit MQ Topic – The publishing service could attempt to create our infrastructure, if we failed to create it on startup of the Topic and the Topic was modified before another attempt to create it was made.

MOD – Rabbit MQ Topic – Added support for serializing and deserializing the Email and Files ESB Message Header properties that were introduced in build 3.7.5.568.

MOD – Rabbit Topic Fails to Start – On startup the Neuron ESB Topic may enter a FAILED state when viewed in Endpoint Health. This could happen if Rabbit MQ throws a brokered exception during the startup process where Neuron ESB creates any necessary Queues and Exchanges. If this occurs in the future, It will continue to enter a FAILED state, but the health timer will be started and future attempts will be made to correct the error and enter a STARTED state instead of the user having to manually attempt to restart the Topic in Endpoint Health.

MOD – Rabbit Topic throws Brokered Exception – During startup or on health intervals, the Neuron ESB publishing service for the Rabbit MQ topic could throw a Brokered Exception like the one below:

Failed to Start the RabbitMQ Publishing Service for 'HR_playa' Topic

RabbitMQ.Client.Exceptions.BrokerUnreachableException: None of the specified endpoints were reachable ---> RabbitMQ.Client.Exceptions.OperationInterruptedException: The AMQP operation was interrupted: AMQP close-reason, initiated by Library, code=541, text='Unexpected Exception', classId=0, methodId=0, cause=System.IO.IOException: Unable to read data from the transport connection: An existing connection was forcibly closed by the remote host. ---> System.Net.Sockets.SocketException: An existing connection was forcibly closed by the remote host

Additionally, there could be a complimentary Handshake Timeout exception found in the Rabbit MQ server log file like this:

2023-03-01 04:03:59.518000-08:00 [erro] <0.4369.19> closing AMQP connection <0.4369.19> ([fe80::fe0f:ebce:c0ae:2cb0]:57156 -> [fe80::fe0f:ebce:c0ae:2cb0]:5672):

2023-03-01 04:03:59.518000-08:00 [erro] <0.4369.19> {handshake_timeout,handshake}

This either could indicate that the Rabbit MQ server is restricted on resources and needs the Handshake Timeout increased to 30 seconds (which should be increased regardless). However, this may also occur because the Rabbit Connections and Channels were not being closed timely in the Neuron ESB Topic. This may result from doing numerous starts and shutdowns of the Neuron ESB Service. Now these are being explicitly closed to eliminate that as a possibility.

MOD – Rabbit MQ Deletion of Dead Letter Queue – Previously if a user specified a different vHost other than the default for the solution, the services would fail to start because we would try to first delete a dead letter queue that did not exist in the current vHost but existed in the default. the error would be:

Unexpected Status Code: 404 NotFound

MOD – Rabbit MQ Clean up – On startup, Neuron ESB will only delete queues and exchanges that are not part of the running solution but will preserve the queues and exchanges for other solutions. There is no need to run each solution in a dedicated vHost.

MOD – Rabbit MQ Shutdown Options – A new property, “Shutdown Handler”, has been added to control how the system will behave when a cancellation event or shutdown event occurs. There are two options, ReQueue or AuditFailure. If the subscribing Endpoint is shut down/stopped in a controlled manner or, the internal consumer threw a Shutdown event, a Cancellation Event will occur, and one of two different Informational log entry will be made like this:

Cancellation Event Occurred. Received message with message ID 'da8ec78b-9145-4fab-9923-160bbf69f171' to 'PlayaSubFinance' Party from the Rabbit MQ channel for 'HR_playa' topic. Will be returned to the Source Queue. RabbitMQ delivery tag of 1. The operation was canceled.

OR

Cancellation Event Occurred. Received message with message ID 'da8ec78b-9145-4fab-9923-160bbf69f171' to 'PlayaSubFinance' Party from the Rabbit MQ channel for 'HR_playa' topic. Message will go to the Neuron ESB Failed Audit database. Rabbit tMQ delivery tag of 1. The operation was canceled.

The informational message that is generated is determined by the “Shutdown Handler” property. Requeue will return the inflight message to the underlying queue whereas the AuditFailure option will forward the message to the Neuron ESB Failed Audit database via the Dead Letter Queue.

The following would generate a Cancellation Event:

- The ESB Service is stopped through the Service Control Manager or through Neuron ESB Explorer.
- The Endpoint or Endpoint Host is stopped via Endpoint Health
- Dispose() is called on a hosted Party
- An internal Rabbit MQ Consumer generated a Shutdown event.

However, if an unexpected exception occurred or the subscribing Endpoint, or the Endpoint Host hosting it crashes, the inflight messages that are currently being blocked within the channel will be forwarded to the Neuron ESB Topic’s Dead Letter Queue. From there, Neuron ESB will transfer the message to the Failed Message Audit Report, removing it from the Dead Letter queue. This will result in an error message being logged like the one below:

A message has been saved to the audit failed database. Topic: HR_playa, Party: PlayaSub, Id: b815967d-2dd9-4893-a186-aa2013607867

FailureDetail - The message was retrieved from the configured Rabbit MQ dead letter queue.

Rabbit MQ x-death header

Key: count : 1

Key: reason : rejected

Key: queue : NEURON.DEFAULT.HR_playa_PlayaSub

Key: time : 3/17/2023 12:35:21 PM

Key: exchange : NEURON.DEFAULT.HR_playa

Key: original-expiration : 86400000

FailureType - Dead Letter

MOD – Defaults for Multi-Threaded – The defaults have changed from 5 to 2 with a maximum limit of 10. Ideally this should never be set to more than 5.

Runtime

MOD – Default HTTP Connection Limit - Modified the DefaultHttpConnectionLimit in appsettings.config from 100 to 200. This is also applied in the ESB Service as well now. Previously this setting just affected the endpoint hosted in the Neuron Endpoint Host.

NEW – Auto Tuning Threadpool for ESB Service - This is identical to the feature for Endpoint Hosts which was released in build 3.7.5.311 and can be used to reduce long startup times of solutions, especially if RabbitMQ topics are used. This is configured at the same time Endpoint Hosts are configured when enabling the Auto Tuning feature in the Configure Server dialog in the Neuron ESB Explorer.

MOD – Auto Tuning Optimization - Optimized the thread tuning routine in the Endpoint Host. The routine now only increases the min worker threads only if it detects that doing so decreases execution time.

FIX – Endpoint Manager Log created twice - The endpoint manager log for each endpoint host would be created twice (one with two date stamps appended to the file name).

FIX – WMI Warning on Startup – On startup, Neuron ESB may log the following error and fail to start. We will now log this as a warning and continue the startup process

2023-04-03 14:56:18.681+02:00 [4] ERROR - Neuron ESB Service Failed to Start - Exception: System.Exception: An error occurred attempting to retrieve the Number of Cores info via WMI. at Neuron.Esb.Internal.ESBHelper.GetNumberOfCores

FIX – Endpoint Host does not startup immediately - Upon startup, in some cases the endpoint host would not start immediately. Instead a warning would be thrown after the client refresh interval * 4 elapsed stating that the endpoint host heartbeat had gone stale (then the endpoint host would start).

FIX – Disabled Single Instance throws Null Exception - If a solution contains endpoint host hosting single instance endpoints (adapter or service connector), if they are not enabled to start (either not meant for messaging or marked as disabled), endpoint health may throw an object reference null exception.

FIX - Disabled Single Instance throws erroneous Stale warnings - If a solution contains endpoint host hosted single instance endpoints (adapter or service connector), if they are not enabled to start (either not meant for messaging or marked as disabled), upon stopping the esbservice warnings may be logged that state the endpoints' heartbeats have gone stale (though they were not active on the machine).

FIX – Daylight Savings Time failover - If a solution is configured for failover and is running in a time zone that implements daylight savings time, upon moving forward an hour the failover mechanism would activate due to the time being stored in the database as a datetime rather than datetimeoffset.

FIX – Excessive logging of Failover Cluster not installed - The endpoint host manager and single instance endpoints log files would excessively log the following if INFO was enabled: "Cluster Network Name: Neuron ESB Service is NOT Clustered. Failover cluster is not installed."

FIX – Erroneous endpoint heartbeat message - If an Adapter Endpoint or Service Connector are marked as Single Instance but the endpoint itself is not enabled, upon the Endpoint Host process dying/failing and restarting, a warning would be logged every client refresh interval that would state that the endpoint heartbeat has become stale and would become the active instance even if there are no other instances of the endpoint running (as they are not enabled).

FIX – Controlled of endpoint incorrectly transferred to another server - The heartbeat check for multi instance endpoint hosts (i.e. primary-failover) could take too long to execute which could cause Endpoint Hosts to failover unexpectedly with a message stating that "the heartbeat has gone stale" on

the failover instance and "control of the endpoint has been transferred to another server on the farm" on the primary.

MOD – Configure Server and Auto Tuning – The Configure Server dialog has been modified to support turning on or off the Neuron ESB .NET Thread Pool Auto Tuning feature introduced in build 3.7.5.311.

The screenshot shows a dialog box titled "Configure DEFAULT Server" with a close button (X) in the top right corner. It has three tabs: "Server", "Logging", and "Performance", with "Performance" being the active tab. The "Performance" tab contains the following settings:

Setting	Option	Value	Current Value
.Net Thread Pool Size:	<input type="checkbox"/> Override	Max Worker Threads: 2000	32767
	<input checked="" type="checkbox"/> Auto Tune	Min Worker Threads: 400	16
		Max IO Threads: 2000	1000
		Min IO Threads: 400	16

At the bottom right of the dialog are "Save" and "Cancel" buttons.

MOD - .NET Thread Pool Override defaults – The default for the minimum number of Worker Threads and IO Threads have been increased to 400 from 200.

MOD – Internal Service Binding Settings defaults – The defaults, found on the Server tab of the Zone, have been modified to accommodate high load environments when using Auditing. Specifically,

- Max Concurrent Sessions has been reduced from 1000 to 500
- Max Connections has been raised from 1000 to 2000
- Max Concurrent Calls has been raised from 500 to 2000

When opening an existing solution, these settings and others will be modified accordingly as follows:

- Open Timeout, Send Timeout and Close Timeout – At minimum these will be set to the default value of 60 seconds. If their current value is greater than 90 seconds, they will be tuned down to 60 seconds.
- Listen Backlog – At minimum this will be set to the default, which is the number of processors * 16. If the existing value is more than twice that, it will be tuned down to the default.
- Max Concurrent Calls – Will be set to the default value of 2000.
- Max Concurrent Instances – At minimum this will be set to the default value of 200. If the existing value is more than twice that, it will be tuned down to the default.
- Max Concurrent Sessions – At minimum this will be set to the default value of 500. If the existing value is more than twice that, it will be tuned down to the default.
- Max Connections - Will be set to the default value of 2000.

MOD – TCP Topic Transport defaults – The defaults, found on the Transports tab of the Zone, have been modified to accommodate high load environments when using TCP based topics. These settings are used anytime a new TCP Topic is created. Specifically,

- Max Concurrent Sessions has been reduced from 500 to 100
- Max Concurrent Calls has been raised from 250 to 2000

When opening an existing solution, these settings and others will be modified for every TCP based Topic as follows:

- Open Timeout, Send Timeout and Close Timeout – At minimum these will be set to the default value of 60 seconds. If their current value is greater than 90 seconds, they will be tuned down to 60 seconds.
- Listen Backlog – At minimum this will be set to the default, which is the number of processors * 16. If the existing value is more than twice that, it will be tuned down to the default.
- Max Concurrent Calls – Will be set to the default value of 2000.
- Max Concurrent Instances – At minimum this will be set to the default value of 100. If the existing value is more than twice that, it will be tuned down to the default.
- Max Concurrent Sessions – At minimum this will be set to the default value of 100. If the existing value is more than twice that, it will be tuned down to the default.
- Max Connections - Will be set to the default value of 500.

Data Mapper

FIX – Double and Float Data Types - Minimum allowed value range checking for double and float value parsing. Previously the Data Mapper did not allow negative values for double and float data types. This has been corrected.

FIX – Data Mapper Service Port change caused remote client to fail – If the user has a remote Neuron ESB Client connected to the server that executes the Data Mapper process step and the ESB Service is restarted while the remote client is connect, the Data Mapper Service hosted by the ESB Service may start on a different port. If that happened, the change would not get propagated to the existing connected remote clients. When they attempt to send a message that would execute the Data Mapper Process step, they may get an error like the one below indicated the original port number:

*Data Mapper transform http call failed with HttpRequestException! Error:
System.Net.Http.HttpRequestException: An error occurred while sending the request. --->
System.Net.WebException: Unable to connect to the remote server --->
System.Net.Sockets.SocketException: No connection could be made because the target machine actively refused it 192.168.2.190:8591*

Services

FIX – OAuth Providers and ExpiresOn - Fixed issue with "ExpiresOn" DateTime. If a user opens a Neuron solution in Neuron ESB Explorer on a computer that has an OS Date format that is different than the OS date format on the computer the Neuron solution was last saved in, and that solution contains OAuth Providers, if the user attempted to view the OAuth provider they would receive a FormatException

stating the date is not a valid value for DateTime. The value will now always be saved and read in a specific format.

FIX – OAuth throws Null Object Exception on startup- Fixed issue with client connectors throwing an exception on startup. The error would occur when starting a client connector configured to use Peregrine Connect as the OAuth provider. The text of the error contains the value "System.InvalidOperationException: Nullable object must have a value."

Event Processor

FIX – Event Processor – This could stop when various exceptions occurred due to incorrect information regarding the consumer and/or connection status.

Monitoring

MOD – Auditing Email and Files collection properties - Added support for auditing the Email and Files ESB Message Header properties that were introduced in build 3.7.5.568.

FIX – REST API for Configuration fails – When using the REST API, /neuronesb/api/v1/Configuration/{instance}/DeploymentGroup/{deploymentGroup}, the expected deployment group information would not be returned. The user would receive an HTTP 500 error.

Workflow

FIX – Concurrent Workflows exceeding Maximum - Cancelling an aborted workflow could cause a semaphore to be released allowing for more concurrent running workflows than the max setting.

FIX – Throw with Message fails - When using the Throw with Message workflow activity, if the ESB Message object was manually created in a code step without setting the Topic property, it could throw an error that Topic is null. This has been corrected.

FIX – Correlated Workflows fail to resume - Correlated Workflows may not resume properly upon the Endpoint Host process dying and restarting due to the improper reloading of correlated messages that were sent in as the host died.

FIX – Correlated Messages create extra Workflow instances - Correlated Messages that are processed during failover may result in extra correlated workflow instances due to the message being processed more than once. This would occur if messages were at a certain processing stage when the Endpoint Host process fails/dies resulting in redelivery from a RabbitMQ topic.

FIX – Workflow Tracking fails when machine is IP Address - If a machine in a Deployment Group was defined by an IP address instead of a machine name and the machine's hostname does not resolve to that IP address, then issuing commands from workflow tracking would not work (e.g. restarting a workflow from workflow tracking).

FIX – Cancelling Workflow causes Semaphore is Full warning - Cancelling workflow instances in the Unloaded state can cause a warning that a semaphore was full while releasing or possibly cause more concurrent workflows to execute than the max instances property set on the workflow endpoint.

Business Processes

MOD – Data Mapping Process Step - If we encounter an error indicating the connection was closed, we'll now attempt to retry the data mapping.

FIX – Detect Duplicates Step – Added function to the stored procedure that determines if a message is a duplicate. Now the stored procedure will also remove any stale entries in the table.

FIX – Audit Process Step – When using the step and the Maximum Instances is set to greater than 1 and the step is used in multiple branches within a Parallel Process step, a null exception could be thrown.

FIX – Execute Process Step – When using the step and the Maximum Instances is set to greater than 1 and the step is used in multiple branches within a Parallel Process step, a null exception could be thrown.

MOD – Audit Process Step logging of Faults – at runtime, if an Audit Proxy faulted, the information would be written as an error and then we would auto recreate it. Not it is written as a warning.

FIX – Audit Process Step – Optimizations have been made to reduce locking and timeouts when under heavy load.

FIX – Execute Process Step – Could sometimes return an error indicating that the Pipeline was null under heavy load.

FIX – Decision and Parallel branches in steps – When under heavy load, a null reference exception could be thrown when evaluating the branch condition.

Neuron ESB Explorer

FIX – Import would fail to import Endpoint Host info - Importing a Neuron solution using Neuron Explorer – Fixed issue where Endpoint hosts were not being imported with no errors being reported. The endpoint host would initially appear to have been imported, but when the solution is saved and closed, and subsequently reopened, the endpoint host would not be included in the solution.

FIX – Edit Test Message dialog - Edit Test Message dialog for process designer would sometimes show "<Error in displaying the body.>" as the body of a message when loading a *.esbmsg file because the underlying BodyType was never appropriately set.

Neuron 3.7.5.607

Topics

FIX – Rabbit MQ Topics – Endpoints subscribing to a Rabbit MQ topic may fail with the error that the underlying Queue did not exist. This could occur if there were more than one Rabbit MQ topic with a similar prefix in their name, for example, HR_, HR_Global and HR_Sales would map as the same internally. On startup, each topic would delete the queues that it thought didn't belong. This has been fixed.

MOD – Rabbit MQ Topics – On startup, each Topic will attempt to create a System queue. If the queue already exists, Neuron will remove any existing bindings to ensure that it does not act as an anonymous subscriber to messages published to the topic.

Samples

FIX – All samples have been upgraded to the latest 3.7.5 and all named pipes topics have been replaced with TCP based topics.

Adapters

FIX - NetSuite Adapter - Fixed metadata generation issue when generating metadata for "item" record types (i.e., Inventory Item). An exception of "Object reference not set to an instance of an object" could be thrown.

FIX - NetSuite Adapter - Fixed metadata generation issue when generating any object containing custom fields. An exception was not thrown, but if the NetSuite role did not have proper permissions, the custom fields would not be included in the generated schemas and samples. Now an exception would be displayed containing the error response from NetSuite.

Event Processor

Several fixes have been implemented to prevent “stuck” messages in queues and abnormal CPU utilization and errors such as:

Unexpected Shutdown event from Rabbit MQ consumer for '<eventname>' Event Type. Initiating source = Peer; Reply Code = 406; Reason = PRECONDITION_FAILED - unknown delivery tag <tagvalue>.

and

*Error occurred attempting to send EPS log entry data to Elastic Search:
System.ArgumentException: Destination array was not long enough. Check destIndex and length, and the array's lower bounds.*

Neuron 3.7.5.594

Business Processes

FIX – Excel to Xml fails with too many tasks – If the Excel to Xml process step is used in a Business Process that’s been assigned to multiple endpoints errors like the one below or Access Violations could occur:

*Neuron.Pipelines.PipelineException: ERROR [08004] [Microsoft][ODBC Excel Driver] Too many client tasks.
ERROR [01S00] [Microsoft][ODBC Excel Driver]Invalid connection string attribute hdr
ERROR [01S00] [Microsoft][ODBC Excel Driver]Invalid connection string attribute imex
ERROR [01S00] [Microsoft][ODBC Excel Driver]Invalid connection string attribute hdr
ERROR [01S00] [Microsoft][ODBC Excel Driver]Invalid connection string attribute imex
ERROR [01S00] [Microsoft][ODBC Excel Driver]Invalid connection string attribute hdr*

ERROR [01S00] [Microsoft][ODBC Excel Driver]Invalid connection string attribute imex

This only occurred with multiple endpoints referencing the same Business Process that contained the Excel to XML process step. Access Violations as well as stack overflow exceptions could have also occurred. This has been fixed.

MOD – Excel to XML properties – Users no longer need to select the version of Excel they need to work with.

MOD – Excel to XML to use OLEDB – The process step has been modified to use the OLE DB provider instead of the ODBC driver.

Neuron Test Client

MOD – Support for Binary files in Bulk Mode – When set to bulk mod, the Neuron ESB Test Client did not allow users to select Binary files to load for sending as the test. This has been enabled.

Topics

FIX – Rabbit MQ Topic does not create System Queue – This was introduced in build 585. When the Neuron ESB Rabbit MQ Publishing Service that represents a Topic started it would not create the Queue required for the internal System Party that's used when users connect a Client Connector or Adapter directly to Business Process. Because of this, during startup, the endpoint may fail with an error that it can't connect to its underlying queue because it doesn't exist. This has been corrected.

MOD – Internal System Party now supports Rabbit MQ sub topics – Previously, when we created the internal Queue for the System Party, routing rules for the root Topic were created, but not for sub Topics. This has been added so that any endpoint directly connected to a Business Process (i.e., not using Messaging) can now publish and receive messages on not just a root Topic, but any sub topic as well.

FIX – Rabbit MQ Topic could delete another Topic's Poison Handler Exchange – This was introduced when we introduced support for the Rabbit MQ Delayed Exchange for Poison Messages in build 583. In short, on startup, or during a configuration change event, Topic "A" could delete the Rabbit MQ Delayed Exchange for Poison Messages that was created for Topic "B" and vice a versa. This could result in an error when a Party attempted to send a message because it had a dependency on that exchange. This has been corrected.

Neuron 3.7.5.585

Monitoring

FIX – Rabbit MQ Topics - If errors or warnings occurred during startup of the topic, they would not be recorded in endpoint health. Only those occurring after the startup process would be recorded. Also, there some errors and warnings that occurred would only be reported in the log files and the Windows Event Log.

Topics

MOD – Rabbit MQ Poison Message Handling - Previously, if the topic was configured to use the NON blocking Poison Message Handling introduced in build 3.7.5.593 but encountered an error because either the delayed exchange plugin was not installed or enabled, the topic, as well as the party would report this as an error and fail to startup. Now this is reported as warning, both Topic and Party will start and the Poison Message Handler will be downgraded to the Blocking type.

MOD – Rabbit MQ Queue Management – On startup, the Neuron ESB Server automatically created all the Exchanges and Queues necessary to represent the Neuron ESB Topics. It also managed the changes to the Rabbit MQ infrastructure in cases where the Topic was either renamed or removed. Each Party, upon connecting, would also run similar code, creating all the necessary Exchanges and Queues to support its subscriptions. The Party was responsible for managing Queue changes at runtime, in cases where the Party was renamed or where Topics were added or removed from its subscription. This has been changed so that the Neuron ESB Server manages the Party modifications. This removes the need for each party to duplicate the infrastructure management that the Neuron ESB server does. That equates to faster connect times and fewer resources utilizing Rabbit MQ connections and operations.

FIX – Rabbit MQ Transport Change – If an existing Rabbit MQ Topic were changed to use the Rabbit MQ transport at runtime, any Party referencing the Topic would not automatically update to the new transport.

FIX – Rabbit MQ Rename or Add – If a Rabbit MQ Topic was either renamed or added to existing Party at runtime, Endpoint Health would reflect the change. However, any connected Parties would recognize the newly added or renamed Topic but would not automatically reconnect to it.

Neuron 3.7.5.583

Topics

FIX – Rabbit MQ – When using Topics configured to use the Rabbit MQ Transport, the Neuron ESB underlying channel regularly checks (every 5 seconds) to see if reconnection to the Rabbit MQ server is necessary. The logging for this function would log verbose messages that included the host name and port of the Rabbit MQ server its connecting to. These were not being properly set and would appear in the logs as empty strings. This did not affect any runtime behavior, only logging was impacted. This has been fixed.

FIX – Rabbit MQ - If a consumer shut down event occurred, but the connection was still good, the Neuron ESB's reconnection timer that runs every 5 seconds would not restore it. Now, even if the connection is open, we check the open status on the consumer channel, if its closed, we recreate the consumers so that messages will continue to be received.

NEW – Rabbit MQ – Added Lazy Queue support to the Transport Properties. Users can now configure a Topic to use Lazy Queues by selecting the option via the new “Queue Mode” property. This property compliments the “Delivery Mode” property that allows users to use either Persistent or Non Persistent Queues. Persistent messages will survive a broker restart, while Lazy means it will write out to disk ASAP

to keep memory pressure low, but without persistent messages will be lost on any restart of the Rabbit MQ server. Persistence and Lazy settings will dramatically decrease memory usage, but possibly increase CPU and Disk I/O.

NEW – Rabbit MQ – NON Blocking Poison Message Handler - Support has been added for Poison Messaging handling that is non-blocking. This feature specifically uses the <https://github.com/rabbitmq/rabbitmq-delayed-message-exchange> . When Poison Messaging is enabled in the transport properties of the topic, a new property, “Handler Type” will appear. If set to “NonBlocking”, this new feature will be used.

MOD – Rabbit MQ Property Defaults - Changed the default Delivery Mode to Persistent to reduce memory footprint. Changed the "Resubmit Unacknowledged Messages" property available if Publish Confirms is selected regardless of if the “Detect Duplicates” property is set to true. Changed the “Prefetch Size” property default from 50 to 20. Changed the default "Retry Cycle Delay" from 10 minutes to 1 minute.

Endpoint Health Monitoring

FIX – Rabbit MQ – Endpoints– When using Topics configured to use the Rabbit MQ Transport, if the underlying Rabbit MQ consumer threw a Shutdown exception via its Shutdown event, it would be logged, but the endpoint would still show started and green in Endpoint Health monitoring. This has been fixed so that an Offline event is also raised which will increment the error value in Endpoint Health as well as change the status to Failed. Once the consumer recovers and reconnects, the Online event will be raised, changing the status of the endpoint back to Started.

FIX – Rabbit MQ – Topics - When using Topics configured to use the Rabbit MQ Transport, if the Topic’s publishing/monitoring service encountered an error on startup, the Topic would still appear green and Started in Endpoint Health. Now the error is recorded, and it appears as Failed.

Business Processes

FIX – Dependency Checking – When checking the dependencies of a Business Process, the check could fail to recognize that other Business Processes were calling it. This has been corrected.

Adapters

FIX – File Adapter – When using Query mode, if the “Delete on Query” was set to false, the file may have its bytes set to zero on disk, effectively deleting the contents of the file, but not the file itself. This happened with some binary files such as PDF files.

FIX – SFTP Adapter – Archive and Delete modes may provide inconsistent results when in Query mode.

MOD - Salesforce adapter - removed support for all API versions before 31.0. Salesforce is disabling this in summer 2023. Added support for versions 52.0-57.0.

Neuron 3.7.5.580

Event Processor

FIX – Locking issue could cause EPS to stop processing messages – A conflicting locking issue was found that could interrupt or stop the processing of EPS and Neuron Log entries.

FIX – FTP Adapter – If set in Query mode and the “Enable Timestamp Comparison” feature was enabled, it would be ignored.

FIX – SFTP Adapter – If set in Query mode and the “Enable Timestamp Comparison” feature was enabled, it would be ignored.

Adapters

MOD – TCP Adapter - Added a "Fixed size" option for Server Socket Adapter. Customers can use this when receiving fixed-sized messages over TCP from 3rd party applications. The developer must know the size of the byte array to receive.

FIX – File Adapter – This only occurred with Query Mode. Invalid validation messages could appear referencing publish mode properties. Also, other validations for query mode have been modified.

FIX – File Adapter – This only occurred in Query Mode. At runtime, if a Query Mode File Adapter was called within a Business Process which was attached to a Client Connector or an Adapter configured in Publish Mode, the following error would be recorded:

“The given key was not present in the dictionary.”

Business Processes

MOD – Publish Step – During design time testing, if a Source ID is not provided in the Message test dialog, the System party will be used internally to publish messages. Previously, users would receive an error indicating the Source ID was not present resulting in the message not being published to the Topic.

MOD – Audit Step - During design time testing, if a Source ID is not provided in the Message test dialog, the System party will be used internally to audit messages. Previously, users would receive an error indicating the Source ID was not present resulting in the message not being audited. Also, if a Topic is not provided, “NONE” will be used. Lack of a Topic name also previously prevented the message from being Audited.

Neuron 3.7.5.573

Neuron ESB Explorer

FIX – Business Process Dependencies Inaccurate – When checking the dependencies of a Business Process, all Workflow Definitions that used an Execute Process Activity would be listed because we were not comparing the Business Process name with the activity’s process name setting. This has been fixed. This was a new feature introduced in build 3.7.5.568.

Adapters

FIX – Pop3 Adapter - Would throw the following error when trying to publish because the Topic was not being properly set. This bug was introduced in build 3.7.5.568

*2023-01-27 15:15:41.638-06:00 [49] ERROR - Publish to ESB failed.
System.NullReferenceException: Object reference not set to an instance of an object.
at Neuron.Esb.Party.SendMessage(ESBMessage message, Boolean duplicateMessage)*

FIX – Microsoft Exchange Adapter - Would throw the following error when trying to publish because the Topic was not being properly set. This bug was introduced in build 3.7.5.568

*2023-01-27 15:15:41.638-06:00 [49] ERROR - Publish to ESB failed.
System.NullReferenceException: Object reference not set to an instance of an object.
at Neuron.Esb.Party.SendMessage(ESBMessage message, Boolean duplicateMessage)*

FIX – NetSuite Adapter - fixed metadata generation issue with "Transaction" entities

MOD – File Adapter - When in query mode, users can use the "Delete on Query" property to prevent the deletion of the file after its retrieved. The default value is to delete the file. Additionally, users can dynamically set the Folder path and the file names to retrieve in a Set Property or C# step by using below:

*context.Data.SetProperty("file", "Path", "c:\\myfolder\\files");
context.Data.SetProperty("file", "FileNames", "myfile.txt;nextfile.doc");*

Topics

NEW – Rabbit MQ Poison Message Handling - Neuron ESB's Rabbit MQ Transport implements Poison Message handling for unhandled exceptions that are thrown from subscribing endpoints (i.e. hosted Parties, Workflow, Adapter and Service Endpoints) or Business Processes assigned to those endpoints. If enabled, then unhandled exceptions will not be immediately audited or logged as errors when they occur. Instead, they will be logged as an initial warning from Neuron ESB's Rabbit MQ client channel as displayed in the example below:

Poison Message: Failed to send Message received from 'PlayaPub' to 'PlayaSub' on Rabbit MQ Topic 'playa' with ESB Message ID '50a84be6-7c27-4179-aded-4d7936529798'. 2 attempts will be made to deliver the message before it is sent to the failed message database. Original Exception: On Receive Event - An unhandled exception occurred while executing 'throw exception' pipeline on step 'C#'. Message ID '50a84be6-7c27-4179-aded-4d7936529798', Topic 'playa', Source Party 'PlayaPub'. The code step "C#" failed due to an error: This is a new exception from a Business Process

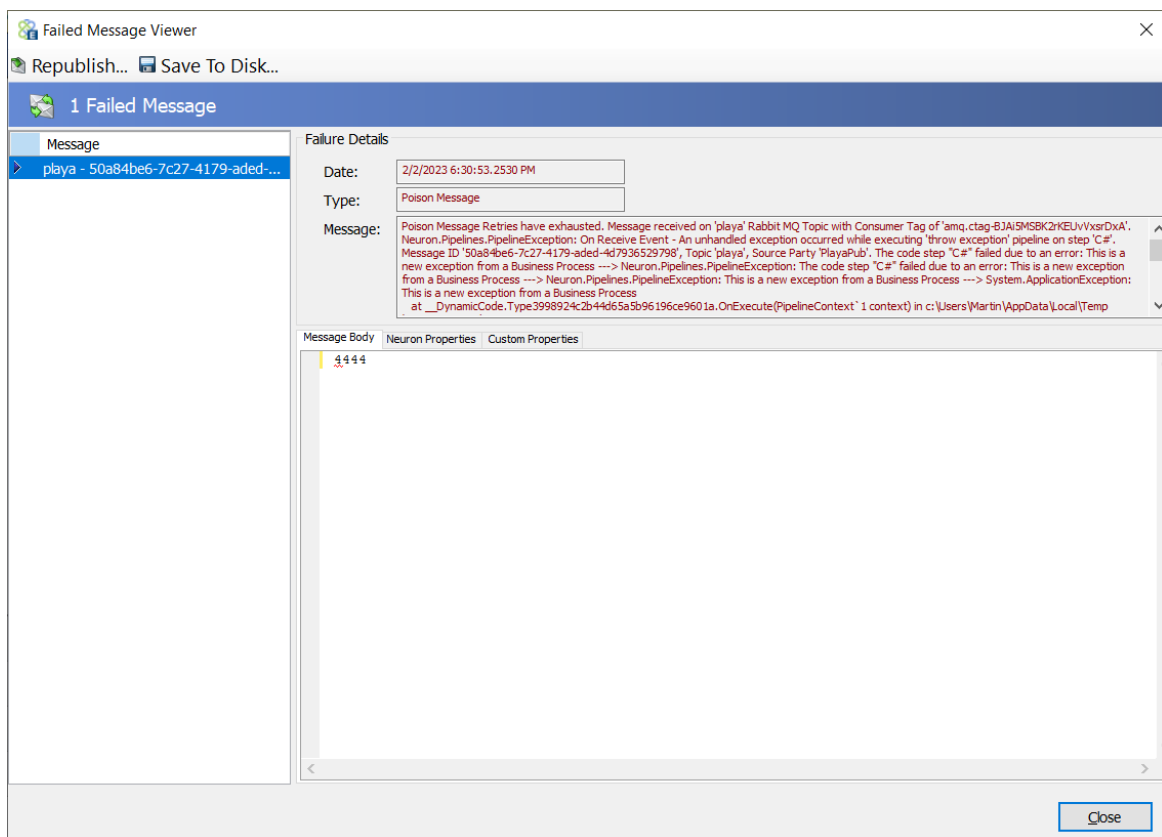
Once the exception is detected the channel will wait for the period specified in the "Retry Cycle Delay" property of the Rabbit MQ Topic Transport. Once the wait has ended, the message will be released once more to the subscribing endpoint. Each unhandled exception that occurs will increment the internal counter until the number of retries reaches the number specified in the "Max Retry Cycles" property of the Rabbit MQ Topic Transport. Each retry will log a warning message like the one below:

Poison Message: Retry 1 of 2. Message received from 'PlayaPub' on Rabbit MQ Topic 'playa' with Consumer Tag of 'amq.ctag-BJAi5MSBK2rKEUvVxsrDxA' and ESB Message ID '50a84be6-7c27-4179-aded-4d7936529798'. On Receive Event - An unhandled exception occurred while executing 'throw exception' pipeline on step 'C#'. Message ID '50a84be6-7c27-4179-aded-4d7936529798', Topic 'playa', Source Party 'PlayaPub'. The code step "C#" failed due to an error: This is a new exception from a Business Process

Once all the retries are exhausted, the original message with the exception information will be forwarded to the Failed Message Audit Report and be removed from the subscribing queue. This will result in an error message being logged like the one below:

A message has been saved to the audit failed database. Topic: playa, Party: PlayaPub, Id: 50a84be6-7c27-4179-aded-4d7936529798
FailureDetail - Poison Message Retries have exhausted. Message received on 'playa' Rabbit MQ Topic with Consumer Tag of 'amq.ctag-BJAi5MSBK2rKEUvVxsrDxA'. Neuron.Pipelines.PipelineException: On Receive Event - An unhandled exception occurred while executing 'throw exception' pipeline on step 'C#'. Message ID '50a84be6-7c27-4179-aded-4d7936529798', Topic 'playa', Source Party 'PlayaPub'. The code step "C#" failed due to an error: This is a new exception from a Business Process

The Poison Message can be viewed in the Failed Message Viewer as illustrated below:



Neuron ESB Test Client

FIX – Message Rate continues to calculate after Bulk Test is over - The Test Client status bar shows message rate during a bulk test. However, even after the test has finished and all the messages have

been received, the message rate never stopped calculating. Now the message rate will stop calculating once the bulk test is complete so that the final number will always be visible.

Neuron 3.7.5.568

Licensing

FIX – Daily Validation Checks generate infinite API calls – This is a serious bug that's been fixed. Every 24 hours a running instance of Neuron ESB will make an API call to the Peregrine License Server. If an exception was thrown anytime in the process, the failure would not be adequately caught. This would result in the timer not being reset but instead immediately loop back and make the API call again. This would result in the Master log file being filled with entries like below if the logging level was set to either Verbose or Informational:

```
INFO - Neuron ESB license key verification process started for machine 'MACHINE1'.  
INFO - Loading License Details - Begin  
INFO - Loading License Details - Reading License File  
INFO - Loading License Details - Loading Data from Licensing File  
INFO - Neuron ESB license key verification process started for machine ' MACHINE1".  
INFO - Loading License Details - Begin  
INFO - Loading License Details - Reading License File  
INFO - Loading License Details - Loading Data from Licensing File  
INFO - Neuron ESB license key verification process started for machine ' MACHINE1".  
INFO - Loading License Details - Begin  
INFO - Loading License Details - Reading License File  
INFO - Loading License Details - Loading Data from Licensing File  
INFO - Neuron ESB license key verification process started for machine ' MACHINE1".
```

FIX – Neuron ESB runtime would fail to start – This would happen if the Neuron ESB Service was configured with a solution and started without EVER opening the Neuron ESB Explorer or using the Neuron ESB Explorer to configure the service. This was a bug introduced in build 548 of 3.7.5. Users would not encounter this error if the Neuron ESB Explorer was used to configure the Neuron ESB Service runtime. If the error occurred, the following would be written to the Master log file and reported in the Windows Event Log:

```
INFO - License File Information: START  
ERROR - Server cannot start up - System.NullReferenceException: Object reference not set to an  
instance of an object.  
at Neuron.Esb.ESBMasterService.ValidateOnServer()  
at Neuron.Esb.ESBMasterService.Start(String instanceName, String environment, String zone,  
Exception& exception, String[] args)
```

FIX – Starting Neuron ESB throws Argument Null Exception – If not using an offline license file and there was no internet access when either the Neuron ESB Explorer was launched or the Neuron ESB Service was started, the following error could be thrown preventing startup:

System.ArgumentNullException: String reference not set to an instance of a String. Parameter Name: s

Adapters

MOD – POP3 and Microsoft Exchange Adapters – This has been modified to change the way in which the retrieved email and its associated attachments were persisted. Previously, they were serialized as bytes and set as the message body of the ESBMessage. Users would need to serialize out the bytes into the defined Neuron.Esb.Mail.EmailMessage type to work with it and retrieve the attachments like so:

```
var email = Neuron.Esb.Mail.EmailMessage.Deserialize(context.Data.Bytes);
context.Data.Body = email.Body;
context.Properties["EmailAttachments"] = email.Attachments;
```

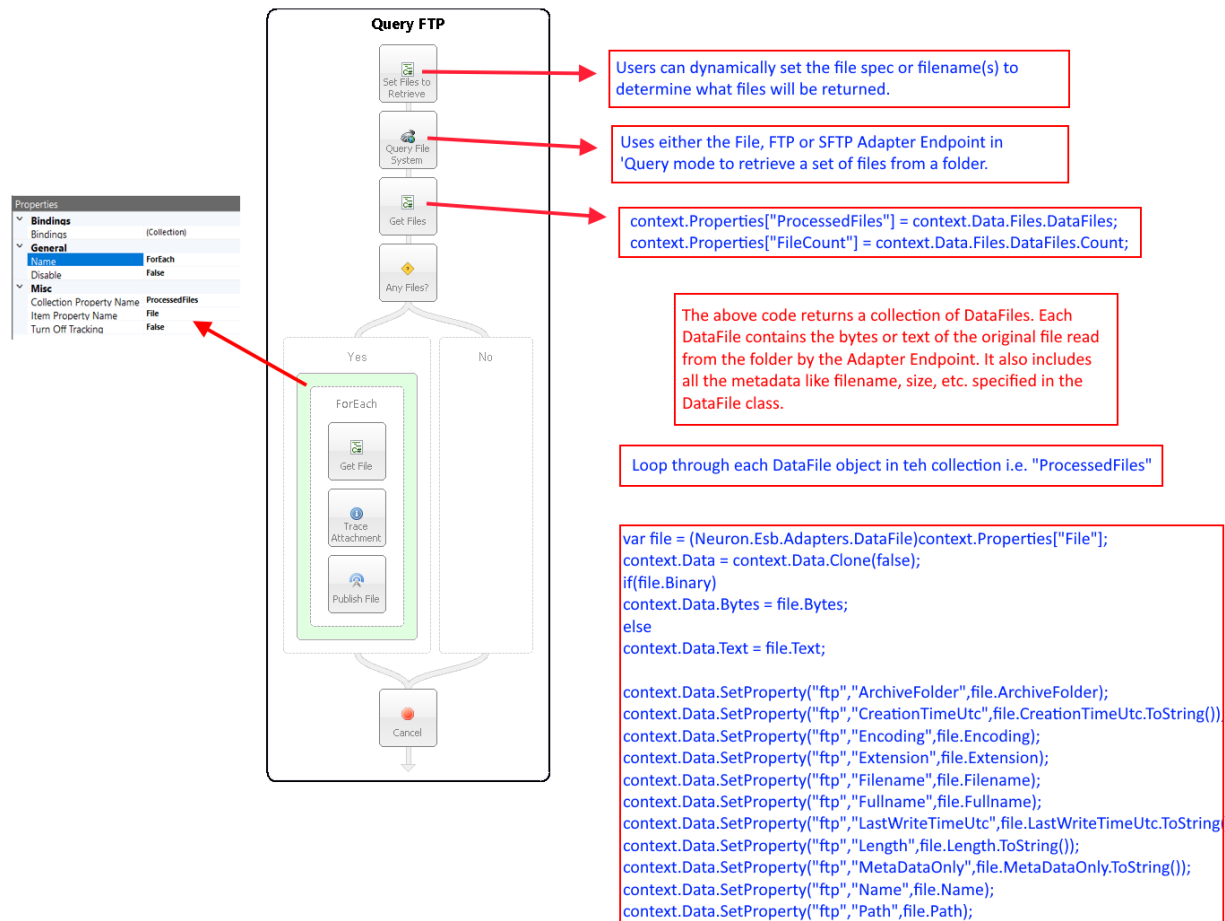
Now, the Neuron.Esb.Mail.EmailMessage is a property of the ESBMessage object named “Email”. The email and associated attachments can now be retrieved like so:

```
context.Properties["EmailAttachments"] = context.Data.Email.Attachments;
```

MOD – FTP Adapter – Added Query mode. This allows users to create an Adapter Endpoint that can be called via Messaging or directly through a Business Process which returns one or more files read from an FTP/S folder and matching either the filename or file extension specifications configured for the adapter endpoint. The Query mode supports all the features that users find on Publish mode. Users can use the FileSpec property to determine the set of files to retrieve. The new Files property of the ESB Message returned from the Adapter Endpoint is of type *Neuron.Esb.Adapters.Files*. This will contain a collection of *Neuron.Esb.Adapters.DataFile* objects. Each DataFile class is defined as:

```
public class DataFile
{
    public DateTime CreationTimeUtc { get; set; }
    public DateTime LastWriteTimeUtc { get; set; }
    public string Encoding { get; set; }
    public string ArchiveFolder { get; set; }
    public Byte[] Bytes { get; set; }
    public string Text { get; set; }
    public long Length { get; set; }
    public string Name { get; set; }
    public string Path { get; set; }
    public bool ReadOnly { get; set; }
    public bool Binary { get; set; }
    public string Extension { get; set; }
    public string Fullname { get; set; }
    public string Filename { get; set; }
    public bool MetaDataOnly { get; set; }
}
```

Using a Business Process, The DataFile objects can be iterated through using a For Each process Step. Below is an updated graphic demonstrating a simple process that iterates through each DataFile, publishing each to a Topic. This can be applied to either the File, FTP/S or SFTP adapters.



MOD – SFTP Adapter – Added Query mode. This allows users to create an Adapter Endpoint that can be called via Messaging or directly through a Business Process which returns one or more files read from an SFTP folder and matching either the filename or file extension specifications configured for the adapter endpoint. The Query mode supports all the features that users find on Publish mode. Users can use the FileSpec property to determine the set of files to retrieve. The new Files property of the ESB Message returned from the Adapter Endpoint is of type *Neuron.Esb.Adapters.Files*. This will contain a collection of *Neuron.Esb.Adapters.DataFile* objects. Each DataFile class is defined as:

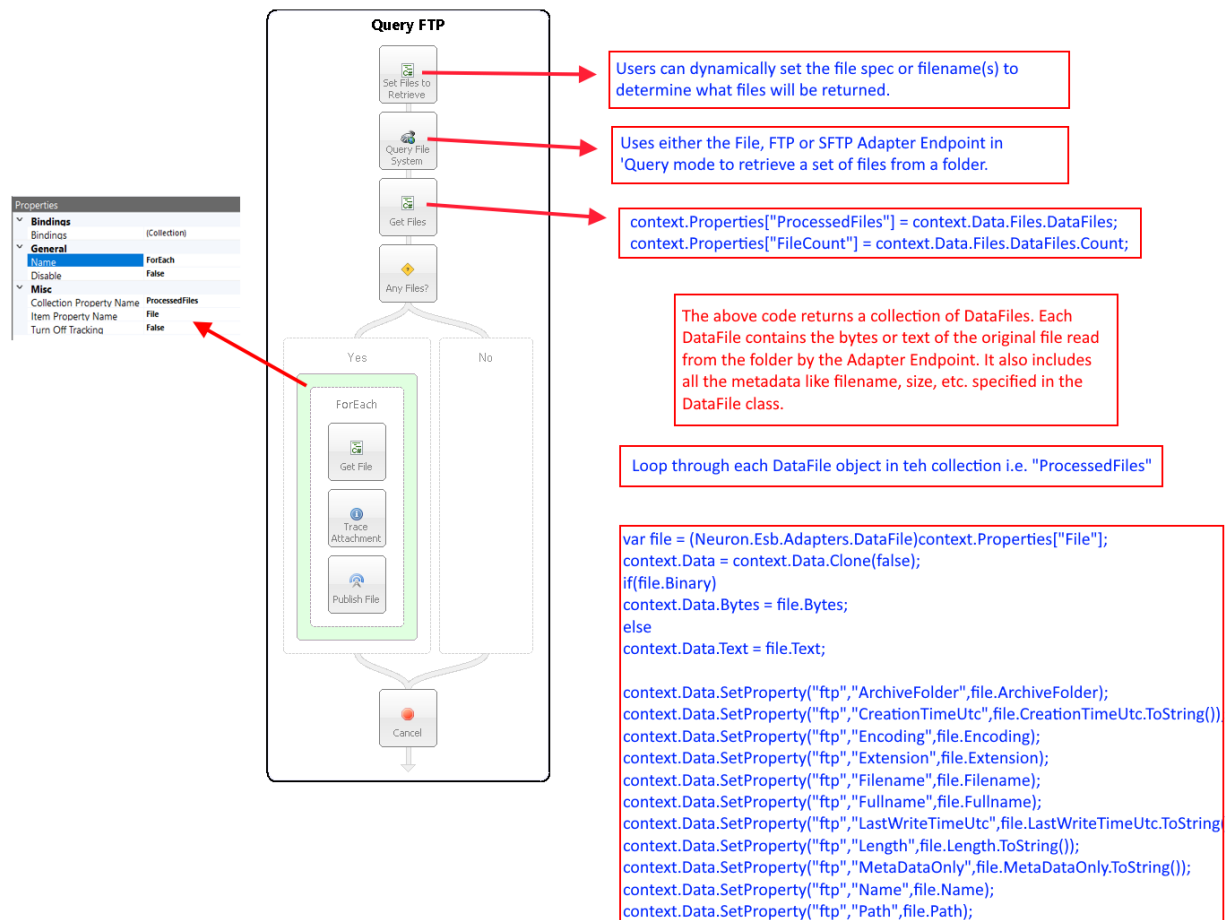
```
public class DataFile
{
    public DateTime CreationTimeUtc { get; set; }
    public DateTime LastWriteTimeUtc { get; set; }
    public string Encoding { get; set; }
    public string ArchiveFolder { get; set; }
    public Byte[] Bytes { get; set; }
    public string Text { get; set; }
    public long Length { get; set; }
    public string Name { get; set; }
    public string Path { get; set; }
    public bool ReadOnly { get; set; }
    public bool Binary { get; set; }
    public string Extension { get; set; }
    public string Fullname { get; set; }
}
```

```

public string Filename { get; set; }
public bool MetaDataOnly { get; set; }
}

```

Using a Business Process, The DataFile objects can be iterated through using a For Each process Step. Below is an updated graphic demonstrating a simple process that iterates through each DataFile, publishing each to a Topic. This can be applied to either the File, FTP/S or SFTP adapters.



MOD – File Adapter Query Mode – The existing Query Mode property has been modified for the File Adapter. This allows users to create an Adapter Endpoint that can be called via Messaging or directly through a Business Process which returns one or more files read from the folder and matching either the filename or file extension specifications configured for the adapter endpoint. The Query mode supports all the features that users find on Publish mode (when the File Detection property is set to Polling). Users can use either the Filename or File Type property to determine the set of files to retrieve. The new Files property of the ESB Message returned from the Adapter Endpoint is of type *Neuron.Esb.Adapters.Files*. This will contain a collection of *Neuron.Esb.Adapters.DataFile* objects. Each DataFile class is defined as:

```

public class DataFile
{

```

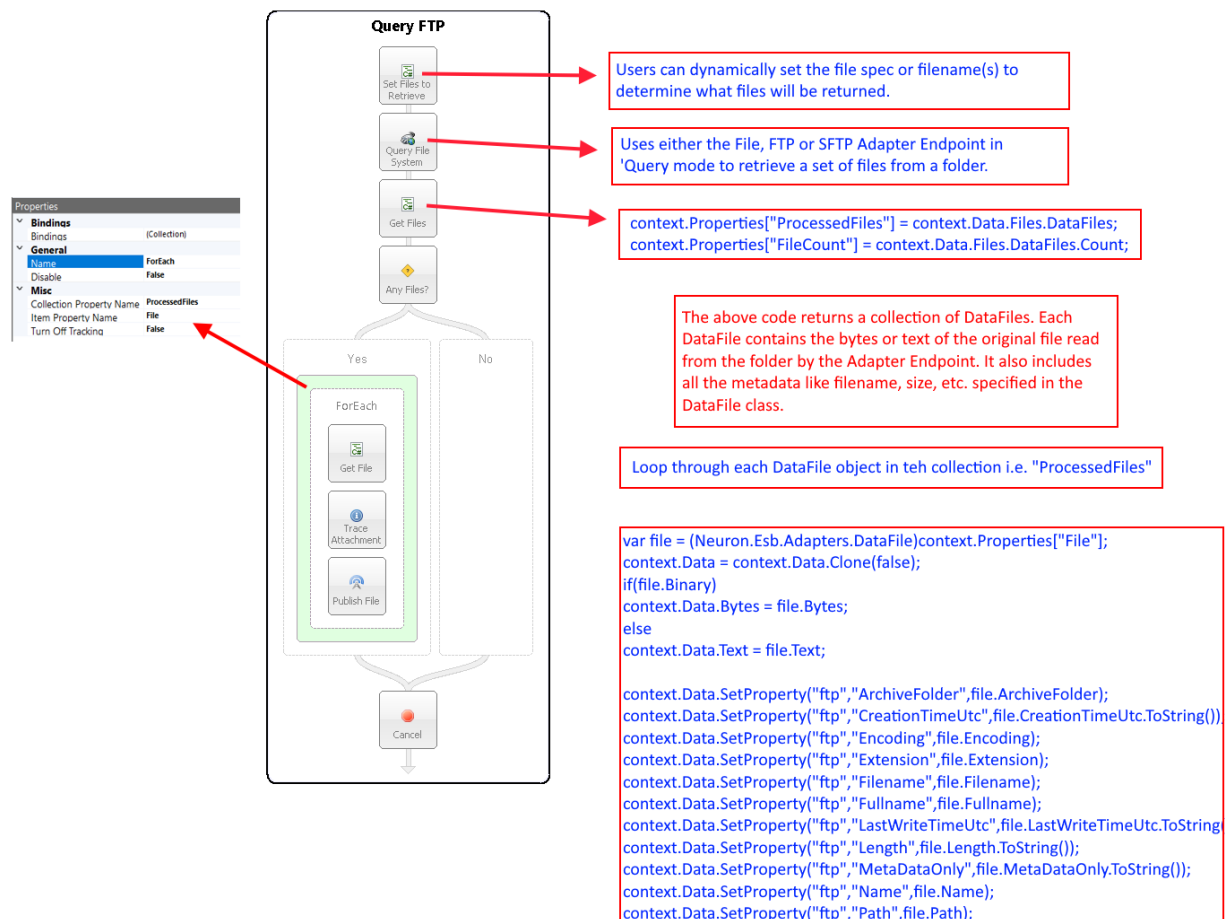
```

public DateTime CreationTimeUtc { get; set; }
public DateTime LastWriteTimeUtc { get; set; }
public string Encoding { get; set; }
public string ArchiveFolder { get; set; }
public Byte[] Bytes { get; set; }
public string Text { get; set; }
public long Length { get; set; }
public string Name { get; set; }
public string Path { get; set; }
public bool ReadOnly { get; set; }
public bool Binary { get; set; }
public string Extension { get; set; }
public string Fullname { get; set; }
public string Filename { get; set; }
public bool MetaDataOnly { get; set; }
}

```

Previously, users had to serialize this collection from the message body into a context property object. Now the File object has been added to the ESBSMessage object so no serialization is required.

Using a Business Process, The DataFile objects can be iterated through using a For Each process Step. Below is an updated graphic demonstrating a simple process that iterates through each DataFile, publishing each to a Topic. This can be applied to either the File, FTP/S or SFTP adapters.



FIX - REST Adapter - fixed Keep Alives support and made most of the configuration properties bindable so that they can be used with Environment Variables.

FIX – Active Directory Adapter - If in publish mode and there was a server unavailable error, the existing connection and notification event handler were not rebuilt if the user had poll on errors set to continue. The system would show that the adapter endpoint was still running and receiving events, even though no more events would be received.

Neuron ESB Runtime

MOD – Disabling dynamic port assignment for Neuron Endpoint Host – On startup, Neuron ESB currently will assign a configured port (default is 51004 specified in the appsettings.config) to the Neuron Endpoint Host service. Before assigning it, it will check if the port is available and, if not, dynamically choose a new port to use for the service. Users can now disable this dynamic assignment and force Neuron ESB to use the statically configured port. To control this we added the “EnableUpdateNeuronEndpointHostPort” flag to the appsettings.config. The default value is “true”, preserving the existing behavior. To disable the dynamic assignment, change the setting to “false”.

Samples

MOD - Updated CorrelatedWorkFlow sample by removing the Orders.In subtopic.

Services

FIX – Starting a Client Connector could fail – if the Wiretap configuration in the appsettings.config file was mis configured, Client Connectors would fail to start, throwing an instance of an object exception.

Repository

FIX – Dependencies tab in Repository throws error - If user created a Data Mapper repository object in Neuron Explorer but didn't create any actual Data Map associated with it, the dependency tabs for any Schemas and Documents would generate an object reference not set to instance of an object error in Neuron ESB Explorer.

Neuron 3.7.5.555

Installation

FIX – FIPS Error during Installation – Modified our installer (i.e., EXE) and MSI so that FIPS enforcement registry key does not affect the installation process. Previously installation would fail with little notification. When using the Bootstrapper installer, we will automatically handle the FIPS issue and let installation succeed. When using the MSI, it will error out with a specific message is using the UI installation, otherwise error message will occur in the log.

Licensing

FIX – Licensing support for offline files and various fixes – our Licensing library has been updated to version 16.1.22319.1. We are now providing more complete support for conditions where there is no network present as well as offline license files. Also, the NeuronQlmInfo.json online license file will be automatically deleted when either patching or deactivating a license.

Business Processes

MOD – ID property – The ID property of Business Process steps is now a hidden property.

New – OAuth Process Step. The OAuth Process Step provides the ability for a developer to retrieve an OAuth token within a business process. This gives developers the ability to use our existing OAuth providers to retrieve tokens that can be used within a business process. A more complete description and directions on how to use it can be found in the documentation:

https://www.peregrineconnect.com/documentation-kb/process-step-reference-guide/#ch_38

FIX – Execute Process Step - The Administrator object used when getting the configuration in the Execute Process step was not being disposed after we were done with it. This could eventually cause timeouts to the underlying Configuration Service.

Adapters

MOD – File Adapter’s Topic Property - When running a file adapter in publish mode, connected directly to a Business Process, users were required to populate the Topic adapter property. this is no longer the case.

MOD - Rest Adapter - Added AllowAutoRedirect property. Developers can set this to false if they don't want the adapter to automatically follow redirect responses, instead opting to handle the redirect in a business process.

Runtime

FIX – Database Required on Startup - If a solution was configured to run that did not have a database associated with the configured deployment group, the following warning would be written to the master log file twice when attempting to start the endpoint host:

2022-11-17 14:56:53.574-08:00 [4] WARN - System.Data.SqlTypes.SqlNullValueException: ConnectionString is Not Initialized

Neuron 3.7.5.543

Event Processor

FIX – Failed to restart if Rabbit MQ error occurred – The Event Processor would stop processing events in the queue if there was a Rabbit MQ specific error. This has been corrected.

Messaging

MOD – Rabbit MQ – The Event Processor and Rabbit MQ Topics have been updated to use Rabbit MQ Client API version 6.4.0. The previous version was 6.2.1.

Adapters

NEW - NetSuite Adapter - Added support for WSDL versions 2021.2 and 2022.1

NEW - NetSuite Adapter - added support for version 2022.1 of the SOAP API

MOD - NetSuite Adapter - Added the reply message property "netsuite.AccountSpecificUrl". After a call to NetSuite, this property on the response message will contain the account-specific URL used when making the call to NetSuite. An example of where this property could be used - when adding a sales order to NetSuite, you need to create a send a URL to that sales order to a different application (i.e., SharePoint).

MOD - NetSuite Adapter - Added support for the "Preferences" header. This header contains per-message overrides for these company-wide settings:

- disableMandatoryCustomFieldValidation
- disableSystemNotesForCustomFields
- ignoreReadOnlyFields
- runServerSuiteScriptAndTriggerWorkflows
- warningAsError

These are only exposed as message properties, as they are overrides to company-wide settings. Developers should only be using these properties when they need them.

FIX - Metadata Retrieval - the wizard wasn't properly checking for existing JSON Schemas, JSON Documents or Swagger documents before adding those generated.

Business Processes

NEW - Lookup Process Step – A new process step has been added that provides the ability to perform simple value lookups from a database table. The table has three fields – Category, Id and Value. The developer provides the Category and Id, with the value being returned. The Lookup Process Step also provides the ability to add, update and delete key-value pairs, giving you the option of building the table over time as messages are processed.

NEW - Lock Process Step – A new process step has been added that will only allow one instance of the same process to execute the contained block of process steps at a time.

Workflow

FIX - The PersistentDelay Activity – This would throw a null reference error upon using the activity in the designer.

FIX - The Timeout Activity – This could not be tested properly at design time especially if it contained a persistent delay activity. This would cause the design time test to be unable to resume the bookmark which would cause the workflow to abort when attempting to resume the bookmark.

Data Mapper

FIX – Padleft and Padright transformations - Padleft/PadRight have been modified to consider the input as total length of field instead of always padding given fix input characters.

FIX – Sequence did not match target schema - Updated output of the map so that it should display an element sequence that matches the sequence in the target schema.

Monitoring

FIX - REST API throws Error = The API (/neuronesb/api/v1/runtime) would throw an error in the case if a solution loaded to ESB service instance had other than .XML file(s) existing in DeploymentGroups or Zones folders which are not valid xml.

Event Processor

MOD – Rabbit MQ – The Event Processor has been updated to use Rabbit MQ Client API version 6.4.0. The previous version was 6.2.1.

Runtime

FIX – Memory Leak with XSLTs - When loading Xslt Neuron Objects for the runtime's configuration. the test input and output messages would be loaded as well. This could cause a lot of memory consumption if the messages were large (i.e. a MB or so). Now we do not load the test messages in the runtime.

Neuron 3.7.5.519

Adapters

FIX - Salesforce Adapter - when generating metadata for outbound messages, XML samples were not being generated.

FIX - NetSuite Adapter - when generating metadata for certain entities (i.e. SalesOrder), when selecting the option to generate sample XML you would receive a "Root element is missing" error.

FIX - Salesforce Adapter - Fixed errors received when using the Metadata Generation Wizard to generate metadata for outbound messages.

FIX - NetSuite Adapter – Fixed errors received when using the Metadata Generation Wizard to generate metadata for different combinations of entities and operations.

Business Processes

FIX - Custom Process Steps - Fix for a dynamic compiling issue when adding custom process steps. The error received is similar to this:

error CS1519: Invalid token '+' in class, struct, or interface member declaration

Runtime

FIX – Neuron ESB Service could shut down with a License failure - Fixed an exception that occurs when the Peregrine Connect license server becomes unresponsive right in the middle of a server-side license validation call. By default, the Neuron ESB runtime performs a validation check of the user license each day. If the Neuron ESB Server cannot connect to the Peregrine License server for more than 15 days, the Neuron ESB service will shut down. However, this bug caused the Neuron ESB Service to believe it was offline for the maximum threshold of 15 days and hence would shut down. Now If this occurs, the validation process will return the expected result as if the server was offline.

Neuron ESB Explorer

FIX – Neuron ESB Explorer crashes when user attempts to open it – When opening the Neuron ESB Explorer, it attempts to create the default location for all Neuron Solutions. The full path for the default location is 'C:\Users\<username>\Documents\Neuron ESB 3'. If the Neuron ESB Explorer could not either access it or, if it doesn't exist, create it, it would shut down with the error below. Rather than shut down, we now inform the user that we cannot access the folder.

Exception Details

System.IO.FileNotFoundException: Could not find file
'C:\Users\<username>\Documents\Neuron ESB 3'.
File name: 'C:\Users\<username>\Documents\Neuron ESB 3'
at System.IO.__Error.WinIOError(Int32 errorCode, String maybeFullPath)
at System.IO.Directory.InternalCreateDirectory(String fullPath, String path, Object
dirSecurityObj, Boolean checkHost)
at System.IO.Directory.InternalCreateDirectoryHelper(String path, Boolean checkHost)
at Neuron.Explorer.Program.CreateSolutionDirectoryIfNotExist()
at Neuron.Explorer.Program.Run(String[] argument)
at Neuron.Explorer.Program.Main(String[] args)

Additional Information

Please review the error information. If you are unsure about how to resolve the error, please
email Neuron ESB Support (NeuronSupport@neudesic.com) and include the error details.

Services

FIX – Service Policy would attempt an additional Retry – If a user configured the Service Policy for zero retries, the runtime would still attempt to retry the service call. Any configuration of the policy retry greater than zero would also be internally incremented by 1. Also, the total number of retries reported in the logging would always be the number of configured retries -1. This has been corrected.

Neuron 3.7.5.510

Workflow

FIX – Correlated Messages appear Stuck - Correlated workflow messages that were queued in the database could become stuck in certain cases if the correlation id used was not unique to a workflow instance, or if the workflow instance completed without receiving the queued message (e.g. if a Receive Message activity was contained in a Timeout activity's body and the timeout elapsed before the message was received). Now, the queue is checked to see if it contained anymore messages that would correlate to the workflow and those messages are reprocessed.

FIX – Timeout Activity fails to work properly - The Timeout workflow activity did not properly execute or handle contained Receive Message activities resulting in hanging workflows if the Receive Message activity caused the workflow to unload.

FIX – Workflow would fail to resume on Failover - In many cases, if a Workflow Endpoint's Endpoint Host failed, Workflows would not properly resume upon failover resulting in stuck or duplicated workflows, orphaned workflow commands in the database, and possible message loss. Now, workflows should properly resume upon failover, workflow commands should only exist in the database for workflows not in a terminal state, and correlated messages that were in the database table should be reprocessed upon failover/startup of the workflow endpoint.

NOTE: If the Workflow Endpoint's topic is RabbitMQ, messages should never be lost, or duplicate processed even upon the Endpoint Host process getting terminated as long as publisher confirms or transactional is used with persistent queues. This is only the case with RabbitMQ topics as it is the only transport with the functionality to provide such behavior. In the case of MSMQ topics, messages should never be lost in a transactional queue, but duplicate message processing may occur upon failover since MSMQ does not provide a mechanism to determine if a message is a redelivery. Any other topic transports do not guarantee against message loss or duplicate processing upon failover (TCP, Named Pipes).

FIX – Correlation fails - Workflow correlated messages may not correlate properly or cause undesired workflow behavior (such as starting a workflow for a message that should have correlated to a running workflow) in the case that a non-unique, non-per-workflow-instance correlation id was used and the messages were processed in quick succession.

FIX – Operations on Workflow could fail - In the case that there are no concurrent workflow slots left, operations performed on workflows from Neuron Explorer's WF Tracking page would not execute until a slot became free. Now the operations are executed without regard to the max concurrent workflow setting unless the operation would start a workflow that is in a non-running state (i.e. unloaded or aborted)

FIX – Endpoint Health UI could freeze on Workflow Operations - Neuron Explorer UI would freeze/timeout if the workflow operation takes time to execute (e.g. canceling an instance locked WF from Workflow Tracking)

FIX – Messages fail to reprocess on Failover event - Messages can get stuck upon failover in the correlated workflow queue. They were not getting reprocessed upon failover due to the lockedByMachine element being null in the database.

FIX – Continual reprocessing of messages - There is a rare issue with workflow that can cause messages to be reprocessed until failure or restart of endpoint host. This issue can occur if an endpoint host failover causes a workflow to resume on a different machine, then the workflow issues a ResumeBookmarkCommand which is picked up by a different machine than the workflow failover resumed on, and then the correlation id is reused. Any messages with that correlation id that are picked up by the workflow host affected will not be processed properly and eventually wind up in the failed messages table.

FIX – Workflow command fails if IP Address is used - Endpoint health in Neuron Explorer would not execute the commands for the endpoint hosts if an IP Address is used instead of a machine name in the machines tab in the deployment group settings.

FIX – Failover fails with more than 1 Primary Host declared - Endpoint Host Failover would not occur properly in the case that there is more than one primary endpoint host resulting in endpoint hosts not failing over or starting up more instances than desired.

Runtime

Single Instance Endpoint Feature

NEW – Single Instance now supported with Endpoint Host - Single Instance endpoints (certain adapters in a specific mode and any service connector service endpoints) can now be ran in endpoint hosts. Previously, they had to be hosted in the ESBSERVICE process by choosing blank in the endpoint host drop down. Neuron will always try to keep a single instance of the endpoint running across all endpoint host instances in the case an endpoint host or the endpoint itself fails.

Endpoint Host Failover Behavior

Neuron will always try to concurrently run as many instances of an Endpoint host that are marked as primary across the server cluster (only 1 instance of a given endpoint host can be up per machine). Only one endpoint host instance should be marked as failover in a cluster/per endpoint host though many multiple primaries are supported, and message ingress will be load balanced across all active instances (including message correlation).

If an endpoint host misses a heartbeat by more than four times the client refresh interval, the failover instance will start up (this includes if the host was shut down manually from endpoint health) or, in the case the Endpoint Host's status is "FAILED", the failover may be detected and initiate much faster.

If an endpoint host instance fails or is manually stopped from Endpoint Health, it will not automatically start back up in the case another endpoint host instance goes down (i.e. fail back over) until it is manually restarted from endpoint health (restarting it this way will not start it up if there is already enough instances running...i.e. the number of instances marked as primary). This gives the user a chance to determine why it failed and to prevent failure loops/server resource consumption.

An endpoint host can be manually switched to another machine by stopping the endpoint host from endpoint health and waiting the four times client refresh interval timespan. Support for choosing which server to switch it to and not having to wait is planned for a future release.

For Workflows, during a failover of the Endpoint Host any workflows that were still running on the failed host will most likely go through the "Instance Locked" state. By default, this is 1066 seconds (approx.) in which the workflow is occupying a concurrent slot and will prevent any correlated workflows from starting that would match that instance's correlation id (same for processing messages which will begin to reprocess upon the time passing). The time that it will remain locked can be controlled by modifying the transientFaultHandling element in the NeuronEndpointHost.exe.config file. This should not be set

below the maximum time the SQL Server may be unavailable for in the case of transient and intermittent issues.

Some examples are:

Incremental Strategy which retries immediately on first failure, with an initial retry interval of 5 seconds, and with a 2 second delay between retries added on after each failure, that retries a max of 5 times. This will lock workflows for 160 seconds: (2 + 7 + 9 + 11 + 16 seconds due to firstFastRetry being true + the total retry time and timeout from the actual database connection settings which is set to 75 total seconds in this example) + a third of the sum in parenthesis as a buffer:

```
<neuron.transientFaultHandling defaultRetryStrategy="incremental">
  <incremental name="incremental" firstFastRetry="true" retryIncrement="00:00:02"
    initialInterval="00:00:05" maxRetryCount="5"/>
</neuron.transientFaultHandling>
```

Fixed Interval Strategy which retries immediately on first failure, with a fixed retry interval of 5 seconds, that retries a max of 5 times:

```
<neuron.transientFaultHandling defaultRetryStrategy="fixedInterval">
  <fixedInterval name="fixedInterval" firstFastRetry="true" retryInterval="00:00:05"
    maxRetryCount="5"/>
</neuron.transientFaultHandling>
```

Exponential Backoff Strategy which retries immediately on first failure, with an initial retry interval of 5 seconds that increases on each retry to a max of 1 minute, with the increase added to the minBackoff and being calculated by $((2^{\text{RetryCount}}) - 1) * (\text{Random Number Between } .8 * \text{deltaBackoff and } 1.2 * \text{deltaBackoff})$, that retries a max of 5 times:

```
<neuron.transientFaultHandling defaultRetryStrategy="exponentialBackoff">
  <exponentialBackoff name="exponentialBackoff" firstFastRetry="true" minBackoff="00:00:05"
    maxBackoff="00:01:00" deltaBackoff="00:00:10" maxRetryCount="5"/>
</neuron.transientFaultHandling>
```

Adapters

FIX - Dynamics CRM Web API Adapter - fixed issue when using windows credentials against an on-premises instance - users would receive "AdapterRestConnector - Reply message: HTTP Error 401 - Unauthorized: Access is denied" error.

FIX - Salesforce Adapter - fixed issue when the request message included SOAP headers. When the headers are not using the correct namespace, the adapter removes them because Salesforce will reject them. However, when attempting to remove the header the adapter would throw an exception.

Neuron 3.7.5.502

Adapters

FIX – Metadata generation for NetSuite produced undefined elements in Schema – This only affected the entity "Opportunity". The correct Search metadata was not being added resulting in undefined elements in the schema.

FIX – MQSeries Adapter – A property exception would occur when selecting the Queues dropdown property in the adapter endpoint property grid.

NEW – Active Directory Adapter new properties – Additional account information properties have been added to Publish mode and Query mode. In Publish mode, these will appear as 3 custom message properties, i.e. AD.AccountStatus, AD.AccountLockout and AD.PasswordExpired. In Query mode, properties will appear as sub elements within the new "AccountStatus_Attributes" element located under the Entity element of the returned query message. A sample appears below:

```
<AccountStatus_Attributes>
  <Status>ENABLED</Status>
  <Lockout>False</Lockout>
  <PasswordExpired>False</PasswordExpired>
</AccountStatus_Attributes>
```

Business Processes

FIX – Child Process error would not propagate to parent Process - If a child Business Process (one called via the Execute Process Step) threw an exception, it would NOT be propagated to the calling Business Process so that it could be captured and reported there.

FIX – GZIP not working with Compression Process Step – If set to use GZIP, the Compression Process Step would throw a null reference exception when trying to unzip a message.

Messaging

FIX – Rabbit MQ Topic would fail to start when using SSL – Users would receive an error indicating the following: A reference to the RabbitMQ Vhost '/' could not be obtained. This has been fixed

Neuron ESB Explorer

FIX – Rabbit MQ Queue Management would fail when using SSL – User would receive the following error on the top of the Queue Management screen: An error occurred connecting to RabbitMQ Management Port, '15671', on 'machine name'. The underlying connection was closed. This has been fixed.

Runtime

FIX – Heartbeat throws SqlNullValueException – The Checkheartbeat function within the master service will throw an error like the one below if the PrimaryHostName field returns null. This can happen due to old data being left over in the EndpointStatus table from a previous version of Neuron.

An exception occurred while monitoring the farm endpoints. Data is Null. This method or property cannot be called on Null values.

System.Data.SqlTypes.SqlNullValueException: Data is Null. This method or property cannot be called on Null values.

at System.Data.SqlClient.SqlBuffer.get_String()

Installation

FIX – Patch PowerShell Script throws hash error – If a user ran the PowerShell script that shipped with the 3.7.5.489 release to update an existing instance of 3.7.5, the error below would be encountered. This was due to an invalid signature on the patch.

```
PS D:\Install\NeuronESB_3.7.5> D:\Install\NeuronESB_3.7.5\InstallPatch.ps1
File D:\Install\NeuronESB_3.7.5\InstallPatch.ps1 cannot be loaded. The contents of file D:\Install\NeuronESB_3.7.5\InstallPatch.ps1 might have been changed by an unauthorized user or process, because the hash of the file does not match the hash stored in the digital signature. The script cannot run on the specified system. For more information, run Get-Help about_Signing..
+ CategoryInfo          : SecurityError (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : UnauthorizedAccess

PS D:\Install\NeuronESB_3.7.5> |
```

Neuron 3.7.5.489

Messaging

FIX – Message Loss - If messages are being sent to a party and the ESB Service, Endpoint Host, or Endpoint hosting the party is shut down gracefully, messages may be lost because they may still be pulled off the underlying topic during shutdown (e.g., may be consumed off a RabbitMQ topic queue) but the handler that processes the message may do nothing with it.

FIX – Double Message Delivery - In the case that Party.Pause() is called to disconnect from the Topic and then Party.Resume() is called later, messages can be double processed due to the handler being attached more than once.

Auditing

FIX – Auditing may throw Null Exception - If a message needed to be audited while the Party was disconnecting from a topic, a null reference error may be thrown due to the audit proxy being closed before the channel was closed.

Services

FIX – OAuth Token becomes invalid if Security Exception is thrown - Service Connectors did not refresh OAuth tokens when a security exception returned from a called service – This has been changed so now if a Security Exception does occur, Neuron ESB will automatically refresh the OAuth Token and retry the service call.

FIX – Failed to recognize the “expires in” OAuth Token value - Fixed issue with Azure Active Directory/ADFS and Thinktechure OAuth providers not storing the access token “expires in” value which is used when determining whether to refresh the access token at runtime.

Business Processes

NEW- Added Thread Safety to Custom Business Process Steps - Added "RequireThreadSafety" attribute to PipelineStep<T> with a default value of false. When set to true, it forces a new instance of the process

step class to be created with each process instance execution. This comes at a 10-15% performance hit. This solves the issue where custom process steps have private variables whose values change during execution of the process step. Set to false (which is the existing default behavior), when a process executes, each process step is instantiated once and reused (this is per process/per "host"). Under heavy load, if a class variable can be changed during execution, it could affect other instances of the process steps. Setting the new "RequireThreadSafety" attribute to true prevents this.

Samples

FIX - OAuth Azure sample - Removed Neudesic Tenant Id from Azure sample and changed the prefix of the client connector id to "api://" from "**Error! Hyperlink reference not valid.**" as that is new default for Azure AD application URI. Added link to more identity server consumer examples.

Adapters

FIX – Socket Adapter could throw Errors and Warnings – The Socket Adapter could throw warnings and errors anytime (roughly a minute apart). These could look like the following:

"The Client socket faulted but successfully reconnected: error code TimedOut"

and/or

"A call to Read to a server socket returned 0 while reading the message header. The client socket adapter still appears connected, but two more attempts to read returned 0. Attempting to reconnect."

and/or:

"The message cannot be sent to the TCP socket because the socket has been disposed."

Installation

FIX – Rabbit MQ and Erlang would fail to install – If a user chose to install Rabbit MQ or Erlang, the installation would fail as the URLs used to download the packages had changed. The URLs have been updated in the installer. The correct URLs did already exist in the existing Readme.html file.

Memory Leaks

FIX – Workflow Memory Leak due to replication of configuration – We found instances where the Neuron ESB Workflow Service would unnecessarily replicate the existing configuration in memory if a user saved the solution that the runtime was using.

FIX – The memory of the Operation Service hosting the REST API would grow on each API call - The NeuronOperationService process would consume memory and not release it for the lifetime of the process when API calls were made against it. This would occur when using any of the APIs available, such as retrieving endpoint health metrics.

NOTE: In certain cases where the Operation Service needs to load the configuration, such as calls to the Client Connectors Configuration API, the memory will jump due to the configuration being loaded into

memory. Any subsequent calls should not increase memory usage and should be released upon CLR Garbage Collection.

KNOWN ISSUE – Memory footprint of Client Context will double first time a solution is saved for a running instance – The Client Context within a Party is reinitialized every time a solution is saved in a running instance. During this process the Client Context still holds on to a copy of the Configuration, so upon saving a running configuration for the first time, the memory footprint will double for the configuration object the first time but remain the same on subsequent saves.

Startup

FIX – Neuron ESB fails to start with Machine name not found error – The Machine Name may not get properly resolved at startup. This has been resolved.

Data Mapper

The following fixes have been implemented in the Data Mapper

- 1) Disable standard spellcheck error for expression text input.
- 2) Corrected display for Source and target document type in auto mapper dialog.
- 3) Load optional 'End Index' value for 'Sub String' action(s) as empty if it's not specified, instead of incorrectly setting it to zero while opening map again.
- 4) Changed default action as 'Copy To' with index zero for 'One to Many' type mapping, instead of previous 'Split' default action.

Utilities and Tools

The following fixes have been implemented in the Tracepoint Utility:

- 1) Conditions were not getting reloaded when editing.
- 2) Overwrite or new config option when saving
- 3) Extension Assembly was not getting set when editing
- 4) Extension Assembly original path is stored in the instead of just the dll name. but at run time the DLL name is used to locate it under WiretapExtensions folder of the Event Processor Home directory.
- 5) Fixed issue in locating the extension type from the specified assembly
- 6) The body XPATH and truncate body processing is now applied (if specified) even after extension processing.

Neuron 3.7.5.472

Adapters

NEW – Added Refinitiv Web Socket adapter

NEW – Drop Box Adapter – Query mode has been added

NEW - Box Adapter – Query mode has been added

Business Processes

FIX – Process Step name could be incorrect when testing - When testing a process step with synchronous threads (i.e., Parallel step), the step name in the output window was incorrect.

Services

FIX – Could not Load File error when using Service Route - When creating a Service Routing Table, when clicking on "Configure Endpoint" in Neuron Explorer, an exception would be thrown due to a missing dll: "Could not load file or assembly 'ActiproSoftware.Editors.Wpf..."

NEW - Added a NetSuite OAuth Provider

NEW – Added Refinitiv OAuth Provider

FIX – Azure/Thinktechure OAuth Providers not storing “expires in” header - Fixed issue with Azure Active Directory/ADFS and Thinktechure OAuth providers not storing the access token "expires in" value which is used when determining whether or not to refresh the access token at runtime. Also modified the runtime to force a refresh of a token when a security exception returned from a called service

Messaging

FIX – Message Loss - If messages are being sent to a party and the ESB Service, Endpoint Host, or Endpoint hosting the party is shut down gracefully, messages may be lost because they may still be pulled off the underlying topic during shutdown (e.g. may be consumed off a RabbitMQ topic queue) but the handler that processes the message may do nothing with it.

FIX – Duplicate Message scenario - In the case that Party.Pause() is called to disconnect from the Topic and then Party.Resume() is called later, messages can be double processed due to the handler being attached more than once.

FIX – Null Reference could occur in Audit Proxy - If a message needed to be audited while the Party was disconnecting from the topic, a null reference error may be thrown due to the audit proxy being closed before the channel was closed.

Neuron 3.7.5.462

Neuron ESB Explorer

MOD – Samples – All samples have been updated

Neuron 3.7.5.452

Data Mapping

FIX - Namespace for XML collection element - When source xml/xsd document imported with collection type element having xml namespace prefix as ‘abc’ for tag “URI1” and at runtime actual xml message element has different prefix, for example, ‘pqr’ for same tag “URI1”. In such cases source values were not transferred or mapped to target field due to change in name space. This has been corrected to that

it finds the source element from message by matching actual tag “URL1” value to find correct element from source message and transfer its value as per mapping detail by ignoring namespace.

FIX - XML attribute mapping and handling namespace for Xml attribute - When mapped xml attribute element has xml namespace prefix, then data mapper was not transferring such attribute value due to prefix namespace is not validated correctly from actual source xml message. This has been corrected so that it finds the correct xml attribute element based on xml namespace in actual xml message so that its value get transferred to target field as per mapping

FIX - Added namespace for child element for xml document - When XML document with attribute is imported in data mapper for source/target, all XML attributes are not correctly mapped to element, so it was not possible to map attributes. This has been corrected so that it imports XML attribute correctly so that one can create correct map. This did work correctly though with XSD schemas.

Services

NEW - Service Routing Tables and XPATH– Added a new property helper that can be used when configuring an endpoint in a service routing table. The new property helper, called BodyXPath, can be used to lookup a value from an incoming XML body using an XPath expression. The syntax of the property helper is {#BodyXPath(<XPath>)}, where <XPath> would be replaced with a string containing an XPath expressing that will return the value to insert. The BodyXPath property helper can be used when setting the Local Path, header values and Url Parameter values.

NEW – Service Routing Tables can route to Adapter Endpoints – Adapter Endpoints that are configured for Subscribe or Query mode can now be added as configurable routable endpoints.

FIX – Condition Evaluation – The issue could appear when using conditions that use the expressions SOAP, HTTPQuery or HTTPHeader. If the incoming message does not contain any values of the selected expression, then the condition would return as true. For example, if you use the expression HTTPQuery:Name Equals (text) Neuron, and if the HTTP request does not contain any query parameters, the condition would have returned true.

Configuration

FIX – On Demand Loading of documents could fail in Adapter Endpoints - When adapters are started up, they are passed a lite configuration that was missing the configuration service address. This address was necessary for on-demand loading of items into the lite configuration like documents from the Document Repository.

Adapters

MOD – SMTP Adapter = Modified to utilize the configuration service when attempting to retrieve an XSLT.

Licensing

FIX – Trial License request would fail – If a user requested a trial license through the installer, it would fail with a message that no-reply@neuronesb.com exists. This has been fixed.

Workflow

FIX – Execute Process Activity logs to incorrect files - If a workflow definition used the Execute Process activity and the desired process to execute used the PipelineContext.Instance.Trace* methods to write to the Neuron logs, log entries could get written to the wrong logs and/or get double logged.

FIX – "For" workflow activity fails to increment – The “For” workflow Activity didn't execute the iteration or initialization properly causing it to be unusable. Now it works, but any existing workflow definitions that contain the broken "For" activity must first remove the activity from the definition using the version of Neuron that contains the broken "For" activity. Then Neuron can be updated and the new working "For" activity can be re-added.

Topics

FIX: Rabbit MQ fails to Audit message if Failure occurs - If a RabbitMQ topic was set to publish confirm transaction type and there are messages still active in the RabbitMQ channel during a disconnect, an exception would occur when the messages were being audited to failed messages and the messages would not be audited.

Neuron 3.7.5.433

Adapters

NEW – IBM MQSeries Adapter - added ability to cache the Queue or Topic on publish mode so we don't connect and disconnect on every poll.

NEW – Microsoft Exchange Adapter – This has been updated to support the optional use of OAuth using the Client Credential Grant Type of the Azure Active Directory OAuth Provider, specifically to support Office 365 in the cloud.

NEW – PublishESbMessage() in Subscribe Mode Adapters – We now support users calling this method in Subscribe mode custom adapters. Previously, if used, this would silently fail as it was designed originally for publish mode adapters.

FIX – Rabbit MQ Adapter Publish mode exception – When using version 3.9 or later of Rabbit MQ server and submitting a sample message using the Rabbit MQ management portal to a queue that the Neuron ESB Rabbit MQ adapter is monitoring in publish mode, the following error would be reported:

Neuron.Esb.Adapters.AdapterPublishException: An error occurred while publishing to the bus. RabbitMQ delivery tag of 1. ---> System.NullReferenceException: Object reference not set to an instance of an object.

at Neuron.Esb.Adapters.RabbitMQAdapter.a(BasicDeliverEventArgs A_0, List`1 A_1)

at Neuron.Esb.Adapters.RabbitMQAdapter.a(Object A_0, BasicDeliverEventArgs A_1)

This is due to a bug within the Rabbit MQ client API stack where the IsHeadersPresent() method no longer works. This has been compensated so that the message will publish successfully.

Messaging

FIX – Rabbit MQ Topics – If the Rabbit MQ Management Portal was configured for SSL, the Neuron ESB Topic would fail to connect. Now we first connect via HTTP, if that fails, we attempt an HTTPS connection.

MOD – Rabbit MQ Topics – If a Party subscribed to a Topic and it received a message from another instance of the same Party, the message would be silently discarded. Now if a Source ID of the Sender is the same as the Receiver, we'll write an informational message to the endpoint log file similar to the following:

ConsumerReceived: The Source ID '{0}' of the Sender is the same as the Receiver. Message received on '{1}' Rabbit MQ Topic with Consumer Tag of '{2}' Will be DISCARDED."

Runtime

MOD – Determining Operating System – Previously Neuron ESB used WMI to determine the Operating System information of the machine. This has been changed to use the System.Runtime.InteropServices.RuntimeInformation API.

MOD – Determine the number of Cores – Previously we used the System.Environment.ProcessorCount for determine the number of cores the underlying machine or virtual machine was configured for. We've changed this as it could provide inconsistent results. We now use WMI

MOD – Licensing Validation – On startup, we used to compare the number of cores that were originally registered for the machine at installation vs the number of cores that exist at the moment of startup. If they were different, startup would abort, and the user would be prompted to deactivate the license and re-activate it. We've removed this startup validation to prevent unexpected production stoppages.

MOD – Dynamic Port Allocation – At runtime, if the default port for the Endpoint Host communication (defined in the appsettings.config file) is in use, we'll make 5 attempts to obtain a lock on it, waiting 5 seconds between attempts. If we fail, we'll find the next available TCP port within the 1000 port range. Previously, we just made one attempt and then assigned a new port. We changed this behavior because sometimes Windows will maintain a port in use temporarily even when the ESB Service has been stopped. This could occur if the ESB Service was quickly restarted.

NEW – Security Checks Added – At startup checks will be made to ensure that the ESB Service and the Neuron ESB Explorer have the appropriate rights to the Solution, log and Temp directories. If not, the appropriate SecurityException error will be thrown and recorded.

Services

FIX – Client Connector using net.msmq Binding fails – If a Client Connector was configured to use a net.msmq binding, it would fail on startup but would not show errors in the Neuron ESB log file for the endpoint. However, if WCF tracing was activated, the following error would be recorded in the WCF trace log for the endpoint:

*System.NullReferenceException: Object reference not set to an instance of an object.
at NeuronEsb.Server.Runner.ESBClientConnector.SetClientConnectorState(Message p1)
at NeuronEsb.Server.Runner.ESBClientConnector.ProcessMessage1Internal(Message message)
at NeuronEsb.Server.Runner.ESBClientConnector.ProcessMessage1(Message p1)*

NEW – Additional Authentication Modes Added - Added support for UserNameForCertificate and UserNameOverTransport authentication modes in custom WCF bindings to support WSE scenarios.

FIX – Service Route Tables – Previously, there was no way to pass through the Incoming REST Method to a Routable Endpoint. Now, if the Routable Endpoint is configured with a blank REST method, the incoming one will be used.

FIX – Service Route Tables – If a user made a copy of a Routable Endpoint by selecting “Copy” from the shortcut menu, a reference copy would be made instead of a clone. Hence if a user changed a property in the original one, the new property value would appear on all the copies. This has been corrected.

NEW – Service Route Tables – When configuring a Routable Endpoint, users can now choose to clear (i.e. remove) the returned message (“Clear Outgoing Message Body” property). This is represented by a check box on the Body tab.

FIX – Service Route Tables – Enabling Auditing on the Response message on the Settings Tab would result in the request body being audited instead of the outgoing response body.

Business Processes

FIX – HTTP Client Utility Process Step - Fixed Http Client Utility dialog so that the Neuron ESB Explorer is its parent and it will no longer disappear behind other windows.

FIX – HTTP Client Utility Process Step - When x-www-form-urlencoded is selected for the body, an extra '&' was being left at the end of the message body.

Neuron 3.7.5.409

Adapters

FIX – MSMQ Adapter generates format name error - The addition of auditing a failed message in build 3.7.0.851 introduced a bug. As part of that previous feature, we attempt to read the QueueName property of the queue we're reading the message from. In some cases, that can throw the following error due to elevated permission requirements:

*Exception occurred reading the queue FormatName:Direct=OS:ITST-LLPAPPBETA\private\$\masterdata. The specified format name does not support the requested operation. For example, a direct queue format name cannot be deleted.
System.Messaging.MessageQueueException (0x80004005): The specified format name does not support the requested operation. For example, a direct queue format name cannot be deleted.
at System.Messaging.MessageQueue.GenerateQueueProperties()
at System.Messaging.MessageQueue.get_QueuePath()*

```
at System.Messaging.MessageQueue.get_QueueName()  
at Neuron.Esb.Adapters.MsmqAdapter.ProcessReceivedMessage(Message msmqMessage)
```

NEW – Rest Adapter. A new adapter that can be used to call Rest APIs has been added to Neuron ESB. Unlike Neuron ESB Service Endpoints, which are based on WCF, this adapter utilizes `System.Net.Http.HttpClient` to perform the Rest calls. This adapter can be used the same way a Rest Service Connector would be used. It's request messages can be created using the Http Client Utility process step, or setting the same message properties in a C# process step. Instead of using a Service Endpoint process step to call the API, you will use the Adapter Endpoint process step. The Rest Adapter can be used to call any Rest API.

When would you use the Rest Adapter instead of a Service Connector? Here are two scenarios we've encountered where WCF cannot call a Rest API:

- Including a body with the GET request
- REST APIs that do not include a "Content-Type" HTTP header on responses

There may be other scenarios where WCF cannot call a REST API, or where `HttpClient` is a better choice. You can use this adapter for all Rest-based calls.

Runtime

FIX – Permission Issue on Configuration – Neuron ESB runtime as well as the Discovery Service runtime require read/write permission to the folder where the Neuron ESB solution is loaded from. Generally, if this is a local folder, there are no issues. However, if a UNC path is used (i.e. the solution actually exists on a remote machine), it's common that the accounts of these services may not be configured with the appropriate permissions. For example, if a UNC path was used for the configuration without the necessary permissions granted to the Discovery service account, the following error could be generated when calling an operation on the Endpoint Health REST API:

```
Could not connect to net.tcp://localhost:0/MasterControl/
```

Neuron ESB as well as the Discovery Service are now checking the existence of the appropriate permissions on startup and will throw a permissions error if the appropriate permissions have not been granted to the service account.

FIX – Added System.Net.Http.Formatting redirect – Added the following redirect to all Neuron ESB related .config files:

```
<dependentAssembly>  
  <assemblyIdentity name="System.Net.Http.Formatting"  
    publicKeyToken="31bf3856ad364e35" culture="neutral"/>  
  <bindingRedirect oldVersion="0.0.0-5.2.7.0" newVersion="5.2.7.0"/>  
</dependentAssembly>
```

Workflow

FIX – Execute Process Activity generates a `NullReferenceException` - If a workflow has an `ExecuteProcess` Activity and the Neuron Process that it executes hits a `Cancel Process Step`, the workflow would abort with a `NullReferenceException`. Now the Result returned to the `ExecuteProcess` Activity will be null in the event the `Cancel Process Step` executes as opposed to throwing the `NullReferenceException`.

Neuron 3.7.5.395

Workflow

FIX – `ObjectDisposed` Error thrown with `Execute Process Activity` - If a workflow contained an `ExecuteProcess` activity, there is the possibility of a `ObjectDisposed` exception occurring when the activity is executed. The chance of this occurring is higher with more concurrent workflow instances (threading related).

Security

FIX – SSL calls could fail – This was a bug introduced with build 3.7.5.389 – when making SSL related calls through a REST based adapter (Salesforce) or Service Connector, the following errors may occur:

`System.IO.IOException: Unable to read data from the transport connection: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.`

`at System.Net.Security._SslStream.EndRead(IAsyncResult asyncResult)`

Or

`System.NotSupportedException: The requested security protocol is not supported.`

`at System.Net.ServicePointManager.ValidateSecurityProtocol(SecurityProtocolType value)`

Neuron 3.7.5.389

Runtime

FIX – Startup Checks can cause unhandled exception - If startup validations routines fail, they could cause the ESB Service to crash rather than gracefully shut down.

Database

FIX - 0046 sql script. The database version was not being updated unless a user ran a workflow.

If a user has ran a workflow and was on the old 0046 script, the database version would be updated to 46.

If a user was on the old 0046 script and did not run a workflow, the database version would remain at 45 and they would get warnings that the database needs to be updated (until a workflow was ran or the new 0046 script was ran by updating the DB through Neuron Explorer).

If a user has not ran a workflow, they can just update the database from Neuron Explorer as normal as the new 0046 script will run since the database version would still be marked as 45.

If a user has ran a workflow, they will want to consider running the ALTER PROCEDURE portion of the new 0046 script, though there is no harm in leaving the old stored procedure from the old 0046 script.

Business Processes

FIX – Audit Step throws Null Reference when Testing - When a Business Process used an Audit Step. during testing in the Process Designer, if a user tested the Business Process multiple times, any test after the first one would fail with a null reference exception.

FIX – Audit steps exceeds pool size - In the Audit Step, the Maximum Instances for Pool Management may sometimes be exceeded during high concurrency (i.e. when multiple threads are executing the same Business Process object).

Retry Process Step – You can now dynamically set the retry count and delay properties for the retry process step by setting the message properties neuron.retryCount and neuron.retryDelay in either a code step or the Set Properties process step.

Adapters

MOD – File Adapter Error Header -Added the folder path information to the error header that gets written if an error is thrown in publish mode.

MOD – File Adapter multiple instances can read the same file – The exclusive lock was ignored so the adapter could process duplicate files if more than one instance of the adapter was running in the environment.

FIX - Dynamics CRM XRM Adapter – fixed issue that could happen when you use a port number on the server name for on-premises CRM deployments. The error received is similar to: “Unable to login to Dynamics CRM, Error was : Metadata contains a reference that cannot be resolved: 'http://crm2015:443/NeuronDemo/XRMServices/2011/Organization.svc?wsdl&sdkversion=9'.”.

MOD - Dynamics CRM XRM adapter – Modified to work with security requirements from Office365 logon to OAuth-based login when using Dynamics 365 online.

Repository

FIX – KeyNotFoundException recorded in Event log when resolving XSLT Includes - Added a new Xml resolver to use for Xslt include lookups. Previously we use a single Xml resolver for both Xsd and Xslt lookups. This caused unnecessary KeyNotFoundException errors to appear in the event log when looking up Xslts.

Neuron 3.7.5.381

Workflow

FIX – Unable to Cancel Workflow - Fixing issue where workflows would not cancel from Workflow Tracking in Neuron Explorer if they are in a running state.

FIX – Unable to Cancel Workflow - Fixing issue where Execute Process workflow activity would not cancel if contained within a timeout activity when the timeout occurs.

FIX – Completed Time not populated in Endpoint Health - fix to show completed time on workflow tracking when a workflow is canceled or terminated.

Business Processes

FIX - Service Endpoint Process Step – Fixed bug when dynamically setting which service connector is executed by the Service Endpoint Process Step using the property Context.Data.Header.Service. If the process step was configured with a Service Endpoint selected in the dropdown list, the value set in Context.Data.Header.Service would be ignored.

Neuron ESB Explorer

FIX – Unable to Test XSLT Documents - Xml Transformations Repository – Fixed error when using Neuron ESB Explorer to test an XSLT containing includes that reference other XSLTs in the repository using the “esb:” syntax. You would receive the error “The transformation failed due to the error below. XSLT compile error.”

FIX - Identity Server OAuth Provider - Enabled the Test function when you are creating the OAuth Provider in Neuron Explorer.

Services

NEW - Added OAuth support for SOAP-based client and service connectors

Installer

MOD – Java Runtime Replacement – During installation, Neuron ESB will now install the Open JDK version 16, rather than Java 1.8. This is required for Data Mapping. Open JDK does not require a runtime license.

Neuron 3.7.5.372

Services

FIX - Azure AD/ADFS OAuth Provider - on-prem ADFS stopped working with latest updates - would receive "Unable to obtain an access token - Authority validation is not supported for this type of authority. Parameter name: validateAuthority. This was a bug introduced with build 3.6.0.1327

Adapters

FIX – File Adapter – When configured for Publish mode and Polling, Polling would not commence. This was a bug introduced with build 3.6.0.1337

Business Processes

FIX – Timeout Process Steps throws Client Context exception - If a Neuron Business Process contained a Timeout step, the steps inside could throw an exception due to the ClientContext being null. For example, if a Timeout step contains a Service Endpoint step, this will cause the Service Endpoint step to throw an exception.

Neuron Runtime

FIX – Environment Variables not being available - If a user configured the Neuron runtime to use Neuron Environment Variables that are defined in the App.config of the ESBSERVICE, if the environment variables are desired inside a Neuron Business Process when connecting with a remote Neuron client (i.e. using the Neuron Client SDK), and if the Neuron Environment Variables were used in a Neuron Binding (e.g. the URL for a Service Connector), the values contained in the App.config were not reflected in the Binding (e.g. reading the URL for a Service Connector in a C# step using the Neuron Configuration object). This bug should have only occurred if the Neuron Client/Party executing the Neuron Business Process is not hosted within the ESBSERVICE or an Endpoint Host.

Endpoint Health

FIX – Workflow “Waiting” Performance Counter - If there was a timeout activity that contained a ReceiveMessage activity, the "Waiting" counter in endpoint health would increment too much.

FIX – Endpoint Host stats incorrect - The Endpoint Host's stats were not being aggregated from the endpoints that it hosted.

Migration

FIX – Opening 3.6 solutions did not migrate Client Connector Settings – When opening a 3.6 solution in the 3.7.5 Neuron ESB Explorer, the Process Mode settings for a Client Connector were not getting updated. This could result in a 3.6 Client Connector which was configured to run a Business Process, would instead be configured for Messaging, with a blank Topic and Party.

Neuron 3.7.5.365 (3.7.5) Features

Neuron ESB 3.7.5 is being released in conjunction with the release of the [Peregrine Connect Management Suite 1.1](#). As an integral part of the Peregrine Connect Integration Platform, Neuron ESB provides critical runtime facilities and features that collaboratively simplify the design, deployment, and management of an organization's enterprise integration needs.

The Peregrine Connect Integration Platform provides a suite of applications that helps organizations meet any integration demand – from on-premise and cloud applications to various data sources and devices. Using the Peregrine Connect Integration Platform any number of workloads can be simplified, deployed, and monitored; from API Management and automated Workflow to scheduled integration flows and Alerting and Notifications.



This release introduces an array of new and complimentary capabilities within Neuron ESB and the Management Suite. Neuron ESB 3.7.5 generates the necessary event data that allows the Management Suite to effectively monitor environments while delivering Alerting and Notifications, API Management, Scheduling, and other features to organizations looking to accomplish their application integration, workflow automation, and API management goals.

If you're using previous versions of Neuron ESB today, please download the Neuron ESB 3.7.5 setup to update your existing installation. Current customers can obtain either the Patch or the full installation from the [Neuron ESB Support website](#). If you are new to Neuron ESB, you can download the full installation from the [Peregrine Connect website](#).

Neuron ESB 3.7.5 adds some exciting new features to the Peregrine Connect Integration Platform, some of which include:

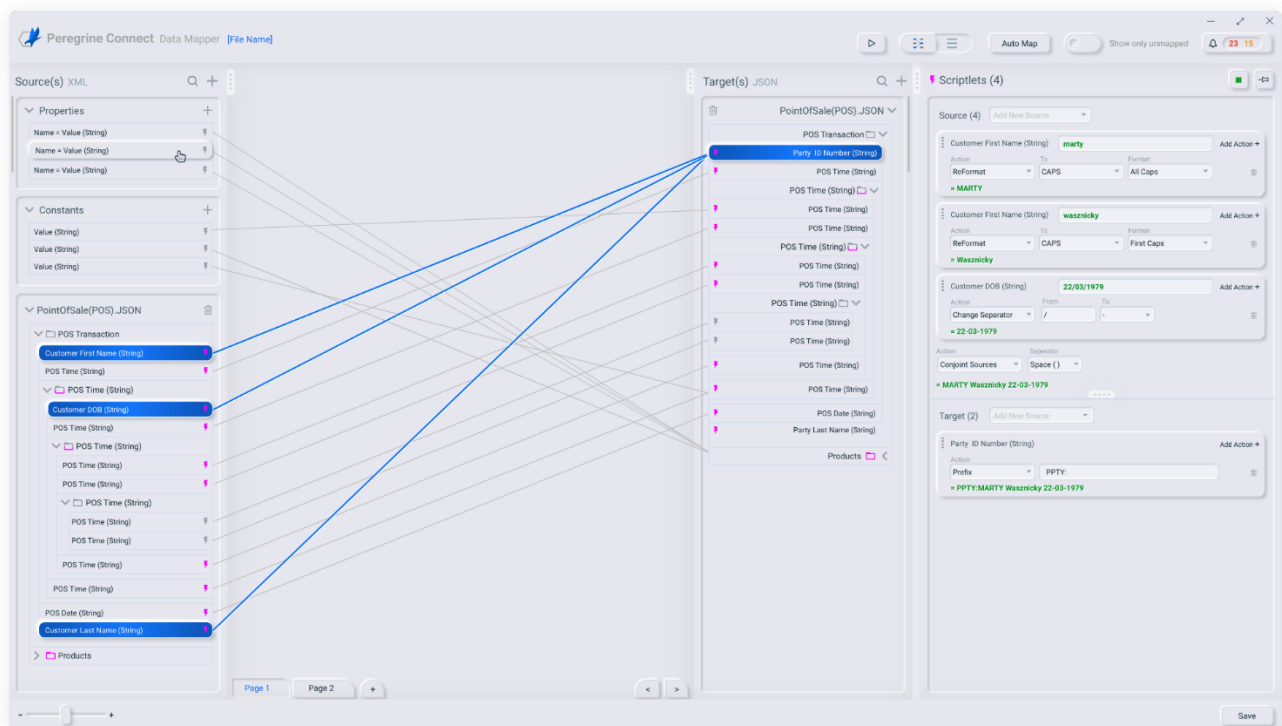
- [Graphical Data Mapping](#)
- [Containerization of Endpoints](#)
- [Service Routing Tables](#)
- [Client-Side OAuth Providers](#)
- [Business Process Scheduling](#)
- [Wiretap Tracing Neuron ESB Messages](#)
- [Rabbit MQ Quorum Queues](#)

All the changes included in the Neuron ESB 3.7.5 release can be found in the [Neuron ESB Change Log](#), which gets installed with Neuron ESB. This article focuses on the major feature additions as well as their Peregrine Connect Management Suite counterparts.

Graphical Data Mapping

Overview

The Peregrine Connect Data Mapper is a new graphical tool that ships with Neuron ESB 3.7.5 to visually create data maps from source document schema or sample to target document schema or sample. It also serves as a tool to extract and define custom Neuron ESB Message properties used in Business Processes, Workflows, Service Endpoints, and Adapters. A Data Map created using the Data Mapper is a JSON structure. It contains the definition of the source and target documents and links between the source fields/custom properties and target fields/custom properties. Environment variables and constants can also be used in a Data Map.



Peregrine Connect Data Mapper – Figure 1- Data Mapper with source and target documents loaded, fields selected, and individual field transformations being tested (in green).

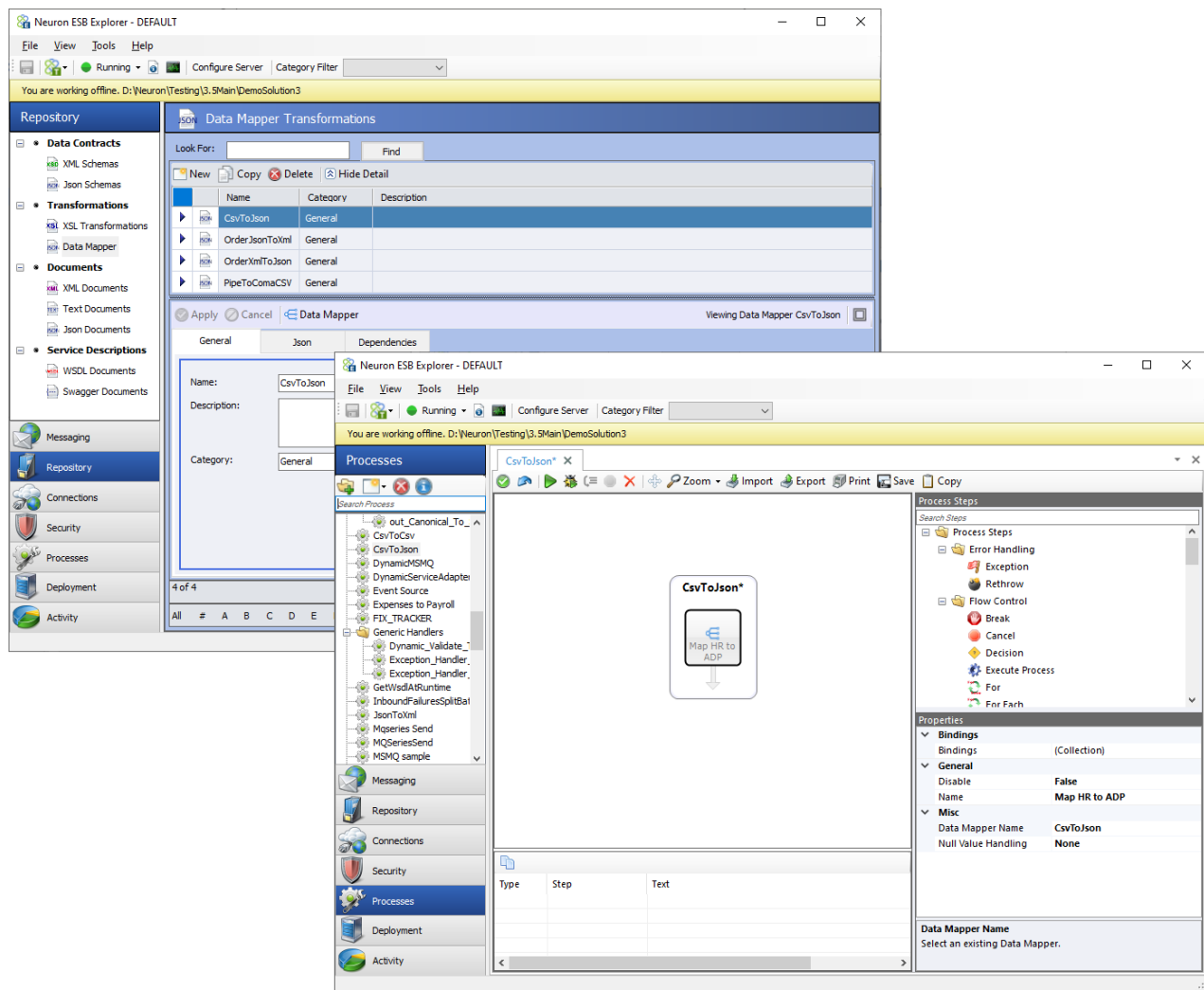
Data mapping is a crucial design step in data migration, data integration, and data transformation projects. It includes mapping data fields from a source format to a target format. Every company deals with massive amounts of data coming from many sources. The data may reside in different formats, and organizations need a powerful, user-friendly tool such as Peregrine Connect Data Mapper to map between data formats. The Data Mapper accelerates the data mapping process and makes it user friendly for business users. It enables businesses to extend their integration and automate/evolve their business processes.

The Data Mapper provides an interactive user interface that simplifies configuring integrations between XML, JSON, and CSV file data sources and targets. You can design your data mapping with the Data Mapper UI canvas and then execute the Data Map via the Neuron ESB runtime engine. Data Maps are stored in the Neuron ESB Solution repository. They can be configured and executed at runtime from Service Routing Tables, Business Processes, and Workflows.

Data maps can range from simple to complex based on the type of hierarchical data structure that source or target schemas represent and the complexity of conversion rules that the target application requires for successful data integration. Data maps can also include multiple source documents that will be merged or joined prior to mapping the result to a target document.

With its drag and drop interface, the mapper can be used by non-technical users. Click and drag a source field onto a target field and your mapping is done. If there is a need to apply additional rules on the map, you can use the built-in functions to transform the data per your business rules. Built-in functions include math, string, conditional, and type conversions.

The Peregrine Connect Data Mapper can be configured using either the Repository Menu option as shown below or by double-clicking on a Data Mapper Process Step within a Neuron ESB Business Process or Workflow. Both options will launch the visual Data Mapper.



Neuron ESB Explorer – Figure 2- The upper picture depicts the new Data Mapper repository within the Neuron ESB Explorer while the lower picture displays the new Data Mapper Process Step within a Business Process. A similar activity also exists for the workflow designer.

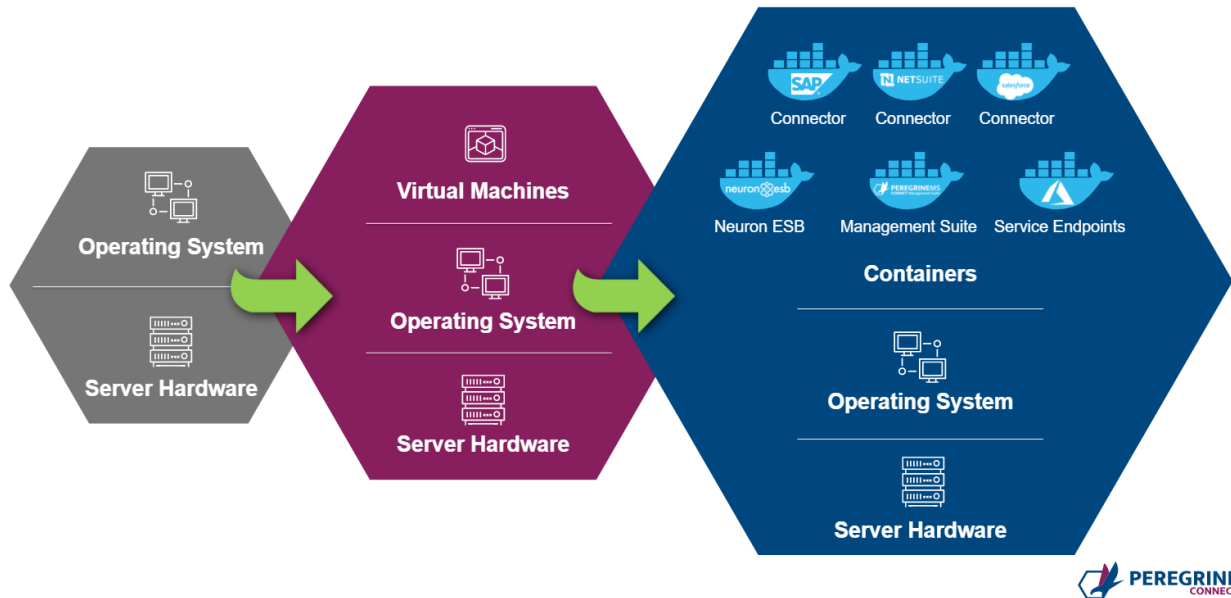
The Peregrine Connect Data Mapper supports several exciting features with a focus on usability, some of which are listed below:

- [End to end Testing of a Data Map](#)
- [Testing of individual Field Transformations](#)
- [Library of Transformations](#)
- [Expressions](#)
- [Auto Mapping](#)
- [Paging](#)

You can learn more about the Peregrine Connect Data Mapper by visiting [Data Mapper Documentation site](#).

Containerization of Endpoints

Previous versions of Neuron ESB provided organizations a Microservice hosting model. However, Neuron ESB 3.7.5 takes a giant step toward Microservice Deployment by supporting Containerization at the endpoint level.



Neuron ESB 3.7.5 has been built on the principal of SOA and Microservices. Hence it is the natural choice to build Microservice-based APIs. In previous versions of Neuron ESB, organizations could easily create REST APIs or SOAP-based services. However, in this release, they can now control the deployment aspect of it as well. Using Neuron ESB 3.7.5, organizations can now choose to run a service (or group of services) as a fully independent service that can be scaled in or out based on consumption using platform services like Docker or Kubernetes.

Some organizations may deploy their applications in a Docker or Docker Swarm host or deploy them to the Kubernetes service. The Neuron ESB runtime and Endpoint Hosts can be deployed individually in Docker to Windows containers on Windows (WCOW). The Neuron ESB database is also capable of being hosted in a Docker container. Neuron ESB 3.7.5 includes an SQLExpress Docker container image for development and testing purposes. However, in production scenarios, organizations are expected to bring their SQL server setup (container or without container) to host the Neuron ESB database. A RabbitMQ container image is also made available with Neuron ESB 3.7.5 to support the Event Processor and Neuron ESB Rabbit MQ based Topics.

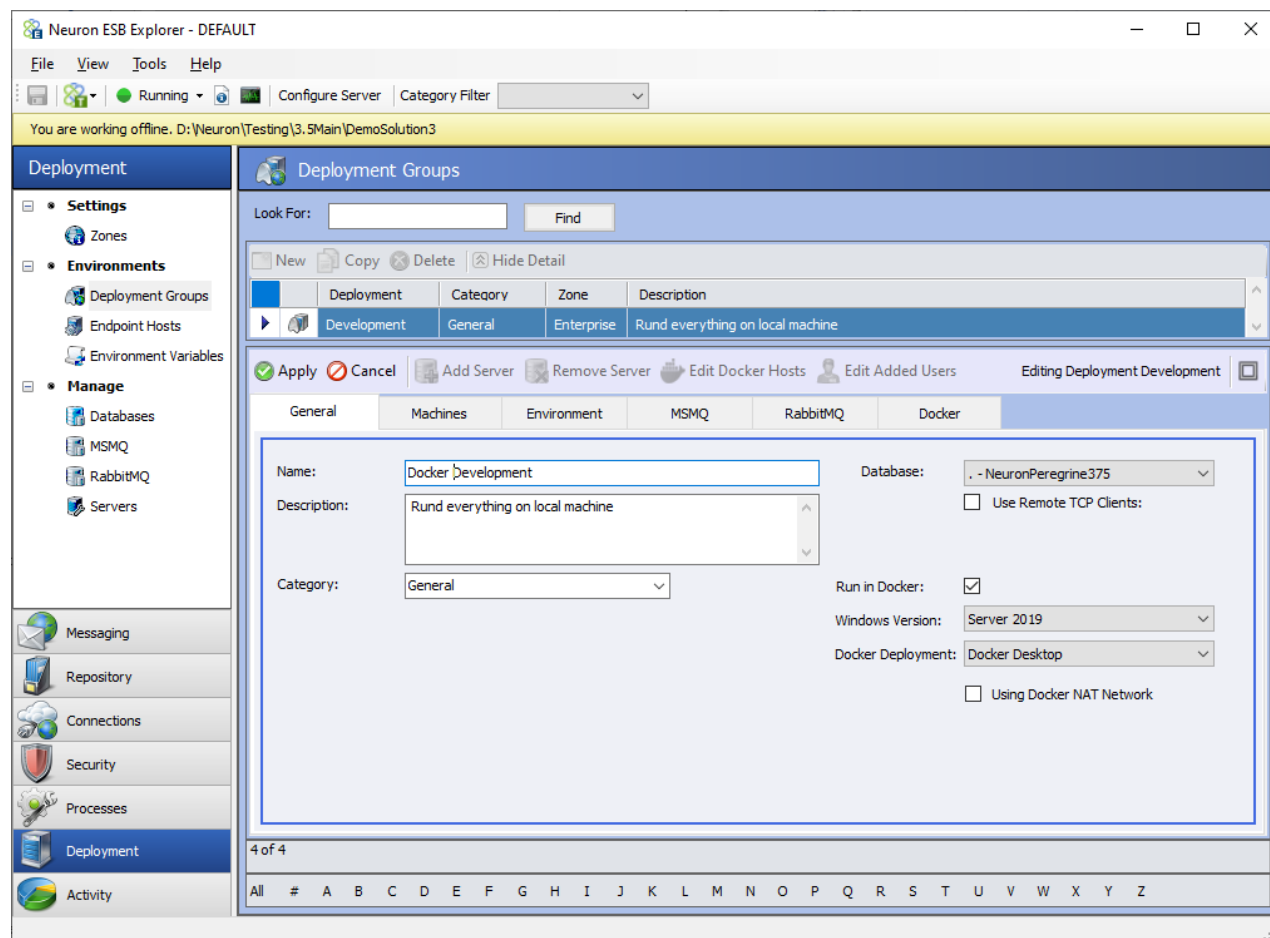
Running Neuron ESB in Docker requires additional configuration to Deployment Groups, Endpoint Hosts, Database Settings, and Security Credentials. Once the solution is configured to run in Docker, the generate Docker files menu option in the File Menu of the Neuron ESB Explorer can be used to generate Docker Compose files from the Neuron ESB Solution. The Docker Compose files can then be deployed to Docker to run a Neuron ESB image that is pulled from Docker Hub.

Neuron Solution

To support the containerization of endpoints, a few new objects have been added to the Neuron ESBSolution. These need to be configured in Neuron ESB Explorer if running Neuron ESB in containers.

Deployment Groups

The “Docker” tab will be visible if the Deployment Group has the “Run in Docker” option enabled. Almost all the configuration for running in Docker is done at the Deployment Group level.

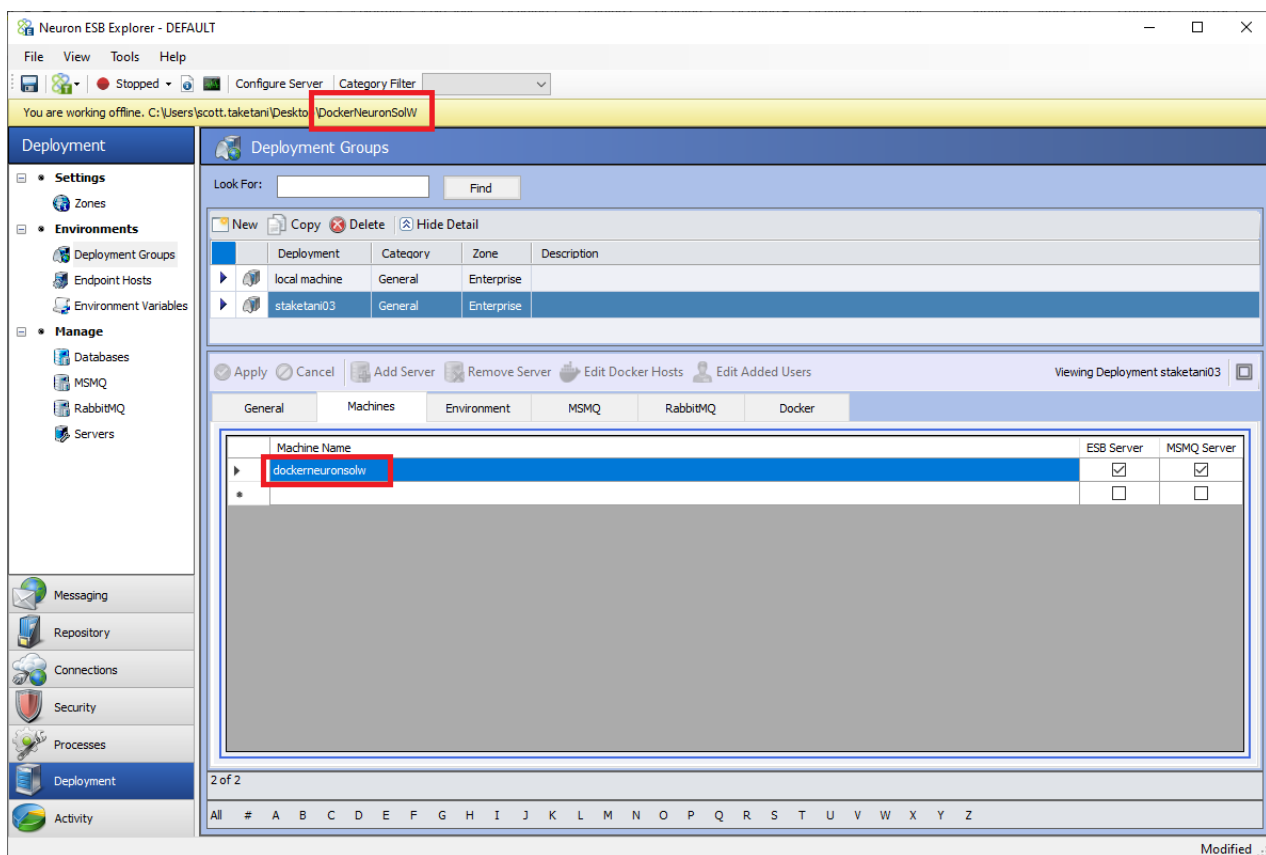


Neuron ESB Explorer – Figure 3- New Docker Container settings have been added to the Deployment Groups. Settings exist on both the General and Docker tabs.

The “Run in Docker” option will cause Docker Compose files to be generated for the selected Deployment Group and any Endpoint Host that runs in the Deployment Group when the “Generate Docker Files” option is clicked in Neuron ESB Explorer’s “File” drop down menu.

There is a drop-down box on the General tab of the Deployment Group to choose which version of Windows Server the desired Docker Host will run on. The correct version must be selected for the correct image to be downloaded at runtime. Another drop-down box shows the Docker Deployment mode with two choices: Docker Desktop and Docker Swarm. If this selection is Docker Swarm, certain features under the Docker tab such as allocated memory and allocated cores are not enabled. If the Docker NAT Network is to be used, the check box “Using Docker NAT Network” should be enabled. The Docker NAT Network is the default network that is installed with Docker on Windows.

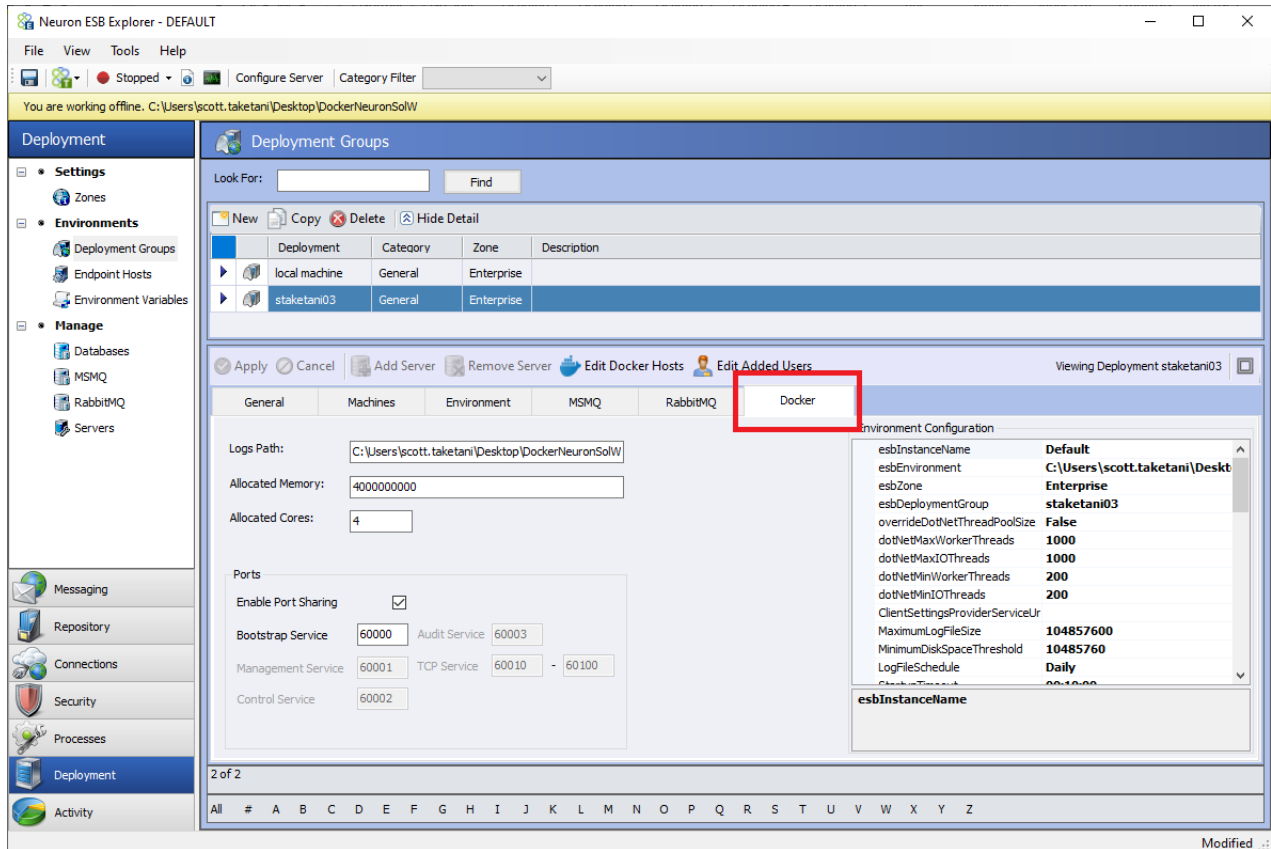
Once the General tab is configured, there should only be one machine listed as an ESB Server under the Machines tab of the Deployment Group as shown below. The ESB Service container’s hostname will be the solution's name without spaces in the generated Docker Compose files. This name should be entered as the machine name.



Neuron ESB Explorer – Figure 4- When configuring for Docker deployment there should only be one machine listed as an ESB Server. The machine name will be the name of the solution without spaces in the generated Docker Compose files.

Docker Settings

A new tab called “Docker” has been added to Neuron Explorer that is only visible if the Deployment Group’s “Run in Docker” setting is enabled.



Neuron ESB Explorer – Figure 5- The Docker tab of the Deployment Group contains all the settings required for the container as well as memory and cpu allocations for the ESB Service runtime.

Each setting within the Docker tab is described below:

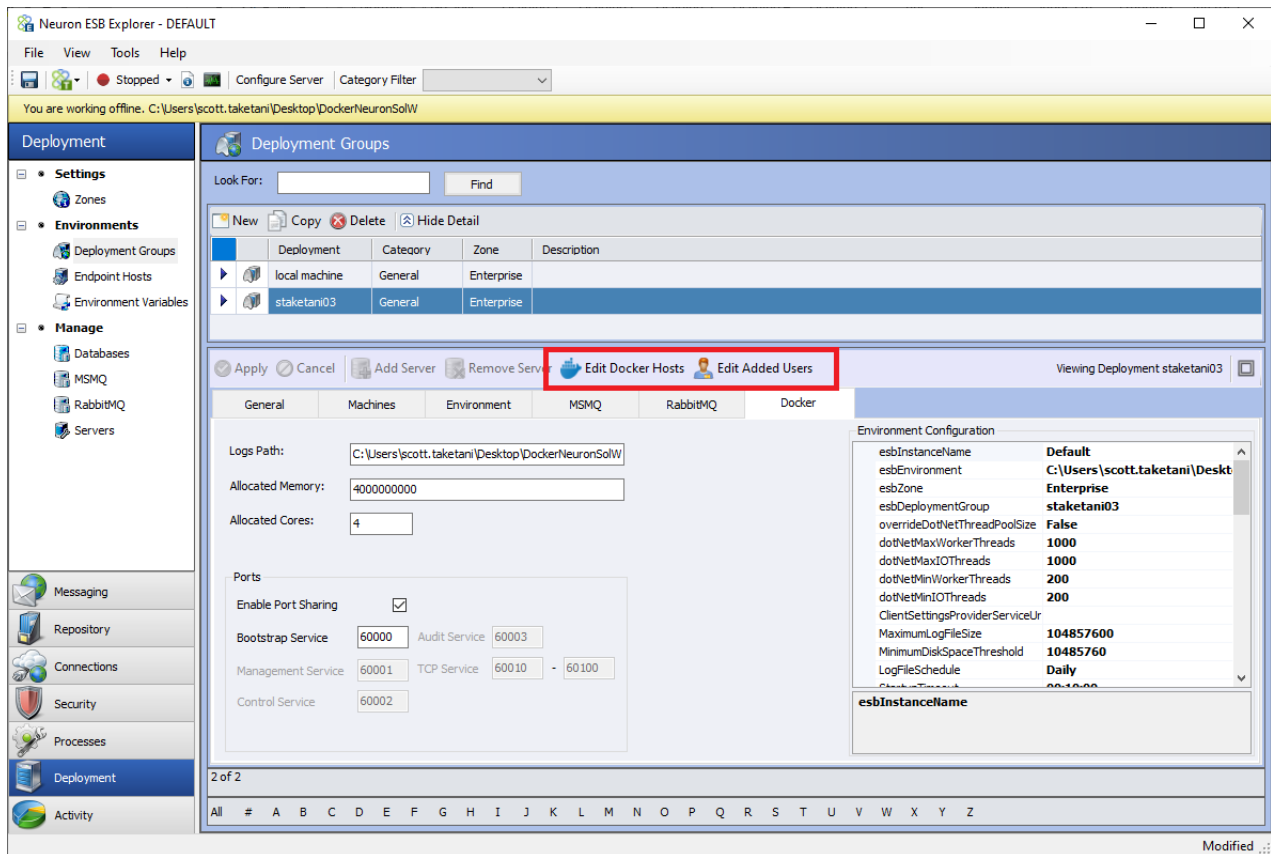
- **Logs Path:** This is the directory path local to the Docker Host machine that log files generated by the Neuron containers will be written to. The ESBService’s container will log to this path utilizing the same naming pattern as running in a non-containerized environment. Endpoint Host logs will reside in the same root directory as the ESBService container’s logs except, they will be written to a subdirectory of that root directory called “EndpointHostLogs.”
- **Allocated Memory:** This is an integer value that will determine how many bytes the ESBService container will be allocated for memory. For information on how to set the Allocated Memory for an Endpoint Host, see the Endpoint Hosts section.
- **Allocated Cores:** This is an integer value that will determine how many processor cores will be allocated to the ESBService container. For information on how to set the Allocated Cores for an Endpoint Host, see the Endpoint Hosts section.

- **Ports:** These settings are the same as the port settings in the Zone section of Neuron Explorer. These settings will override the Zone settings when running in a Docker container. That means port settings are configurable by Deployment Group rather than on a solution level when running containerized.

Note: If you reserved some ports from Hyper-V to run Neuron on Host also, you need to change the ports in the solution to other values that are not reserved for use outside Hyper-V.

- **Environment Configuration:** These are the equivalent of the “appSettings.config” file found in the Neuron Instance’s installation directory. For container configuration, you can override all the settings shown here. Refer to the Neuron ESB Documentation for appsettings.

Also, there are two new tool strip buttons called “Edit Docker Hosts” and “Edit Added Users” that are enabled on the Docker tab.



Neuron ESB Explorer – Figure 6- The “Edit Docker Hosts” and “Edit Added Users” buttons have been added to facilitate Docker configuration.

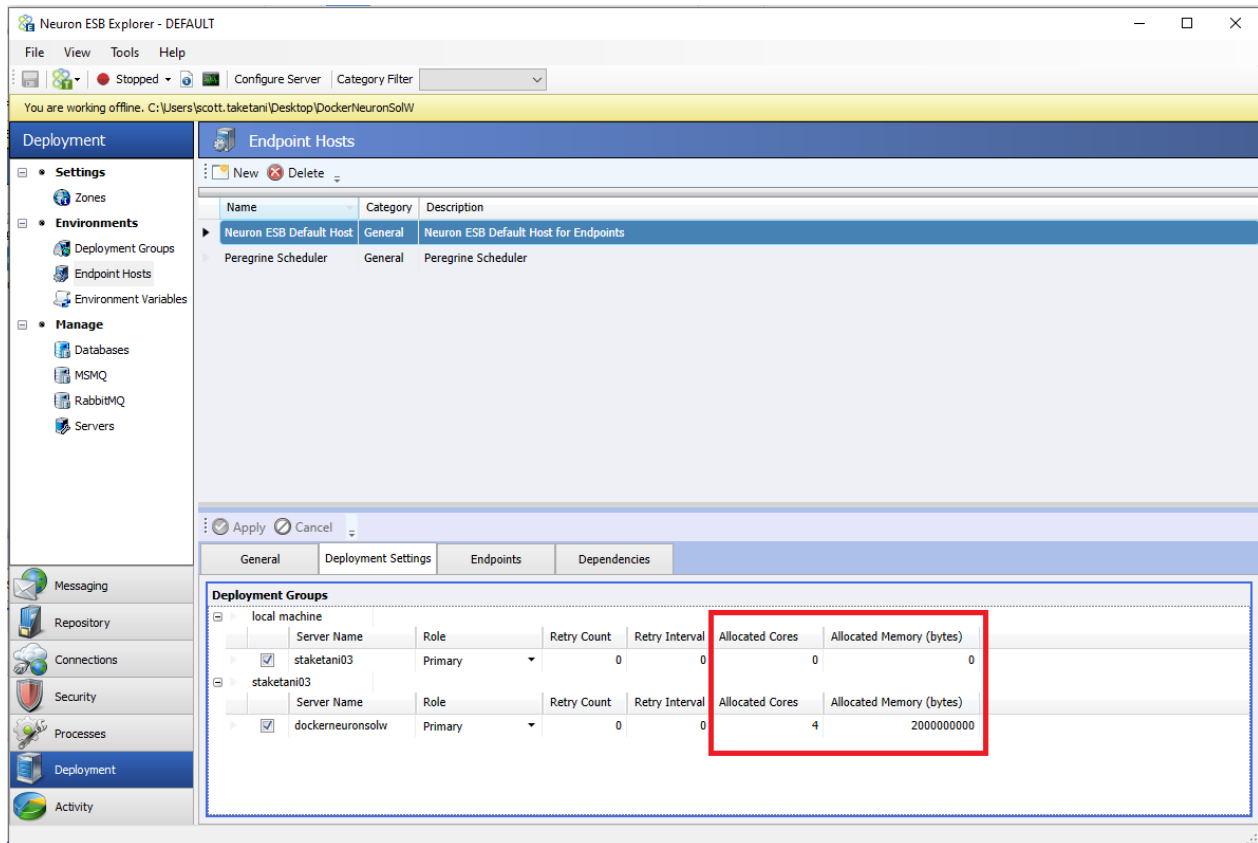
- **Edit Docker Hosts:** This is where the user should define each Docker Host’s machine name. This is mainly used to determine which machines to query for the Endpoint Health screen in Neuron Explorer.

- **Edit Added Users:** This is where users can define which credentials to add to the container on startup. The credentials are defined in the Security area in Neuron Explorer. The only type of credentials that can be added are “Windows Container” type credentials.

The purpose of adding these users to the container’s Windows Users and Groups is mainly for connecting with Neuron Explorer in online mode. The Neuron credentials added to the list are added to the chosen Windows Group (i.e., either “Users” or “Administrators”) inside the container.

Endpoint Hosts

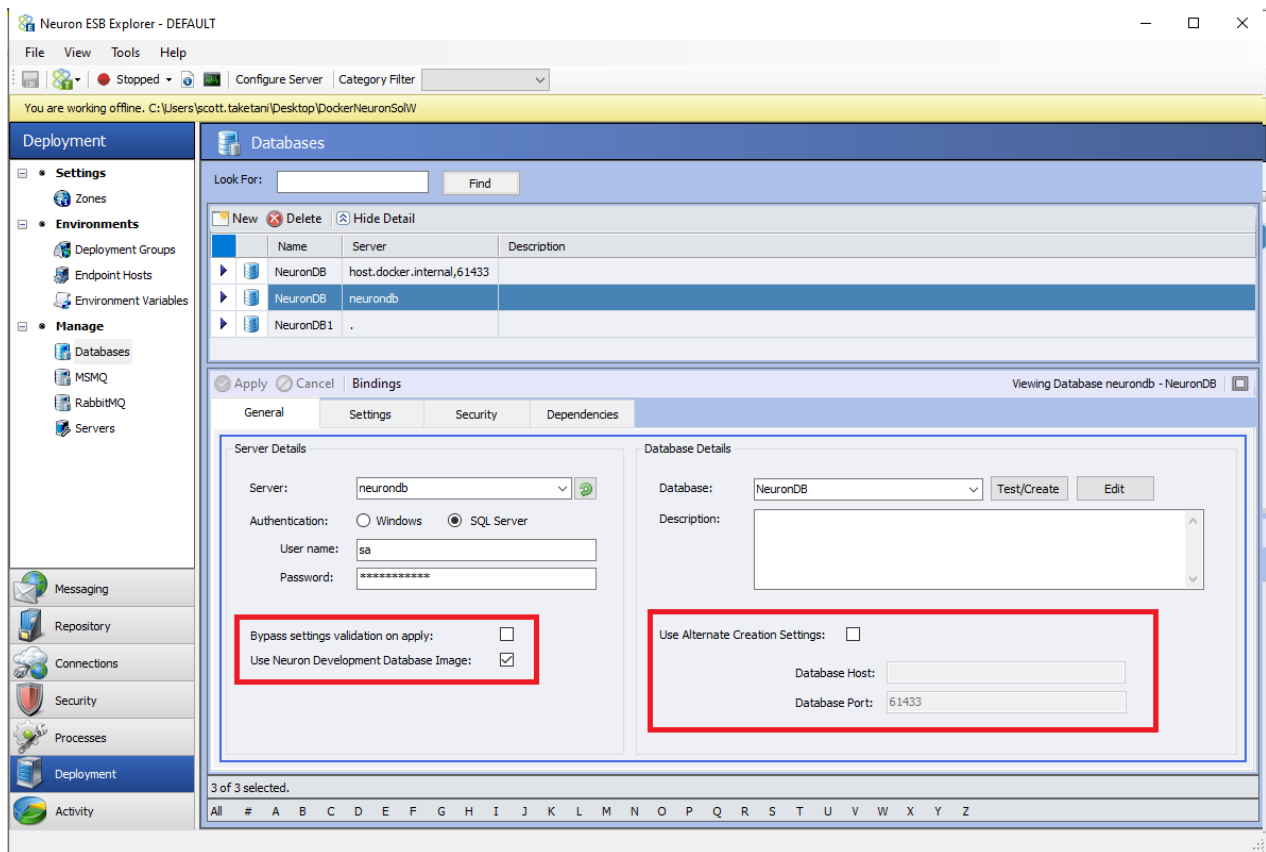
One of the key advantages of running Neuron ESB in containers is that each Endpoint Host can run in their own isolated, individual container. Organizations can assign any number of endpoints to an Endpoint Host regardless of type. For Endpoint Hosts, there are two new settings defined when mapping to deployment servers. The settings are “Allocated Cores” and “Allocated Memory.” The “Allocated Cores” setting defines how many of the Docker Host machine’s cores are allocated to run the Endpoint Host’s container when running on the mapped Docker Host machine for that Endpoint Host. “Allocated Memory” is how much memory will be available, in GB, to the Endpoint Host container. The Docker Compose file that is generated will contain these settings.



Neuron ESB Explorer – Figure 7- Memory and CPU allocations for individual Endpoint Hosts that will be containerized are allocated at the Endpoint Host level.

Database

A few new options have been added to the Database settings in Neuron ESB Explorer. These are required because the database server hostname can be different when running in a container alongside the Neuron ESB containers, as opposed to if the database server is runs on a different machine than the Docker Host.



Neuron ESB Explorer – Figure 8- Modifications have also been made to the Neuron ESB Database configuration to support Docker containerization

Settings

- **Server:** This should be the name of the machine hosting the SQL Server instance (if running in a container on a different host machine than Neuron, this should be the host machine's name and port, if not 1433, that was mapped from the SQL Server container to the host machine).
- **Authentication:** The authentication method used for the desired SQL Server.
- **Bypass Settings Validation on Apply:** If enabled, when the Database settings are applied using the "Apply" button in Neuron Explorer, the standard database settings validation is skipped. This is useful when the SQL Server is unavailable when the settings are applied (e.g., the container running the database server is not up).
- **Use Neuron Development Database Image:** see the section below on using the development Neuron SQL Express image for details.
- **Database:** The name of the database to be used/created.

- **Use Alternate Creation Settings:** This enables different settings for the SQL Server host and port when using the “Test/Create” button to create the database. This should be used in the situation that the SQL Server hostname supplied in the “Server” text box is different than what should be used to test/create the database from where Neuron Explorer is being used. An example of this scenario would be that Neuron Explorer resides on machine “A” while the SQL Server container and Neuron container are running on machine “B” in the same instance of Docker. This is because the “Server” setting is used during runtime to create the connection string that is used to communicate from the Neuron container to the SQL Server container whereas the Alternate settings are used only when Neuron Explorer is creating the database.

Creating the Database

Normally, database creation is done by Neuron ESB Explorer using the “Server” setting in the connection string. This is fine when Neuron ESB Explorer can reach the database using the same connection string as the Neuron ESB runtime. But suppose the runtime connection string differs from the connection string used to create the database, as in the scenario described under “Use Alternate Creation Settings” above. In that case, alternate settings need to be specified during database creation.

Also, when using the Development Neuron SQL Express image and running that image in the same Docker container host, alternate settings may be necessary for database creation depending on the version of Docker (versions below 2.4.0.0) since it is the same scenario described above.

After saving the database record, go to the deployment group and select this database. Then generate Docker Files from the file menu. This will create a docker-compose file named like <yourdeploymentgroup>_docker-composedb.yaml in the Neuron ESB Solution folder under the “docker\<Deployment Group name>”.

You need to host this container image before you can use it to Test/Create a database. For hosting that image, you will need to run the following command.

`docker-compose -f <yourdeploymentgroup>_docker-composedb.yaml up` (Note: Change the file name to the one that is generated for you.)

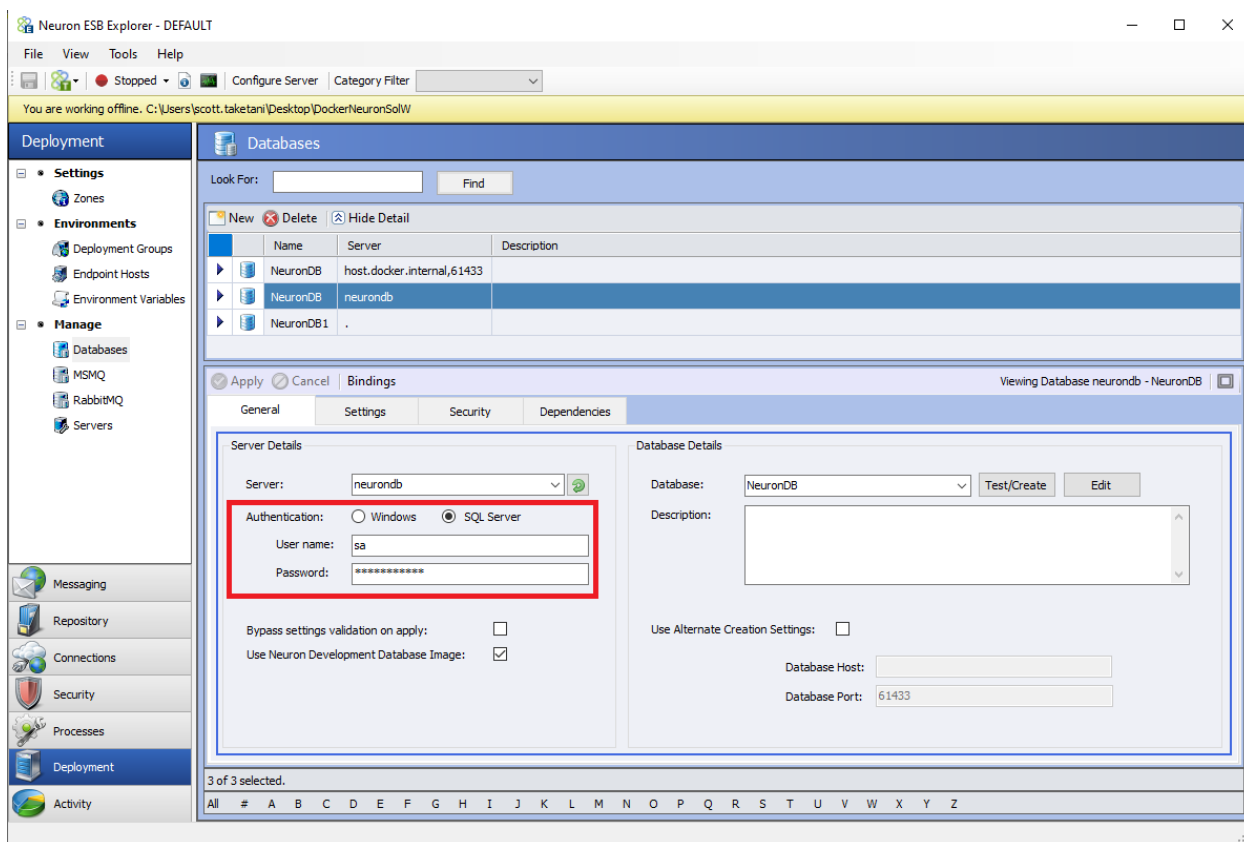
Using Development Neuron SQL Express Image

The purpose of the Development Neuron SQL Express image is to make it easier for developers to test a Neuron ESB Solution running in a container. This setting, when enabled, will cause the database settings to be automatically filled in when the “Apply” button is clicked in Neuron ESB Explorer. When the Docker Compose files are generated, if the database being used by a Deployment Group has this setting enabled, a Docker Compose file will be generated for a database container as well. This database container will use a Neuron ESB supplied SQL Express image.

Note that the database itself must still be created once in the Development Neuron SQL Express container. This can be accomplished by clicking the “Test/Create” button in Neuron ESB Explorer with the database container running (and accessible by Neuron ESB Explorer). The database’s data files are

contained in the Neuron ESB Solution in the “docker\<Deployment Group name>\Database Files” folder. When using the Neuron provided database container, the “Database Files” directory is mounted in the container to the default directory that would normally contain the data files.

Also note that when using the Development Neuron SQL Express image, any changes made to the specified database will be persisted (as well as any changes made to the SQL Express instance itself...e.g. a login created to access the SQL instance will be persisted if the container is restarted).



Neuron ESB Explorer – Figure 9- When using SQL Express, SQL Server security must be used.

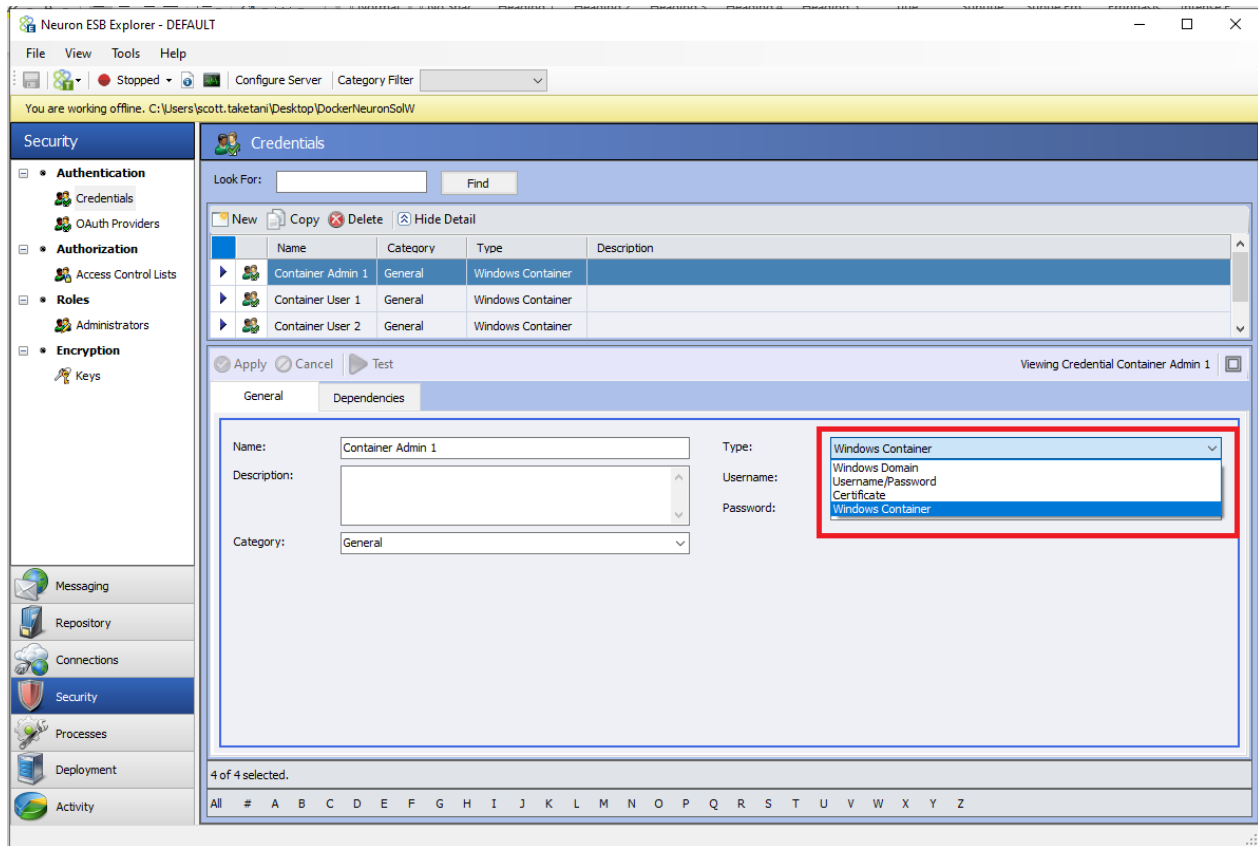
Furthermore, the SA password will be what is set in the database’s settings in Neuron ESB Explorer. The “Username” setting must be “sa”, and the “Password” will be stored in plain text in the Docker Compose file for the Development Database container. This is a reason why the Development Database should only be used in a development/test environment.

Security

A new credential type has been added to the “Credentials” area of the “Security” tab in Neuron ESB Explorer. The following information in this section applies to a Neuron ESB instance running in a Docker Container.

Credentials

The new credential type that was added is called “Windows Container”. This credential type can be used when running Neuron ESB in a Docker container.



Neuron ESB Explorer – Figure 10- The new “Windows Container” credential type has been added to the Credentials section of the Neuron ESB Explorer.

The credential itself is a username and password. This credential type can be used for the following purposes:

- In an Access Control List to secure a Client Connector with HTTP Basic Authentication
- In a Role to secure access to the running Neuron Solution
- To add a local Windows User/Administrator to a Neuron container’s underlying Windows Operating System

Access Control Lists

Access Control Lists utilizing the “Windows Container” type credential can be used to secure a Client Connector with HTTP Basic Authentication in the same manner as other credential types that are allowed for Client Connector authentication.

Roles

By default, there are three Roles defined in a Neuron Solution: Administrators, Users, and Everyone. These are the Roles used to secure access to the Neuron ESB Solution when running in a container. Securing the Neuron ESB Solution will prevent unauthorized users from viewing and or modifying the solution with Neuron ESB Explorer unless their credentials are added to the “Edit Added Users” dialog in the Deployment Group Docker settings.

The way this works is that the selected credentials in the “Edit Added Users” dialog are added as either a local user or administrator in the Neuron ESB Container’s underlying Windows Operating system. This occurs when the ESB Service or Endpoint Host is starting in the Neuron ESB container.

Since the default Roles use the specified local Windows Groups, a user will only be able to connect to the running solution if they know the username password combination of a credential that has been added in the “Edit Added Users” dialog.

To connect to the running solution, users will need to specify the correct credential in Neuron ESB Explorer’s connect dialog. Note that the “Domain” field should be the name of the container’s hostname, which will be the name of the Neuron ESB Solution without spaces (if using the generated Docker Compose files as is).

Client Connectors

When generating the Docker Compose files for a Neuron ESB Solution, any Client Connectors defined to run will cause a port mapping to be added to the Compose file. Thus, the host’s port and the container’s port will be the same (e.g., if the Client Connector URL is “http://0.0.0.0:8001/MyService” then a port mapping of “8001:8001” will be added). This means that the Docker Host must have the defined port for each Client Connector available.

```
# endpointhost for neuronessbdefaulthost
dockerneuronsolw_neuronessbdefaulthost:
  image: neuronessbperegrineconnect/neuronessb:localbuild_2019
  entrypoint:
    - /ProgramData/Microsoft/Windows/Start Menu/Programs/Startup/NeuronStartup.bat
    - endpointhost
  networks:
    default:
      aliases:
        - dockerneuronsolw_neuronessbdefaulthost
  env_file: staketani03_config.env
  hostname: dockerneuronsolw_neuronessbdefaulthost
  environment:
    endpointHostName: Neuron ESB Default Host
    bootStrapPort: 60000
    controlPort: 60002
    logLevel: INFO
    logPath: c:\\app\\logs
    esbhostName: dockerneuronsolw
    HOSTNAME: dockerneuronsolw_neuronessbdefaulthost
    portSharingEnabled: "True"
    IsScheduler: "False"
  volumes:
    - C:\\Users\\scott.taketani\\Desktop\\DockerNeuronSolW:C:\\Users\\scott.taketani\\Desktop\\DockerNeuronSolW
    - C:\\Users\\scott.taketani\\Desktop\\DockerNeuronSolW\\docker\\logs\\EndpointHostLogs:C:\\app\\logs
    - C:\\Users\\scott.taketani\\Desktop\\DockerNeuronSolW\\docker\\staketani03\\Startup Scripts:C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Startup
    - C:\\Users\\scott.taketani\\Desktop\\DockerNeuronSolW\\docker\\staketani03\\Custom Assemblies:C:\\app\\Custom Assemblies
    - C:\\Users\\scott.taketani\\Desktop\\DockerNeuronSolW\\docker\\staketani03\\Custom Pipelines:C:\\app\\Pipelines
  ports:
    - 8001:8001
  cpu_count: 4
  mem_limit: 2000000000
  depends_on:
    - "dockerneuronsolw"
```

Neuron ESB Explorer – Figure 11- An example of port mapping within the Compose file produced by the Neuron ESB Explorer.

In Docker Compose port mappings, the left side is the port that will be used on the Docker Host machine, whereas the right side is the port that it gets mapped to in the container.

Security

Client Connector's running in the Neuron ESB Containers can be secured using an Access Control List (for HTTP Basic Authentication only), or a proxy can be stood up in front of the container if transport-level security is desired. In the latter case, the proxy should terminate the TLS/SSL and pass the request unencrypted to the Client Connector endpoint.

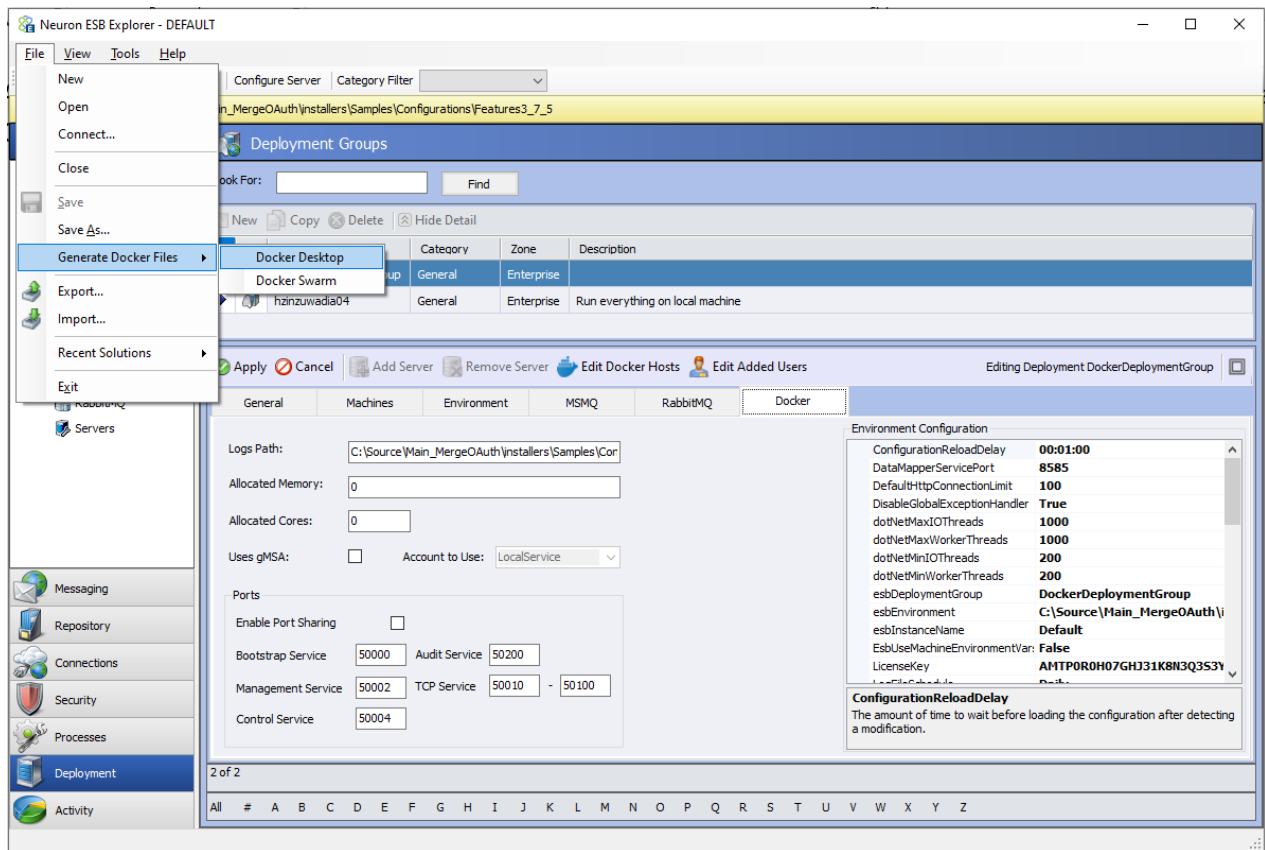
If the proxy is a container running in the same Docker Host and uses the same network adapter, the port mapping for the Client Connector should be manually removed from the generated Docker Compose file. In this use case, the proxy container has access to the internal Docker network and should forward the request using the hostname of the container running the Client Connector and the Client Connector's port defined in its URL.

If the proxy is running on the Docker Host machine (not in a container), the listening port for the proxy and port for the Client Connector must be different. The proxy should forward the request to the Client Connector's port, and the generated port mapping should remain as is in the Docker Compose files.

Generate Docker Files

Overview

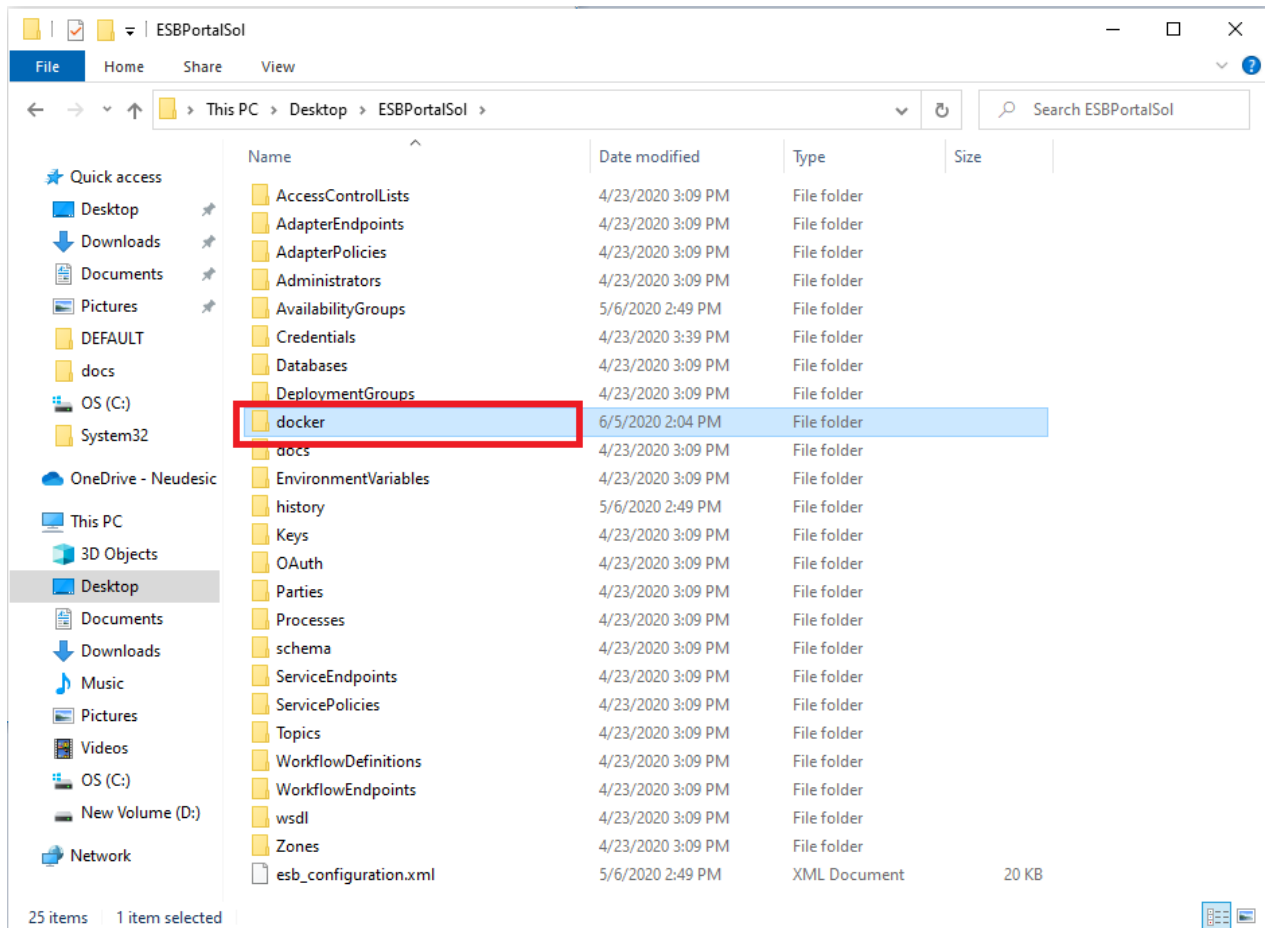
A new "File" menu option has been added in Neuron ESB Explorer that will automatically create the necessary Docker Compose or Docker Swarm (Docker Service) files to run a Neuron Solution in Docker.



Neuron ESB Explorer – Figure 12- Generating Docker files can be done directly from the menu within the Neuron ESB Explorer.

Generated Files

The necessary files generated from the “Generate Docker Files” feature will be located in the Neuron ESB Solution’s “docker” directory. For each Deployment Group that has the “Run in Docker” option enabled, a directory will be created in that “docker” directory. The files to run the Deployment Group are contained within those directories.



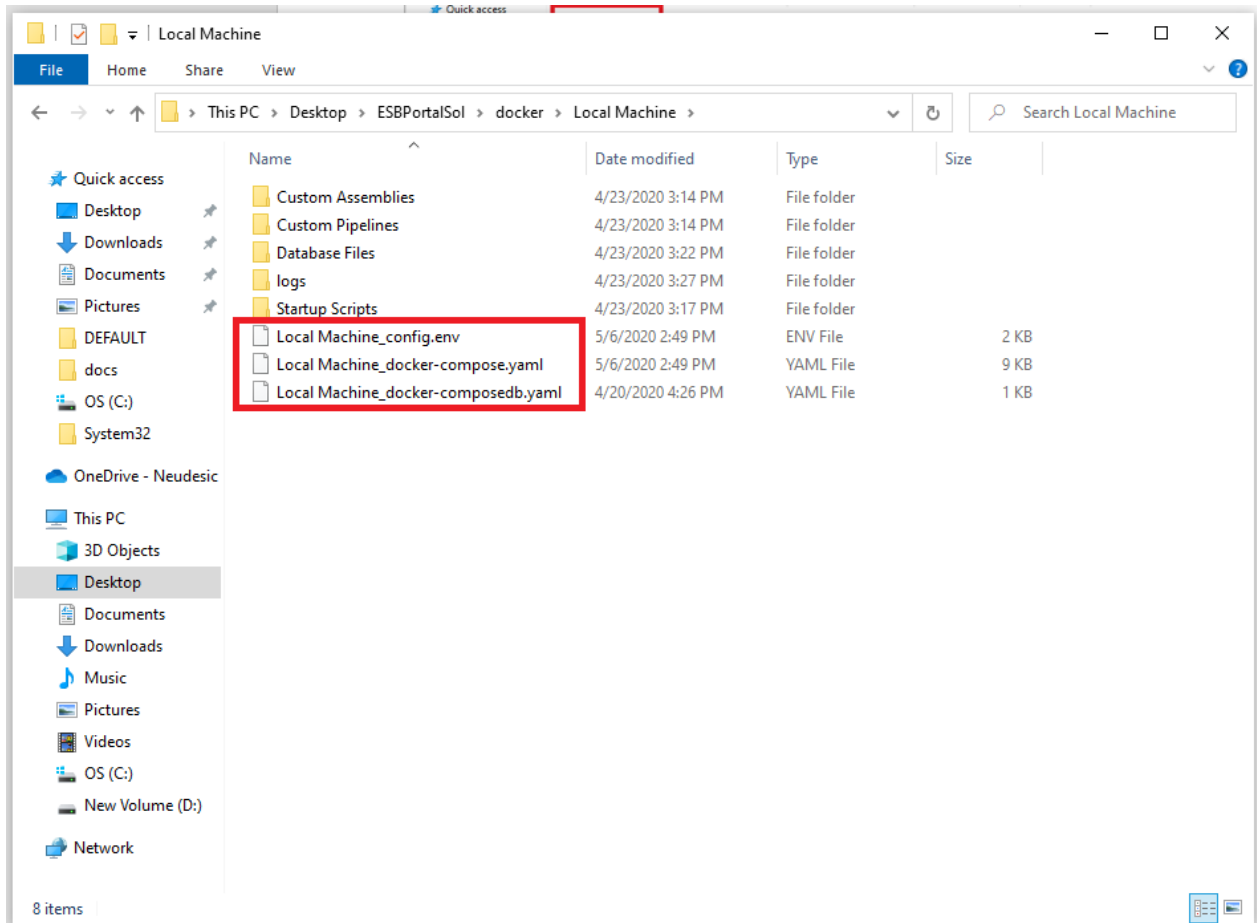
Neuron ESB Explorer – Figure 13- Generating Docker files are stored in the docker subfolder of the solution that is loaded within the Neuron ESB Explorer.

There will be one main Docker Compose file for each Deployment Group. The name of the file will take the name “<DeploymentGroupName>_docker-compose.yaml”. This file contains the Docker configuration that defines Neuron as a Docker service.

There will also be one “config.env” file for each Deployment Group. This file contains the information found in the Docker Settings for the Deployment Group in the Environment Configuration property grid. The information here will be passed into the Neuron ESB containers at runtime. In case of Swarm deployment, this file will be prefixed with Swarm/

An Event Processor yaml file will be created per Deployment Group. The name of this file will be in the form “<DeploymentGroupName>_EPS.yaml.” The Event Processor is used to insert information into the Peregrine Connect Management Suite’s Elasticsearch data store, which the Management Suite uses for reporting Neuron ESB runtime information. Therefore, if using the Event Processor in a container, the container should be started before trying to start an environment from the Management Suite.

An optional file is also created if the database used by the Deployment Group is set to use the Neuron ESB Development Database Image. This file will be used to deploy the Development Database if the user has the option enabled on the database in Neuron ESB Explorer.



Neuron ESB Explorer – Figure 14- There are three generated Docker files produced by the Neuron ESB Explorer depending on if generating for Docker Desktop, Docker Swarm, or Kubernetes.

Ports

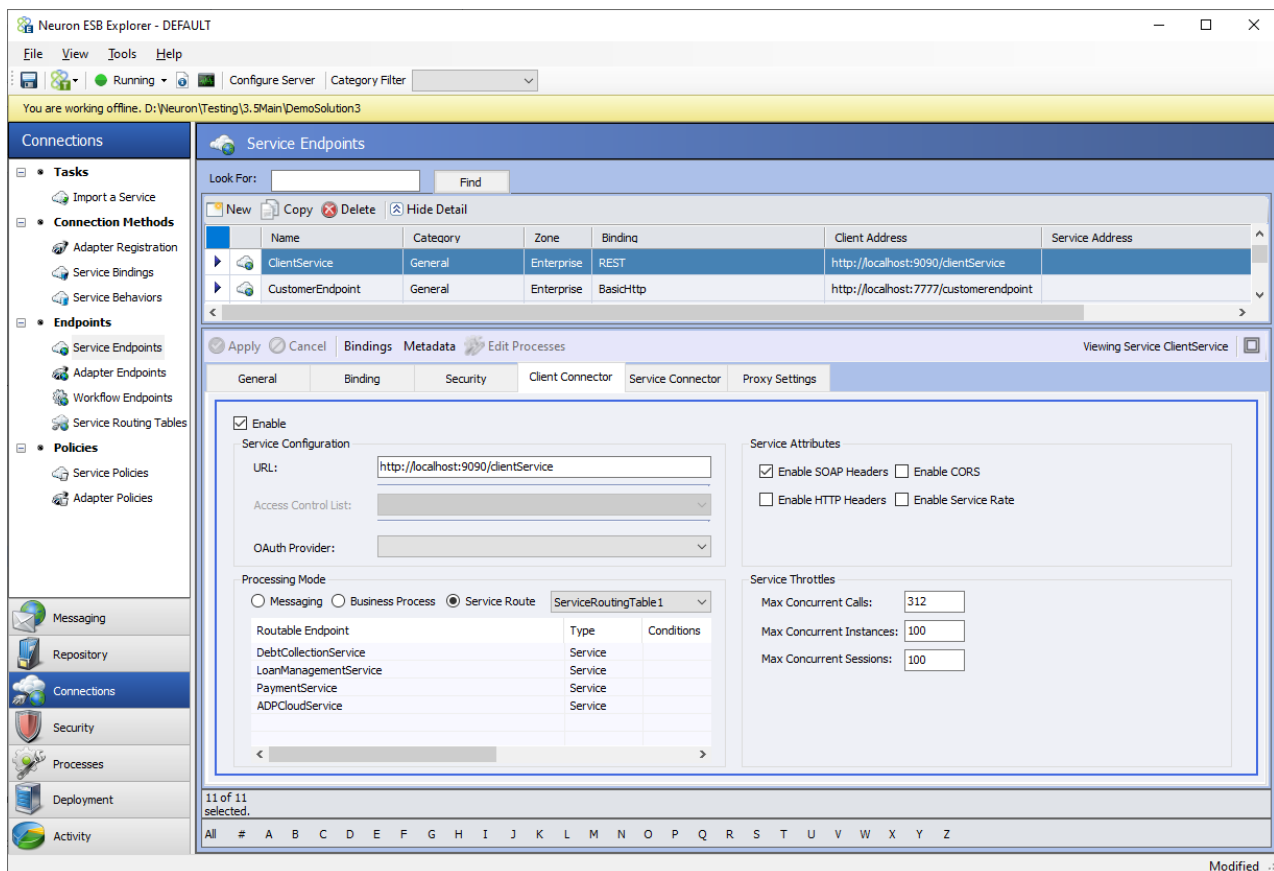
The ports that the Neuron ESB Solution needs are mapped to the Docker Hosts ports. This means that if the Neuron ESB Solution is set to use a port, the port must be available on the Docker Host itself. Due to the way Docker implements ingress to the containers, anything outside of the Docker Host can use the Docker Host's machine name and port number to connect to the container if that port is mapped.

For example, the bootstrap port for the Neuron ESB Service is 50000. To connect to the solution using Neuron ESB Explorer's connect feature (if connecting from a remote machine), the user should supply the Docker Host's machine name as the server, as well as the appropriate credentials (Users and Administrators can be added through "Edit Added Users" in the Docker settings for the Deployment Group in Neuron ESB Explorer).

In the same scenario, but instead of connecting from the Docker Host machine, the server name provided to the connect feature can be “host.docker.internal”, the Docker Host’s machine name, or “localhost.”

Service Routing Tables

Over the years of providing integrated solutions to a various of organizations, some common integration scenarios have emerged. Neuron ESB now provides a zero-code solution for implementing these scenarios using a new feature of Neuron ESB Client Connectors called Service Routing Tables.



Neuron ESB Explorer – Figure 15- A Client Connector configured with a Service Routing Table. When a Service Routing Table is selected, its high-level information (e.g., endpoint, type, and condition) is displayed in the grid below it. Double-clicking on the grid will navigate the user to the Service Routing Table to view or edit.

Service Routing Tables are a powerful and simple alternative to using Business Processes to create pass-thru and dynamic routing activities within an organization. Service Routing Tables provide a very low latency way of implementing basic scenarios of mocking, mapping, and routing services without the overhead of a Business Process. Though Business Processes are far more customizable, they have overhead because of all the context available in the Business Process to customize message processing.

In a Service Routing Table, one or more service endpoints can be included and then the Service Routing Table can be associated with a Client Connector. When configured this way, all calls to the Client Connector are routed to the backend service endpoint configured in the Service Routing Table.

Some of the scenarios that can be configured using Service Routing Tables are listed below:

- Authorization and Access Control
- Dynamic Routing
- Message format conversion
- Stubbing / Mocking or Service virtualization
- Protocol Conversion
- Service Availability and Policy
- Blue-Green Deployment and Testing
- Message Enrichment / Content Masking
- Request Filtering
- URL Rewriting
- Message Transformation

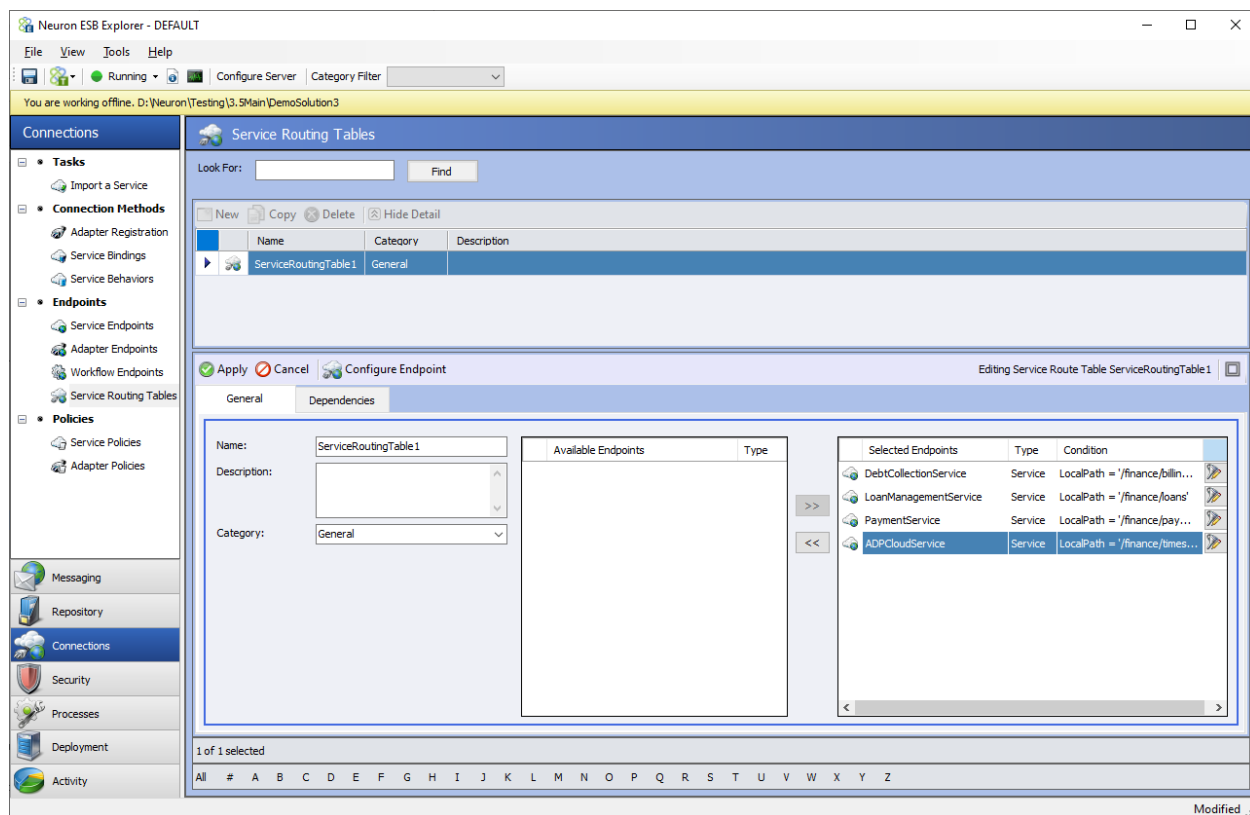
Service Routing Tables are a powerful feature allowing organizations to perform complex operations using a no-code approach. For example, new clients can use javascript friendly REST APIs to interoperate with legacy WCF services or existing WCF clients can upgrade to new REST APIs without changing their code. Other examples can encompass the use of OAuth Tokens and API keys. For instance, standard Query strings like API Keys could be automatically appended to a service endpoint or, route to different endpoints or append alternative paths/query strings based on the scope in the incoming OAuth Token. Countless opportunities exist because of the flexibility we built into the Service Routing Table feature.

Although the Service Routing Table functionality is designed as a simple and low latency solution while providing for many real-life use cases, it may not cover every need. Therefore, it may not suit an organization's exact needs for the above use cases. In those outlier cases, configuring the Client Connector with a Business Process and implementing the exact logic that's needed may be the best option. Business Processes provides the ultimate flexibility to manipulate the message and header with a variety of useful steps, including custom C# code.

Configuring Routes

To configure a Service Routing Table, first Service Connectors must be added in your solution. When Service Connectors are referenced with a Service Routing Table the Service Endpoint itself does not need to be enabled nor does the Service Connector need to be enabled for Messaging. Internally, upon startup of the solution, the Service Connector will not be "live", e.g., hosted in either the ESB Service or Neuron Endpoint Host process taking up valuable processing time or memory. These Service Connectors are called inline within the Client Connector after the route is resolved.

Once Service Connectors have been added to the solution, they can be added to a Service Routing Table. Select Service Routing Tables from the left-hand menu under Connections. Click the New button to get to the screen shown below.



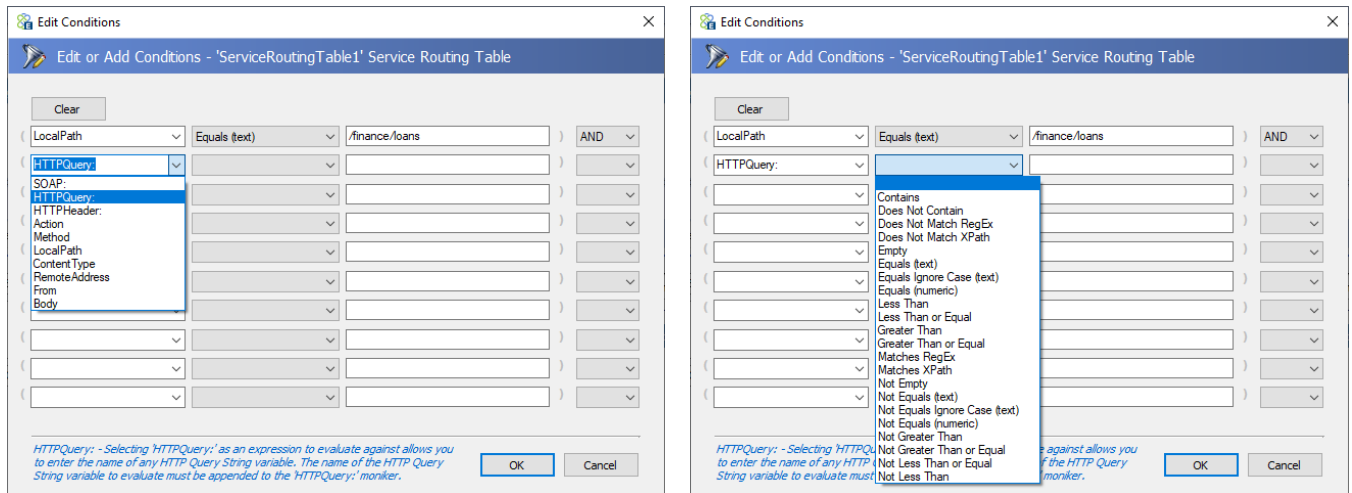
Neuron ESB Explorer – Figure 16- A Service Routing Table located under Connections -> Endpoints. All Service Connectors will be listed on the Available Endpoints window. All selected endpoints are listed in the right. Double-clicking, right-clicking or clicking on the “Configure Endpoint” button will launch the Configure Endpoint dialog for the selected endpoint.

You can see the available Service Connectors under the Available Endpoints in the bottom portion of this screen. From there move the Endpoints you want to include in the routing table using the >> arrow button. You can also remove the Service Connectors by using << arrow button.

Condition Editor

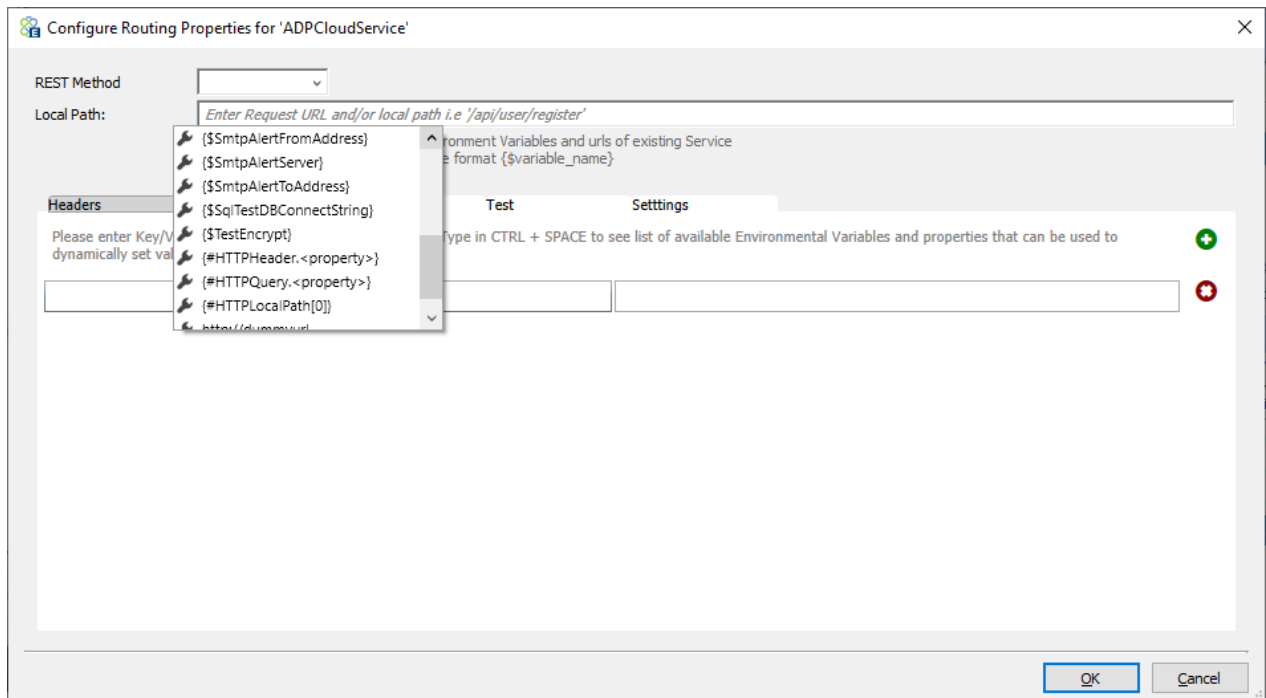
A Service Routing Table can contain more than one service endpoint and multiple copies of the same endpoint. The incoming message will be routed to the first service endpoint that satisfies the Condition configured for the selected endpoint. Clicking on the Conditions icon in the last column titled Condition, the conditions editor can be opened.

Users can select from a variety of properties or even the incoming message body to check and the condition to match in the first two drop downs. Whereas a free form expression to evaluate can be entered in the third box and a join clause (AND, OR, or XOR) in the last dropdown box.



Neuron ESB Explorer – Figure 17- Clicking the Conditions icon will launch the Edit Conditions dialog. Users can create complex conditions that serve as routing rules for the individual endpoints within a Service Routing Table. The conditions are resolved at runtime to determine which endpoint to route the incoming request to.

Once the service routing endpoints are added, each endpoint can be configured in the right-hand Selected Endpoints box. Click configure endpoints to display the screen shown below.



Neuron ESB Explorer – Figure 18- The Configure Routing Properties dialog launched for an endpoint within the Service Routing Table. Here users can configure almost any property of the targeted endpoint using expressions, variables, static text, constants in any combination thereof.

A big part of the flexibility that's been built into the Service Routing Table feature is the ability to configure almost every property of the Service Connector by using any combination of text, Environment Variables, pieces of the original request body, incoming SOAP or HTTP Headers or HTTP QueryStrings as well as the any part of the local path or even the original URL. In any property where this is supported, users can view the available list of options by pressing the CTRL+Space key combination on the keyboard. In the case of the Local Path property, users can either append to the existing local path, add a new local path, or overwrite the existing URL of the Service Connector entirely.

For Dynamic Routing scenarios, users can add copies of the same Service Connector to the list of Selected Endpoints and then provide a unique condition. The Configure Routing Properties dialog shown above allows the user to apply completely different local paths or URL options to the same Service Connector. In that case, they could do the same with HTTP headers and even send unique request bodies.

REST Method or SOAP Action

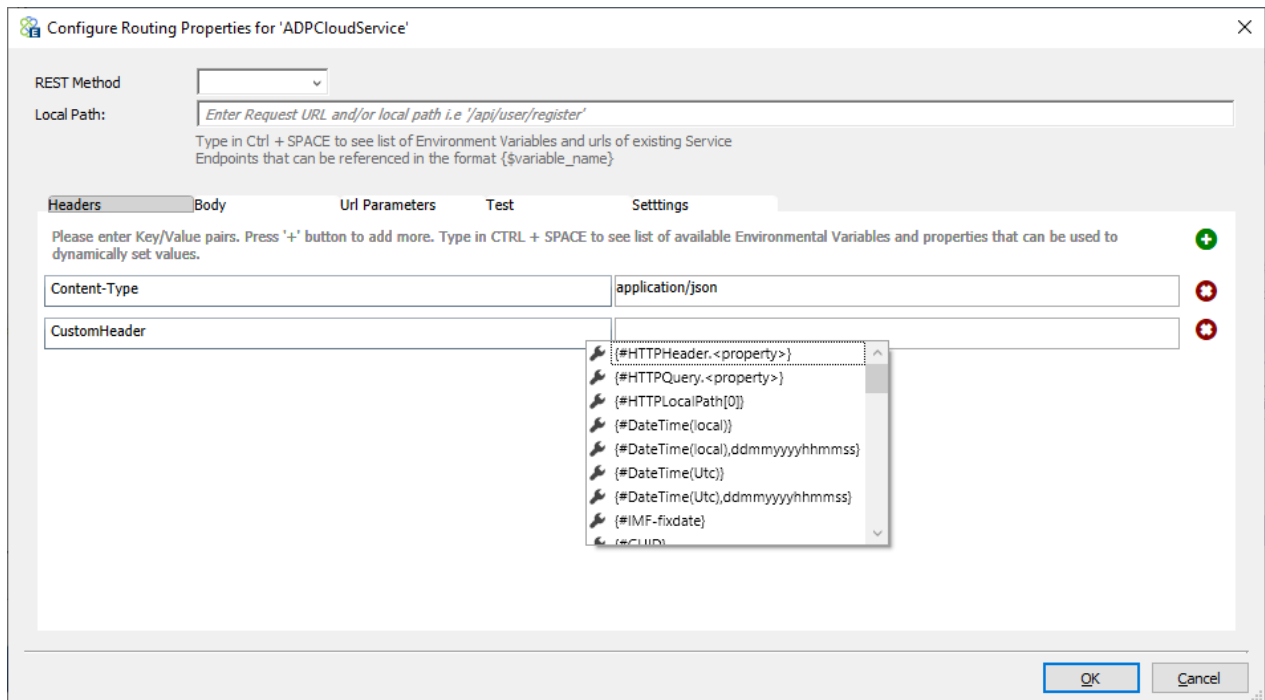
If the selected service is a WCF or SOAP service, the SOAP Action will be the first field in the top row, whereas if it is a REST service the first field in the top row will be Method. For a SOAP service, the SOAP action is a required field, and for REST service, the Method is a required field.

Local Path

The second field is the local path. The local path may include a partial URL to be appended to the base service URL or contain the full-service URL with the local path appended. In the local path and SOAP Action, the CTRL+Space key combination can be used to select some token substitution as shown above. These tokens and/or text can be concatenated to create the final property value.

Headers

Users can enter one or more key-value pairs in the headers section, and these headers will be added as HTTP headers. When adding headers, users may use token substitution in the value column. Neuron ESB Environment Variables, Neuron ESB Message properties, Constants, HTTP Query, and Header properties will appear in the drop-down list. You can use the icons for + and x signs to add /remove key-value pairs in this table.



Neuron ESB Explorer – Figure 19- The Configure Routing Properties Dialog Headers tab. This is where HTTP Headers for the outbound message can be configured and manipulated.

Body

Next, you can customize the message body. For example, you can choose to transform the incoming message using a Data Map, or you may choose to compose a new message based on existing properties.

The Message body can be Form Body, x-www-form-urlencoded, or raw body. In case of first two options, a grid of key-value pairs like the headers section will appear, whereas for raw body a choice of Text, XML, or JSON in the drop-down will be presented. Users can also clear the Message body (if case you are using the REST and GET method, this is required).

Optionally a request Data Map and a response Data Map can be applied to incoming / returned messages. All the existing Data Maps in the solution will appear in the dropdown selection.

Configure Routing Properties for 'ADPCloudService'

REST Method: POST

Local Path: /finance/billing

Type in Ctrl + SPACE to see list of Environment Variables and urls of existing Service
Endpoints that can be referenced in the format {variable_name}

Headers | **Body** | Url Parameters | Test | Settings

☐ Form Body ☐ x-www-form-urlencoded ☒ Raw XML ☐ Clear Incoming Message Body

Request Data Map: OrderXmlToJson Response Data Map: OrderJsonToXml

Type in CTRL + SPACE to see list of available Environmental Variables and properties that can be used to dynamically set values.

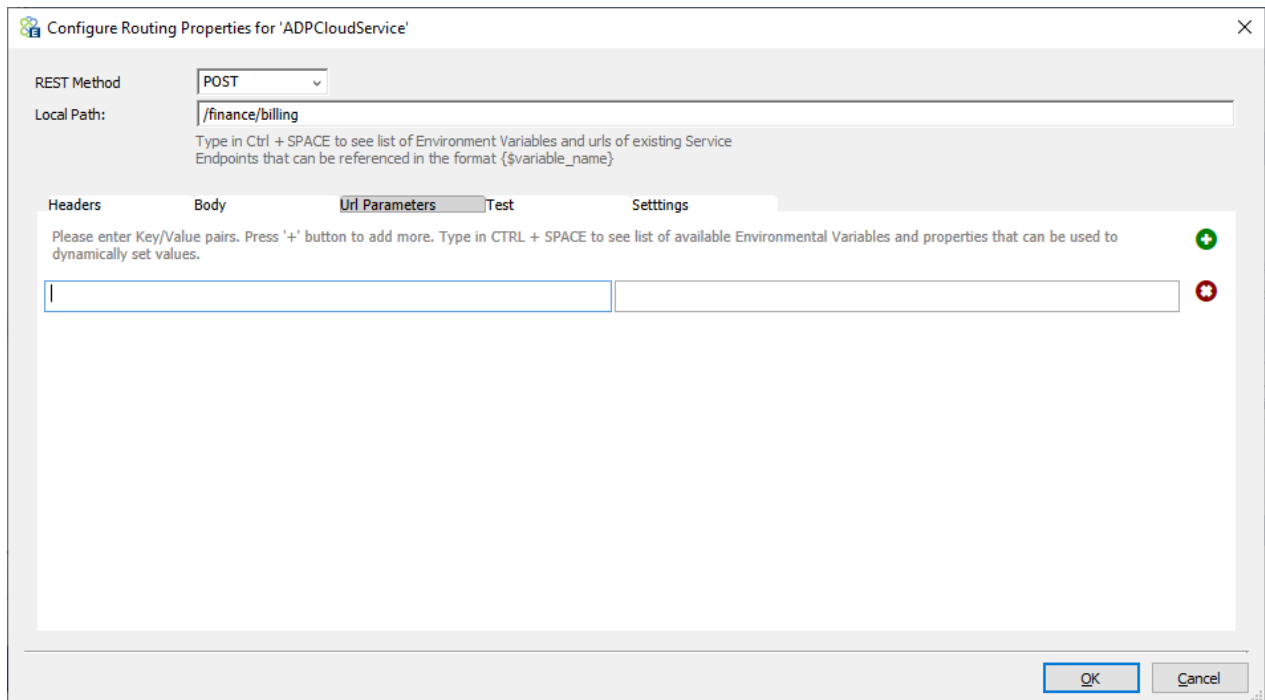
- {#HTTPHeader.<property>}
- {#HTTPQuery.<property>}
- {#HTTPLocalPath[0]}
- {#DateTime(local)}
- {#DateTime(local),ddmmYYYYhhmmss}
- {#DateTime(Utc)}
- {#DateTime(Utc),ddmmYYYYhhmmss}
- {#IMF-fixdate}
- {#UUID}

OK Cancel

Neuron ESB Explorer – Figure 20- The Configure Routing Properties Dialog Body tab. This is where the incoming body can be overwritten for the outbound request. Data maps for request/response can also be specified here.

Url Parameters (Query strings)

This is very similar to the Headers section. Key-value pairs entered here are appended to the service endpoint URL as query strings in the format “?key1=value1&key2=value2” etc. Users can use token substitution with various options to assign the value of query strings.

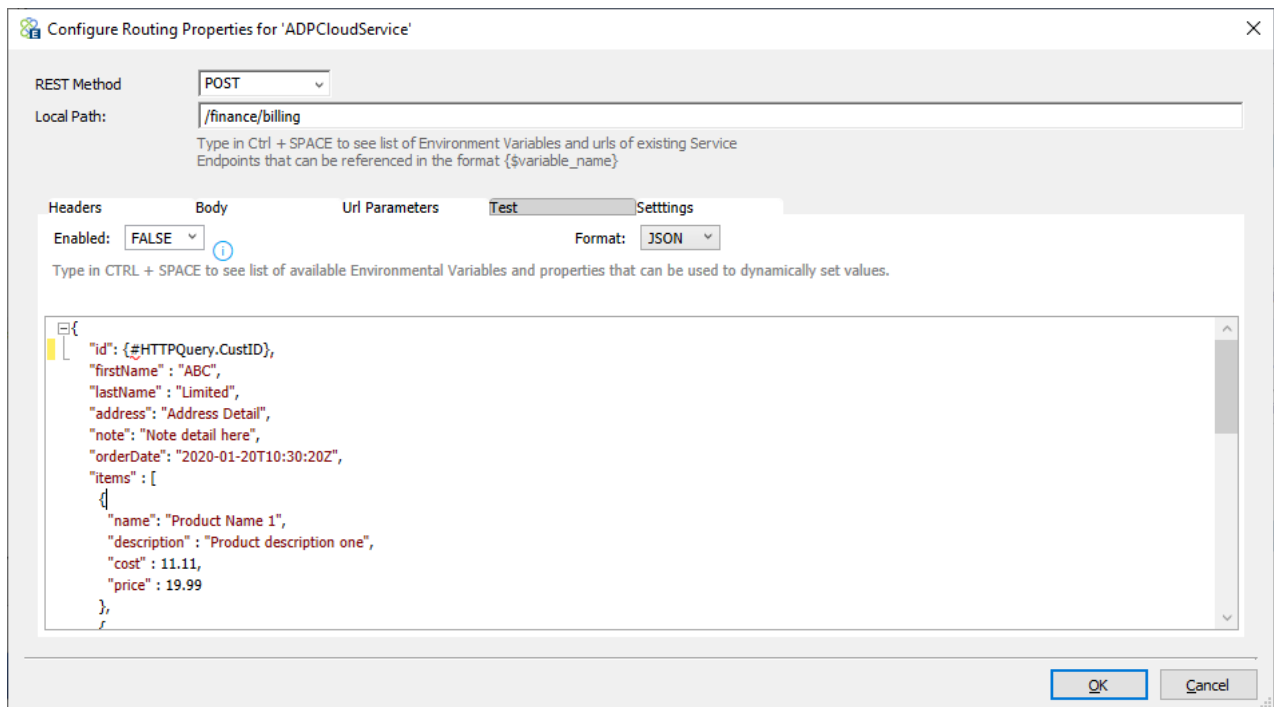


Neuron ESB Explorer – Figure 21- The Configure Routing Properties Dialog Url Parameters tab. This is where Url Parameters can be added or overwritten for the outbound request.

Test

Sometimes users need the Service Routing Table to return a canned response if the service endpoint it is going to connect to is not yet available. This can be useful if there is a need to mock out services in a specific environment where the service itself is unavailable, or a need to unit test against specific message bodies expected to be returned by the service. In that case, the Test tab can be used to define a test response to be returned instead of calling the service endpoint configured.

Users can disable or enable this feature by selecting an option from the drop-down box. However, a Neuron ESB Environment Variable can also be used to configure the Enable/Disable feature. This can be useful for instance if the Test feature should be enabled in one environment (e.g., TEST/QA) but not enabled in the Production environment. Users can use the dynamic token substitution to construct the message body.

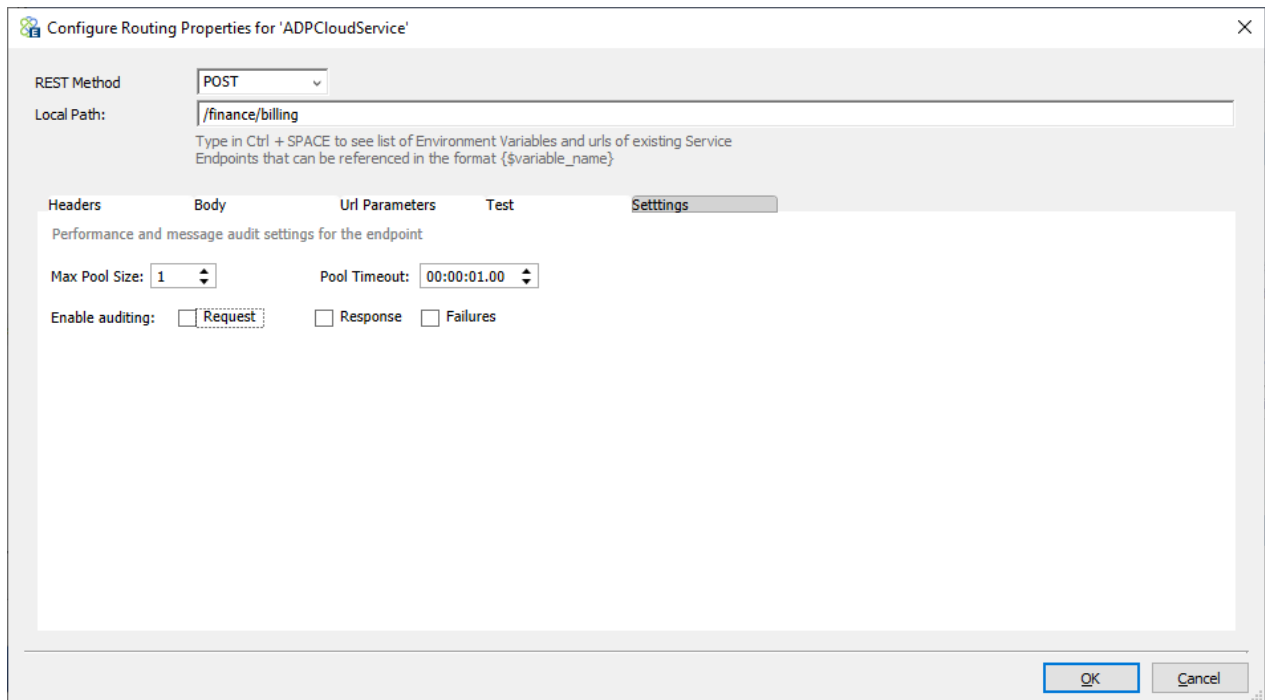


Neuron ESB Explorer – Figure 22- The Configure Routing Properties Dialog Test tab. This allows users to mock the response for service. When enabled, the underlying service to be routed to will not be called. Instead, the body within the test window will be returned as the response. The Enabled property can be configured with an Environment Variable.

Performance and Audit Settings

On the last tab of configuring the service endpoint, users can choose to Audit requests, responses, or failures during the execution of the Service Routing Table. Users can also adjust the value of pool size and timeout depending on load and performance requirements.

Creating a connection pool provides low latency because of the savings in overhead of creating the service connection. However, too many service pool connections can exhaust TCP connections on the server, so the pool size must be balanced against the maximum number of open connections. Pool timeout lets users control how long the request can wait to get a pooled connection when the server is too busy.



Neuron ESB Explorer – Figure 23- The Configure Routing Properties Dialog Settings tab. This allows users to Audit the request, response and/or failures that occur. Tuning of the auditing service can also be configured here.

Neuron ESB is a zero-code solution for many common use cases in the implementation of Service-Oriented Architecture. Neuron ESB features are distilled and tuned from years of experience and customer feedback. Service Routing Tables add brand new capabilities with a few clicks to enable low latency high-performance SOA solutions. To learn more, check out the training/tutorials and documentation at <http://PeregrineConnect.com>.

Wiretap Tracing Neuron ESB Messages

Frequently, it is necessary to capture the in-flight messages for testing and debugging purposes. In some other cases, its required for specific business activity monitoring in real-time. The existing Neuron ESB capabilities can be done by rewiring the Business Process to include the message Audit process step. Alternatively, the solution can be configured for verbose logging to capture some of these details. A third option for WCF services is to use WCF Tracing. These options are useful in development and test environments, but they are not recommended for production time tracing.

Neuron ESB version 3.7.5 adds capabilities for capturing in-flight messages into searchable Elastic Search Index when used in connection with the Peregrine Connect Management Suite. In Service Oriented Architecture, this capability is referred to as Wiretap pattern. This has been added because some customers need this capability to be turned on and off in production without modifying the configured solution and restarting the Neuron ESB runtime service. Also, auditing and tracing everything can grind performance to a halt. The Wiretap capability offers a solution to this requirement. The Wiretap Tracing can be turned on or off for specific periods of time and for specific steps/points in a Business Process or Service Route Table.

This feature is only available when a Client Connector is calling a Business Process directly or is configured with a Service Routing Table. A configuration file is created using the new Trace Configuration Utility. The configuration file specifies exactly what messages to capture and how much detail to capture for these messages.

Users may specify which Client Connectors to include for wiretapping and for how long. Along with each Client Connector, the exact points in the process or service route where to capture the messages can be specified. In addition, conditions can be added, user defined custom properties to extract from messages, schema, and sample document. The captured messages are stored in an Elastic Search Index that is setup when the Peregrine Connect Management Suite Environment for the Neuron ESB instance is configured. These messages can later be searched and analyzed using various data analysis tools. Users can even add their own extension assembly, which will receive wiretapped messages. In this assembly, users may modify or forward messages to other enterprise monitoring systems.

The following sections show how quickly tracing can be setup for one or more Client Connectors. Tracing provides many options that can be fine tune with what and how much to capture. The result of the configuration is saved into a configuration file in the current Neuron ESB Instance folder. This configuration is also referred to as the Tracking Profile.

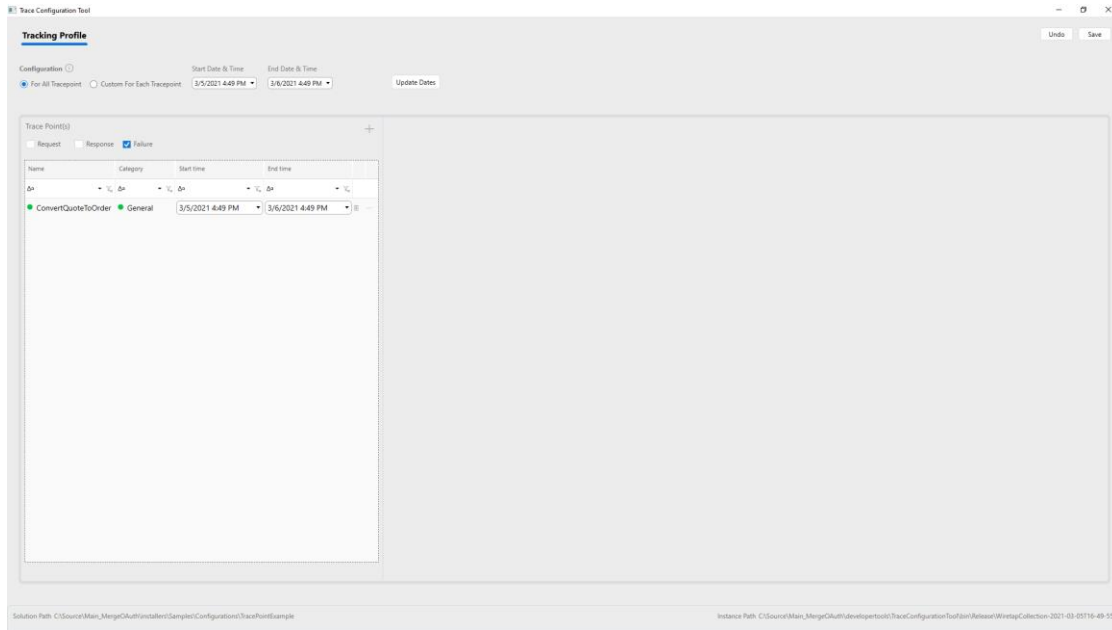
Basic Wiretap configuration

Basic Wiretap configuration to capture messages for Client Connector is straightforward. Launch the TraceConfigurationTool.exe from the Neuron ESB Instance Folder. The tool reads the currently configured solution and loads Client Connectors defined in the solution for configuration.

For basic configuration the radio button For All Tracepoints is selected under Configuration. With this setup,

Users can simply specify start date/time, end date/time and select the Client Connectors they want to trace by adding the plus button in the grid under Trace Points(s). Request, response, failure or any combination thereof can be selected.

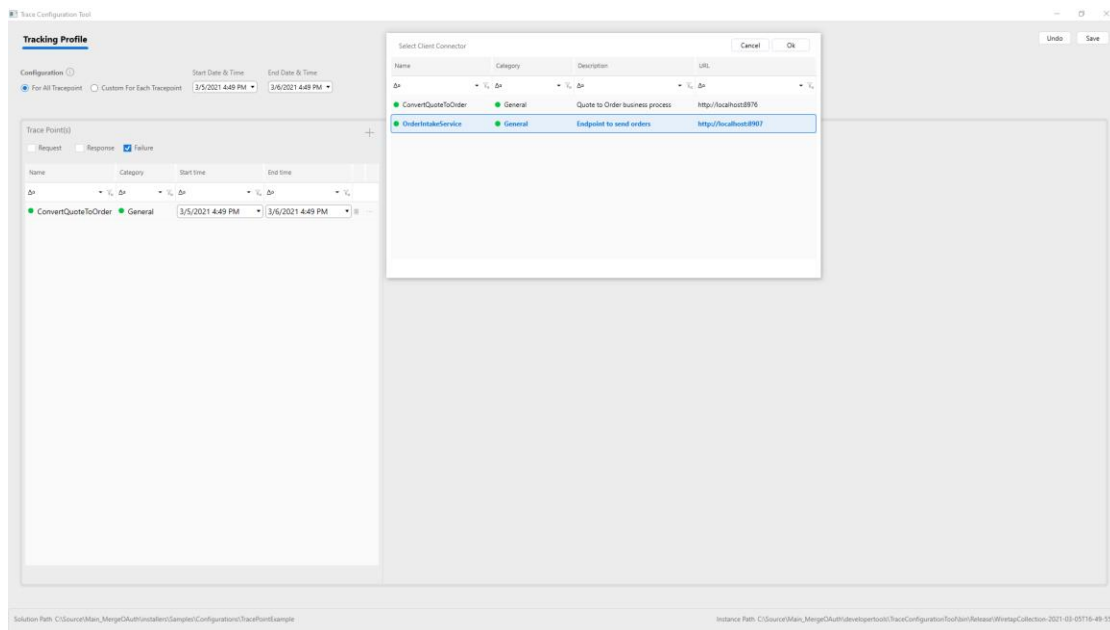
If updating an existing configuration and the start and end dates need to change, the values can be changed on the date & time boxes in the top row and click update dates to update dates on all the selected Trace Points.



Peregrine Connect Tracing Configuration Tool – Figure 24- Basic Configuration of Wiretap Tracking Profile. This tool is launched from the Instance directory of the Neuron ESB installation.

Adding a Client Connector

Highlight the Client Connector to add for Tracking and click Ok. For each client connector, select checkboxes to trace requests, response, or failures in the first part of the grid on the left-hand side of the screen below.



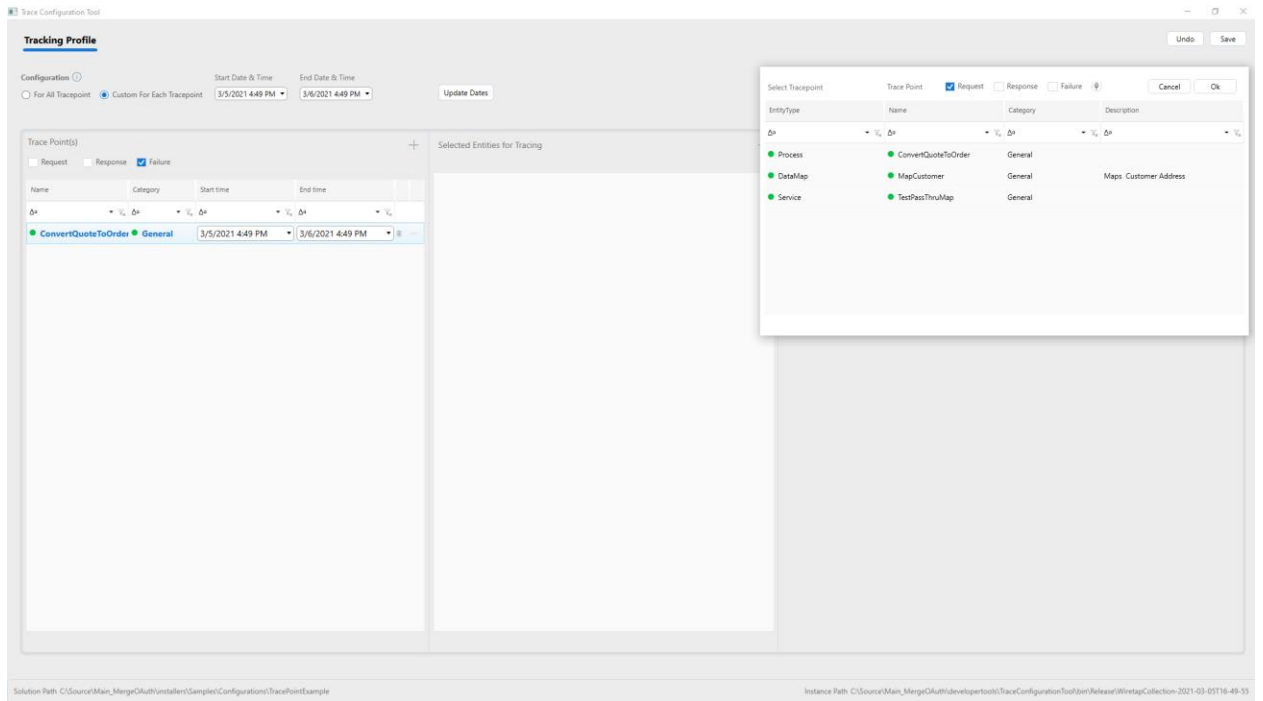
Peregrine Connect Tracing Configuration Tool – Figure 25- Selecting a Client Connector for Tracking Profile. This tool is launched from the Instance directory of the Neuron ESB installation.

Configuring Wiretap for Business Processes

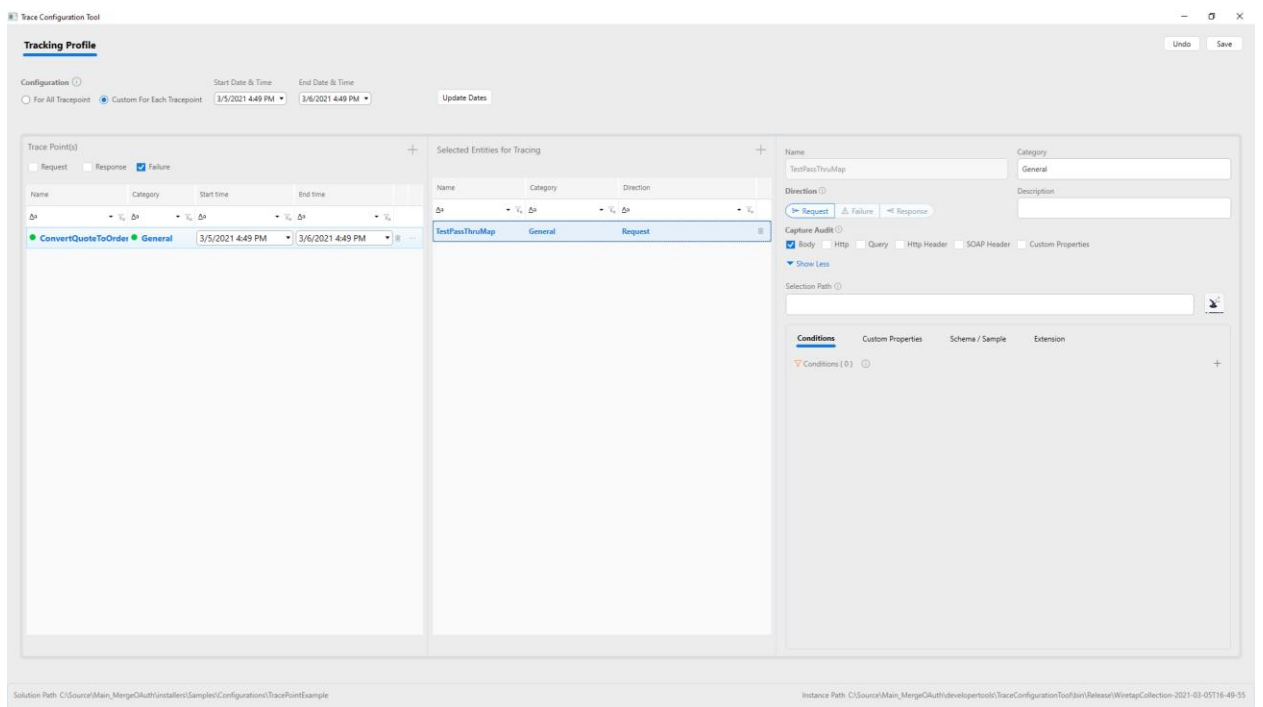
A more detailed configuration can be used to capture messages based on conditions or at specific points in the associated Business Process.

Selecting the Custom for each Tracepoint Radio button allows the entities associated with the Client Connector to be configured for message tracking. If the Client Connector is associated with a Service Routing Table, the selection is restricted to the service endpoints in the Service Routing Table. If the Client Connector is associated with a Business Process, the selection shows all the major Process steps in the Business Process. The major Process steps include the following:

- Adapter Endpoint
- Service Endpoint
- Data Mapper
- XSLT Transform
- Execute Process



Peregrine Connect Tracing Configuration Tool – Figure 26- Selecting Process Steps in a Business Process for Tracking Profile. This tool is launched from the Instance directory of the Neuron ESB installation.

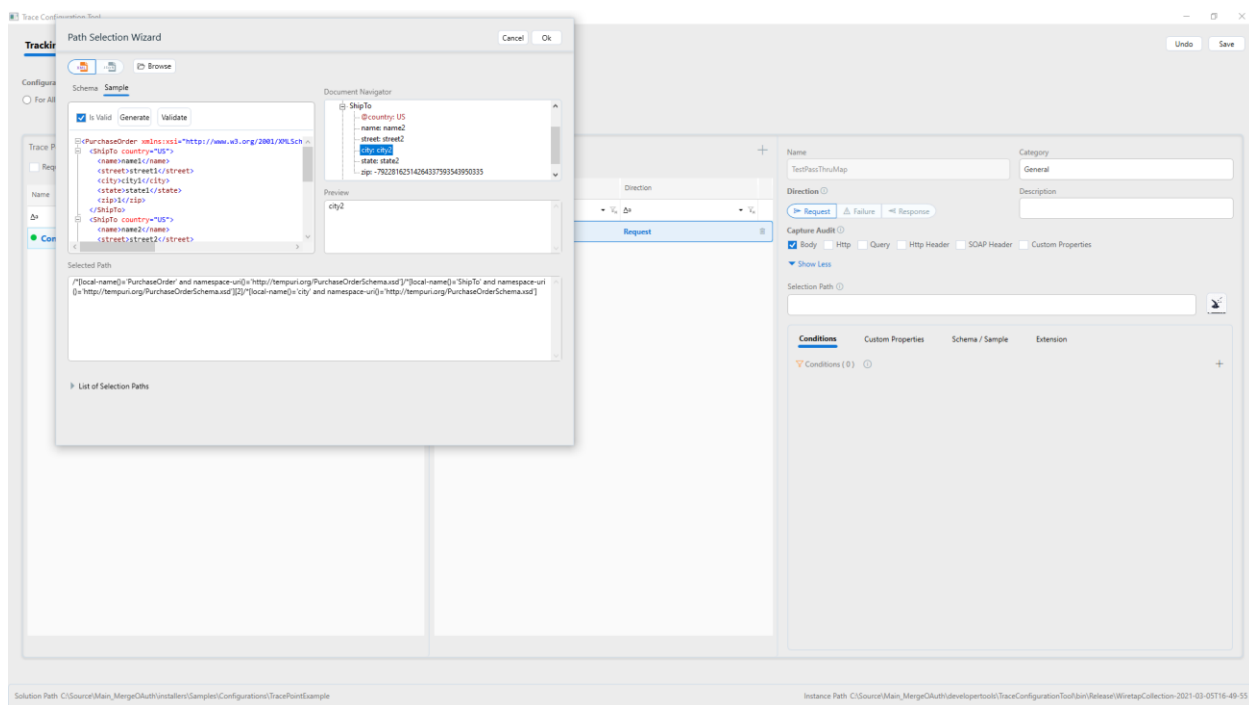


Peregrine Connect Tracing Configuration Tool – Figure 27- Configuring details for the Tracking Profile. This tool is launched from the Instance directory of the Neuron ESB installation.

Each Process step can be configured whether to capture messages from Request, Response or Failure. If needed all of them can be captured by adding multiple entries for each Process step and change the Direction radio button to Request, Response, or Failure. What to capture can be specified using the checkboxes to indicate capturing body, http headers, query strings, etc. Capture only the minimum needed so as to control the size of the indexes and save storage space.

Customize what to capture

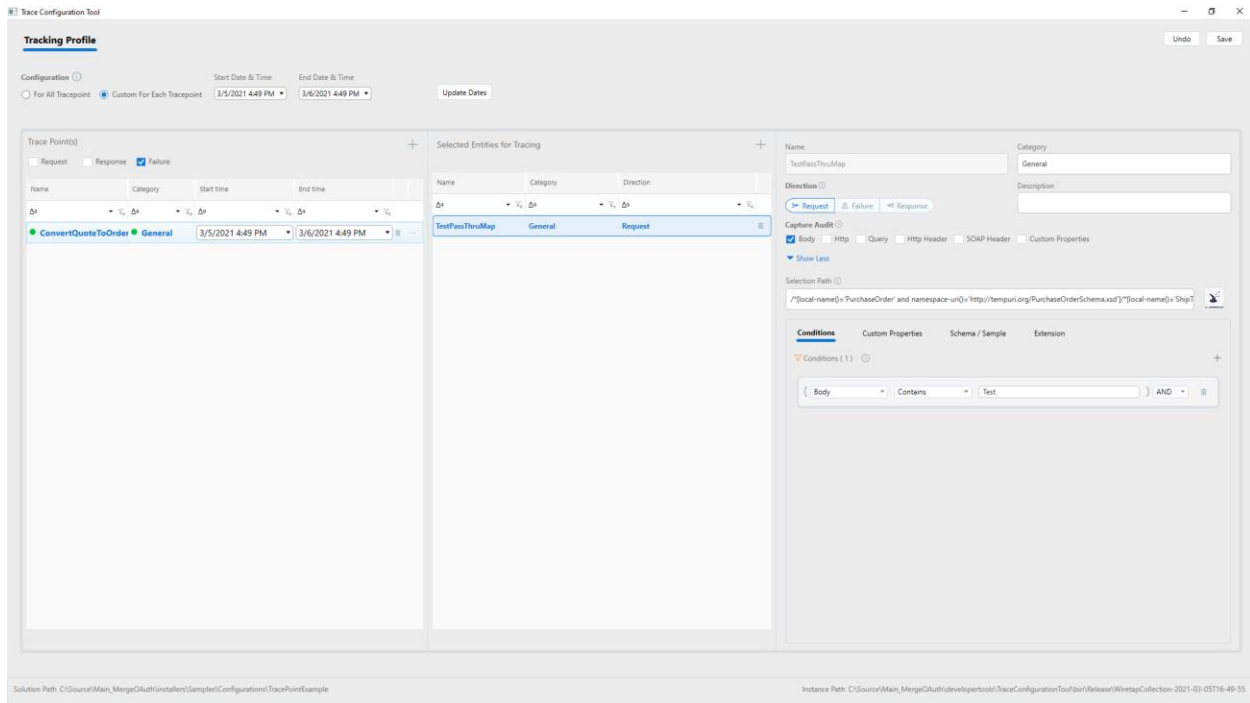
When body is selected, a further selection path can be provided from the message using XPath or JSON. Only parts of the message body that are selected by the XPath or JSON Path specified. To aid in creating the selection path, a tool is provided for visually selecting the XPath or JSON Path from sample message/schema.



Peregrine Connect Tracing Configuration Tool – Figure 28- Configuring details for the Tracking Profile Steps – Selection Path for the Body. This tool is launched from the Instance directory of the Neuron ESB installation.

Add conditions for tracking

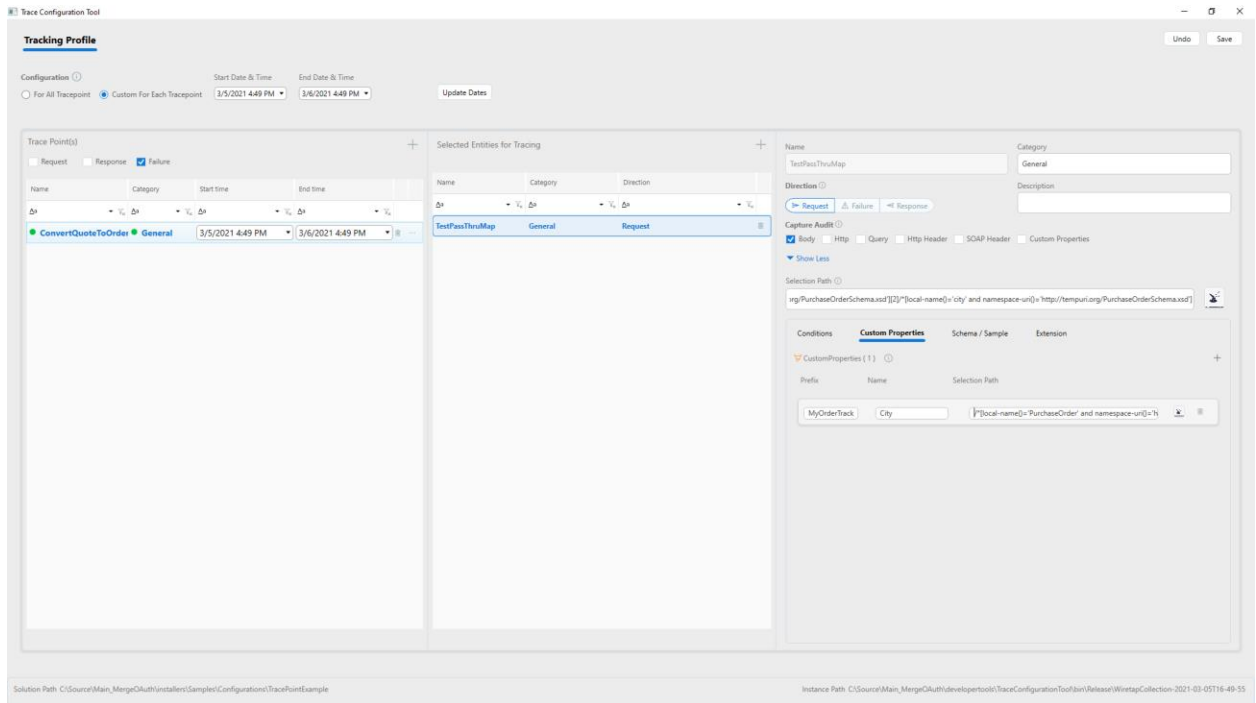
Using the Conditions tab under the expanded details of service endpoint for tracking, one or more conditions can be specified. Only messages that satisfy the condition will be stored in the tracking index.



Peregrine Connect Tracing Configuration Tool – Figure 29- Configuring details for the Tracking Profile Steps – Conditions for capturing. This tool is launched from the Instance directory of the Neuron ESB installation.

Add custom specific properties

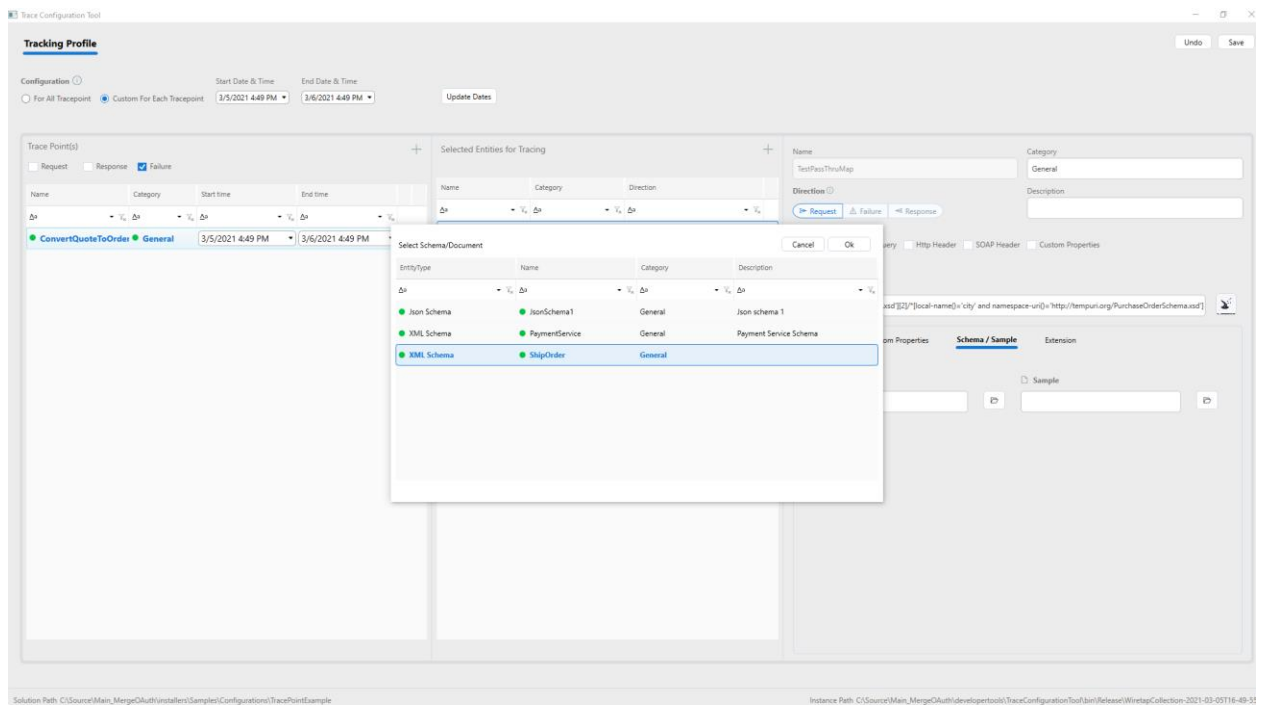
Using Custom Properties allows the capture of specific items into its own entry in the tracking index. Using Custom Properties this way makes it easy to analyze data out of the index searches.



Peregrine Connect Tracing Configuration Tool – Figure 30- Configuring details for the Tracking Profile Steps – Custom Properties to capture. This tool is launched from the Instance directory of the Neuron ESB installation.

Associate schema and sample document

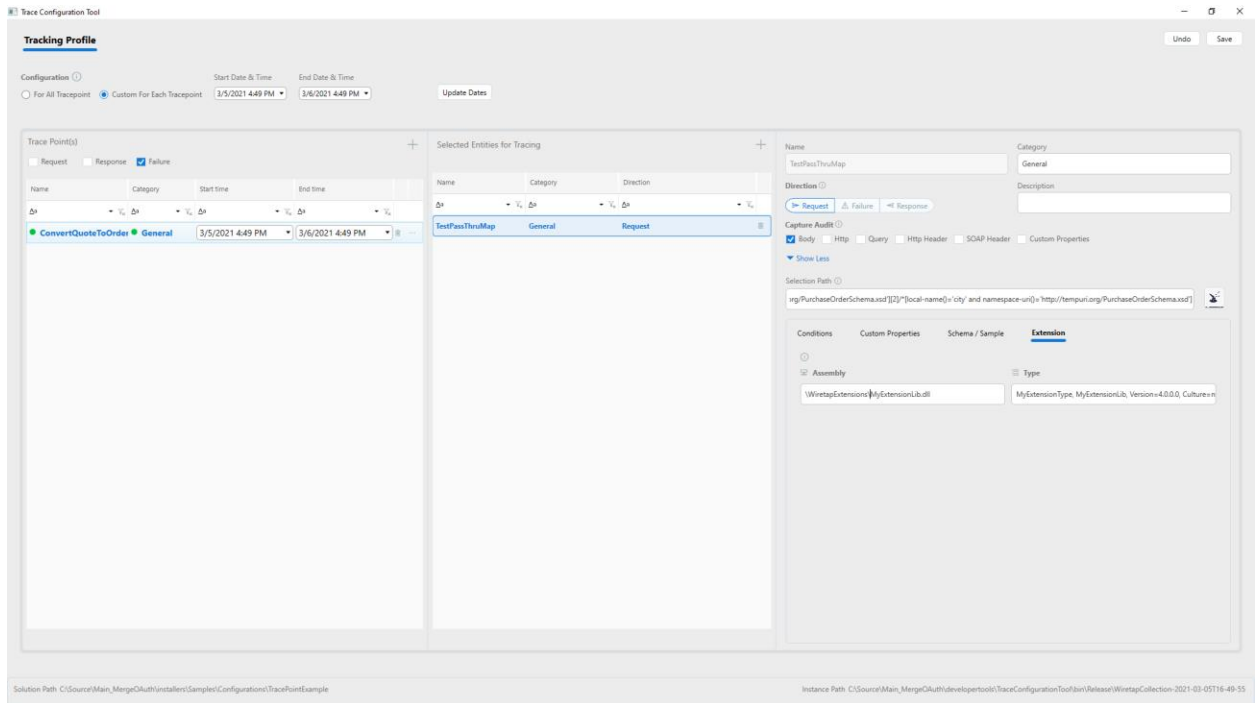
When a schema and sample document is associated with the captured message, the unstructured data in the message body can be used along with the schema by using advanced analysis tools like Kibana and Logstash. An additional extension for Business Activity Monitoring is coming soon, and it will utilize this information to provide visual data analysis.



Peregrine Connect Tracing Configuration Tool – Figure 31- Configuring details for the Tracking Profile Steps – Associate Schema/Document. This tool is launched from the Instance directory of the Neuron ESB installation.

Wire up your custom extension

A custom extension can be added to capture and modify the message that is logged. This is useful for compliance scenarios. The captured messages will be run through the extension before being logged. This way the extension can scrub or mask the private information if desired (e.g., credit card numbers, social security numbers, etc.). Refer to Wiretap Extension whitepaper for more details.



Peregrine Connect Tracing Configuration Tool – Figure 32- Configuring details for the Tracking Profile Steps – Custom Extension. This tool is launched from the Instance directory of the Neuron ESB installation.

Configuring Wiretap for Service Routing Tables

For Service Routing Tables, the configuration is similar except that capture requests/responses or failures can be captured for each service endpoint in the Service Routing Table associated with the selected Client Connector.

Editing Wiretap configuration

When opening the TraceConfiguration Utility from the instance folder, it opens the configured wiretap configuration. The dates may be changed, Client Connectors may be added/removed, and configuration properties can be changed and save. If starting from scratch, find and delete the Trace configuration file in the instance folder. By default, this file is located under the wiretapcollection folder. The exact location is based on the setting in appsettings.config file in the instance folder.

After that, when you TraceConfiguration Utility, a new configuration is created.

Either the Endpoint Host or the Client Connectors previously configured will have to be restarted after any changes to the configuration.

Enable / Disable Wiretap

To disable or enable this functionality, either change the start and end dates on each Client Connector or delete the configuration file. Either the Endpoint Host or the Client Connectors previously configured will have to be restarted after any changes to the configuration.

Wiretap load considerations

The Wiretap feature should be used with a limited time and for selected number of Client Connectors as needed. Since processing all the tracked messages consumes processor, memory, bandwidth, the load, message size, and latency requirements must be considered when using this method of tracing messages. Do not use this method for auditing messages regularly. For that purpose, use the Audit process step or the Audit functionality on the Service Routing Table settings.

Viewing Captured Wiretap information

The information captured using the Wiretap feature is stored into Elastic Search Indexes named

DeploymentGroup_Application_Machine_Instance_tracepointmessagewrapper_mmddyyyy

You can use a tool such as Elastic Search Head to query these indexes, as shown in the screens below. Elastic Search Head is a Google Chrome extension that you can download and point to your Elastic Search Server.

The screenshot displays the Elastic Search Head interface in a web browser. The top navigation bar includes the Elasticsearch logo, a search bar, and a 'Connect' button. Below the navigation bar, there's a 'Browse' section with a list of indices. The 'Indices' section is expanded, showing a list of indices with their respective scores and fireinstanceid values. A specific index, 'dev_features3_7_5_hzinzuwadia04_default_tracepointmessagewrapper_04042021', is selected, and its details are shown in the main panel. The details include the index name, type, id, and a list of fields such as 'source', 'type', 'id', 'score', and 'fireinstanceid'. The 'source' field is expanded, showing a JSON object with various fields like 'type', 'id', 'score', 'fireinstanceid', 'ApplicationName', 'Instance', 'Machine', 'SchedulerName', 'SchedulerInstanceId', 'Job', 'Trigger', 'ScheduledFireTimeUtc', 'ActualFireTimeUtc', 'Recovering', 'Status', 'Duration', 'ErrorMessage', 'Job key', 'MessageId', 'TransactionId', 'RequestMessage', 'ResponseMessage', 'JobType', and 'JobDataMap'.

Elastic Search Head – Figure 33- Viewing captured messages using Elastic Search Head.

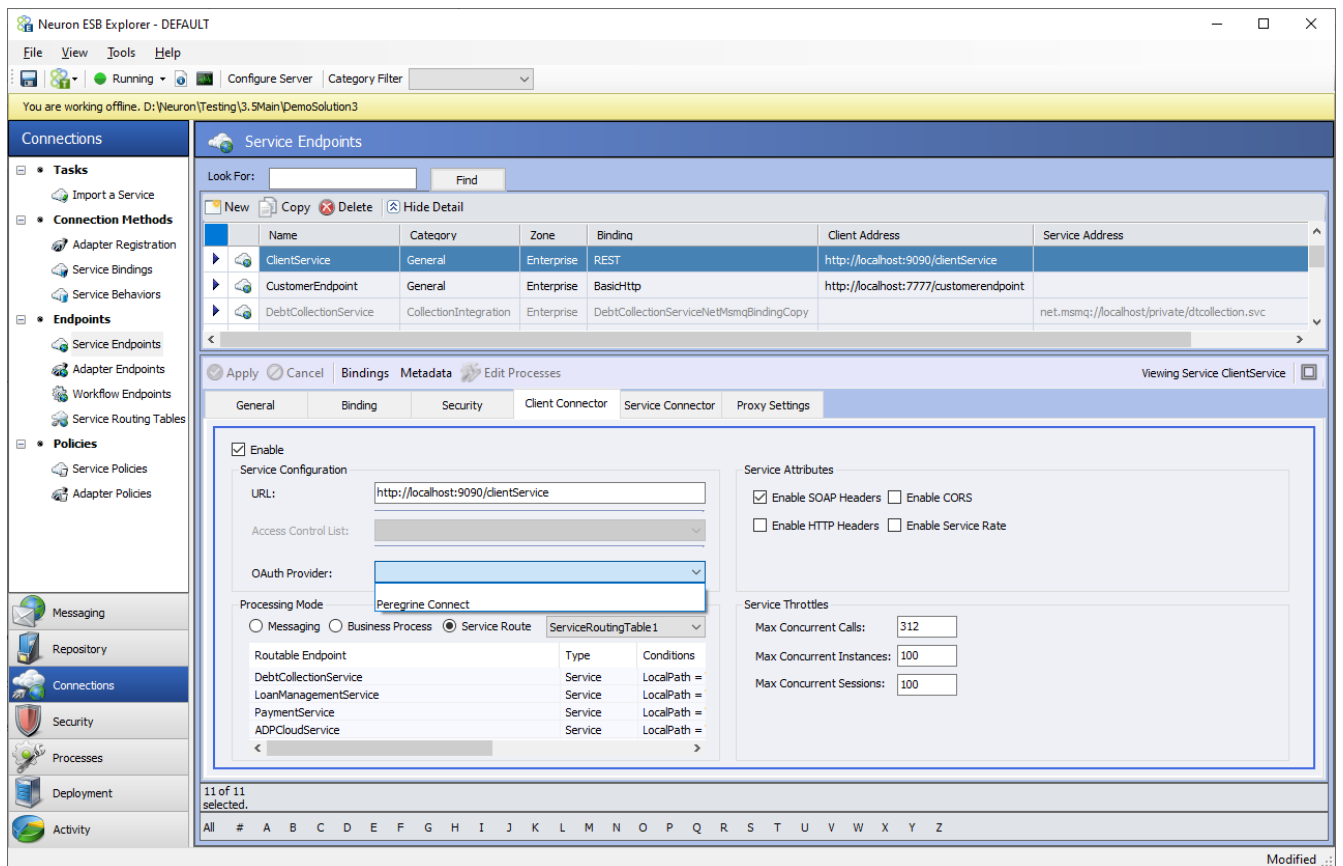
Client-Side OAuth Providers

Previously, securing access to Neuron Client Connectors relied on the options offered by WCF, such as Basic, NTLM, Windows, Digest, and Certificate based authentication. Modern REST services use new standards called Open ID connect and OAuth for access control and authorization. These standards are cross-platform and cross-application standards providing great interoperability in both Intranet and Extranet environments.

Now the Neuron ESB client connectors with REST binding have an option to secure them with an OAuth provider. The OAuth providers can be any of the popular identity providers such as Azure, Amazon, Google, Okta, Thinktecture identity server, etc.

When a Client Connector is secured with OAuth, every caller needs to obtain a token from the Identity Provider and include that in the Authorization Header in the format Bearer <token>. The Neuron ESB runtime validates this token and makes the token as well as Application Id and Consumer Id available in the message context as custom HTTP headers. These properties can then be used in Neuron ESB Business Process to make decisions based on them.

If you are using Peregrine Connect Management Suite, the built-in Peregrine Connect OAuth Provider can be used. If using this provider, the only thing that needs to be configured in the Neuron ESB Explorer solution is the OAuth Provider property on the Client Connector tab. This property should be set to “Peregrine Connect” as shown below. After that, the rest of the OAuth configuration is done within the Peregrine Connect Management Suite. The details for that setup can be found in the API management section of [Peregrine Connect Management Suite 1.1](#).



Neuron ESB Explorer – Figure 34- A Client Connector’s OAuth Property drop-down box. By default, populated with “Peregrine Connect” OAuth Provider. If other providers are configured within the Security section of the Neuron ESB Explorer, they will be displayed here.

The following sections detail how an external OAuth provider other than Peregrine Connect can be provisioned.

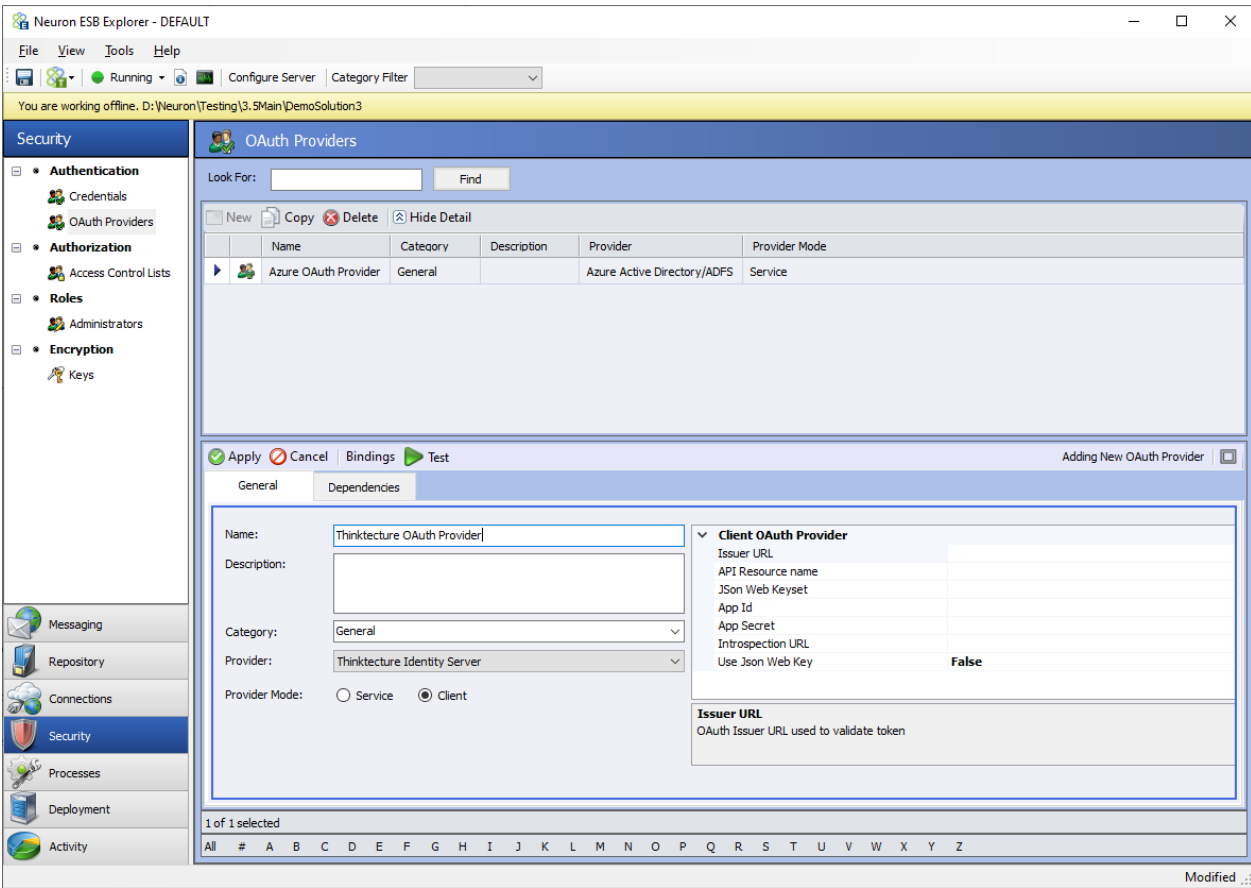
Adding or Editing an External OAuth Provider

An external client mode OAuth Provider must be added before it can be used to secure a Client Connector. This can be done by navigating to the Security section of the Neuron ESB Explorer and then selecting the OAuth Providers menu under the Authentication section. The screen below displays the fields and options for a selected client mode OAuth provider. NOTE: service mode OAuth Providers has been available since version 3.6 of Neuron ESB. The documentation for service mode OAuth providers can be found here:

<https://www.peregrineconnect.com/article/introducing-oauth-support/>

In the screen below, start with changing the radio button for Provider mode to client and then select the provider from the drop-down box as shown. There are three options for Azure, AWS, and Thinktecture Identity Server respectively.

Azure and AWS are special because they add some tweaks to the OAuth implementation. In case of Azure there is an additional policy configuration for Business to Consumer websites. In the case of AWS, there is no provision for Introspection Endpoint. The Thinktecture Identity Server provider is the most generic implementation, and it works with many other providers such as Okta, Google, etc., including the on-premise implementation of the open-source Thinktecture Identity Server.



Neuron ESB Explorer – Figure 35- The configuration properties for the client mode Thinktecture Identity Server OAuth Provider are displayed in the right-hand side grid.

A name, description, and category can be assigned to the external OAuth provider like all other Neuron ESB artifacts. The remainder of the configuration properties for the providers can be found in the grid below.

Field Name	Description	Applies To
Issuer URL	This is the URL that will identify your unique provider in Azure, AWS or any other Generic cloud base identity server provider and	All providers

the Identity Server Host in case of On-premise server.

API Resource Name	This is the unique name assigned to one or more client connectors group that you are secure with the same scheme. Also known as application name.	All providers
JSON Web Keyset	Set of public keys to be used for decrypting encrypted messages by the OAuth server with their private key. This value can be generally downloaded from the well-known endpoint of the Issuer URL. When you click ... the button next to the field value, you can open the dialog box to copy this value from. Example screen shot is shown below.	All providers
Policy	In the case of Azure Active Directory B2C this is an additional setting to identify your subscription.	Only Azure
App Id	App Id and App Secret are used to validate token using the Introspection method. Consult Identity Server documentation for details.	Only Thinktecture Identity Server
App Secret		Only Thinktecture Identity Server
Introspection URL	This is the URL used by Neuron Runtime to validate the token that is presented in the Authorization Header.	Only Thinktecture Identity Server
Use JSON Web Key	When set to true, Neuron Runtime will not validate using Introspection URL but rather use the JSON Web key to validate	Only Thinktecture Identity Server

token. Works in scenarios where
Neuron runtime doesn't have
Internet Access.

Edit OAuth Config

File

Edit

Discovery URL (from Issuer URL)

https://login.microsoftonline.com/687f51c3-0c5d-4905-84f8-97c683a5b9d1/v2.0/.well-known/openid-configu

Jwks URL (from Discovery URL)

https://login.microsoftonline.com/687f51c3-0c5d-4905-84f8-97c683a5b9d1/discovery/v2.0/keys

Enter the public Key as JWKS value downloaded from discovery URL

[{"kty": "RSA", "use": "sig", "kid": "huN95lvPfehQ34GzBDZ1GXGimM", "x5t": "huN95lvPfehQ34GzBDZ1GXGimM", "n": "6lIdKm5Rc_vMKa1RM
TtUv3mtj52wLRrJqu13yGM3_h0dwru2ZP53y65wDfz6_tLQjoYuRCuVsjoW37-0zXUORJvZ0L90CAX-
58lW7NcE4bAzA1pXv7oR9kQw0X8dp0atU4HnHeaTU8LZxcjJO79_H9cxgwa-
clKfGxllcos8TsuurM8xi2dx5VqwwqNMB2s62l3MTN7AzctHUiQCIX2jArGjAhs-mxS1wmyMlyOSipdodhjQWRAcseW-
aFVyRTFVi8okl2cT1HjPXd0b1WqYSOzeRdrLUcA0oR2Tzp7xzOYJZSGNnNLQqa9f6h6h52XbX0iAgxKgEDIRpbJw", "e": "AQAB", "x5c":
["MIIDBTCCAe2gAwIBAgIQPCxPbySVSLZOggeWRzBWQjANBgkqhkiG9w0BAQsFADAtMSswKQYDVQQDEyJhY2NvdW50cy5hY2Nlc3Njb
250cm9sLndpbmRvd3MubmV0MBA4XD01MDYwNzAwMDAwMFoXD01MDYwNzAwMDAwMFowLTERMCKGA1UEAxMiYWYWNjb3VudHMuY
WNjZXNzY29udHJvbm5kb3dzLn5ldDCCASlwdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAOpZXSpUUXP7zCmtUTP07VL97Z
rY+dsC0ayard8hjN/4dHcK7mT+d8uucA38+v7Swo6GLkQribI6ft
+/tM11DkSb2dC/dAgF/ufjVuzXBOGwMwNaV7+6EfZEMNF/HadGrVOB5x3mk1PC2cXlyTu/fx/XMYMGvnJSnxsZZXKLPE7LrqzPMYtnceVa
sM6jTAdrtOpdzEzewM3LR1IkAol9oiQKxowIbPpsUtcJsjCMjkoqXaHYy0FkQHLHvmhVckUxYvKJjdnE9RyYz13cdG9VqmEjs3kXa6y1HANK
Edk86e8czmCWUUhjZzS0KmvX+oeoedl219lMSoBA5UaWycCAwEAAAMhMB8wHQYDVR0OBBYEFFXP00DFhf3RS6oRijM5Tb
+yB8CMA0GCSqGSIb3DQEBCwUAA4IBAQB9GtVikLTbJWlu5x9YCUTTzKzNhi44XXogP/v8VylRSUHI5YTMdnWwvDlt/Y1sjNonmSy9PrioEjc
li11U8nicveafMwIq5Vln
+gEY2lg6KDJAgAvA88CXqwfHHvtmYBovN7goolp8TY/kddMTf6TpNzN3ICTM2MK4Ye5xLLVGdp4bqWCOJ/qjwDxpTRSydYlkLUDwqNjv
+sYfOEJpYAB4rTL/aw3ChJtiaA4MtXEt6OjbUtbOa21IShLzvNRbYK3+ukbrhmRI9lemJEeUls51vPule+jg
+Ssp43aw7PQjxt4/MpfNMS2BfZ5F8GVSVG7qNb352cLLeJg5rc398Z"], "issuer": "https://login.microsoftonline.com/687f51c3-0c5d-4905-

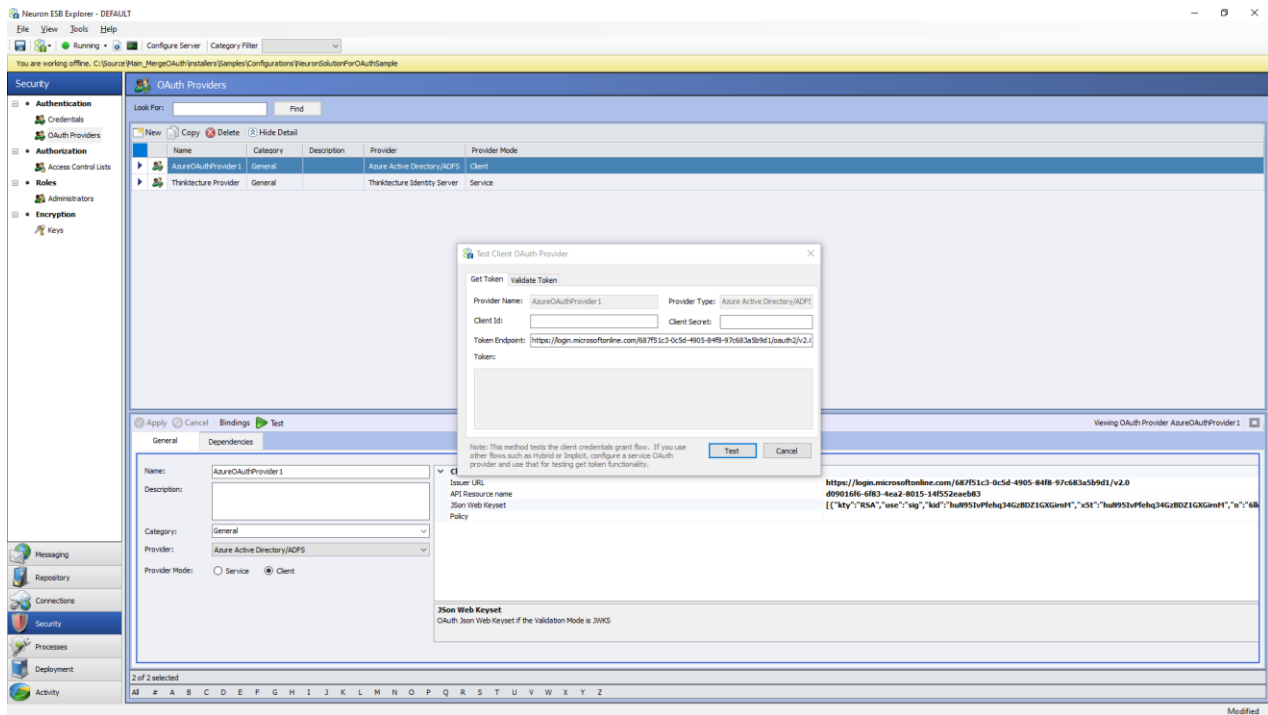
OK

Cancel

Neuron ESB Explorer – Figure 36- The JSON Web Keyset Dialog. This is launched when editing the JSON Web Keyset property of a client mode OAuth Provider.

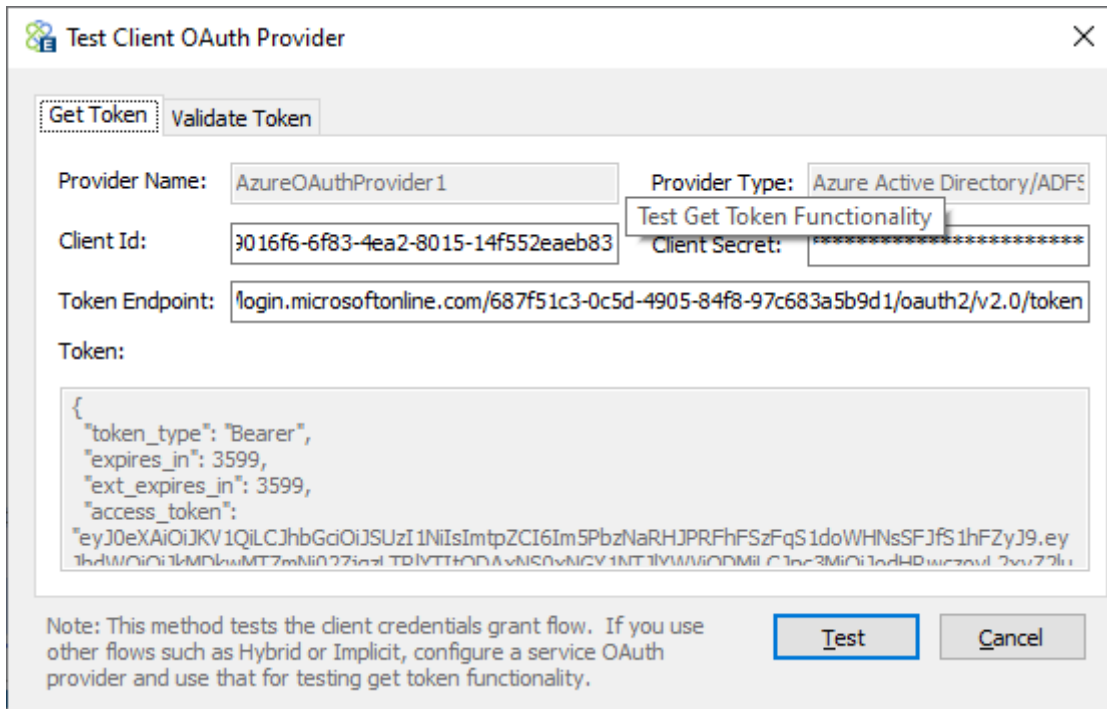
Testing External OAuth Provider

The external OAuth provider can be tested by using the Test button, as shown below.



Neuron ESB Explorer – Figure 37- Client mode OAuth Test Dialog. This is launched when clicking the Test button. An OAuth Token can be retrieved and validated.

On the “Get Token” tab, a client id, client secret, and token endpoint from the provider must be entered so that a token can be issued and returned using the first part of the above screen and clicking the Test button. Once the token is returned, it will be visible in the Token property window shown below.



The dialog box is titled "Test Client OAuth Provider" and has a close button (X) in the top right corner. It contains two tabs: "Get Token" (selected) and "Validate Token".

Fields in the "Get Token" tab:

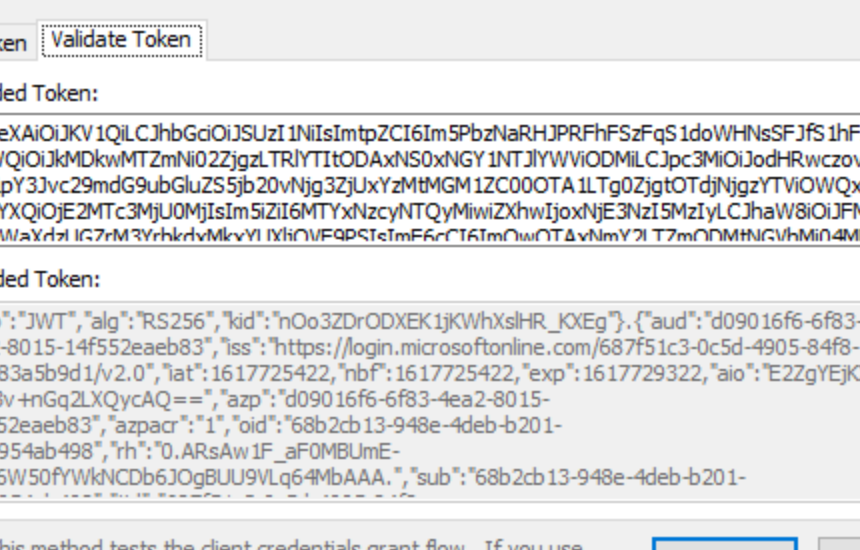
- Provider Name: AzureOAuthProvider1
- Provider Type: Azure Active Directory/ADFS
- Client Id: 3016f6-6f83-4ea2-8015-14f552eae83
- Client Secret: Test Get Token Functionality
- Token Endpoint: /login.microsoftonline.com/687f51c3-0c5d-4905-84f8-97c683a5b9d1/oauth2/v2.0/token
- Token: A JSON object containing token details.

Note: This method tests the client credentials grant flow. If you use other flows such as Hybrid or Implicit, configure a service OAuth provider and use that for testing get token functionality.

Buttons: Test, Cancel

Neuron ESB Explorer – Figure 38- Client mode OAuth Test Dialog. This is launched when clicking the Test button. An OAuth Token can be retrieved and validated. The Get Token Tab is where the Token is retrieved to when clicking the Test button.

To validate the token, copy the token’s text from the Token property window, navigate to the “Validate Token” tab, and paste the contents of the token into the “Encoded Token” text box, and the decoded text of the token will appear in the “Decoded Token” text box below it.



Test Client OAuth Provider

Get Token Validate Token

Encoded Token:

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Im5PbzNaRHJPRHFsZSZeFqS1doWHNsSFJfs1hFZyJ9.eyJhdWQiOiJkdMDkwMTZmN022jgzLTRYiTiODAxNS0xNGY1NTJlYWViODMiLCJpc3MiOiJodHRwczovL2xvZ2l1Lm1pY3Jvc29mdG9ubGluZS55b20vNjg3ZjUxYzMtMGM1ZC00OTA1LTg0ZjgtOTdjNjgzYTViOWQxL3YyLjAiLCJpYXQiOiJ2MTc3MjU0MjIsIm5iZiI6MTYxNzcyNTQyMiwiZmxwIjoxNjE3NzI1MzYlLCJhaW8iOiJFMlplbnVUWVns1hWAXdZlIG7rM3YrhkdvMkvYlIXhQVFP9PStsImF66rCI6ImQwOT4vNmY2lT7mODMhNGVhMiN04MDF1lTFN
```

Decoded Token:

```
{\"typ\": \"JWT\", \"alg\": \"RS256\", \"kid\": \"nOo3ZDrODXEK1jKWhXslHR_KXEg\"}, {\"aud\": \"d09016f6-6f83-4ea2-8015-14f552eae83\", \"iss\": \"https://login.microsoftonline.com/687f51c3-0c5d-4905-84f8-97c683a5b9d1/v2.0\", \"iat\": 1617725422, \"nbf\": 1617725422, \"exp\": 1617729322, \"aio\": \"E2ZgYejKXVw sPfk3v+nGq2LXQycAQ=\", \"azp\": \"d09016f6-6f83-4ea2-8015-14f552eae83\", \"azpacr\": \"1\", \"oid\": \"68b2cb13-948e-4deb-b201-c03a954ab498\", \"rh\": \"0.ARsAw1F_aF0MBUmE-JfGg6W50fYwKNCDb6JOGBUU9VLq64MbAAA.\", \"sub\": \"68b2cb13-948e-4deb-b201-c03a954ab498\"}
```

Note: This method tests the client credentials grant flow. If you use other flows such as Hybrid or Implicit, configure a service OAuth provider and use that for testing get token functionality.

Test Cancel

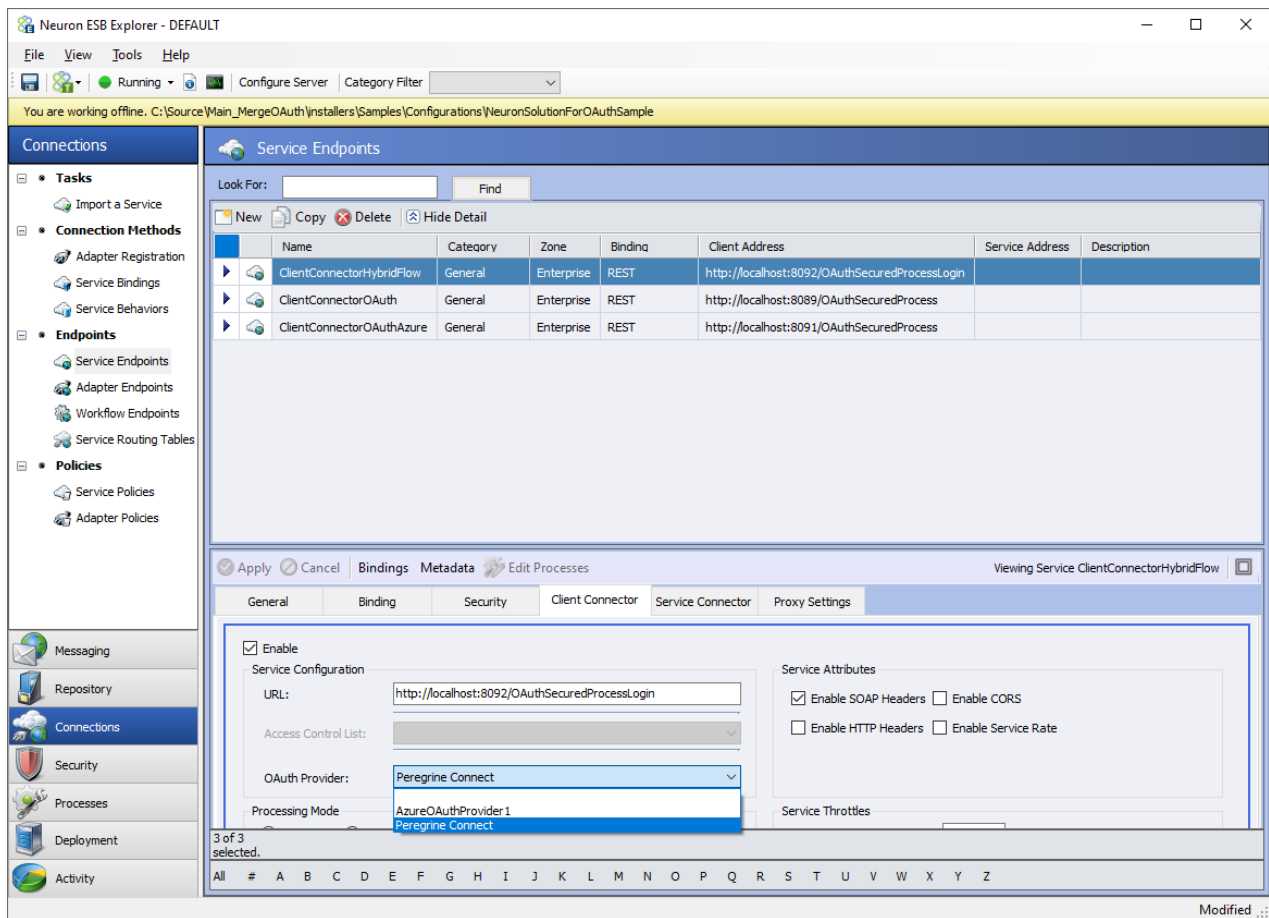
Using OAuth Provider with Client Connector

Once a client mode OAuth Provider has been configured it can be assigned to a Client Connector. Having a REST binding with Security Model set to “None” or “Transport None” is a pre-condition to use an OAuth provider with a Client Connector.

Assigning OAuth Provider

The following screen displays the Client Connector portion of the Service Endpoint Screen to select the OAuth provider. The OAuth provider box will show all the configured OAuth providers and built-in Peregrine Connect OAuth Provider. Do not select the Peregrine Connect OAuth provider if the Peregrine Connect Management Suite is not being used.

Select the OAuth provider, apply and save the Service Endpoint.



Neuron ESB Explorer – Figure 40- Client mode OAuth Provider being selected for a Client Connector.

Removing or changing OAuth Provider

To remove the OAuth provider associated with the Client Connector, change the drop-down box to no selection select another OAuth provider. Remember if the OAuth provider is changed, the consumers of Client Connectors will have to change the way they obtain their tokens.

Using Tokens in Business Process

Neuron ESB runtime places three new headers in the HTTP Headers of the ESBMessage that is passed on into the business process. They are Token, ConsumerId, and OAuthAppId.

Read Token contents

The structure of the token may vary slightly based on the provider. Test your implementation to get the exact structure. The following shows a typical OAuth token.

```
{
  "nbf": 1617456481,
  "exp": 1648560481,
```

```
"iss": "http://hzinzuwadia04:8080/authsvc",  
"aud": "Portal",  
"client_id": "clientforesbservices",  
"jti": "C18704D59E3C6D4F2E9A0346B9ADC12A",  
"iat": 1617456481,  
"scope": [  
  "Portal"  
]  
}
```

Implement Logic based on Scopes

The values of the token contents, Consumer Id, and App Id can then be used in Business Process steps to check their contents and make decisions based on the Token Contents, Consumer Id, or App Id. In addition, the Token contains scopes and audience. These properties can be checked to restrict certain steps of the Business Process or filter the results based on some security rule.

Client OAuth Samples

To make Client-Side OAuth easier to understand, Neuron ESB Ships with a new sample demonstrates the ability to use Client-Side OAuth Providers. The sample is comprehensive in that it walks the user through configuration and setup as well as accessing the Token information within a Business Process. The sample can be accessed through the Sample Browser of the Neuron ESB Explorer. Its documentation can be found here:

<https://www.peregrineconnect.com/documentation-kb/securing-client-connectors-with-oauth-sample/>

Business Process Scheduling

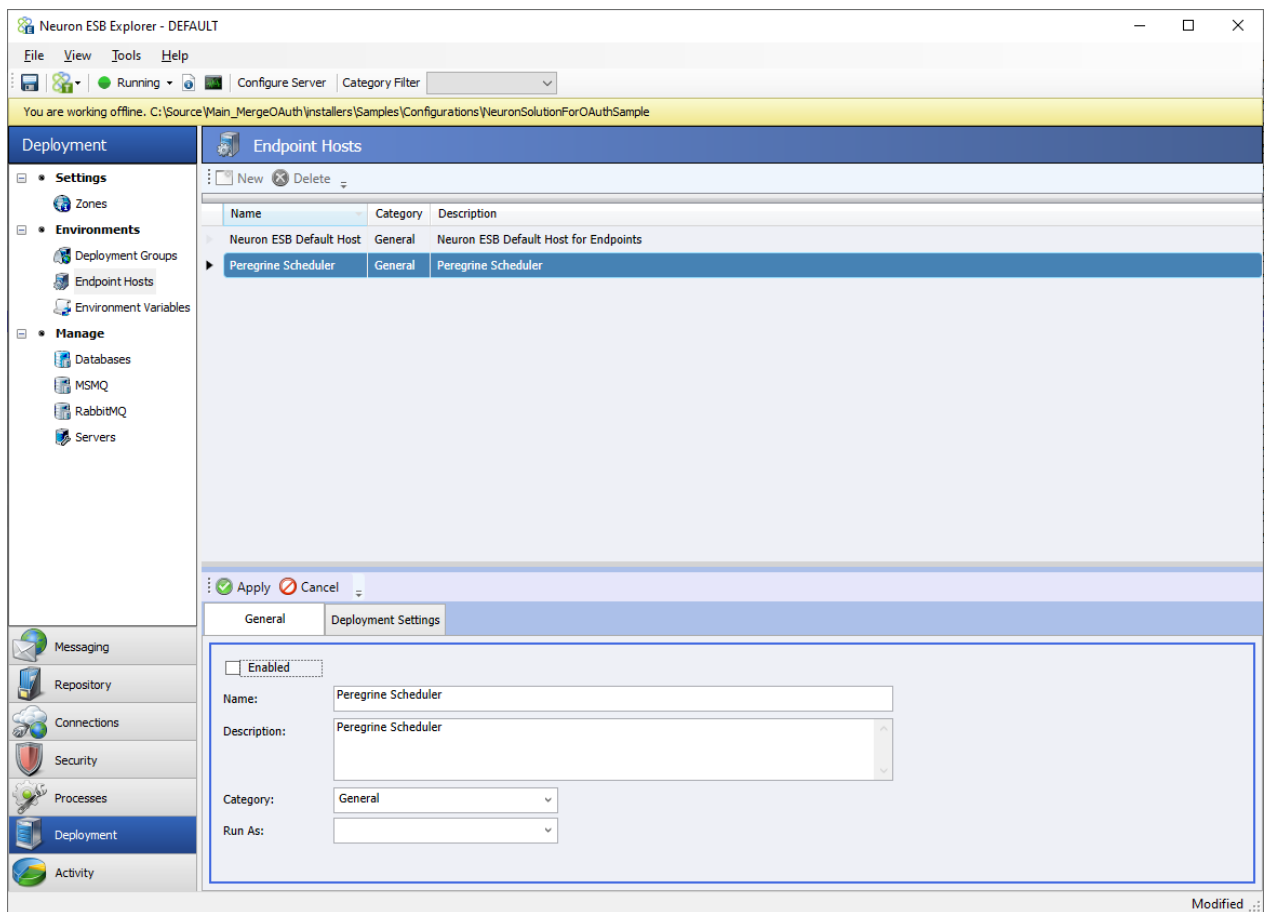
An exciting new feature that ships with the Neuron ESB 3.7.5 in conjunction with the [Peregrine Connect Management Suite 1.1](#) is the Job Scheduler. The Peregrine Management Suite Job Scheduler uses the Quartz job scheduling framework (<https://www.quartz-scheduler.net/documentation/>) that lets organizations schedule Neuron ESB Business Processes to be executed later, either just once or periodically. In the Job Scheduler, users can add new jobs, triggers, remove old ones, and create complex schedules for executing any job. Examples are e.g., tasks that run daily, every other Friday at 7:30 p.m., or only on the last day of every month or year. When the Job Scheduler is stopped due to an unexpected event, it will store the jobs in the Neuron ESB database. They will also survive Job Scheduler and runtime restarts and maintain their state. When the Job Scheduler is restarted, it will run all the jobs it should have run while it offline.

The Job Scheduler includes several features, including

- Support for Custom Jobs
- Scheduling any Business Process
- Reporting History
- Calendars
- Triggers
- CRON expressions
- Passing Messages and custom data to Jobs

While the Peregrine Connect Management Suite is used to configure Business Processes to run as scheduled Jobs, the actual runtime for the Scheduler is shipped with Neuron ESB 3.7.5 in the form of a special, dedicated Neuron Endpoint Host named “Peregrine Scheduler.” This Endpoint Host cannot be deleted, but it can be disabled.

Both Neuron ESB 3.7.5 and the Peregrine Connect Management Suite are required to use the new Job Scheduler functionality.

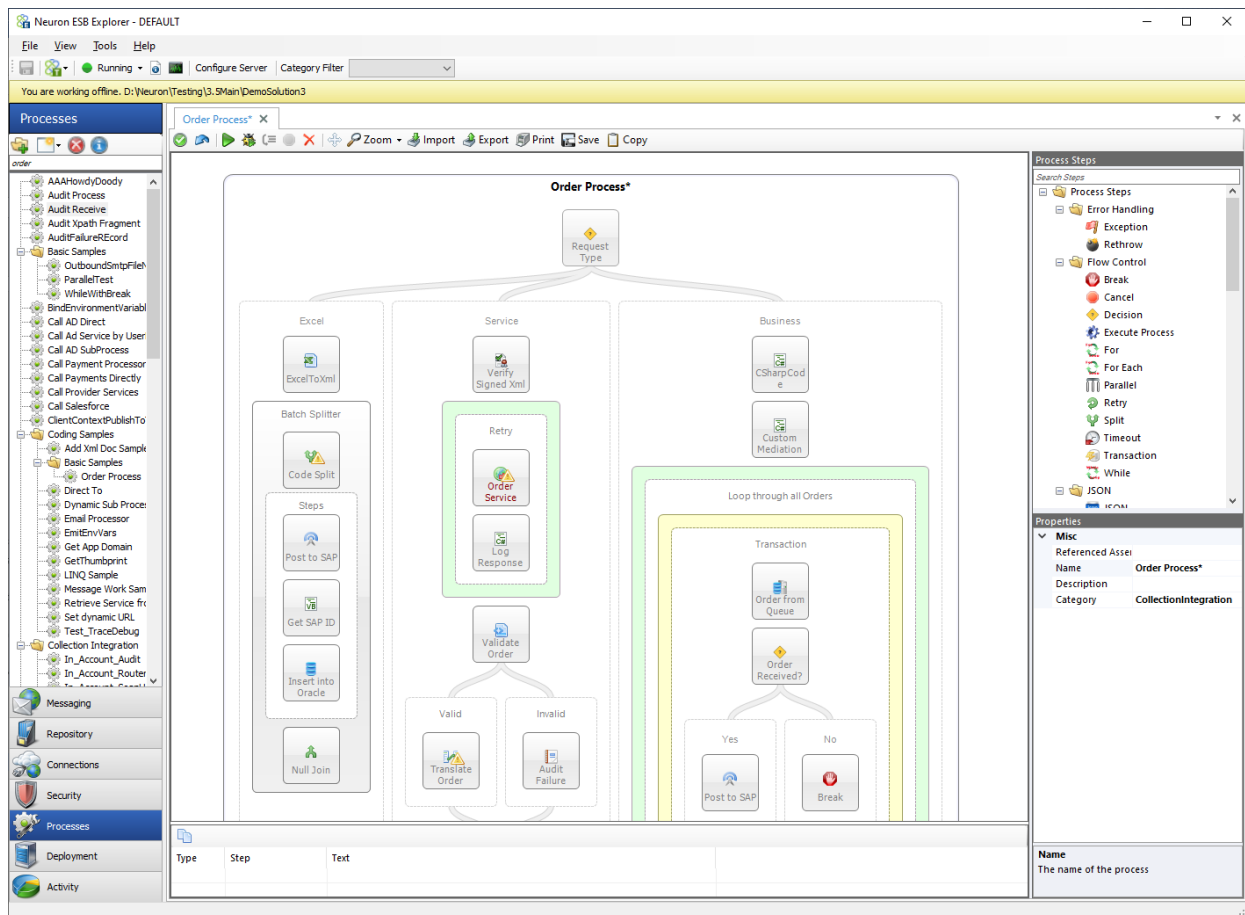


Neuron ESB Explorer – Figure 41- Endpoint Host screen displaying the Peregrine Scheduler. This is the runtime host for the Job Scheduler. All Jobs configured using the Peregrine Connect Management Suite are executed within this Host.

It is often a requirement of integration that data sources be queried for updated information regularly. This can be for many reasons, and most developers can think of at least three right off the top of their head.

Neuron ESB ships with a number of adapters and allows for the development of custom adapters. When used in conjunction with their polling and availability functionality, can easily accomplish this task. However, what happens when you wish to have the same polling on a schedule functionality for web services? What about when you need to have an adapter stop polling on specific days such as holidays and don't want to shut down the adapter manually? This is where the Job Scheduler comes into play and provides you the flexibility and functionality to address these scenarios.

The Job Scheduler allows any Business Process within a Neuron ESB solution can be scheduled as a Job. Scheduled Business Processes are not limited in functionality in any way, nor do they need to be appended with a Cancel Process Step. That means that adapter and service endpoints can be called within them. Even messages can be published to Topics and later processed by Workflows.



Neuron ESB Explorer – Figure 42- Sample Business Process. There are no limitations to what a user can configure within a Business Process for it to run as a scheduled job.

The Job Scheduler also injects custom properties and messages that can be defined at the Job level. Once defined in the Job Scheduler, they can be accessed and modified at the Business Process level. Information modified during the Business Process execution is accessible to subsequent executions of the same Business Process job.

For instance, the code snippet below shows how a property named “ConnectionString” can be retrieved from the Job Scheduler and used within a Business Process C# Process Step.

```
//Parse the value returned by web service

dynamic currencyObj = JObject.Parse(context.Data.Text);

Decimal AUDRate = System.Convert.ToDecimal(currencyObj.rates.AUD.Value);

Decimal USDRate = System.Convert.ToDecimal(currencyObj.rates.USD.Value);

Decimal GBPRate = System.Convert.ToDecimal(currencyObj.rates.GBP.Value);

Decimal CADRate = System.Convert.ToDecimal(currencyObj.rates.CAD.Value);

string connString;

//Get the connection string from Job Data Map of Scheduler.

Context.Instance.Scheduler.Properties.TryGetValue("ConnectionString",out
connString);

if (string.IsNullOrEmpty(connString))

    throw new ApplicationException ("Connection String Job Property is required
to save currency data");

//Make the database call to save the data.

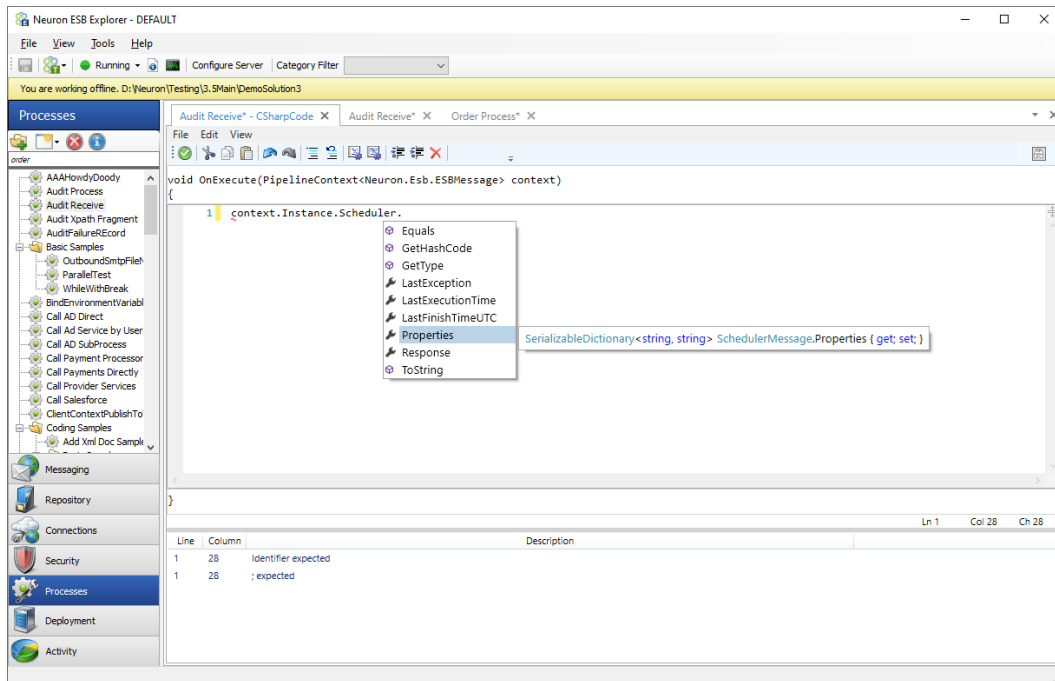
using (var conn = new SqlConnection(connString))
//@"Server=.\SQLEXPRESS;Integrated Security=true;Database=JobSchedulingDemo;")
{

    var cmd = new SqlCommand("insert into dbo.CurrencyRates (timestamp, Base,
Date, USD, AUD, CAD, GBP) values (@timestamp, @Base, @Date, @USD, @AUD, @CAD,
@GBP)", conn);
```



```
cmd.Parameters.AddWithValue("@timestamp", DateTime.Now);  
cmd.Parameters.AddWithValue("@base", "EUR");  
cmd.Parameters.AddWithValue("@Date", DateTime.Now);  
cmd.Parameters.Add("@USD", System.Data.SqlDbType.Decimal).Value = USDRate;  
cmd.Parameters.Add("@AUD", System.Data.SqlDbType.Decimal).Value = AUDRate;  
cmd.Parameters.Add("@CAD", System.Data.SqlDbType.Decimal).Value = CADRate;  
cmd.Parameters.Add("@GBP", System.Data.SqlDbType.Decimal).Value = GBPRate;  
  
conn.Open();  
  
cmd.ExecuteNonQuery();  
  
conn.Close();  
}
```

Additionally, the properties of the last job run can be accessed and modified within a Business Process, as shown below in the C# editor.



Neuron ESB Explorer – Figure 43- The C# Code Editor within a Business Process. This uses IntelliSense to display the Scheduler properties that are accessible within a Business Process. These properties can be accessed at runtime and modified so that their new values are available to the Business Process on subsequent executions.

There are many reasons why users may want to modify the Job Properties or add a Job Property from a scheduled Business Process. For example, in the previous code snippet the Last AUD rate could be added as a custom property by adding it to `context.Data.Scheduler.Properties` as shown below.

```
context.Data.Scheduler.Properties["AUDRate"] = 2.45;
```

Taking this approach, the last AUD Rate would always be available to the Business Process anytime it was executed.

The Job Scheduler is a powerful piece of technology shipping with Neuron ESB 3.7.5 and the Peregrine Connect Management Suite. To learn more about the Job Scheduler, please review the [Peregrine Connect Management Suite Job Scheduler documentation](#) and the samples we ship.

Rabbit MQ Quorum Queues

Neuron Topics using RabbitMQ as the transport now includes support for the Quorum queue type. The Quorum queue type offers increased data safety and equal or better throughput for the underlying topic's queues when compared with classic, durable, mirrored queues.

Quorum queues are implemented by RabbitMQ using a durable, replicated FIFO queue based on the Raft consensus algorithm (more information on Raft can be read here: <https://raft.github.io/>). They are desirable when data safety is a top priority and should be used as the default option where replicated queues are desired.

More information about RabbitMQ and their Quorum Queue implementation can be found here:

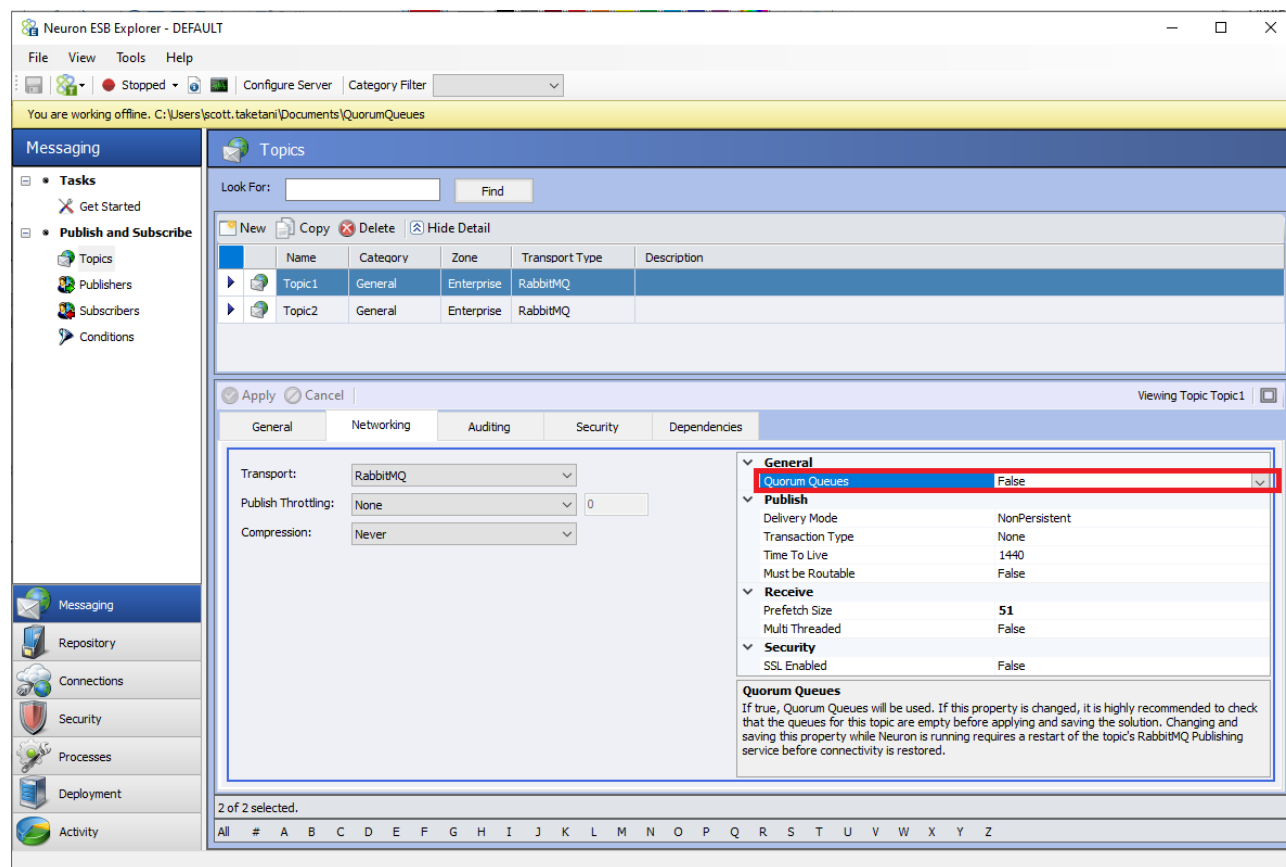
<https://www.rabbitmq.com/quorum-queues.html>

Requirements

1. A RabbitMQ server(s) with minimum version 3.8.0
2. A RabbitMQ cluster with at least 3 RabbitMQ nodes
3. A Neuron ESB installation with minimum version 3.7.5.327

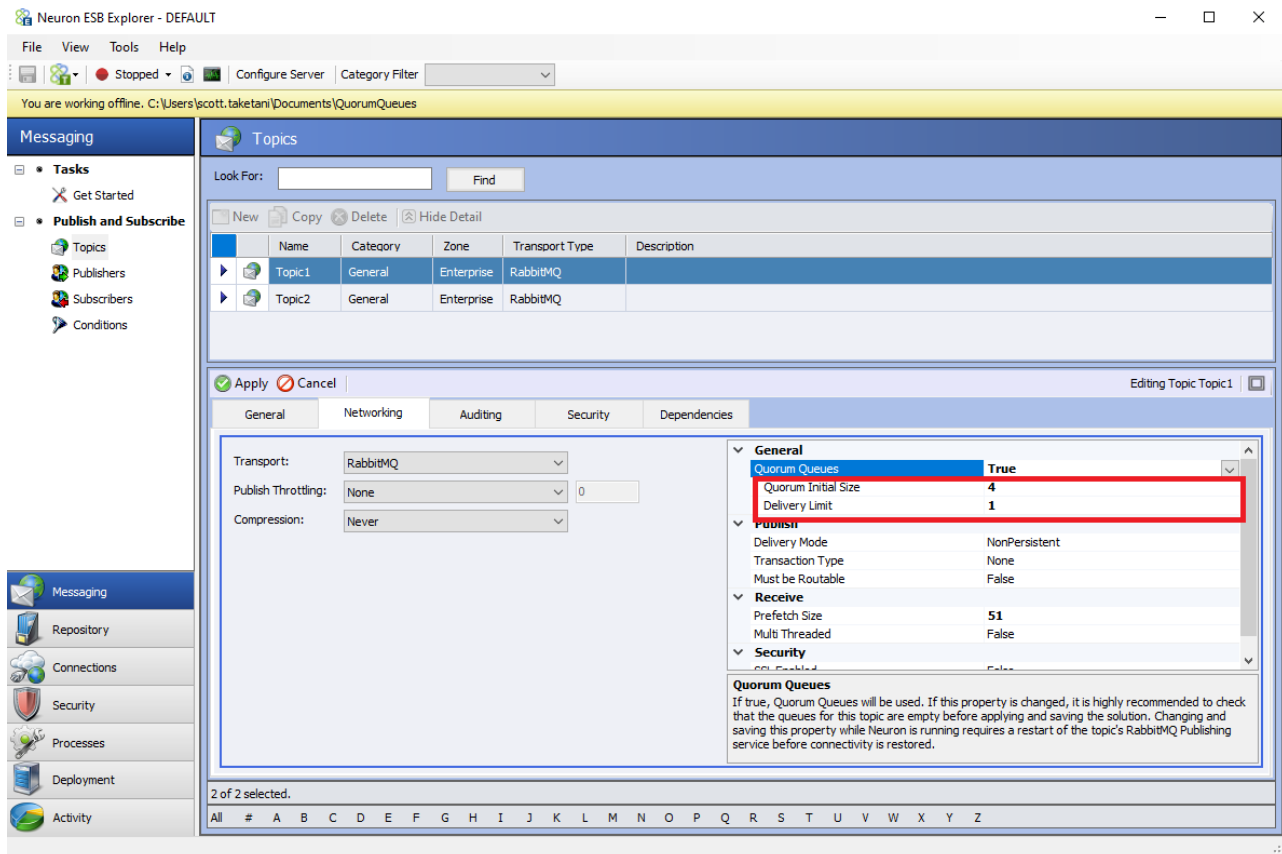
Setting up the Neuron Solution

The following screenshot shows a new property for RabbitMQ Topics in the Networking tab called “Quorum Queues.” By setting this property to true, Neuron will attempt to create the underlying topic’s queues using the “x-queue-type” RabbitMQ argument set to “quorum” (as opposed to “classic”).



Neuron ESB Explorer – Figure 44 - The network properties of the Rabbit MQ Topic are displayed. A new property called “Quorum Queues” can be set to true or false. Default is false.

After enabling the Quorum Queues property, the properties “Quorum Initial Size” and “Delivery Limit” will be shown, and the property “Time to Live” under the Publish section will be hidden (“Time to Live” is not supported as of now for Quorum type queues):



Neuron ESB Explorer – Figure 45 - The network properties of the Rabbit MQ Topic are displayed. A new property called “Quorum Queues” can be set to true or false. If set to True, two other properties become visible below it: Quorum Initial Size and Delivery Limit.

New Quorum Queue Properties

Property Name	Description
Quorum Queues	This setting determines if Quorum Queues will be used for the underlying Topic’s RabbitMQ queues. This does not affect the type of queue used for the Dead Letter Exchange. If a Quorum Queue is desired for the Neuron Instance’s Dead Letter Exchange, the setting “UseQuorumQueueForDeadLetters” is found in the appSettings.config file should be set to true.
Quorum Initial Size	The number of nodes that will participate in the Quorum and the number of nodes that the queues will be replicated to initially
Delivery Limit	Used for poison message handling. This is the number of attempts that will be performed to redeliver a message if the first attempt to deliver fails. This is implemented as a Queue Policy on the Neuron Topic’s underlying queues.

If the Quorum Queues property is changed, applied, and saved in a running Neuron Solution, the Neuron Topic's RabbitMQ Publishing service must be restarted either in Neuron Explorer's Endpoint Health page, or by restarting the entire ESB Service. Otherwise, any Neuron Parties subscribed to the topic will remain disconnected and any Parties attempting to connect to the topic will not succeed. Any messages published during this time will fail and be audited to the Failed Messages database table.

After changing queue types, restarting the Publishing service will cause the existing queues to be deleted and recreated with the desired type if there are no messages still in the queue. If there are messages in the queue, the queues will not be deleted, and the topic will fail to reconfigure. Therefore, it is highly recommended to drain the queue before saving the Neuron Solution if the queue type has changed (i.e., the "Quorum Queues" property was switched).

After performing the restart of the Publishing Service, any Neuron Parties that were disconnected should now automatically reconnect.

Delivery Limit

The Delivery Limit mechanism is used for poison message handling in place of Time To Live. This property is implemented in RabbitMQ as a Queue Policy (the policy property is called "delivery-limit" and is of type Number). The creation and deletion of this policy is handled automatically by Neuron and can't be used for Classic type queues.

The format of the created policy's name is as follows:

NEURON.<Neuron Instance Name>.<Topic Name>_DeliveryLimitPolicy

The policy's definition only includes the "delivery-limit" property. The policy's matching pattern (which determines which queues to apply it to) is of the format:

*NEURON.<Neuron Instance Name>.<Topic Name>_.**

There will be a policy created for every Quorum Queue type Neuron Topic.

Other Properties

All other properties should behave the same as when using classic type queues. Please see the following article for more information on RabbitMQ Topics in Neuron: <https://www.peregrineconnect.com/article/rabbit-mq-topics/>

Dead Letter Exchange

The Neuron Instance's Dead Letter Exchange for RabbitMQ Topics can also be set to use a Quorum Queue. This is not controlled by a Neuron Solution setting but rather an entry in the appSettings.config file found in the Neuron Instance's directory (default installation path: "C:\Program Files\Neudesic\Neuron ESB v3\DEFAULT").

UseQuorumQueueForDeadLetters

This property determines if a Quorum Queue will be used for the Dead Letter Exchange. Unlike the property in a RabbitMQ Topic's Network properties, switching this setting has no effect until the ESBSERVICE is restarted since the Dead Letter Exchange is per Neuron Instance.

However, like the Topic level property, the Queue will be automatically managed upon starting/restarting the Neuron Instance.

DeadLetterQuorumInitialSize

Similar to the Topic level property, this setting determines the number of nodes that will participate in the Quorum and the number of nodes the queue will be replicated too initially.

Other Considerations

The number of RabbitMQ nodes in the RabbitMQ cluster should be an odd number. This is because Quorums need to have a majority to work properly for the voting mechanism. Although this number can be even, if it is, then the Quorum mechanism may not work properly if a majority cannot be established.

Also, the guidance from RabbitMQ states that seven nodes should be the upper limit for the Quorum size and above five nodes cause performance to drop. Please see the Performance Characteristics section found here: <https://www.rabbitmq.com/quorum-queues.html#performance>

Additional Changes

Adapters

NEW – Salesforce Adapter - Durable Messaging with the Streaming API – The Streaming API feature has been updated to support durable messaging. A new property has been added to the Salesforce adapter endpoint with the name “Durable Messaging” and the default value of true. When set to true, the adapter will save the replay ID from the most recently received event from each subscribed channel. If the Neuron ESB service or the Salesforce.com adapter endpoint is restarted, the adapter will request any events that were created after the last received replay ID (for each channel the adapter subscribes to). Salesforce will only save events for either 24 or 72 hours, depending on the event type. For more information regarding message durability with the Streaming API, see https://developer.salesforce.com/docs/atlas.en-us.api_streaming.meta/api_streaming/using_streaming_api_durability.htm. When the Durable Messaging property is set to false, the adapter endpoint will only receive new events created from each time the adapter endpoint starts.

NEW - Salesforce adapter - Added support for versions 49 & 50 of the SOAP and REST APIs.

NEW - NetSuite Adapter - Added Support for version 2021.1

MOD – Microsoft Dynamics CRM Plugin – Support for version 4 of Microsoft Dynamics CRM has been removed.

MOD – Microsoft Dynamics CRM Subscription Adapter – Support for version 4 of Microsoft Dynamics CRM has been removed.

MOD – Microsoft SharePoint Server – Version 2010 and lower have been removed.

MOD – Microsoft Dynamics CRM Adapters/Plugins – All have been upgraded to use latest Microsoft Dynamics 365 SDK libraries.

NEW – HubSpot Adapter – Support for HubSpot has been added.

NEW – Box Adapter – Support for Box.com has been added.

NEW – DropBox Adapter – Support for DropBox has been added.

NEW – Alpha Trust Adapter – Support for Alpha Trust has been added.

Configuration

NEW – Import/Export Command Line switches – The following command line switches have been added to support the import/export of Data Maps and Service Route Tables:

Service Route Tables: -sr or -service-route

Data Maps: -datamap

Samples

NEW - Using Service Routing Tables - Service Routing tables provide a very low latency way of implementing basic Neuron ESB scenarios of mocking, mapping, and routing services without the overhead of a Business Process. This sample demonstrates using Service Routing Tables in a variety of use cases.

<https://www.peregrineconnect.com/documentation-kb/service-routing-tables-sample/>

NEW - Business Process to Workflow - The business process to workflow sample will show you how to receive information inside of a business process, in order to return an immediate response to a client, while passing the processing of long running tasks over off to a workflow.

<https://www.peregrineconnect.com/documentation-kb/business-process-to-workflow-sample/>

NEW - Peregrine MS Custom Job Scheduling – The Peregrine Connect Management Suite Scheduler can schedule any job within an enterprise or even outside Neuron ESB. Using the Custom Job function, you can implement your solution with Visual Studio or any .NET language. The custom job defined when using the Peregrine Connect Management Suite can be configured, triggered, and monitored. This sample illustrates a how to implement the custom job functionality. In addition, it will demonstrate how to check log folders for specific log files containing keywords and emails. This sample can be used as a guide for any other functionality you choose to implement.

<https://www.peregrineconnect.com/documentation-kb/scheduler-custom-job-sample/>

NEW – Peregrine MS Job Scheduling using Business Process - The Peregrine Connect Management Suite Scheduler sample will show you how to setup the Job Scheduler and run Business Process jobs hosted in Neuron ESB 3.7.5.

<https://www.peregrineconnect.com/documentation-kb/scheduled-jobs-sample/>

NEW - Securing Client Connectors with OAuth - This sample will show you how to setup a Client Connector secured with OAuth. The easiest way to setup a Client Connector with OAuth is with the help of Peregrine Connect Management Suite (PeregrineMS). You may also setup OAuth using an external provider such as AWS, Azure, Google or any other generic OAuth provider such as Thinktecture's Identity Server 4. In this sample, both PeregrineMS and Azure AD providers are illustrated. This sample shows you how to get a token and use it to call a Client Connector that is secured with OAuth.

<https://www.peregrineconnect.com/documentation-kb/securing-client-connectors-with-oauth-sample/>

Services

MOD – Service Point Manager properties – For all Service Endpoints called by Neuron, the following Service Point Manager Properties are set differently from their defaults:

- UseNagleAlgorithm = **false**
- Expect100Continue = **false**
- EnableDnsRoundRobin = **true**
- ReusePort = **true**

More about these properties can be found here: <https://docs.microsoft.com/en-us/dotnet/api/system.net.servicepointmanager?view=netframework-4.8>

Neuron 3.7.0.851

Workflow

FIX – Workflow Serialization Exception - If a workflow contained a Service Connector activity and the ThrowExceptionOnFault property was enabled, if a service returned a fault and the thrown ESBException was not handled with a TryCatch activity, it would cause the workflow to abort with a SerializationException due to a non-serializable member in the ESBException rather than aborting due to the ESBException.

Services

FIX - Client Connectors – An issue when trying to use a WSDL with a schema with <import> directives to schemas stored in the Neuron ESB repository. Previously the referenced schemas would not be ignored, resulting in problems when trying to import the WSDL in applications like Visual Studio or SoapUI. Now, when you have a WSDL that references schemas stored in the Neuron ESB repository, you need to change the schemaLocation attribute to use the “esb:” prefix, follow by the name of the schema in the repository. For example, if your WSDL contains a schema that looks like this:


```

<wsdl:types>

  <xsd:schema targetNamespace="http://tempuri.org/Imports">

    <xsd:import schemaLocation="https://localhost:8080/test?xsd=xsd0"
namespace="http://tempuri.org/" />

    <xsd:import schemaLocation="https://localhost:8080/test?xsd=xsd1"
namespace="http://schemas.microsoft.com/2003/10/Serialization/" />

    <xsd:import schemaLocation="https://localhost:8080/test?xsd=xsd2"
namespace="http://schemas.datacontract.org/2004/07/MyDataTypes" />

  </xsd:schema>

</wsdl:types>

```

You would change it to this:

```

<wsdl:types>

  <xsd:schema targetNamespace="http://tempuri.org/Imports">

    <xsd:import schemaLocation="esb:tempuri" namespace="http://tempuri.org/"
/>

    <xsd:import schemaLocation="esb:serialization"
namespace="http://schemas.microsoft.com/2003/10/Serialization/" />

    <xsd:import schemaLocation="esb:mydatatypes"
namespace="http://schemas.datacontract.org/2004/07/MyDataTypes" />

  </xsd:schema>

</wsdl:types>

```

Where tempura, serialization and mydatatypes are the names of the schemas in the repository.

When an application like Visual Studio or SoapUI try to import the schema, Neuron will automatically convert the schemaLocation attribute values to valid URLs that can be used by the application. To the application, the above schema will look like the first example above, replacing “esb:tempuri” with <https://localhost:8080/test?xsd=xsd0>.

You can also make similar changes inside the referenced schemas. For example, if the ESB schema named mydatatypes has in import directive to another schema in the repository, modify it’s schemaLocation attribute to use the “esb:” prefix followed by the ESB schema’s name.

Neuron ESB Explorer

FIX – Workflow with Switch Activity generates an error at design time – The following exception would popup if a Workflow Definition had a "Switch<T>" activity:

Microsoft.CSharp.RuntimeBinder.RuntimeBinderException: 'object' does not contain a definition for 'Count'

FIX – MSMQ Memory Leak – When querying message counts within the MSMQ management screen in the Neuron ESB Explorer, a memory would occur if the user continually queried for messages when processing many MSMQ messages. This could cause MSMQ to report the following error and subsequently kill all client channels:

Exception: An error occurred while sending to the queue: There are insufficient resources to perform this operation. (-1072824281, 0xc00e0027). Ensure that MSMQ is installed and running. If you are sending to a local queue, ensure the queue exists with the required access mode and authorization.

Business Processes

FIX – Memory Leak when running ForEach Loop with Large number of records – The follow exception could occur when running a large number of records through a foreach loop. This was due to internal tracking limitations that have been removed:

*System.OutOfMemoryException: Exception of type 'System.OutOfMemoryException' was thrown.
at System.Text.StringBuilder.ToString()
at Neuron.Esb.ESBMessageHeader.AddExecutionSegment(String name)
at Neuron.Esb.Internal.PipelineRuntimeHelper.ApplyPipelines*

Added "Turn Off Tracking" property to while, foreach, for and splitter. The default value is true. If true, the execution of the process steps within these control flow structures will not be tracked using the AddExecutionSegment() of the ESB Message. Additionally, the route tracking feature is limited to 250KB per Neuron ESB Message, regardless if "Turn Off Tracking" is set to True.

FIX – Custom Process Steps throw Serialization Exception when referenced remotely – The PipelineStepDataContractResolver did not resolve types for custom process steps when run remotely.

FIX – Custom Process Steps Null Exception at Design Time- If Custom Pipeline steps are present in the Pipelines folder in the instance's directory, Neuron Explorer could popup an exception when a process was opened or a new process was created. The exception would be regarding a Composition Container and an object being null which made it impossible to open or create a process. This was due to a sorting function that orders the process steps in the process designer's toolbox.

Messaging

FIX – Rabbit MQ Topic causes infinite message redelivery - If an ESBMessage contained items in the ESBMessage.Header.RequestHeadersToPreserve property and the ESBMessage contained the same

items in the CustomProperties collection, and the ESBMessage was used to create a reply exception ESBMessage (e.g. a service connector tries to return the reply for an endpoint that does not exist), the following exception could occur :

*Exception has been thrown by the target of an invocation. ---> System.ArgumentException: An item with the same key has already been added.
at System.ThrowHelper.ThrowArgumentException(ExceptionResource resource)
at System.Collections.Generic.Dictionary`2.Insert(TKey key, TValue value, Boolean add)
at Neuron.Esb.Internal.MessageConditionContext..ctor(ESBMessage message)
at Neuron.Esb.Internal.ConditionHelper.EvaluateMessageCondition(ESBMessage message, Condition condition)
at Neuron.Esb.Administration.ESBTopic.TopicMatch(String publisherTopic, Subscription[] subscriptions, ESBConfiguration config, ESBTopicContext topicContext, ESBMessage message, Boolean forReceive, Boolean forSend)
at Neuron.Esb.Party.SendMessage(ESBMessage message, Boolean duplicateMessage)
at Neuron.Esb.Party.SendMessage(ESBMessage message)
at NeuronEsb.Server.Runner.ESBServiceConnector.<Receive>b__17(ESBMessage x)
at NeuronEsb.Server.Runner.ESBServiceConnector.process(MessageEventArgs e, Action`1 reply)
at NeuronEsb.Server.Runner.ESBServiceConnector.Receive(Object sender, MessageEventArgs e)*

FIX – Rabbit MQ Multi-Threaded Receive Corrupts Message - If a topic is RabbitMQ and the Multi-Threaded property is set to true, messages could become corrupt when received by subscribers. The Rabbit MQ Topic has been modified to implement one channel per consumer for receiving messages based on the number of threads in the RabbitMQ topic's networking properties.

FIX – Rabbit MQ password fails if multiple servers used – If multiple Rabbit MQ servers are defined in deployment group and each has a different password, when an endpoint attempts to connect to a Topic, it may fail with a credential failure as the first entered password would be used for all servers when attempting a connection.

MOD – Rabbit MQ Topics Fail with Handshake Continuation Timeout – If a solution has a large number of Rabbit MQ Topics and parties, a Handshake Continuation Timeout Exception may occur if the user's machine could not process the Rabbit MQ Connections fast enough. To alleviate this, the client side Handshake Continuation Timeout has been extended from 10 to 30 seconds. However, users should also modify the Rabbit MQ Server configuration to match. Specifically, its recommended that the following Rabbit MQ server configuration entries be modified with the values below:

handshake_timeout:should be increased to 30 seconds

More information can be found at: <https://www.rabbitmq.com/networking.html>

OAuth Providers

NEW – Active Directory Provider - Added support to the Active Directory OAuth Provider to use a certificate for authenticating the token requestor.

Neuron ESB Runtime

MOD – Neuron Endpoint Host fails to start if machine name cannot be resolved to local machine - Adding error logging so that if no entries listed in the Machines tab of the Deployment Group can be associated to the current machine trying to start Neuron, an error will be logged. Previously, if this situation occurred, nothing was logged even though the endpoint hosts could not start.

FIX – Log Entries may be written to the incorrect log file – There was an issue where the EventInfo header on some warnings and errors could be incorrect. The incorrect header would be for a different endpoint than the one the event log entry originated from.

MOD – Neuron Endpoint Host lack of stack trace - When stopping an Endpoint Host, Neuron just logged the error message and not stack. This was problematic if we got a file load assembly error on stop. Previously, the only error information logged would be this:

Can't perform operation Stop Operation on Endpoint Host WMS BR05 Endpoint Host, because The located assembly's manifest definition does not match the assembly reference. (Exception from HRESULT: 0x80131040)

FIX – Neuron Endpoint Hosts receiving incorrect operations requests - Sometimes requests and replies sent and received by the HttpClient used to communicate with the Endpoint Hosts could get redirected to the wrong client and Endpoint Host.

NEW – Auto Tuning Threadpool for Endpoint Hosts - This can be used to reduce long startup times of solutions, especially if RabbitMQ topics are used.

On startup of the Endpoint Host process, if "StartupAutoTuneThreadpools" is true in the appSettings.config, Neuron ESB will try to monitor a system.threading.timer callback for completion time of the callback. If the completion time is off by no less than .25 seconds of its expected completion time (clock speed and core count is used to calculate the timespan that must be exceeded to cause a threadpool tune, but won't be shorter than .25 seconds), then the Threadpool minimum worker thread settings is increased by roughly the value contained in the appSettings.config key "NumThreadsToAddPerTune".

This value in the appSettings.config is an approximate number because it will be multiplied by an amount equal to $(1 + \text{<seconds that the callback execution missed by>} + \text{<cpu cores>} + \text{<processor speed>} / 10)$. This allows for more threads to be added if the time that the callback execution missed by is larger or if the cpu is quite robust.

The default value for "NumThreadsToAddPerTune" is set to 300. This value can be decreased if it is found that the threadpool tuning is adding too many threads or increased if it is found that the number of threads being added is still not enough. The recommended amount to change this by is +/- 100 but can be changed by smaller amounts if desired.

The Endpoint Host logs will contain final thread tuning counts for each endpoint host at the end of the Neuron startup. These can be used as rough estimates for calculating and setting the .NET threadpool overrides in the Neuron ESB server's configuration.

By default, the threadpool minimum worker thread value is set to the number of cores of the cpu. This is usually not sufficient for a Neuron ESB Solution that utilizes RabbitMQ for topics (if the number of RabbitMQ topics + number of any party/client instances planning to connect to the RabbitMQ topics exceeds the core count, then the threadpool minimum worker threads value should be increased).

An alternative method to the auto tune feature to estimate the min worker threads setting needed for solutions running RabbitMQ topics is to use the RabbitMQ web management interface. Start the Neuron ESB service with the desired solution (thread pool auto tuning can be enabled if the solution takes too long to start), then go to the RabbitMQ web management interface and note how many channels are being used by the instance of Neuron ESB (this step is easiest if no other applications are using RabbitMQ). This number is roughly the desired minimum worker threads setting. The number might seem high depending on the number of RabbitMQ topics but is a good estimation to use since the threadpool might become exhausted for a given Endpoint Host process during RabbitMQ server failures/outages.

Message Auditing

FIX – Sporadic Object Reference Exception occurs when Auditing Message – If using the Audit Process Step in a Business Process with Maximum instances set to 1, the following exception could occur. The exception is not predictable or easily replicated and would rarely occur with or without significant load:

System.NullReferenceException: Object reference not set to an instance of an object. at Neuron.Esb.Pipelines.EsbMessageAuditPipelineStep.MessageAudit.AuditReceive(String subId, ESBMessage message, Boolean asynchronous, Boolean includeBody, Boolean includeProps) at Neuron.Esb.Pipelines.EsbMessageAuditPipelineStep.CallAuditService(MessageAudit auditProxy, Boolean singleInstance, PipelineContext`1 state, ESBMessage messageToAudit, ESBClientContext clientContext, Object currentException, String failureType, String failureDetail) at Neuron.Esb.Pipelines.EsbMessageAuditPipelineStep.OnExecute(PipelineContext`1 state)

Adapters

FIX – FTP/SFTP/FTPS/ODBC/MQSeries/File Adapters – If a ThreadAbortException occurred during the Publish mode of the adapters, the exception may not trigger the disconnect and offline events of the endpoint. In that case, the Neuron ESB runtime would still believe the endpoint was alive even though the polling thread had failed. If the endpoint was configured for single instance mode, this failure would prevent the endpoint from starting on the next machine as the current machine would still believe its alive.

MOD – MSMQ Adapter – If an error occurred while publishing an incoming message to a Topic when in publish mode and the transaction property was set to false, the original MSMQ message would not be preserved. This behavior has been changed so that the message is now audited to the Neuron ESB Failed Message database.

MOD – Active Directory Adapter – Added support for SSL

MOD – Dynamics CRM XRM Adapter – Updated to the latest Dynamics 365 SDK

MOD – Dynamics CRM Subscription Adapter – This has been removed

MOD – Dynamics CRM 4 Plugin – This has been removed

Database

NEW – New SQL Database Update – SQL database script 0043_UpdateTo3_7.sql. This script creates the default "Network Service" login and user for the database.

Neuron 3.7.0.820

Topics

FIX – MSMQ Topic throws Null Reference Exception when MSMQ service is unavailable – If the MSMQ Service was shut down, or the MSMQ Server became unavailable, the following error message would be generated when Neuron would try to reconnect:

*Exception occurred when creating the message channel. System.NullReferenceException: Object reference not set to an instance of an object.
at Neuron.Esb.Channels.Msmq.MsmqChannel.GetNetMsmqQueueAddress*

Because of this, if the MSMQ server came back on line, Neuron would still fail to reconnect, forcing the user to restart the ESB Service. This has been corrected.

FIX – Rabbit MQ - Party.Connect() was not returning Topic errors – When using the Party API, the Connect() function would return a collection of errors if there was an issue connecting to the Topic. However, if the Topic was Rabbit MQ, the errors collection would not get populated. If the user was not capturing the OnLine() event or had logging enabled for the Client API, the user would not know if the Connect() function succeeded or failed. If the user attempted to Send a message, they would get a generic “Topic is not Connected” error. If the Party was just instantiated and connected to receive messages, the client application would never receive any messages successfully published to the Topic since they were never connected. This behavior has been corrected and now any connection errors will populate the Errors collection returned by Connect(). This only occurred when using Rabbit MQ Topics.

Neuron ESB Explorer

FIX – Republishing to Rabbit MQ Topic would fail – If a user attempted to republish a message to a Rabbit MQ Topic from either the Message History or Failed Message report, it would fail with a generic “Topic not connected” exception. This was because the following redirect was missing from the Neuron ESB Explorer’s app.config file:

```
<dependentAssembly>  
<assemblyIdentity name="System.Runtime.CompilerServices.Unsafe" publicKeyToken="b03f5f7f11d50a3a"
```

```
culture="neutral" />  
<bindingRedirect oldVersion="0.0.0.0-5.0.0.0" newVersion="5.0.0.0" />  
</dependentAssembly>
```

Additionally, the generic error message has been replaced with the correct underlying error message.

FIX – Solution Modified message when Solution is closed - If a user opened a solution in Neuron Explorer, then closed the solution using the File menu option, then modified the solution outside Neuron Explorer, a change was still detected for that solution even though it was closed. This would cause a popup to be displayed stating the solution has been changed and if the solution should be reloaded.

Neuron Runtime

MOD - Added SQL Error Code 596 to list of codes we should retry the SQL connection on – If Neuron previously received the following error message, it would not attempt to retry the connection to recover:

Cannot continue the execution because the session is in the kill state

FIX – Environment Variable is not set – This issue could occur if there are a large number of endpoints being hosted in a given endpoint host. The error that would occur states that an environment variable has not been set and would happen if bindings were being used to set properties on any type of endpoint that supports bindings (if the binding was using an environment variable). The issue usually would occur during startup of the endpoint host when all hosted endpoints were being started up.

Adapters

MOD – Xceed Libraries Version Update – All Xceed libraries have been updated from version 6.8 to 6.9. This affects all the FTP/SFTP/FTPS Adapters

MOD – SFTP Adapter and Public Key Algorithms – The following Public Key Algorithms have been added to the SFTP Adapter:

- aes128-gcm
- aes256-gcm
- aes128-gcm@openssh.com
- aes256-gcm@openssh.com

Neuron 3.7.0.811

Workflow

FIX – C# Editor throws exception - If a user tried to open a C# class code editor in workflow (i.e. right click on the activity and select "edit code"), they could run into an error that states that assembly loading has failed and that the enumeration may not execute since the collection was modified.

FIX – VB.NET/C# Code Editor Displays incorrect line numbers – Incorrect line numbers would be displayed if a design time error occurred.

FIX – VB.NET Workflow Activity error – The VB.NET Workflow Activity would through a compile error anytime when the editor was opened at design time.

FIX - **Transform – XSLT** and **Validate – Schema** Activity fails to load documents with “esb:” notation - If users have an XSLT or XSD that references an external document using the “esb:” notation, they would get exceptions when the Activities are invoked. For the Transform – XSLT step the exception was “System.Xml.XslLoadException” and for the Validate – Schema Activity the exception would state that one of the imported or included types has not been declared.

Business Processes

FIX – VB.NET/C# Code Editor Displays incorrect line numbers – Incorrect line numbers would be displayed if a design time error occurred.

FIX - JSON Schema Validation step - When an exception was thrown in the "Valid" branch, execution would jump to the "Invalid" branch. This was due to the step catching all exceptions and not just the invalid schema exception.

FIX - **Transform – XSLT** and **Validate – Schema** fails to load documents with “esb:” notation - If users have an XSLT or XSD that references an external document using the “esb:” notation, they would get exceptions when the process steps are invoked. For the Transform – XSLT step the exception was “System.Xml.XslLoadException” and for the Validate – Schema step the exception would state that one of the imported or included types has not been declared.

FIX – ODBC Process Step generates “The SqlConnection property has not been initialized” - An empty connection string could be used in the case that there is more than one ODBC process step running in the same Windows process AND one of the ODBC process steps is being called with a dynamic connection string (i.e. the connection string is passed in on the ESBMessage custom property "odbc.ConnectionString"). This would cause an exception to be thrown with the message "The SqlConnection property has not been initialized". The exception would appear to be thrown randomly from any ODBC step that did not use dynamic connection strings that ran in the same Windows process as the ODBC step that did use dynamic connection strings.

Neuron Runtime

FIX – Endpoint Host stuck in Starting State - If an Endpoint Host started from a failover, and it contained a workflow endpoint, then if a user restarted the workflow endpoint from Endpoint Health in Neuron Explorer, the Endpoint Host would get stuck in the "Starting" state even though it is running.

Neuron ESB Explorer

FIX – Neuron ESB Explorer crashes when using C# code editor – When editing within a C# code editor within a Business Process, the Neuron ESB Explorer could run into a condition causing an unhandled exception to occur and crash the Neuron ESB Explorer. The exception that could occur is:
System.InvalidOperationException: Collection was modified; enumeration operation may not execute.

Installation

FIX – Readme.HTML - Readme.html had documented incorrect redirect directives for RabbitMQ. Also incorrect Xceed assembly versions and the 0042 SQL script reference was missing.

FIX – Patching would cause Rabbit MQ Topics to fail – When running the existing Patch to update an existing instance of Neuron ESB to the latest version, the System.Runtime.CompilerServices.Unsafe.dll file was not getting installed correctly. This would cause all endpoints and test clients to fail to connect to RabbitMQ Topics.

Neuron 3.7.0.793

Workflow

FIX - Multiple Workflow Instances started for each message - If a workflow host started to experience exceptions of type "InstancePersistenceException" (which includes "InstanceOwnerException" and "InstancePersistenceCommandException") and the workflow endpoint was restarted (e.g. from Neuron Explorer's endpoint health page or upon changing the Neuron configuration), the Workflow Subscriber would not get cleaned up.

This could lead to messages being received by multiple instances of the subscriber, which could cause multiple workflow instances to be kicked off for each message received.

An "InstancePersistenceException" is usually caused by outages in SQL Server connectivity that persist for longer than 10 minutes when using the default transient fault handling settings (in versions older than 3.6.0.1282, it could occur at around 5 minutes).

Document Repository

FIX – “The Configuration Address is missing “ error when retrieving document at runtime - If a user tried to obtain an Xslt using GetXslt() from the ESBConfiguration object in a C# workflow activity, an error would occur stating "The Configuration Address is missing.". This happened only at runtime and would cause the workflow to abort.

Neuron 3.7.0.785

Environment Variables

FIX – Environment Variable values with Unicode characters not saved correctly – if the values entered for an Environment Variable contained specific Unicode characters, they would not be persisted correctly. Previously, the Neuron ESB Explorer was using “ISO-8859-1” for encoding rather than Unicode. This has been modified to only use Unicode encoding. For instance, if a user entered the following value, “testčšžćďěöß”. Once the solution had been saved and reopened, the value would be presented as this: “testcszcděöß”

Business Processes

FIX – Binding Change does not enable Apply – If Environmental Variable Bindings were added or edited on a Process Step, the activity would not enable the Apply Button. The user would have to change some other property in the property grid before the Apply button would be enabled. This has been fixed.

Neuron Test Client

FIX – Connecting to Rabbit MQ Topics would Fail – If a user attempted to use a Party that connected to a Rabbit MQ based Topic. The Topic would show up in the Test Client as offline and the following error would be generated.

```
There was a Broker Unreachable error connecting to the RabbitMQ channel for topic "Topic1":  
None of the specified endpoints were reachable  
RabbitMQ.Client.Exceptions.BrokerUnreachableException: None of the specified endpoints were  
reachable ---> System.IO.IOException: connection.start was never received, likely due to a  
network timeout  
at RabbitMQ.Client.Framing.Impl.Connection.StartAndTune()  
at RabbitMQ.Client.Framing.Impl.Connection.Open(Boolean insist)  
at RabbitMQ.Client.Framing.Impl.AutorecoveringConnection.Init(IFrameHandler fh)  
at RabbitMQ.Client.ConnectionFactory.CreateConnection(IEndpointResolver endpointResolver,  
String clientProvidedName)  
--- End of inner exception stack trace ---  
at RabbitMQ.Client.ConnectionFactory.CreateConnection(IEndpointResolver endpointResolver,  
String clientProvidedName)  
at Neuron.Esb.Channels.Amqp.AmqpChannel.CreateConnection()
```

This was due to an incorrect redirect in the Test Client's app.config file.

Runtime

FIX – Log Cleanup – At runtime, Neuron ESB will delete log files and log folders that exceed the Log Retention threshold defined in the Configure Server section of the Neuron ESB Explorer. By default, this is 10 days. However, Neuron ESB was first deleting the directories and then iterating through and deleting older individual log files that remained. This could result in newer log files be deleted because the original log folder, though created beyond the threshold, was still the current log folder because Neuron ESB had not been restarted. Neuron ESB creates a new Log Folder using a Timestamp only on startups. The logic has been changed to first delete older individual log files in all existing Log directories. After which all older Log directories which exceed the threshold would be deleted provided that they did not contain existing log files.

MOD – Newtonsoft JSON library – updated from 12.0.2 to 12.0.3

MOD – EasyNetQ Management Library – Updated to 1.3.0

Workflow

FIX – Unable to update referenced assembly in designer if assembly is updated - If a workflow code activity referenced an assembly, then the assembly was updated, users were unable to update the reference using the workflow designer. An exception in Neuron Explorer would occur when trying to manage the assembly references.

Now if the referenced assembly cannot be found or loaded, Neuron Explorer will attempt to find any version of the assembly in the instance's folder and any subfolders. If found, the user will have the option to use that assembly and update all references in any code activities. Note that although if the references are updated, if the assembly is located in a subfolder in the instance directory, the probe path must still be set in the app's config (e.g. EsbService.exe.config) for any code activity that reference that assembly.

If no version of the assembly can be found, the user will have the option to remove all references to the assembly in any code activities.

NOTE: This functionality is only available if the assemblies are referenced at the workflow level.

FIX – Endpoint Host at 100% CPU Utilization on failure of SQL Server - If an endpoint host was hosting a workflow endpoint and the SQL Server was not available for over 5 minutes, the endpoint host could start consuming a lot of CPU and Memory. Additionally, for all Endpoint Hosts that workflows are assigned to, the “PoolBlockingPeriod” for the Connection String to SQL Server is now automatically set to “NeverBlock”. This issue would only present itself if the SQL error code returned was not in the list of transient errors that is maintained within the Neuron ESB runtime. The SQL error codes that will cause the SQL Connection to retry using the transient fault handling mechanism are:

40501, 10928, 10929, 10053, 10054, 10060, 10061, 40197, 40540, 40613, 40143, 1225, 1205, 233, 64 -2, 20, 53, 983, -1, 3906

Lastly, the default transientFaultHandling Incremental strategy has been changed to retry for a minimum of 10 minutes with a 15 max retry count (16 if firstFastRetry is true) and 5 second initial interval and retry increment (600 seconds at minimum).

FIX – Throw with Message Workflow Activity - The arguments were not marked as required so it could cause a null exception if no "Message" argument was supplied. This would occur when a workflow aborted due to the activity which caused the workflow command to not be removed from the database. This would cause the workflow to start a new instance after a while since the command would be considered "timed out" and set to be reprocessed.

Neuron Configuration

FIX – Import of *.ESB files overwrites existing Endpoint Host – First issue, if all the adapters were not included in the *.esb file imported, the import file would add them, prompting the user to overwrite

existing ones. This also occurred for the default Neuron ESB Endpoint Host and Peregrine Scheduler host. This has been corrected so that no other entities are added to imported *.esb configuration.

Adapters

MOD – The Password property is now visible when Private Key is selected.

FIX – Metadata Generation for SAP and Dynamics CRM/XRM Adapters - 1) Response Xml samples were not being generated, and 2) If the response Xml sample was generated, it was identical to the request Xml sample.

FIX – NetSuite Adapter Metadata Generation – The following issues only affect version 2020_1 and newer:

- Search operations were not correctly generated
- Generated samples did not validate against the generated schemas
- Modified naming convention of the generated schemas to include the NetSuite version used to generate them

NEW – NetSuite Adapter and 2020_1 Support – Support has been added for the NetSuite 2020_1 release.

NEW – NetSuite Adapter and 2020_2 Support – Support has been added for the NetSuite 2020_2 release. User credentials are not support starting with this release. Customers that want to use this version or later will need to use Token-based Authentication as their security model

Licensing

NEW -Offline Licensing Support at Runtime – This is now available for Subscription based customers.

Neuron 3.7.0.764

Licensing

FIX – Updated License Key Fails – If a user received a new activation code, deactivating the old one within the Neuron ESB Explorer and then activating the new code, the Neuron ESB Explorer would reflect all the correct changes. However, when the ESB Service would start, it would still read and record the original Activation Code.

Runtime

MOD – Timers – Removed unnecessary stopping of timers within heartbeat interval routines. This could have had a negative impact on performance.

MOD – Parties would stop reconnecting to management service after 10 minutes - Previously, all connected parties would call the central Management Service hosted by ESB Service every 10 to 15 seconds to update its client statistics as well as check for configuration changes. If for whatever reason, the connection to the management service failed, the Party would attempt to continuously reconnect for up to 10 minutes, logging warnings on each attempt. This has been modified so that the Party will no

long stop attempting to reconnect after 10 minutes. However, after 10 minutes, the Party will only log a warning every 15 minutes thereafter, until it successfully reconnects.

MOD – Timers – Changed the timer in the TCP channel and Rabbit Channel to use `System.Threading.Timer`.

Test Client

FIX – Recording duplicate Event Log entries – This would occur due to an extra Logger element in the Test Clients' .config file.

FIX – Null Exception after ESB Service goes down – If the Test Client was Connected to the ESB Service and the ESB service went down for longer than 10 minutes, once it came back up, a null exception would be thrown if an attempt was made to send a message from the Test Client.

Messaging

MOD – TCP Clients would stop reconnecting after 20 minutes – Previously, if a Party was connected to a TCP Topic and that Topic went offline or down, the Party would attempt to continuously reconnect for up to 20 minutes, logging warnings every few seconds. This has been modified so that the Party will no longer stop attempting to reconnect after 20 minutes. However, after 10 minutes, the Party will only log a warning every 15 minutes thereafter, until it successfully reconnects.

FIX - Rabbit MQ Topics/ Endpoints not recovering - if Rabbit MQ was offline when Neuron ESB was started up, the Topics would be marked as Failed in endpoint health. However, any dependent Adapter, Service or Workflow Endpoints would appear green and Started in endpoint health, even though they had failed to start due to their inability to communicate with the Rabbit MQ Topic. If Rabbit MQ was later started up, the Topics did not automatically recover and return to a Started state. This was the same for any Party, Adapter Endpoint, Service Endpoint or Workflow Endpoint associated with the Topic i.e. They did not auto reconnect to the Rabbit MQ Topic. Since the underlying Topic would never have been connected to on the original startup, once Rabbit was started, the endpoints would not attempt to reconnect. Additionally, if someone attempted to send a message to the Topic, Neuron would think the client was connected when it was not...hence they would receive a null exception.

FIX – Null Exception returned on Send when Disconnected – if a user attempted to send a message to Topic via a Party when the Topic was offline, a null exception would be returned. Now the user will get a message indicating that the Party is not currently connected to the Topic.

Services

FIX - Null Exception returned when sending a request to Client Connector - if a user attempted to send a message to the URL hosted by a client Connector when its configured Topic was offline, a null exception would be returned, including the entire stack trace. Now the user will get a message indicating that the Party is not currently connected to the Topic without the stack trace.

Monitoring

FIX – Endpoint Health not recording some Workflow Errors – If there were errors and warnings generated at startup, they may not have been reflected in Endpoint Health.

FIX – Workflow Endpoints would not reflect issues if dependent Topic went offline – If the dependent Topic for a Workflow Endpoint went offline, or the workflow failed to connect to it on startup, the Workflow would not register any errors or warnings in Endpoint Health and would be displayed as Started and Green.

Neuron 3.7.0.757

Services

FIX – HTTP Status Code 422 – If a service endpoint returned a 422, it would cause the following the error:

'Enum value '422' is invalid for type 'System.Net.HttpStatusCode' and cannot be serialized.'

We now intercept that and report it as a 400 HTTP Status code

FIX – HttpBasic Security Model – This is now supported with REST endpoints when using a Username credential. Previously, we were not creating the necessary HTTP Authentication Header for it.

Workflow

FIX – Starting aborted workflow would fail on failover server – When using the Neuron Explorer, if a workflow was running on a server and it aborted, then an attempt was made to start it on a failover server, it would not work. Now we send the restart command to whichever server the workflow endpoint is currently running on (only in the case of primary-failover setup for the endpoint host that the workflow endpoint is assigned to).

Adapters

FIX – Sales Force Adapter - Salesforce adapter was not honoring Send Timeout property. Would default to 2 seconds instead.

FIX – ODBC Adapter Metadata Generation - When clicking the Execute button on the wizard page "Choose the Operations", the user would receive the error "The SQL statement failed to execute", with the exception details stating "Object reference not set to an instance of an object".

MOD – Sales Force Adapter - Added MaxReceivedMessageSize property for outbound messages mode – Previously, Salesforce could sending messages that could being rejected by the Salesforce Adapter because they were larger than the default value.

MOD – NetSuite Adapter – The adapter has been updated to use the HMAC-SHA256 for all Token Based Authentication. Previously, the HMAC-SHA1 algorithm was used.

Runtime

FIX – Environment Variable to control Enable/Disable - If an environment variable was used to control the enable/disable flag of an endpoint and, that endpoint was hosted within an endpoint host, the following error would occur:

The <adapter name> failed to start. Error occurred attempting to set the 'Enabled' property using the '{\$NameOfEnvironmentVariable}' Environment Variable. The environment variable NameOfEnvironmentVariable has not been set

FIX – Endpoint Host Control Port not updating Configuration – Event though the endpoint host process would get the updated port in the appSettings.config, the ESB Service would not since a refresh was not called on the config. This could cause an error to be logged when stopping/starting from the Peregrine Connect Management Suite.

Neuron 3.7.0.747

Workflow

FIX – Workflow Performance Counters – There were 2 performance counters that were not being properly disposed of.

FIX – Receive Message not receiving correlated message on time - The Receive Message activity would not receive the correlated message until the timeout period elapsed that is set in the "Timeout" property. This occurs if the Workflow was unloaded while waiting for the message.

FIX – Endless loop of Workflow Endpoint during Change event - The workflow endpoint could endlessly loop upon changing the running workflow definition and then saving the configuration as the updated configuration was not being properly retrieved for the workflow's party.

FIX – Missing Logging Information - The party used by a workflow endpoint would not log anything.

FIX – Previously saved 3.6 Workflows not starting in 3.7 - When a configuration was saved in 3.6, then a user upgraded to 3.7 or later, and if there were workflows started in 3.6, when the workflows resumed in 3.7, they would not resume and would go into the "error processing" state.

Services

FIX – WebHttp Binding for Service Connectors – Support has been enabled for using WebHttp binding in scenarios where the REST binding may fail.

MOD – Environment Variables and Service Policies – Environment Variables can now be used to determine the Service Policy to execute for a Service Endpoint on startup.

Adapters

MOD – Environment Variables and Adapter Policies – Environment Variables can now be used to determine the Adapter Policy to execute for an Adapter Endpoint on startup.

Samples

FIX – Sample Update for 3.7 Configurations – Some samples had previously not been updated for 3.7.

Neuron 3.7.0.743

Services

FIX – Incorrect Service Policy may be executed – If a user has more than one Service Policy defined in the solution and Service Endpoints are bound to the different policies, an incorrect policy may be executed at runtime.

Business Processes

FIX – Transform XSLT Process Step honoring xsl:output Encoding attribute – The step has been modified to honor the Encoding attribute of the XSL:Output element of the XSLT document.

FIX – Parallel Process Step throws Exception – under uncertain conditions, the parallel step may throw a System.NullReferenceException after all the steps within all the branches had been executed successfully. This was due to a thread race condition.

Adapters

FIX – NetSuite Adapter – Fixed error that would occur when generating metadata for the search operation in some of the NetSuite object listed under the “Transactions” category. This fix has been applied to metadata generation for adapter endpoint that utility NetSuite release 2018.2 and newer.

FIX - NetSuite Adapter – Added support for setting Search Preferences in NetSuite Adapter endpoints that will be used when calling search operations on NetSuite objects. The Search Preferences that can be set are:

- Page Size – the number of records to return per page
- Body Fields Only – when set to true, only body fields of the object being searched are returned. When set to false, sub list values are also returned.
- Return Search Columns – This property is only relevant when you are using an advanced search to execute a saved search. When set to true, only the columns defined in the saved search are returned (unless overridden by the advanced search query, which then only those columns are returned). When set to false, the return columns are ignored and the entire record is returned.

Neuron 3.7.0.735

Adapters

MOD – FTP/FTPS/SFTP Supports non routable IP Addresses – When opening a passive FTP data connection, the FTP server communicates an IP address and port to connect to, in response to client's the PASV-command. However some of FTP servers return non-routable IP addresses, i.e. instead of their public IP they return the private/internal IP of the server. The adapters have been modified to now identify the following address ranges as private non-routable addresses:

- 10.0.0.0 through 10.255.255.255
- 172.16.0.0 through 172.31.255.255
- 192.168.0.0 through 192.168.255.255

and for IPV6, any address with the prefix 'fd'.

When these are encountered in a response from PASV, the adapters will now ignore these addresses and use the server's IP address instead *IF* the server's address is not itself a private address as well.

Neuron Database

MOD – Neuron ESB Database Creation – Neuron ESB will no longer set the size of the database file or log file at creation time.

Neuron ESB Runtime

MOD – Document Repository causes Memory growth – Neuron ESB no longer will front load documents located in the Repository at startup. Instead, WSDL, SWAGGER, XSLT and XSD documents will be loaded on demand and cached. For users that need to retrieve these documents in custom code within the Business Process designer or Workflow designer, the following method signatures have been added. Examples follow below.

for business processes:

- `context.Data.Text = context.Configuration.GetWsldDocument(docName).Xml;`
- `context.Data.Text = context.Configuration.GetSwaggerDocument(docName).Xml;`
- `context.Data.Text = context.Configuration.GetXslt(docName).Xml;`
- `context.Data.Text = context.Configuration.GetXsd(docName).Xml;`

for workflow:

- `message.Text = configuration.GetWsldDocument(docName).Xml;`
- `message.Text = configuration.GetSwaggerDocument(docName).Xml;`
- `message.Text = configuration.GetXslt(docName).Xml;`
- `message.Text = configuration.GetXsd(docName).Xml;`

MOD – Logging Version of Operating System – The method in how this is done has been modified. The method used the OSVersion off of the System.Environment object which would not reflect the current version of the OS if the machine had been updated.

Neuron 3.7.0.725

Topics

MOD – Rabbit MQ Upgraded to Client version 6.2.1 – The Rabbit MQ Client used by Neuron ESB has been upgraded from 6.1 to 6.2.1.

Adapters

MOD – ODBC Adapter – Added support for populating the internal datatable using ExecuteReader rather than the standard fill command. A new property has been added named “Use Data Adapter”. The default value of “True”, provides the previous behavior using the Fill command. When set to false, the

“Command Behavior” property becomes available to set which is used in conjunction with the ExecuteReader command. Setting “Use Data Adapter” to false should be used when attempting to access a resultset where multiple resultsets are returned. In this scenario, using the Fill command could return an empty resultset.

Services

FIX – Dynamically setting the URL of REST endpoint using OAUTH - If using the Addressing.To property to dynamically change the URL of a service connector at runtime, if the Service Connector also had an OAUTH provider assigned to it, the user would receive an Authentication exception similar to the one below.

Neuron.Esb.EsbService.RestPolicyStatusException: The REST Service call returned the HTTP Status Code '401' that violates the established service policy 'REST Policy'

Business Processes

FIX – Debugging Error when using Audit Step - If a user attempted to debug a Neuron business process in the Neuron ESB Explorer Business Process designer and the process contained an Audit step, the audit step would fail due to the instance name not being set.

Licensing

MOD – Retail and Standard editions – The installer licensing software was not recognizing Standard and Retail license editions

MOD – Offline Activation during Installation – The Computer Code can now be entered into the installer screen during the installation process so that the user does not need internet access during installation. The computer code can be provided by Neuron ESB Product Support.

Endpoint Health

FIX – UI Controls out of focus - Issue: When selecting endpoints/hosts/services in Neuron Explorer's endpoint health screen, sometimes other objects would appear to be selected automatically, particularly when new heartbeats came in.

FIX – Toolstrip button commands would fail for Endpoint Host - Issue: The toolstrip "Restart Service" and "Stop Service" buttons would fail to work when an endpoint was selected in the Endpoint Hosts part of Endpoint Health.

Neuron 3.7.0.707

Adapters

FIX – Adapter Policy adding Retries – If an Adapter Policy was set for 0 retries, at least 1 retry would be attempted by the runtime. This has been corrected.

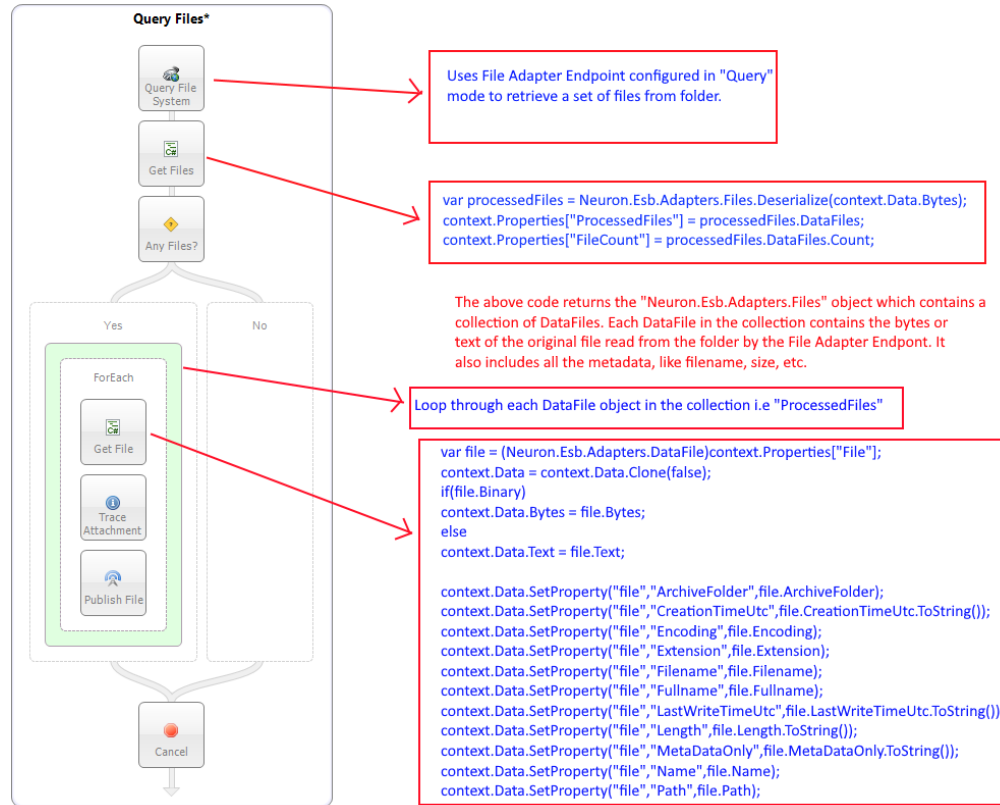
MOD – MSMQ Adapter – The queue path can now be set dynamically at runtime in a Business Process by setting the “msmq”, “QueuePath” message property, identical to the MSMQ Business Process step functionality

MOD – FTP/FTPS/SFTP Adapter – Underlying Xceed libraries have been updated from version 6.5 to version 6.8. This also applies to the Compression Business Process step.

NEW – File Adapter Query Mode – A Query Mode property has been added to the File Adapter. This allows users to create an Adapter Endpoint that can be called via Messaging or directly through a Business Process which returns one or more files read from the folder and matching either the filename or file extension specifications configured for the adapter endpoint. The Query mode supports all the features that users find on Publish mode (when the File Detection property is set to Polling). The message body returned from the Adapter Endpoint is of type *Neuron.Esb.Adapters.Files*. This object will contain a collection of *Neuron.Esb.Adapters.DataFile* objects. Each DataFile class is defined as:

```
public class DataFile
{
    public DateTime CreationTimeUtc { get; set; }
    public DateTime LastWriteTimeUtc { get; set; }
    public string Encoding { get; set; }
    public string ArchiveFolder { get; set; }
    public Byte[] Bytes { get; set; }
    public string Text { get; set; }
    public long Length { get; set; }
    public string Name { get; set; }
    public string Path { get; set; }
    public bool ReadOnly { get; set; }
    public bool Binary { get; set; }
    public string Extension { get; set; }
    public string Fullname { get; set; }
    public string Filename { get; set; }
    public bool MetaDataOnly { get; set; }
}
```

Using a Business Process, The DataFile objects can be iterated through using a For Each process Step. Below is a graphic demonstrating a simple process that iterates through each DataFile, publishing each to a Topic.



New – File Adapter reads for a specific file (s) – Users could only configure the set of file extensions to look for when the File Adapter was configured for Publish mode. The adapter would then, either through polling or monitoring of folder events, read the files which matched the extension specifications populated in the “File Type” property. For example, the following would be a valid entry for “File Type”:

**.txt;*.xlsx;15 ??eudesic W9.p?f;*.png*

The adapter, regardless of the “File Detection” property setting (which could be either Polling or FolderEvents) would retrieve all files which matched the list of extensions. However, now the Publish mode and Query mode of the adapter supports retrieving one or more files (separated by a semicolon), that match specific full names. For Publish mode, this feature is ONLY accessible when the File Detection property is set to Polling, and the File Retrieval Method is set to FileName.

MOD – File Adapter Overwrite on Archive – An Overwrite option has been added to the archive feature of the adapter. Previously, if the file name already existed in the archive folder, attempting to move the file with the same name would throw an error.

MOD – MQSeries Adapter Cipher Specs - Added the following additional cipher specs to the MQSeries adapter and process step:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_GCM_SHA256

- TLS_RSA_WITH_AES_256_GCM_SHA384
- ECDHE_ECDSA_AES_128_CBC_SHA256
- ECDHE_ECDSA_AES_256_CBC_SHA384
- ECDHE_ECDSA_AES_128_GCM_SHA256
- ECDHE_ECDSA_AES_256_GCM_SHA384
- ECDHE_RSA_AES_128_CBC_SHA256
- ECDHE_RSA_AES_256_CBC_SHA384
- ECDHE_RSA_AES_128_GCM_SHA256
- ECDHE_RSA_AES_256_GCM_SHA384

Messaging

FIX – Parties Not Renewing Topic Connections with changes of Topic Network Properties – For the most part, parties would recycle and renew their Topic Connections if they detected that a change occurred in the Network Properties. However, this did not happen in all cases. Now we check each individual property to see if it has been changed and if so, we recycle the Party's connection to the Topic so that it can use the new Network Properties.

NEW – Rabbit MQ and Multi Threaded Receive – Rabbit MQ Topics provide single threaded consumption of messages to subscribing Parties out of the box. This behavior also provides ordered delivery out of the box. Previously, the only way to increase throughput would be to create multiple instances of the Endpoint or Party. This would in turn create multiple consumers for each underlying queue. This had limits as it would also be creating competing consumers. In this release, a “Multi Threaded” property has been added to Rabbit MQ Topics (default value is False). If set to true, the “Number of Threads” property will become visible. It can be set from 1 to up to 20, though the default value is 5. This will indicate how many threads the Neuron ESB subscriber will use to dispatch messages received from the underlying consumer. Then can significantly increase subscriber throughput when using Rabbit MQ based Topics.

Neuron ESB Explorer and Tools

FIX – ImportConfig.exe and Remote Powershell sessions – When using the invoke-command in Powershell to execute the ImportConfig command on a remote computer a NativeCommandError would occur. This has been fixed.

Topics

MOD – Rabbit MQ Upgraded to Client version 6.1 – The Rabbit MQ Client used by Neuron ESB has been upgraded from 5.1 to 6.1. This has the impact of adding the following dependencies to any application hosting the Client API which is using Rabbit MQ based topics:

- System Buffers.dll
- System.Memory.dll
- System.Numerics.Vectors.dll
- System.Runtime.CompilerServices.Unsafe.dll
- System.Threading.Channels.dll

- System.Threading.Tasks.Extensions.dll

These dependencies are installed with Neuron ESB and can be copied directly from the Neuron ESB instance folder. Additionally, the following redirect has been added to the ESB Service, Neuron Endpoint Host, Neuron ESB Explorer and Neuron ESB Test Client's app.config files:

```
<dependentAssembly>
  <assemblyIdentity name="System.Threading.Tasks.Extensions" publicKeyToken="cc7b13ffcd2ddd51"
    culture="neutral" />
  <bindingRedirect oldVersion="0.0.0.0-4.2.0.1" newVersion="4.2.0.1" />
</dependentAssembly>
```

Auditing

FIX – Resubmitting Binary Messages fail – Once a message has been saved in the Audit system, it can be retrieved and saved to disk or, resubmitted to an endpoint or publisher through the Message Viewer in either the History or Failed Message reporting feature of the Neuron ESB Explorer. If the message was of type Binary, as in the case of a message from the POP3 adapter that contains attachments, resubmits would fail with a System.Runtime.Serialization.SerializationException exception. This has been corrected.

Runtime

MOD – Endpoint Host Communication Port – The default Endpoint Host communication is 51004. This could be modified in the appsettings.config file. Previously, on startup if the port was in use an error would be thrown. Correction would be to manually find a free port, edit the appSettings.config file and attempt to restart the Endpoint Hosts. This has been modified to be dynamically assigned. Neuron ESB will attempt to use the default port and, if in use, will scan a range of 1000 ports to find a port that is available.

MOD – Scheduler Remoting Port – The default Scheduler Remoting port is 49155. This could be modified in the appsettings.config file. Previously, on startup if the port was in use an error would be thrown. Correction would be to manually find a free port, edit the appSettings.config file and attempt to restart the Scheduler Host. This has been modified to be dynamically assigned. Neuron ESB will attempt to use the default port and, if in use, will scan a range of 1000 ports to find a port that is available.

Neuron 3.7.0.671

Health Monitoring

FIX – REST API call to ClientConnectors returns 500 – When a user made a REST API (i.e. <http://{NeuronMachine}:51002/help/index>) call to retrieve the list of Client Connectors an HTTP Status Code of 500 would be returned. The specific API is:

VERB: GET

URL: /neuronesb/api/v1/Configuration/{instance}/ClientConnectors

NEW – REST API for retrieving Failed and Audited Messages – 4 new API calls have been added to the Activity section of the Neuron ESB REST API (**Error! Hyperlink reference not valid.**) that allows users to return either a single Failed Message or Audited Messages, or a collection of either. A database updated is required for one of the APIs (i.e. /neuronesb/api/v1/Activity/{instance}/Message/ID/{messageid} requires a new store procedure, 'GetMessageFromAuditByMessageId') The specific APIs are as follows:

VERB: GET

URL: /neuronesb/api/v1/Activity/{instance}/FailedMessage/ID/{messageid}

Returns one or more failed messages that match the provided message id for the given instance.

VERB: GET

URL: /neuronesb/api/v1/Activity/{instance}/FailedMessage/Sequence/{sequenceid}

Returns one failed message that matches the provided sequence id for the given instance.

VERB: GET

URL: /neuronesb/api/v1/Activity/{instance}/Message/ID/{messageid}

Returns one or more Audited messages that match the provided message id for the given instance.

VERB: GET

URL: /neuronesb/api/v1/Activity/{instance}/Message/Sequence/{sequenceid}

Returns one Audited message that matches the provided sequence id for the given instance.

NEW – REST API Logging and Exception Shielding – Logging has been implemented. All errors and, Debug, Verbose and Info statements will now be written to the NeuronOperationService-{date}.log file located in the logs directory of the Discovery Service (default location: C:\Program Files (x86)\Neudesic\Neuron ESB v3\logs). Previously, if an error occurred the entire error and stack would be returned to the client. Now the client will receive a message (i.e. HTTP ReasonPhrase) as well as an HTTP status code of either 500, 200, 204, or 404. This has been done for the Configuration and Activity Sections of the REST API.

FIX – REST API Configuration Not reflecting current configuration – The configuration that the REST API was loaded at the start up of the discovery service. If a running instance's configuration was changed by restarting the instance, the REST API would still reflect the original configuration that was associated with the instance on the startup of the discovery service.

MOD – Locking in Audit and Failed Message Reports – if using the Neuron ESB Explorer to query Audited or Failed Messages and the existing database has many records and the query is done at a time of active runtime auditing, locking errors could occur. We've added the NO LOCK hint to all reporting based stored procedures to alleviate this.

Auditing

FIX – Custom SOAP Headers Note Stored – When a message had custom SOAP headers and the message was audited, the custom SOAP Headers were not being serialized correctly into the Neuron ESB Audit database.

Runtime

FIX – Run As for Endpoint Host throws Access Denied Error – When attempting to run an Endpoint Host under specific user credentials (other than LocalSystem) an access denied error would occur. This has been fixed. However, if ESB Service is running under something other than LocalSystem, the service account must be given the following permissions in the Security Policy for Windows (LocalSystem account should have these already). This is now documented in the Readme.html file.:

- Log on as a service
- Act as part of the operating system
- Adjust memory quotas for a process
- Replace a process level token

Adapters

FIX - Salesforce Adapter - When using the SOAP API, the first call to the API would call login to retrieve a session ID that would be used on subsequent calls. The session ID would have an expiration time, and when it expired the adapter would call login again to get a new session ID. The expiration time was incorrectly being stored as the current time, causing the adapter to believe the session ID had expired on the next call. The adapter was calling login for every request sent to the SOAP API. This has been fixed and the expiration time is being appropriately saved based on the session settings that are set in Salesforce-> Setup->Session Settings.

Neuron ESB Explorer

FIX – Environment Variables – If a user chooses to delete an environment variable, then the values of the next environment value in the list will be emptied out.

Neuron 3.7.0.662

Adapters

FIX - SharePoint Adapter - Addressed issue when the error

“Neuron.Esb.Adapters.SharePoint.SharePointAdapterException: Failed updating list Vendor Program Announcements, WSS returned the following error Message: Column 'xxx' does not exist. It may have been deleted by another user.” Is returned. This error may have occurred on some SharePoint sites and not others (i.e. a test site vs. production site), when using the same column name. This issue has been addressed by using the internal column name as set by the SharePoint site.

Services

FIX – System.Net.Http dll not found error – the following error would be reported if starting an Endpoint Host or calling an REST endpoint:

System.IO.FileNotFoundException: Could not load file or assembly 'System.Net.Http, Version=4.2.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a' or one of its dependencies. The system cannot find the file specified. ---> System.IO.FileNotFoundException: Could not load file or assembly 'System.Net.Http, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a' or one of its dependencies. The system cannot find the file specified.

Licensing

NEW – Licensing – Support has been added for Retail OEM edition. Previously this was not recognized by the Neuron ESB Explorer or runtime.

Messaging

FIX – MSMQ Topics – Neuron would log the following error even if transactions were set to false. This should only be logged if transactions are true:

"The channel may have entered a FAULTED state because the Receive Error Handling property on the MSMQ Topic has been set to Fault. As long as a poison message remains in the queue, the channel cannot be recreated successfully. Please ensure that the underlying queue representing the topic is cleared of Poisoned messages."

FIX – MSMQ Topic Poison Message Processor - The poison message handler could be started if the ReceiveErrorHandling property was set to Move while transactions were disabled.

Neuron 3.7.0.656

Service Endpoints

FIX – Service Policy REST Status Codes - When a REST response with a status code that is listed in the "Include" list of Http Status codes in the policy contains a body with content, the following exception would be thrown: "This message cannot support the operation because it has been written."

Neuron 3.7.0.647

Service Endpoints

FIX - Client Connectors – when an incoming SOAP request is routed to a service connector, and the service called by that service connector returns a SOAP Fault that includes custom SOAP headers, the client connector would not return those custom SOAP headers to the client. This issue has been fixed. Now, when a client connector is configured to Enable SOAP Headers, it will return SOAP headers that are included with a SOAP Fault that is returned by a call made by a service connector.

FIX – Service Policy REST Status Codes - When a Rest Service Policy was configured to exclude http status codes that WCF considers communication exceptions, the Rest Service Policy would not work correctly. Now when a status code like 404 is added to the list of excluded http status codes, responses with that code will be excluded from the policy's retry and failure logic. Also, the Neuron ESB Message Header FaultType property will now be set to None when the REST Service Policy is set to exclude a protocol or communication exception.

OAuth Providers

FIX - OAuth providers cannot create SSL/TLS secure channel - modified the OAuth providers to negotiate which version of SSL/TLS to use when making OAuth calls. Previously, users could encounter the following error: "The request was aborted: Could not create SSL/TLS secure channel."

Logging

FIX – Service Connector Logging Inconsistencies – Some log entries could be written to unrelated Service Connector log files rather than the Service Connector log file they were targeted for.

FIX – Message Audit Log Entries – Some log entries could be written to log files not related to the target log file.

Neuron 3.7.0.634

Business Processes

NEW – MQSeries Process Step – Added the ability to configure the process step using either Username and Password or SSL. SSL requires that the binding property be set to Client. This makes the process step consistent with the MQSeries Adapter.

FIX – C# Process Steps - When editing code in a c# or c# class process step, the intellisense completion would disappear once a user has pressed any key after it pops up. The expected behavior is that the completion list should remain open and filter based on the characters entered.

Neuron ESB Explorer

FIX – Checking local MSMQ Queues could cause crash – When using the Neuron ESB Explorer to check for the existence of local queues, the Neuron ESB Explorer could crash, reporting a fault in msftedit.dll.

Adapters

New - Salesforce adapter - Added support for versions 47 & 48 of the SOAP API.

FIX – Salesforce Metadata generation - fixed salesforce related schemas generated so that messages would validate against them. Also modified the metadata generation for outbound messages. The wizard now will query the Salesforce account for configured outbound messages and only display those as options for metadata generation. Also fixed the generated schemas so that outbound notifications received from Salesforce will validate.

Neuron 3.7.0.630

Adapters

NEW – AlphaTrust Adapter – This is a new adapter that has been added to provide connectivity to AlphaTrust Document Process Automation system.

NEW – Rabbit MQ Adapter – Added support for dynamically setting the Rabbit MQ replyto and replytoaddress properties on outbound messages. The specific 4 properties which can be set on the outbound Neuron ESB Message are (where “rmq” is the adapter prefix property):

- `rmq.ReplyTo`
- `rmq.ReplyToAddress.ExchangeName`
- `rmq.ReplyToAddress.ExchangeType`
- `rmq.ReplyToAddress.RoutingKey`

FIX – Dynamics CRM Plugin – The installers were missing the CRM SDK assembly, `Microsoft.Xrm.Sdk.Workflow.dll`.

NEW – Adapter Metadata Generation Wizard – Added support for adapters that can generate JSON schemas as part of their metadata.

Business Processes

MOD – Flat File Parser – The dependent FileHelpers library has been updated from 3.1.5 to 3.4.1.

Neuron ESB Installer

FIX - .NET required framework would fail to install – During the installation of Neuron ESB, if the installer detected that the required version of the .NET Framework (i.e. version 4.7.2) was not installed, it would attempt to download the installer from Microsoft's web site and install it. Unfortunately, Microsoft changed the URL which caused the installer to fail during download. The URL has now been corrected.

MOD – Installation Powershell Script Patch – The description section of the patch README file still had a reference to 3.6.

Neuron ESB Runtime

MOD – Memory consumption increases with number of Service Endpoints – If a solution had hundreds (upwards of 600) of Service Endpoints defined, on startup, the Endpoint Host hosting the services could easily consume up to 6 or 7 gigabytes of memory on startup. This has been corrected to not take up more than 1 gigabyte of memory.

Neuron 3.7.0.603

Adapters

FIX - Azure Service Bus Adapter - improved performance when using batching in publish mode.

FIX - Azure Service Bus Adapter fix - When polling for messages from a queue that has sessions enabled, an exception would be thrown when there are a large number of message sessions. The exception would include the text "The service was unable to process the request; please retry the operation."

FIX - SharePoint Subscription Adapter - Fixed issue where SharePoint returned the error "Column 'xxx' does not exist. It may have been deleted by another user. This would occur when attempting to create/update a list item with custom column names.

FIX - SharePoint Subscription Adapter - Attempting to update an existing attachment in a list may fail.

NEW - SharePoint Subscription Adapter - Allows the addition of the Neuron ESB message body as an attachment when creating a list item.

MOD - REST based Adapters - When using a REST based adapter, if an unauthorized exception was returned by the service being called, an error similar to the following would be reported:

"The <adapter endpoint>, '<name of adapter>', failed to send the message. Calling the 'https://myserver.api.crm4.dynamics.com/api/data/v9.1' url after retrieving a new Token still resulted in a 401: Unauthorized error. cd46265e-14f8-43c6-b4ec-c7486675a51b"

This message has been modified to now include the error description returned by the underlying HTTP call.

MOD - Dynamics XRM SDK - Upgraded From 8.2 to 9.0.2.21.

FIX – MQSeries Adapter – Could throw an Object Not Found exception when listing the Queues in the Neuron ESB Explorer

MOD – MQSeries Adapter – The .NET API for IBM MQ (i.e. amqmdnet.dll) has been updated from version 8.0.0.12 to 9.1

NOTE: Starting with Version 8 of IBM MQ, MSMQ Topics with Transactions enabled are no longer supported when an MQSeries Adapter Endpoint (either subscribing or publishing to that topic) also has Transactions enabled. IBM has essentially discontinued support for this type of Distributed Transaction Scenario. For example, if a message is published to an MSMQ Topic with transactions enabled, and an MQSeries Adapter Endpoint (with Transactions set to True) is subscribing to the message, the following error would be generated:

*WebSphere MQ Adapter - WebSphere MQ error ', 2012 : Reason Code: 2012', occurred while trying to send message to Queue, 'X'.
CompCode: 2, Reason: 2012*

The immediate way to correct the error would be to either turn off transactions on either the Topic or the Adapter Endpoint.

Logging

FIX – Master Log file ignored Verbose Logging – If Verbose logging was configured for the Neuron ESB Runtime, DEBUG statements would not be written to the Master log file. This has been corrected.

Licensing

FIX – License Validation would fail on startup - During startup, the Neuron ESB Service could throw an error indicating the license is not activated, even after it has been activated against the Peregrine Connect License server.

Installation

MOD – Bootstrapper Application – The Neuron ESB Installer’s bootstrapper application, “InstallNeuronESB.exe”, is now being digitally signed. Previously, if a user launched it, the Windows Operating System would display a message stating that the publisher was unknown. Previously, we were only signing the MSI package.

Configuration

FIX – Swagger Docs Not Accessible at Runtime – Although the Swagger Document collection of the Neuron ESB Configuration object was accessible at design time, it was not accessible at runtime. An empty collection would always be returned.

Business Processes

FIX – Compression Process Step – When in Verbose logging mode, the Zip Encryption Password (if entered) would be written to the log file.

Neuron 3.7.0.576

Business Processes

FIX – C# Class Process Step - When adding an assembly reference in the code editor from the GAC, it would throw “Value cannot be null” error on the first one added. Additions from the GAC after that would work, but the first referenced assembly would not be accessible in the code editor.

Auditing

FIX – Message Viewer Fails with Older Messages – After upgrading the Neuron ESB Database, the following error may be reported if viewing messages that were Audited prior to build 3.6.0.1176. This was due to format changes made to the serialization of the ESB Message. This has been corrected.

System.Runtime.Serialization.SerializationException: There was an error deserializing the object of type Neuron.Esb.ESBMessage. Element 'item' was not found

Neuron ESB Runtime

FIX – Custom Counters File View of Memory Error - Customers may encounter an error like the follow: Could not initialize performance counter for ‘<some name>’. Custom counters file view is out of memory.

Neuron ESB Explorer

FIX – Connect to Remote Servers – Previously users could not connect to remote Neuron ESB servers. Also, previously the status displayed in Neuron ESB Explorer Toolbar was from the local server even if a user was connected to a remote server. Now it displays the remote server's status.

FIX – Securing Parties with ACL – Users were unable to select an Access Control List to secure a Publisher or Subscriber

Adapters

NEW - Salesforce Adapter - Added support for setting SOAP headers when using the SOAP API. Certain Salesforce objects may require SOAP headers for processing. You can now add these SOAP headers to the Salesforce request message in the same manner as you would for adding SOAP headers to a Service Connector request. For example, to add the AssignmentRuleHeader SOAP headers, use this code in a C# step in a process:

```
string headerKey = "urn:partner.soap.sforce.com" + "/" +
    "AssignmentRuleHeader";

string headerValue = "<AssignmentRuleHeader
    xmlns=\"urn:partner.soap.sforce.com\"><useDefaultRule
    xmlns=\"urn:partner.soap.sforce.com\">true</useDefaultRule></Assi
    gnmentRuleHeader>";

if (!context.Data.Soap.Headers.ContainsKey(headerKey))
    context.Data.Soap.Headers.Add(headerKey, headerValue);
else
    context.Data.Soap.Headers[headerKey] = headerValue;
```

You must use the namespace “urn:partner.soap.sforce.com” as the beginning part of the key. Any SOAP headers that are included in ESB Message’s Soap.Headers collection that do not use urn:partner.soap.sforce.com will be ignored by the adapter.

FIX – SFTP Adapter – When using a Proxy that does not require user credentials, an authentication error would be returned.

MOD – Azure Service Bus Adapter – If chunking was enabled for large message support, it would only be activated if Azure returned a MessageSizeException. However, in some circumstances, if Azure is misconfigured, a QuotaExceedException could be thrown. We now check that exception and, if it is due to message size, Neuron will chunk and send the message as expected.

FIX – Rabbit MQ Adapter - Updated Rabbit MQ adapter to use same serialization logic for the ESB message when in Mandatory routing mode that the Neuron ESB Topic channel uses. this was done so that all the properties are represented.

Installation

MOD – Certificate Signing – The Neuron ESB installer has been configured to use an updated code signing certificate. The powershell scripts shipped with Neuron ESB have also been updated with the new signature using the new certificate.

MOD – Installer detecting .Net 4.7.2 – The Neuron ESB installer has been modified to detect the presence of the Microsoft .NET Framework 4.7.2. If not installed, the user will be prompted to install it before continuing with the installation of Neuron ESB.

Workflow

FIX – Audited Messages from Workflow Tracking – Unable to audit pending messages from workflow tracking. The necessary parameters were not being set since the removal of the binary body from auditing. Now we are setting the necessary parameters.

Endpoint Health

FIX – Status incorrect when using REST API - Fixing endpoint host and workflow endpoint status when using the StartAll and Shutdown endpoint host REST API. Previously, the endpoint host would show Stopped status when shutdown was called via the REST API, but would not show Started when StartAll was called afterwards. Also, workflow endpoints would show as Started when Shutdown was called on the endpoint host and would stay in the Started state even though the endpoint was stopped. This only happened when using the external REST API, rather than the UI to shutdown or start the Endpoint Host.

Neuron 3.7.0.546

This build requires a database update. To update the database, all workflows must be in a completed state, or a state where the workflows will never be resumed (i.e. cancelled, terminated, PURGE-PENDING, or error processing).

Workflow

MOD – Normal and Request/Reply Workflows do not automatically resume - If a user had normal workflows that unloaded due to stopping the ESB Service gracefully, or from persistent delays/receive message activities, the workflows would not automatically resume when the ESB Service and its associated Endpoint Hosts were restarted. Now they are restarted upon startup of the endpoint host.

MOD – Unable to restart Workflows - Users can now start and restart workflows with the "Instance Locked" state from workflow tracking.

FIX – Receive a message during Cancellation - The receive message activity could receive a message while it was being cancelled (by a Timeout activity, for example) but the message would be lost due to the cancellation. Now it is audited as a failed message.

FIX – Workflows may not resume during Failover - During endpoint host failover, if the server failed over back to the originally failed server, some workflows would not resume properly.

FIX – MultipleLockOwnersNotSupported exception - If a user cancelled, aborted, or suspended a workflow from workflow tracking, another workflow lock owner would be created by workflow foundation. This was due to the mentioned operations creating a new workflow instance store object, which creates the extra lock owners. This could result in a "MultipleLockOwnersNotSupported" message and most subsequent workflows would abort with that message if the workflow has a persistence step (including a persistent delay or timeout).

To implement this fix, all workflows must be in a non-resumable state and all System.Activities.DurableInstancing tables must be truncated while the ESB Service is stopped. The following SQL Script can be run to truncate the necessary tables:

```
TRUNCATE TABLE [System.Activities.DurableInstancing].[IdentityOwnerTable];
TRUNCATE TABLE [System.Activities.DurableInstancing].[InstanceMetadataChangesTable];
TRUNCATE TABLE [System.Activities.DurableInstancing].[InstancePromotedPropertiesTable];
TRUNCATE TABLE [System.Activities.DurableInstancing].[KeysTable];
TRUNCATE TABLE [System.Activities.DurableInstancing].[LockOwnersTable];
TRUNCATE TABLE [System.Activities.DurableInstancing].[RunnableInstancesTable];
TRUNCATE TABLE [System.Activities.DurableInstancing].[ServiceDeploymentsTable];
TRUNCATE TABLE [System.Activities.DurableInstancing].[InstancesTable];
```

FIX – Workflows would not automatically resume when failing over to another machine. If a user had an endpoint host with a workflow endpoint, with the endpoint host in a primary-failover setup, upon failing over, the workflows would not be resumed correctly, if at all. Now they are being resumed properly. Note: Users may see more warnings upon failover. Also, upon failover, users may see the endpoint host stay in the starting state for approx. 5 minutes. This is due to the lock mechanism implemented by Workflow Foundation. There is a 5 minute period where a workflow instance is still "locked" by the endpoint host that failed. This also means messages sent to the workflow endpoint won't be processed until all workflows that were running in the failed endpoint host are reloaded. Messages can still be sent to the workflow endpoint during the 5 minute period, but no new workflows will start until the period has elapsed.

FIX – Workflow Message Loss during Shutdown - In some scenarios, such as if there were messages being processed by the workflow endpoint and the ESB Service was stopped gracefully, the messages in memory could be lost. Now they will be audited to the failed messages table.

FIX - In some cases, the scheduled jobs for persistent delays would not be deleted from the database upon execution. An example case would be if the workflow did not complete a timeout activity, which uses a persistent delay internally, due to an activity inside the timeout aborting.

Neuron Runtime

FIX – Endpoint Host Restart Counter was not being reset after a successful restart – If an Endpoint Host goes into restart mode and then eventually restarts, the restart counter was not reset. Hence, if it went in restart mode again, it may not retry as many times since the retry counter never got reset to zero after successful startup.

FIX – Memory Leak – A possible memory leak could occur at runtime when we attempt to clone our configuration and a decryption exception occurs during the cloning process.

Auditing

FIX – Message Body would always be audited when IncludeMessageBody property was set to false - There was an issue with Audit Process Step where we weren't honoring the "IncludeMessageBody" and "IncludeCustomProperties" options. They would always be audited.

Adapters

FIX – SFTP Adapter – an empty value for the name could appear when an error connecting would be reported. i.e. "SFtp adapter failed to send the message with file name, (), to the SFTP Server 'xx.xx.xx.xxx'".

FIX – Adapter Policy set to 1 retry would not fire - If a policy had a retry of 1, it would not be fired.

FIX – SFTP Adapter – If a load balancer is in front of the SFTP Adapter configured for Publish mode, if the load balancer fails and then is restored, the SFTP Adapter may not recover because it has cached some of the connection resources. This issue could result if an appliance like NetScaler is used. Previously we were only disconnecting after every poll, not we completely dispose of the SFTP resources.

FIX - SAP Adapter – A memory leak would occur when receiving IDocs from SAP that contain namespaces as part of SAP names (i.e. segments).

Neuron ESB Explorer

MOD – SetProperty Validation added - Added validation in SetProperty() method to prevent a "." from being used as the prefix or start of a prefix for a custom property.

FIX – Message Viewer would truncate custom property names - When viewing custom message properties using the message viewer either from Message History or Failed Message reports, the "name" component of the custom property could be truncated.

Neuron 3.6.0.1181

Monitoring

FIX - Endpoint health - Neuron Explorer would not show the correct stats for Message Rate, Messages Processed, Rate, and Completed for items hosted in the ESB Service and Endpoint Host.

Runtime

MOD – Subscriber not found during startup – Modified runtime to detect subscribe mode adapters and to start them before publish mode adapters. Subscriber not found errors could result on startup otherwise.

Adapters

NEW – MQSeries Adapter – Added the ability to configure the adapter using either Username and Password or SSL. SSL requires that the binding property be set to Client.

NEW – Rabbit MQ Adapter – Users can now enter multiple machine names in the "Machine Name" field if using multiple rabbit mq servers configured for HA.

Neuron 3.6.0.1176

Adapters

MOD – IBM MQ Series Adapter – Enclosing multiple name value data pairs within a dummy root element when reading MQRFH2 headers from a message retrieved from a Queue.

FIX – Metadata Generation Wizard – When producing documents for the Salesforce adapter an “a object reference not set the instance of an object error” could sometimes be thrown.

Messaging

FIX – Key not found error - Key not found error that would result if a party that had different types of topics (i.e. different transports) associated with it were changed

Business Processes

FIX- Sign and Verify XML Process Steps - We were not able to retrieve the RSA key as private key. The following error would be generated: Invalid algorithm specified.

Samples

FIX – Client Samples and Log4Net – Many of the samples shipped with Neuron ESB did not contain the proper Log4Net configuration within their app.config files.

Auditing

MOD – Storing of Binary ESB Message – Neuron ESB no longer stores a compressed binary representation of the Neuron ESB message when auditing the message. We previously did this in addition to our table properties. We’ve not expanded the table properties and removed this. This change requires a Database schema update. For large messages, this should reduce database size.

Runtime

FIX – Failover using SQL Server Always On - In the case of a Neuron instance using an Always On SQL Server database, during SQL server failover the instance store used for workflow could fail to renew its lock in the database. This would cause an error to occur that states that an instance handle has become invalid. Now we have created a custom SQL workflow instance store class that utilizes our transient fault block so that upon failing to connect during a lock renewal, the instance store will retry. Also, the time between when the host lock is scheduled for renewal and the time that the lock expires has been increased to 1 minute. This effectively gives the database a minute to come back up in the case the database goes down right when the host lock is set to renew.

Neuron 3.6.0.1168

Workflow

FIX – Error could occur when SQL Failover occurs - Occasionally, after a SQL Server Always On cluster database failover occurred, any persistence commands for workflow would fail and cause the workflow to abort. The issue would only occur if the lock expired during the failover (i.e. the workflow host couldn't renew the lock in the durable instancing table since the database was unavailable). This was due to the lock renewal period being too short on the workflow instance object. This has been increased from 1 minute to 5 minutes. The error thrown would state “the instance owner registration for owner ID ‘<GUID>’ has become invalid”.

FIX – Unloaded workflows unable to load – This can occur when workflow activity execution tracking is disabled on the Workflow Endpoint. When Neuron attempts to resume an unloaded workflow, an exception would be thrown stating that a returned database value was null. This was due to the activity

data not being available to compare against the currently executing activity. This issue would cause unloaded workflows to get stuck as unloaded.

FIX – SQL Server could not be reached if database connection pool settings were too low – If the max connections in the database connection pool was too low to support the concurrent number of workflows running, errors could occur indicating the SQL server could not be reached. This is been modified to our Reliable SQL Connection pattern.

FIX – SQL Always On Failover would fail to failover to Secondary database - This fixes an issue where Neuron ESB would record continuous errors happening in workflow during an SQL server failover when using an Always On Availability Group. The errors would be in the format of the following:
"Unable to access availability database '<DatabaseName>' because the database replica is not in the PRIMARY or SECONDARY role. Connections to an availability database is permitted only when the database replica is in the PRIMARY or SECONDARY role. Try the operation again later." There could also be an error thrown saying the secondary database is read-only in certain Always On Availability Group configurations.

FIX – SQL Deadlocks could occur when checking for messages – Neuron ESB periodically polls the SQL database to see if there are previously suspended workflow endpoint messages available to be processed. This could cause deadlocking to occur if there was a lot of IO on the WorkflowQueuedMessage table. This same issue may also cause the “Receive Message” activity to fail under high database load since it wouldn’t be able to read the correlated messages off the table.

Database

FIX – Purge and Backup jobs could cause SQL deadlocks – The Neuron ESB purge workflow tracking and purge backup database jobs have been modified to generate the SQL to execute the deletes one by one on the tables to reduce locking. This fixes an issue where SQL exceptions could occur when the jobs ran and there was a lot of database activity occurring at the time.

Adapters

FIX - Salesforce Adapter - The adapter includes two options for which data format to use when making REST calls to Salesforce.com – XML or JSON. Under certain circumstances the selected option would be ignored and only return JSON-formatted data. This has been resolved.

NEW - Salesforce Adapter – Added support for SSL/TLS when receiving Salesforce.com’s outbound messages in publish mode.

FIX - SAP Adapter - previously, the adapter would fail when receiving IDOCs from SAP that contain namespaces in either the IDOC or segment names.

FIX – Object reference not set to an instance of an object - When attempting to read custom properties from custom adapters, the Neuron ESB Explorer could display the following error:

Object reference not set to an instance of an object.

This could happen when a custom adapter exposes an object as a custom property and that object was never initialized with a default value.

FIX – Active Directory Adapter – When attempting to clear a value from a property using the Edit transaction type, the following error would be thrown:

DirectoryServicesCOMException

The attribute syntax specified to the directory service is invalid

NEW – Active Directory Adapter - Modified the edit and create transaction types to support binary content by adding support for the binary=true attribute on the element to be edited/created.

FIX – IBM MQ Series Adapter – A thread abort could be thrown when the adapter is disconnecting on shutdown.

MOD – IBM MQ Series Adapter – Added support for multiple name value data pairs when reading MQRFH2 headers from a message retrieved from a Queue.

MOD – SFTP Adapter - Modified to only support version 3 of the protocol rather than attempt to negotiate with servers for the version.

Business Process/Workflow Activities

FIX – Service Endpoint Step/Activity - modified the "Throw Exception On Fault" property to throw an exception when the service endpoint binding is REST and the exception is a communication exception. Otherwise the REST binding will not throw an exception on fault.

Neuron ESB Explorer

FIX - Samples – fixed the issue where the Client API samples would throw an exception regarding a missing Log4Net configuration section in each of the Visual Studio projects.

FIX – Neuron ESB Test client – when doing bulk send and receive, the msg/sent/received per second would not increment.

FIX – Import Config would fail to import Business Processes - Processes were not importing due to comparing the wrong values.

Neuron ESB Runtime

FIX – Restarting TCP Topics from Endpoint Health results in Error – When attempting to restart an TCP Topic from Endpoint Health when there are Endpoint Hosts defined and running could result in an Port in use exception. This occurred because the Endpoint Host would inherit the handle of the TCP Topic's server port. This has been resolved.

Neuron 3.6.0.1132

Business Processes

FIX – Websphere MQ Process Step – Modified to support showing list of Queue Managers if using Version 8.x from design time property grid.

FIX – Websphere MQ Process Step – When passing an MQRFH2.NameValueData string whose length is not a multiple of 4, the following error is would be returned by the application retrieving the message:

Invalid folder size of "67" detected within a WebSphere MQ MQRFH2 header. The folder size defined in an MQRFH2 header must be a multiple of four bytes, and must not be negative. The message being processed does not conform to this rule.

NEW – Adapter Endpoint Process Step – Users can now dynamically set the value of Service header property of the ESB Message object with the name of the Adapter Endpoint to execute at runtime. If set, the Adapter Endpoint Process Step will use that value for the “Adapter Endpoint” property. To work, the Adapter Endpoint Process Step must NOT be configured manually at design time. The “Adapter Endpoint” property must remain blank. Below is an example of setting the property at runtime using a C# Process Step:

```
context.Data.Header.Service = "FileAdapterEndpointOut";
```

MOD – Salesforce Adapter – SSL 3 support has been removed from the adapter.

Adapters

FIX – IBM MQSeries Adapter – Modified to support showing list of Queue Managers if using Version 8.x from design time property grid.

FIX – IBM MQSeries Adapter – When passing an MQRFH2.NameValueData string whose length is not a multiple of 4, the following error is would be returned by the application retrieving the message:

Invalid folder size of "67" detected within a WebSphere MQ MQRFH2 header. The folder size defined in an MQRFH2 header must be a multiple of four bytes, and must not be negative. The message being processed does not conform to this rule.

NEW – NetSuite Adapter – Added Send Timeout as a property that can be set at design time

NEW - Salesforce Adapter – Added support for the Salesforce Bulk API. The Bulk API is a Rest-based API that can be used to asynchronously insert, upsert, update, delete or query many records from Salesforce. To use the Bulk API, in the Salesforce Adapter endpoint properties grid set the OAuth Connection property to True, and then set the Web API property to Bulk. These settings will allow you to make any calls to the Bulk API as described here: https://developer.salesforce.com/docs/atlas.en-us.218.0.api_async.meta/api_async/asynch_api_intro.htm. You are responsible for creating and closing the job, adding batches to the job and any monitoring you wish to add inside your solution. While using the Bulk API can be done within a business process, it is recommended that you use Neuron ESB Workflows for creating jobs, adding batches and monitoring the progress of the batches.

Workflow

FIX – Correlated Send/Receive Workflow Sample - The correlated send and receive sample did not work due to the removal of subtopic support for workflows. This has been fixed by changing the topic for the "OrderResponder" party from "Orders.In" to "Orders"

FIX – Correlated Workflow Sample - The Correlated Workflow sample was wrong. The receive message activity was missing and there was an extra message variable. The message argument was of type string, which was also incorrect.

FIX - Normal Correlated Send and Receive Workflow - These did not work the way they were intended when running in a parallel activity, or when there were more than one send and receives in a workflow. Users would see stuck messages and other undesired behavior such as extra workflows.

FIX – Correlated Normal Workflow Behavior - The behavior of Normal Type Correlated Send and Receive workflows was changed. The behavior is now as follows: Every message sent in will start a new workflow instance unless the message has the corresponding correlation set elements set in the message in which case the message will try to correlate to a running workflow instance. If a running workflow with the correlation Id cannot be found, a new workflow instance will be started.

MOD – Correlated Workflow Fails to process Correlated Message – Inbound messages to a Correlated Workflow may fail with the exception below because a process before the Receive Message activity is taking longer than 10 seconds to execute, delaying the Workflow from entering a Wait state. This timeout has now been extended to 3 minutes.

Exception Type: NeuronEsb.Server.Runner.Workflow.WorkflowException

Exception Message: Message "596a1cbb-1d77-4b91-951c-979ae803eb25" was received on topic "<topic>" from party "<party>" for workflow endpoint "<endpoint>", but has failed while reprocessing it due to the retry count exceeding the limit. This also means the workflow state could not be determined possibly due to the sql server being overloaded

Monitoring

FIX - Workflow Endpoints Stats incorrect - The Terminated state was incorrect and would be reported as Completed. Suspended messages also were reported incorrectly. Active and Cancelled state also were not incrementing and decrementing correctly. The Neuron ESB Explorer will also now display a Terminated state column.

FIX – Logging to incorrect Log File – Under certain circumstances, log entries generated from the execution of a Business Process may be recorded in the wrong Neuron log file.

Neuron ESB Explorer

FIX – Register Adapter reports error - When clicking new to register an adapter, user would receive the following error:

Error Loading Assembly. File: path to the adapters folder

The system cannot find the file specified. (Exception from HRESULT: 0x80070002)

Neuron 3.6.0.1117

Adapters

FIX - Dynamics AX Adapter – Fixed issue with design-time loading of available service URLs in the AIF. Depending on your security settings, you may receive an error. A “Property Validation Error” dialog would appear, and when you click on “Show Details”, this is the error received:

System.ServiceModel.Security.SecurityNegotiationException: A call to SSPI failed, see inner exception. ---> System.Security.Authentication.AuthenticationException: A call to SSPI failed, see inner exception. ---> System.ComponentModel.Win32Exception: The target principal name is incorrect

This was due to the adapter only using an SPN Endpoint Identity Type when retrieving the list of services. The adapter now uses the “Endpoint Identity Type” and “Endpoint Identity Value” at design time when retrieving the list of Service URLs. If you still receive the above error after upgrading to this version of Neuron ESB, then the Endpoint Identity Type and Endpoint Identity Value are incorrect. Often, the correct values are “Dns” as the Endpoint Identity Type and no value for the Endpoint Identity Value.

FIX – Sharepoint 2016 plugin – Fixed UI to align text with checkboxes

FIX - Microsoft Exchange Adapter - To work with latest Microsoft exchange office 365 online servers – User may get receive the following when autodiscovery is set to true: If an Autodiscover server returns a redirect HTTP status code, this method will generate an AutodiscoverLocalException with the Message property set to a string such as "Autodiscover blocked a potentially insecure redirection to <https://autodiscover.contoso.com/autodiscover/autodiscover.xml>".

MOD - SFTP Adapter – When Uploading a file to an SFTP server an error, similar to the one below, may occur:

Xceed.SSH.Sftp.SftpStatusResponseException was caught

HRESULT=-2146233088

Message=Sftp.OpenFile("/Demo/mydoc.txt") got a status: SSH_FX_OP_UNSUPPORTED:
Unsupported operation.

This error can occur if, when connecting to the server, the server re-negotiates the connection, requesting that Neuron ESB use version 6 of the SFTP protocol. Unfortunately, some SFTP Servers claim they support version 6 of the protocol, but in practice they fail to support even basic functionality of that protocol. Since the SFTP Server re-negotiated the connection to version 6, we use flags and parameters supported for that specific version which in turn the server does not support. Moving forward, the SFTP adapter will not renegotiate and will support Version 3 of the SFTP protocol by

default. However, users can override the version and/or choose the old behavior of Negotiating the protocol by modifying the new “Protocol Version” property.

FIX - SAP Adapter - Fixed issue with sending IDOCs to SAP. IDOCs containing multiple segments of the same type would fail to upload.

FIX – Azure Service Bus Adapter – if the ContentType property of the BrokeredMessage object is null, a null exception would be thrown when receiving the message while in Publish mode.

NEW - SharePoint Adapter – Added support for setting multi-valued managed metadata properties for list and document items. The format of the property value is the property label, “|”, and then the guid of the managed metadata item. For multiple values these would be separated by a semicolon:

```
Happy|b62f3710-e8e0-40b2-b4d6-c844f2e74243;Blue|7e0b1e7e-d490-4448-84c4-7d354ec0b917
```

The “wss id” does not need to be included in the meta data properties, the adapter will automatically add that.

The ability to set the same list or document item managed metadata properties to a folder if that list or document item is saved to one has also been added. This is controlled by a new Boolean adapter endpoint property called “Set Folder Properties”. When set to true:

- For single-value managed metadata properties, the value set on the list or document item will replace whatever value the folder currently has
- For multi-valued managed metadata properties, the value(s) set on the list or document item will be added to the values the folder current has

Business Processes

FIX – Custom Process Steps fail to load – After loading custom DLLs into Pipelines folder, an error similar to below may appear when attempting to open a Business Process referencing a custom process step.

The 'name of process' Business Process could not be displayed. The composition produced a single composition error. The root cause is provided below. Review the CompositionException.Errors property for more detailed information.

- 1) Unable to sort because the IComparer.Compare() method returns inconsistent results. Either a value does not compare equal to itself, or one value repeatedly compared to another value yields different results.

FIX - Split-Join Process Step – Fixed issue where context properties being set when using code based splitting prior to the split-join step are not available inside the split-join step. Now the context properties can access with each split message. Be advised when setting or modifying contexts properties – when the Synchronous property is set to True, the last split to change the context property will be the result you see in that property after the join. If Synchronous is set to false, the result of the context property after the join will be unpredictable.

FIX – Timeout Process Step – If used within a Split Join and the Split is set to run asynchronously, any process steps placed within the Timeout step will appear not to run or run just once. That is because an exception was being thrown internally but was not being reported by the runtime. The exception condition has been fixed and error reporting has been extended so that the runtime will capture all internal exceptions.

FIX – Execute Process Step – If used within a Split Join and the Split is set to run asynchronously, any process steps placed within the Business Process that the Execute step is configured to run may appear not to run. That is because an exception was being thrown internally but was not being reported by the runtime. The exception condition has been fixed and error reporting has been extended so that the runtime will capture all internal exceptions.

FIX - Service Endpoint Process Step – If the Throw on Fault is set to True, and the service called is SOAP based, if an error is thrown, the message body contained the original request message rather than the error information returned from the called SOAP service.

Workflow

MOD - Persistent Delay Activity – The “FromTimeoutActivity” argument in the persistent delay activity has been removed from the User Interface Property Grid.

FIX – Adapter Endpoint Activity Memory Leak – When processing multi megabyte messages through the Adapter Endpoint workflow activity, memory will gradually increase overtime. The amount of time for the increase would be determined by the number of messages and their size. .NET Garbage collection was not cleaning up these messages. This is now resolved.

Endpoint Health

FIX – Workflow Endpoint - When a workflow endpoint is stopped within Endpoint Health, it will stop but it did not update its state appropriately.

FIX – Endpoint Host – If an Endpoint Host has been restarted through Endpoint Health, it would fail to update the statistics of the Endpoint Host like processId, lastheartbeat etc.

Service Endpoints

MOD – ADP OAuth Provider – After 60 minutes the OAuth Token would expire and not renew. This was due to ADP throwing a protocol exception rather than a message security exception. In previous versions of Neuron ESB, the Token automatically renewed when expired and would attempt the call again. Now, although the Token is still automatically renewed, the attempt to call again will not be made. The protocol exception will be thrown instead. The user will need to configure a Service Policy to catch the specific protocol exception and configure the policy to retry the call.

MOD – HTTP Query properties – These are now case insensitive.

FIX – CORS Fails with Custom Header – If a custom header is passed to a Service Endpoint, an error like the one below may be returned.

Access to fetch at 'http://someurl' from origin 'null' has been blocked by CORS policy: Request header field client is not allowed by Access-Control-Allow-Headers in preflight response.

Neuron 3.6.0.1094

Service Endpoints

FIX – Client Connectors become disabled when in Business Process mode – A Client Connector may become disabled if a previously referenced Party is disabled. This only occurred if a Client Connector was once previously associated with the disabled Party, but was later changed to run a Business Process directly.

FIX – Transfer-Encoding on POST throws exception – When calling a REST service with the POST verb, the following error below may be recorded by the Global Exception Handler. This only occurred if using the Service Endpoint Process Step to call a REST service:

Exception: The 'transfer-encoding' header must be modified using the appropriate property or method.

Parameter name: name

Method: ThrowOnRestrictedHeader

Adapters

NEW - SAP Adapter - Significantly improved performance when loading the list of RFC operations from the OTHER category.

FIX – SAP Adapter – When using transactions, the connection object that was cached was never being closed. This could result in SAP returning an error indicating that the Maximum number of conversations has been exceeded.

NEW - SharePoint Adapter - Added support for using content types when creating/updating document libraries or lists.

NEW - SharePoint Adapter - Added feature to allow adding and updating items to a list or library folder. If the designated folder does not exist, the adapter will automatically create it. The folder name can either be set as an adapter endpoint property or passed-in using the message property “sps.FolderName” with the name of the folder you want to create.

FIX - SharePoint Adapter - Fixed problem when trying to update a list item using the adapter and an exception was returned with the text “Failed updating list Neuron List, WSS returned the following error Message: Guid should contain 32 digits with 4 dashes (xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx)”.

NEW - Azure Service Bus Adapter - Added the ability to send and receive messages larger than 256K or larger than whatever the Azure Service Bus queue limit is defined as. To support this, the Queue being referenced MUST have Sessions enabled. For Publish mode adapter endpoint, there is a new property named, “Receive chunked messages”. If set to true, the Queue referenced must have sessions enabled.

All messages received will be checked to determine if they are part of a batch of messages, and if so, they will be aggregated into their original message. For Subscribe mode adapter endpoints, there is a new property named, "Chunk large messages". If set to true, it will allow users to send messages larger than what the target queue is configured to receive. This is done by splitting the messages into smaller components.

Workflow

FIX – Stuck messages for Correlated Workflows - In the case that a message came in for a running correlated type workflow at the same time the workflow was unloading, the message could get stuck in the pending messages queue since the state of the workflow was detected as running, when in fact it was unloaded. Now we check to see if there are any pending messages while unloading and reprocess them if there are.

FIX – Stuck messages for Correlated Workflows - In the case that a correlated workflow is running in an endpoint host setup to run on multiple machines with the hosts in a primary-failover configuration, and the originating endpoint for a message sent to the workflow is ran on more than one machine (i.e. the message sent to the workflow can originate on any machine), messages/workflow commands could get stuck in the pending messages queue. This occurred because the machine property on the message would be for the originating machine and not the machine that the workflow endpoint is running on.

FIX – XML/Message Transformation - The Workflow transformation activities "Transform XML" and "Transform Message" would fail under certain circumstances when the output XML would not result in valid XML (i.e. outputting HTML fragments). The error message received would state "Token Text in state Start would result in an invalid XML document. Make sure that the ConformanceLevel setting is set to ConformanceLevel.Fragment or ConformanceLevel.Auto if you want to write an XML fragment." These same transformations would work with the Transform - XSLT process step.

FIX - XML Schema Validation - in 3.6 the schema validation workflow activity would fail when the schemas contained an include or import directive with no error information. This functionality previously worked in 3.5.4.

FIX – Stuck Workflow Commands when Persistent Delay exists - In the case that a solution is configured to run an endpoint host on multiple machines marked as primary, workflow commands could get stuck in the pending messages tab if the workflow definition contained a persistent delay activity set to delay for more than a minute. This would cause a number of workflows to not continue after the delay activity. Also, the following warning message could be seen in the event log "Could not remove workflowId '<workflowInstanceId>' from delayedWorkflows collection".

FIX - Persistent Delay takes longer under heavy workloads - In the case that the Persistent Delay activity was being used in the workflow, the Quartz Scheduler used to resume the workflow could start behaving very slowly. This resulted in the delay activity delaying far longer than expected. This occurred mostly under high workflow loads. This fix requires a Neuron database update to version 31.

Neuron Test Client

FIX – BodyType property not set – When sending text (non xml) messages, the ESB Message Header property, BodyType, was not being set correctly.

Business Processes

FIX - XML Schema Validation - in 3.6 the schema validation process step would fail when the schemas contained an include or import directive with no error information. This functionality previously worked in 3.5.4.

FIX - HTTP Client Utility – The Selected method (i.e. GET, POST...) was not getting set during execution a URL value was not supplied.

Endpoint Host

FIX – Restart Options for Endpoint Host - Endpoint host would ignore the retry count and retry interval properties if both were set to any value of less than 2. It would default both to the value of 3. Retry Count can now be set to 0. If Retry interval is set to any value less than 1, it will default to 3.

Neuron 3.6.0.1079

Adapters

FIX – SAP Adapter – When using the Meta Data Harvesting Wizard to generate schemas, IDOC groups containing escaped characters would not expand to display IDOC types.

Security

NEW – Salesforce OAuth Provider – A new Salesforce OAuth Provider has been added to the list of OAuth providers that users can select and configure within the Neuron ESB Explorer.

FIX – ServiceNow OAuth Provider – When testing the Client Login within the Neuron ESB Explorer, if the test was successful, the following message would be displayed, "Azure Resource Owner Password Credentials OAuth Test Successful". This has been corrected to read, "ServiceNow Password Credentials OAuth Test Successful".

FIX – Thinkecture OAuth Provider – This provider would not initialize properly. This has been corrected.

Endpoint Health

FIX – Neuron Endpoint Host Logging stops – If a Neuron Endpoint Host was restarted through Endpoint Health, all logging that would typically occur within its respective Endpoint Manager log file would cease.

FIX – Exception thrown after restart from Endpoint Health – If a Client Connector was restarted manually using Endpoint Health, it would restart successfully. However, on receipt of the next incoming request message, it could throw an error similar to the one below:

Exception'. Message ID 'e38ef787-f8bc-42ce-b974-b926d280320b', Topic '', Source Party 'System'. Object reference not set to an instance of an object.. Message is being sent to Failed Audit System. Party ='System', Message ID ='e38ef787-f8bc-42ce-b974-b926d280320b'

This could only occur if a Client Connector was configured to execute a Business Process directly, rather than attached to a Publisher (Processing Mode set to Business Process), and the Business Process contained either an Adapter Endpoint, Service Endpoint, Audit or Execute Process Step. This occurred because on manual restart, the existing Business Process process steps were not being disposed of properly.

FIX – Neuron Endpoint Host Fails on Restart – The Neuron Endpoint Host may fail to restart if restart is initiated directly from Endpoint Health. An error similar to the following may be reported:

Unable to create the communication API for the Endpoint Host 'Neuron ESB Default Host' on URL 'http://localhost:51004/DEFAULT/NeuronESBDefaultHost'.

System.Exception: Unable to create the communication API for the Endpoint Host 'Neuron ESB Default Host' on URL 'http://localhost:51004/DEFAULT/NeuronESBDefaultHost'. --->
System.NullReferenceException: Object reference not set to an instance of an object.

This has been corrected and the Neuron Endpoint Host should restart

FIX – Neuron Endpoint Host does not restart on all servers - if an attempt is made to restart the Neuron ESB Endpoint Host by selecting “Restart Service on All Machines” on the context menu of Endpoint Health, the Endpoint Host would only restart on one machine, regardless of how many machines were configured in the Deployment Settings of the Endpoint Host. Additionally, the following warning would be logged if more than one machine was configured in Deployment Settings:

“Cannot Restart Endpoint Host because it is already running on machine : <Machine Name>”

FIX – Neuron Endpoint Host does not restart server - if an attempt is made to restart the Neuron ESB Endpoint Host by selecting “Restart Service” on the context menu of Endpoint Health, the Endpoint Host may not restart on the machine and instead the following warning would be logged.

“Cannot Restart Endpoint Host because it is already running on machine : <Machine Name>”

FIX – Restarting Neuron Endpoint Host would fail with NON Netbios names – If a value other than the NetBios name of the machine was used in the deployment groups, the Neuron Endpoint Hosts could not be restarted using Endpoint Health.

FIX - KeyNotFoundException on shutdown - When stopping the Neuron ESB Service, an error similar to the following could be logged:

“Operation Request: Failed to execute Operation Type 'Stop' against 'ClientConnector' ID of 'b225da48-73bb-4759-891d-b91e199f737b' in Endpoint Host 'Neuron ESB Default Host'.
System.Collections.Generic.KeyNotFoundException: The given key was not present in the dictionary.”

Neuron 3.6.0.1072

Audit Service

MOD – Audit Service and missing FailureType – When using the Audit Process Step or Workflow Activity, the Action can be set to Failure. In such circumstances, the Step/Activity should always be placed in the Catch block of an exception handler. In those cases, the Neuron runtime will automatically capture and populate the Failure Type and Failure Details before the message is audited. However users can use the Step/Activity outside of a catch block when the action is set to Failure. However, the user MUST set the failure type and failure details properties. When those are not set, users will see an error similar to one below in the Audit Service Neuron log file as well as the Neuron ESB event log:

2019-02-18 00:20:37.782-08:00 [405] ERROR - Neuron ESB Global Exception Handler - ProvideFault - Message Audit Service: The failure type is required

Parameter name: failureType

2019-02-18 00:20:37.797-08:00 [405] ERROR - Failed to write the message to the Neuron ESB Audit Database for message with 'fa04e783-373c-41c2-b46a-bd1974c2251d' Message ID. Source Party, <PartyID>, Source Topic, <TopicName>. The failure type is required

Parameter name: failureType

System.ArgumentNullException: The failure type is required

Parameter name: failureType

This would indicate that the properties have not been populated and the result would be that the message would not be audited. The Audit Service has been modified to populate the properties, if they are missing, rather than NOT audit the message, the properties will be provided the following value: “NOTPROVIDED”, so as not to prevent the auditing of the message.

Rabbit MQ Topics

FIX – Logging of Fault would fail - In the case where the Rabbit MQ channel faulted on startup of the Topic, the exception was not be logged properly, instead an index outside the bounds exception would occur.

Workflow

FIX – Quartz Transaction Deadlock – When running large amounts of workflows, the following error may be experienced:

2019-02-21 13:01:39.020-08:00 [ESBHostScheduler_QuartzSchedulerThread] ERROR - Couldn't rollback ADO.NET connection. Transaction not connected, or was disconnected

System.Data.DataException: Transaction not connected, or was disconnected

```
at  
Quartz.Impl.AdoJobStore.JobStoreSupport.CheckNotZombied(ConnectionAndTransactionHolder  
cth)
```

```
at  
Quartz.Impl.AdoJobStore.JobStoreSupport.RollbackConnection(ConnectionAndTransactionHold  
er cth)
```

2019-02-21 13:01:39.020-08:00 [ESBHostScheduler_QuartzSchedulerThread] ERROR - An error occurred while scanning for the next trigger to fire.

Quartz.JobPersistenceException: Couldn't acquire next trigger: Transaction (Process ID 252) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction. ---> System.Data.SqlClient.SqlException: Transaction (Process ID 252) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

This has been resolved.

FIX – Correlated Workflows not completing - In the case of a primary-primary workflow setup for load balancing, on high load, messages could get stuck reprocessing forever in the workflow host service with log entries stating the workflow state is empty. This results in correlated workflows (or normal workflows that use correlated receive and send patterns) not completing and messages getting lost.

FIX – Correlated Workflows not completing or messages not being processed correctly - In the case of a primary-primary workflow setup for load balancing, on high load, workflow state could return as empty. This would cause undesired behavior such as correlated workflows (or normal workflows that use correlated receive and send patterns) not completing, messages getting stuck in pending messages, and messages not being processed properly.

FIX – Failover results in duplicate workflows or workflows not running - For endpoint hosts, when in a primary-failover scenario, the workflow endpoints would not properly pick up and reprocess the workflows that were running on the primary in the event the primary went down. This occurred if the Neuron instances were not the same name. In the case that they were the same name, old workflow commands from the primary would not be cleaned up on failover. This resulted in duplicate workflows.

Neuron ESB Explorer

MOD – Reporting and logging ReflectionTypeLoadException - Modified how Neuron reports and displays ReflectionTypeLoadException information to the user and logs. Previously, we were not iterating through the collection of loader exceptions and displaying useful information. We also were not recording all the information within the Application Event Log. A ReflectionTypeLoadException exception could occur if dependent assemblies or types were missing from the path directory when launching the Neuron ESB Explorer.

FIX – Logging – All Errors, Warnings, and Informational and Verbose messages can now be logged to the Application Event Log. The level of logging can be controlled through the Neuron ESB Explorer’s config file (i.e. NeuronExplorer.exe.config), specifically in the “configuration\log4net\appender\filter\levelMin” node. The default minimum logging level is set to WARN. Users can configure this to either INFO or DEBUG.

Adapters

NEW – ODBC Adapter – Added support for binary Timestamp datatypes.

NEW – Dynamics AX Adapter – Added the ability to disable caching of the internal proxy used for communication. By default, “Connection Caching” is set to true, which means the same proxy will be used for subsequent calls. Setting the property to false will ensure a new proxy is created on every send. Additionally, the Security Properties have been renamed. When upgrading, users may have to reset the credentials on the adapter endpoint as it will fail to start up if security was previously enabled.

Business Processes

NEW – ODBC Process Step – Added support for binary Timestamp datatypes.

Samples

FIX – WSDL Service Sample – The Payment Service’s app.config file was misconfigured, containing a trailing slash after the payment service baseaddress. This has been corrected.

Neuron 3.6.0.1058

Adapters

FIX – File Adapter – The incorrect file size could be reported for files when a Publish based file adapter is consuming a large batch of files submitted concurrently.

MOD – SharePoint Adapter - added support for setting managed metadata properties (TaxonomyFields) in lists and libraries. You can set a managed metadata property the same way you would set other list or library properties, exception that instead of just providing a value, you need to provide the value of the Taxonomy term label (i.e. “My Property”) and the Guid of the taxonomy term in this format: label|Guid.

Business Processes

NEW – WebSphere MQ – This is a new process step that allows users to send messages to an MQSeries Queue, with all the same capabilities of the existing Neuron ESB WebSphere MQ adapter (including the same “wmq” message prefix). However, this process step also allows users to Query a queue to return either the first message in the queue or a message by its message id or correlation id property. When in Query mode, the message can be either read (removing the message permanently from the queue) or browsed (returning the message and its properties while leaving the original message on the queue i.e. peek). When querying by either Message Id or Correlation Id, their respective properties must be set on the inbound message before the query like so:

```
context.Data.SetProperty("wmq", "MQMD.MsgId", "323456");
```



```
context.Data.SetProperty("wmq", "MQMD.CorrelId", "xyz323456");
```

FIX – Service Endpoint Process Step - If there was a service fault or timeout exception, the proxy was not getting cleaned up. Normally this would not be an issue. However, if the service timeout exceeded a number of minutes, this could inadvertently corrupt the underlying proxy. In that case, if the service that was called came back up, the proxy may continue to throw timeouts.

Workflow

FIX – Service Endpoint Workflow Activity - If there was a service fault or timeout exception, the proxy was not getting cleaned up. Normally this would not be an issue. However, if the service timeout exceeded a number of minutes, this could inadvertently corrupt the underlying proxy. In that case, if the service that was called came back up, the proxy may continue to throw timeouts.

Neuron 3.6.0.1050

Neuron Explorer

FIX – Import/Export – If the Zone was chosen to export to a *.esb file, when importing, the Zone would not be visible in the Import dialog or present within the *.esb file.

FIX – Import/Export – In the Import/Export dialog, Endpoint Hosts were labeled Availability Groups.

FIX – Import/Export – Swagger Documents and Oauth Providers were previously not supported and visible within the UI.

FIX – Import/Export - When writing out response files, the “availability-group” switch has been changed to “endpoint-host”. Also, the following switches have been added, --swagger, --oauth-provider.

FIX – Import/Export - Added dependency support to adapter and service endpoints for endpoint hosts. If an endpoint is selected that’s running under an endpoint host, the endpoint host will be auto selected.

Command Line Tools

FIX – Import/Export - Added support for Oauth Providers and Swagger docs. Availability Group switch has been changed to Endpoint Hosts. Command line switches are now:

--endpoint-host, --swagger, --oauth-provider

FIX – Import – If a *.esb file was being imported from the command line and it did not have the default Neuron Host Endpoint within it, the import would fail and list the following issue:

ImportConfig Error: 1 : Error occurred creating the metabase reference for 'c:\temp\SaveTest.esb'. Entity=Neuron ESB Default Host: The 'Neuron ESB Default Host' was added for Endpoints.

Workflow

FIX – Correlated Workflows - If there is more than one instance of neuron marked as primary for the endpoint host running correlated workflows (the hosts can be on different machines), each instance should pick up and process the messages for the correlated workflows running on the given instance. However, there could be a case where if a workflow was unloading due to, for example, a receive message activity, the state of the workflow could be corrupted causing new workflows to start up instead of a message correlating back to a running workflow. The workflows expecting the message would also stay in the unloaded state since the receive message activity never received the message to allow it to continue.

FIX – SQL Deadlocks - If there is more than one instance of neuron marked as primary for the endpoint host running correlated workflows (the hosts can be on different machines), each instance should pick up and process the messages for the correlated workflows running on the given instance. However, there could be a SQL deadlocking issue that occurred due to the SERIALIZABLE transaction level being used. This has been changed to READ COMMITTED along with using exclusive row hold locks. A database update is required.

Service Endpoints

FIX – Swagger Documents - Added a filter to load only custom swagger documents stored in the repository when configuring the client connectors for metadata. Previously we were listing those Swagger documents which we ship with Neuron ESB.

NEW – SOAP Header Support - Modified SOAP header support to not require a namespace as part of the key in the SOAP Headers dictionary. Existing processes will not need to change. Also modified the headers to allow the mustUnderstand attribute to be set on the header's root element.

Adapter Endpoints

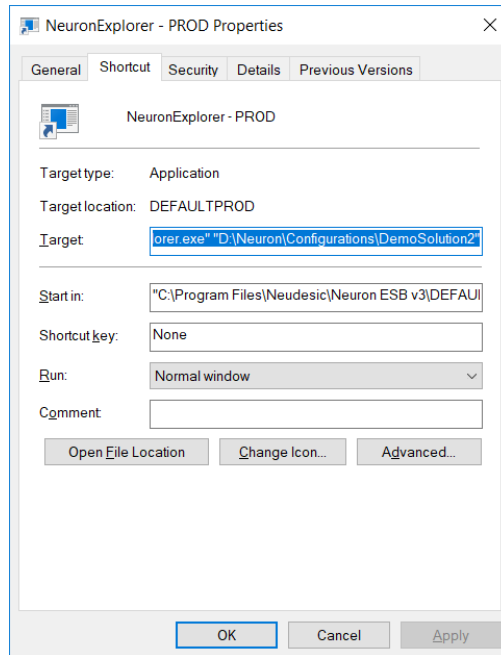
FIX – Meta Data Generation Wizard – When attempting to connect using either Dynamics CRM XRM, NetSuite or ServiceNow adapters, their respective proxy server configuration was not displayed/used.

FIX – MQSeries Adapter – When setting MQRFH2 headers via C#, if the NameValueCCSID property was set to something other than 1208, it was ignored. Also, added string padding implementation to properties set that have a string data type to ensure the proper length before the string value is set for a property.

Neuron 3.6.0.1043

Neuron Explorer

NEW – Start with Solution – Desktop shortcuts can now be created passing the path of the Neuron Solution to the Neuron Explorer executable. This will launch the Neuron Explorer and auto open the designed solution. For example, the following screen shot shows the property window of the desktop shortcut:



The value of the target property is set with the location of the Neuron ESB Explorer executable followed by the path of the solution to open:

"C:\Program Files\Neudesic\Neuron ESB v3\DEFAULTDev\NeuronExplorer.exe"
 "D:\Neuron\Testing\3.5Main\DemoSolution2"

Workflow

FIX – Execute Process Activity – Would not bind to Environment Variables used within the called Business Process.

FIX – Batch of Messages to Correlated Workflows Deadlock - If a user sent in a large batch of messages to a correlated workflow a deadlock issue could occur resulting in workflows not receiving all the messages (the messages would fail to the Neuron ESB audit table). This fixes the deadlock issue. A database update is required.

FIX - Batch of Messages causes Workflow to return Null value - If a user sent in a large batch of messages to a workflow instance at once, the resulting collection of commands that is implemented by the workflow engine could potentially return a null value due to concurrency locking.

FIX – Subsequent changes to Workflow Definition not recognized at runtime - If a user made a change to a workflow definition and then saved it, the first time the workflow definition change would be reflected in the runtime once the changes were detected. On the second time after saving (without restarting the ESB Service), the changes would not get picked up.

FIX – Restart of Correlated and Request/Retry Workflows fail - If a user attempted to restart a correlated or request reply type workflow from Workflow Tracking, it would not work.

FIX – Subsequent Execute Process exceptions are reported incorrectly - If a user had a workflow with an execute process activity that throws an exception and they ran it, then changed the exception that is thrown in the process, then tried to restart the workflow from workflow tracking, in the workflow tracking item, the exceptions reported in the Tracking tab that occurred before the attempted restart would be overwritten with the new exception.

Auditing

FIX – SQL Truncation Warnings generated with Auditing - The SQL data type length was incorrect for the ReplyToPartyId in the Neuron ESB database. It should be 100 instead of 36. This would cause warnings in the previous release when using a Party Id that is longer than 36 characters and that party is used in the ReplyToPartyId message header when auditing the message.

Adapters

FIX – REST based Adapters - If a REST adapter returned a response that did not have a content-type header, it threw an Object not set exception.

FIX - MQSeries adapter - if user selected the bindings option, they could receive the following error at runtime: Machine name missing. Cannot resolve to DNS entry.

FIX – MQSeries Adapter – When sending a message to MQSeries, the MQRFH2 was not using the existing value of the CodedCharSetId of the MQMD header by default, resulting in a CCSID value of 437. This has been corrected. However, users can dynamically set the MQRFH2 header (overriding the default value) by using the syntax below:

```
context.Data.SetProperty("wmq", "MQHRF2.CodedCharSetId", "1208");
```

Neuron 3.6.0.1034

Service Endpoints

FIX – JSON interpreted as XML – If the incoming call to a REST based Client Connector had an HTTP content type of application/json+patch, then the following error would be reported.

The XmlReader used for the body of the message must be positioned on an element.

FIX – Transfer Encoding set to Chunked – If the Transfer Encoding was set to Chunked on the outgoing response message, the calling client would not receive the response.

FIX – Swagger Port change not applied at runtime - Changing the port for Swagger configuration in the appsettings.config file would not have any effect. The swagger help documentation configured on client connector would not redirect to the updated new port.

Adapters

FIX – Rabbit MQ Adapter – Users can now enter a number from -1 to infinite for the Time to Live value. This was previously restricted to a range of 0 to 1440. If -1 is entered, Neuron ESB will NOT set the Time to Live Value.

NEW – MQSeries Adapter – Support has been added to allow users to specify IPC rather than TCP for all local calls to a Queue Manager. The Binding property has been expanded to support 3 options: Server, Client and Bindings. When Bindings is selected, the server, port and binding channel will not be passed to the open queue manager function, forcing the adapter to use the MQSeries IPC protocol for communication rather than TCP over a binding channel.

Neuron 3.6.0.1031

Workflow

FIX – Execute Process Activity – Returns the following error message at runtime: “The process named ‘X’ was not found in the ESB configuration.” This was introduced in the 1030 build.

FIX – C# Workflow Activity - When testing a workflow that contains any C# activity in the workflow designer, an object reference error would be thrown. This is regression from a fix in build 1030.

Adapters

FIX – Rabbit MQ Adapter – Users can now enter a number from zero to infinite for the Time to Live value. This was previously restricted to a range of 1 to 1440.

MOD – Rabbit MQ Adapter – Users can now specify that the adapters should attempt to create the queue if the queue doesn’t already exist when in Subscribe mode. A new property called “Create Queue” will be visible when the user sets the “Routing Mode” property of the adapter to “Queue”. If the Create Queue property is set to True (default is false), Neuron will attempt to create the queue before sending the message.

Neuron 3.6.0.1030

Neuron ESB Runtime

MOD – Reducing startup time for multi core machines - All Endpoints and Topics are now started in parallel utilizing all available cores on the machine, reducing overall CPU utilization and total time required for starting complex solutions on machines with multiple cores

MOD – Synchronization of startup – There were conditions where the endpoints assigned to Endpoint Hosts could signal they are ready for processing before other Endpoint Hosts were fully up. This has been corrected.

FIX – Single Instance Not Failing over when hosted in ESB Service - If a user had a single instance endpoint hosted in the ESB Service rather than in an Endpoint Host, the endpoint would not failover upon the primary instance stopping.

FIX – 1. Memory Leak when Adapters set for Single Instance and misconfigured – If a user has an adapter endpoint configured for Single Instance mode and set to run under the ESB Service (rather than an Endpoint Host) and the adapter endpoint’s properties are not configured appropriately to allow it to successfully start, errors similar to the one below will be continually generated. In the example below,

an Environmental Variable was being used to configure a port on an FTP Adapter Endpoint, but the value had not been set:

The FTP_PaymentPickup failed to start. Error occurred attempting to set the 'Port' property using the '{\$PaymentPort}' Environment Variable. The environment variable PaymentPort has not been set

In this case, the start of the adapter endpoint would fail, causing control to revert to the next server, which would also fail to start if running the same configuration. From there the startup process would continually repeat itself several times a minute. If left unattended for a significant period of time (30+ hours), any connections to a TCP based Topic may fault, placing the adapter endpoint in an unstable state where it's unable to shut down internally, locking its resources. The symptom of this would be continued nonstop logging of the following:

2019-01-05 19:25:13.768-07:00 [197] INFO - Attempting to stop...

At this point, new instances of the party would be created on every attempted start of the adapter endpoint. Eventually, after several days, this could result in the exhaustion of winsock resources and memory for the host process. With this fix, if an error conditions results in the Adapter Endpoint failing to start, the adapter endpoint will clean up its resources if the connection to TCP based Topics fault.

FIX – 2. Memory Leak when Adapters set for Single Instance and misconfigured – If a user has an adapter endpoint configured for Single Instance mode and set to run under the ESB Service (rather than an Endpoint Host) and the adapter endpoint's properties are not configured appropriately to allow it to successfully start, errors similar to the one below will be continually generated. In the example below, an Environmental Variable was being used to configure a port on an FTP Adapter Endpoint, but the value had not been set:

The FTP_PaymentPickup failed to start. Error occurred attempting to set the 'Port' property using the '{\$PaymentPort}' Environment Variable. The environment variable PaymentPort has not been set

In this case, the start of the adapter endpoint would fail, causing control to revert to the next server, which would also fail to start if running the same configuration. From there the startup process would continually repeat itself several times a minute. If the error condition is left unattended and unresolved, the system would experience a gradually increasing memory leak.

Logging and Monitoring

FIX – Workflow Endpoints status set to Stopped in Endpoint Health – If there are only Workflow Endpoints assigned to an Endpoint Host, they will always display in a Stopped state within Endpoint Health, even if started. This only occurred if there were no other types (i.e. Adapter or Service Endpoints) assigned to the same Endpoint Host.

FIX – Not logging correct number of Endpoints that were started under an Endpoint Host – When set for informational logging, Neuron should log a message similar to the one below with the correct number of endpoints. Before this fix, the endpoint count was always logged as zero:

2018-12-20 14:59:40.304-05:00 [1] INFO - Operation Request: Operation StartAll Completed successfully against '7' endpoints in Endpoint Host 'Neuron ESB Endpoint Host'

Workflow

FIX – Unloaded Workflows generating Temp Files - Code activities would be compiled again if the workflow came back from being unloaded causing temp files to be created at a high rate if the user had workflows that go unloaded then resume and hit a code activity. Now we have a collection and workflow extension that allows the type and method info to be stored for code activities which are restored upon execution of the code activity.

FIX – Workflows processing children of Subtopics – Workflows would process children of sub topics that they did not have specific subscription to, but that the underlying Subscriber object did. For instance, if a workflow was configured to receive messages from the “TopicA.Orders” topic, it would also receive messages from any subtopic of Orders such as “TopicA.Orders.Purchases” IF the underlying Subscriber object had a subscription to “TopicA.Orders.Purchases” as well as “TopicA.Orders”.

FIX - Semaphores errors reported - Semaphores weren't being released properly in all situations such as when a workflow aborted due to an unhandled exception in certain rare cases. For example, if the workflow aborted from an unhandled exception (e.g. the temp directory is full causing c# activities to not compile/execute causing some timeout exception to occur in the actual workflow application itself (in Microsoft's code)).

FIX – Quartz Scheduler errors reported in log - A quartz scheduler error could occur when attempting to store job and trigger data. The error would state that the trigger already exists with the given name. Now we do a delete and add effectively updating the trigger.

FIX – Correlation within a Timeout within a Loop fails - If there is a workflow with a receive message activity within a timeout activity within a loop, if the receive message step causes the workflow to go unloaded, then when a message is received to resume the workflow, that message would start another workflow instead of correlating to the receive message.

Adapters

FIX – SAP Adapter – When configured as an IDOC listener, if the same IDOC segment was defined in two different IDOCs, but each one has a different customization, the second IDOC would fail during serialization. This has been corrected.

Business Processes

FIX – Transform Process Step throws Exception when Testing – When testing the Transform – XSLT process step within the Business Process Designer, an Object reference not set to an instance of an object exception would be thrown. This was a regression in Transformation - XSLT that was introduced

when we fixed the step to use the output settings from the XSLT provided in build 3.6.0.991. This did not affect runtime.

Service Endpoints

MOD – REST Endpoints with Policy would not log response message with status code – In some cases, if a Service Policy associated with a REST endpoint executed and Audited the failure, the response message (if any) returned from the server hosting the REST service may not be audited with the Status Code and Status Description.

Neuron 3.6.0.1012

Business Processes

MOD – Audit Process Step – Support has been added to audit messages with extended HTTP Status Codes. If we detect that the message has an extended code not supported by .NET (i.e. 207, 422, etc.), we will clone the message before Auditing it. Before the Audit (but after the clone), we'll reset the HTTP Status Code to the nearest hundred starting with the first digit of the extended code (i.e. 207 would become 200). The original extended code will be prepended to the HTTP Status Description of the ESB Message.

FIX – HTTP Client Utility Process Step – If the full URL path was set in the HTTP Client Utility, the subsequent call to a Service Endpoint would not use the URL but instead use the URL it was originally configured with.

FIX – HTTP Client Utility Process Step – If the full URL path was set using either Environment Variables, Context Properties, Global Variables or Message Properties, the underlying properties of the ESB Message object would not be set correctly. This would affect LocalPath, To and the Addressing.To properties.

FIX – HTTP Client Utility Process Step – If JSON was inserted in the Body property when set to Raw, the braces would be stripped out of the resulting body when passed to the service.

MOD – Tracing to Business Process Design window during testing – Several steps were not doing this including ODBC, Audit, Service and Adapter Endpoint Process steps. Also, the Neuron ESB Explorer has been configured to output Debug level traces by default.

MOD – Service Endpoint Process Step – If a MessageSecurityException or ProtocolException is generated during the sending of a message using REST, Neuron ESB will automatically attempt to retry the service call after recreating the underlying proxy, message and HTTP headers.

FIX – Adapter Endpoint Process Step – if executed within a Process directly tied to a Client Connector rather than to a Publisher, a Topic Null Exception could occur:

FIX – Audit Process Step – When using during design time testing, if the proxy created faulted and another attempt to test is made, the user may receive the following warning message:

MAXIMUM number of Audit Proxies '1' have already been created. WAITING for an Audit Proxy to become available.

Neuron ESB Runtime

MOD – Single Instance Failure – When an Endpoint marked as single instance would fail over to another service, it could be due to a database error that would be logged in the master log file, but not in the actual log file of the endpoint affected.

MOD – Single Instance Failover - An Endpoint marked as single instance would fail to start if the Endpoint is assigned to run under an Endpoint Host. This has been corrected so now it starts. However, the single instance configuration will be ignored. Single instance behavior must now be configured using the Deployment Settings tab of the Endpoint Host.

FIX – CPU Usage Spikes and Slow startup – When running on a machine with 2 or fewer cores with a configuration that included a dozen or more Endpoint hosts, the runtime may experience delays in starting and CPU usage of 100%.

Neuron ESB Explorer

FIX – Deleting a Sub Topic – If a user attempted to delete a sub topic that was positioned before the last sub topic in the grid, a System.IndexOutOfRangeException could be generated when the user attempts to Apply the change.

Service Endpoints

FIX – Client Connector URL not matched to incoming URL – During an incoming REST based API URL request, the incorrect Client Connector could be matched to the incoming request if there were more than one Client Connector URL that starts with the incoming URL. For example, an incoming request url of `http://mymachine/test/v1/summary?GPID=5` could be matched with either a Client Connector URL of `http://mymachine/test/v1/summary/benefits` or one with a url of `http://mymachine/test/v1/summary` . Now, Neuron ESB will do an exact match first and, only if there is not an exact match, fall back to find a Client Connector url that contains the incoming url.

MOD – Service Endpoint – If a MessageSecurityException or ProtocolException is generated during the sending of a message using REST, Neuron ESB will automatically attempt to retry the service call after recreating the underlying proxy, message and HTTP headers.

Adapter Endpoints

FIX - Dynamics AX Adapter – Fixed issue that when a stored credential was selected in the property grid of the adapter endpoint, if you clicked on the dropdown list of Service URLs you would only see the list if the currently logged-in user has permission to call the Query Service in the Dynamics AX server.

FIX - Dynamics AX Adapter – Fixed issue that when running the Metadata Generation wizard, if you selected the Dynamics AX Adapter from the Registered Adapter dropdown list, an exception would be thrown stating that the AOS Server Name property could not be blank.

FIX - SAP Adapter – fixed issue when using the Metadata Generation Wizard to create sample XML for customized IDOCs.

NEW - SAP Adapter – Added support for transactions in SAP. When an SAP Adapter endpoint is configured as transactional, any message that is processed by the adapter endpoint without the property sap.TransactionId will create a new connection to SAP and create a transaction ID that is associated with that connection. The transaction ID will be included with the response message from SAP as the message property sap.TransactionId. This transaction ID must be added as the message property sap.TransactionId on all following message that are part of the transaction (including commit and rollback). This will ensure the calls to SAP occur over the same connection. You can then send a BAPI_COMMIT_TRANSACTION or BAPI_ROLLBACK_TRANSACTION to SAP to commit or rollback all the previous transactional messages. The lifetime of the connection is controlled by the adapter endpoint property Transaction ID Expiration.

Workflow

FIX – C# Workflow Activity – When opening the C# Workflow Activity in the Workflow Designer, the following error may occur:

System.InvalidOperationException: Collection was modified; enumeration operation may not execute.

FIX – Workflow Processing Message Twice - After saving a Neuron ESB solution for a running instance with workflow endpoints, multiple workflow host services could start up for a single endpoint. This would mean a single message gets processed twice (at the same time) by a given workflow endpoint. This issue occurred only if the Party that the workflow endpoint is assigned to did not get disposed before the OnConfigurationChanged event executed in the WorkflowHostService, which could be the case since it runs asynchronously. The issue had a chance to occur only when the workflow endpoint itself had a configuration change (e.g. changing the endpoint to run a different workflow, changing the endpoint's party, etc.) as the workflow endpoint only gets restarted upon such changes.

MOD – Service Endpoint Activity – If a MessageSecurityException or ProtocolException is generated during the sending of a message using REST, Neuron ESB will automatically attempt to retry the service call after recreating the underlying proxy, message and HTTP headers.

Monitoring

FIX – Endpoint Health – Pending counter for Workflow incorrect - The “Pending” counter for workflow endpoints in the Endpoint Health Monitor in Neuron Explorer could increment in some cases but not decrement when it should causing the counter to be incorrect. This issue shouldn't affect the runtime or the execution of workflows. This has been corrected.

MOD – Endpoint Health – A Process ID column has been added to show the process id of the Endpoint Host.

FIX – Clear Errors and Warnings generates UI error – In Endpoint Health, if a user right clicks on a row that has errors in it, and selects "Clear Errors and Warnings", the user will get the error below, but the errors are successfully cleared:

System.ArgumentException: Key not found: 'Errors'

Neuron 3.6.0.995

Business Processes

FIX - Timeout Process Step would fail with unhandled exception – If a C# step was placed within the Timeout Process step block and the Configuration or Log object was accessed, an unhandled exception would be thrown.

FIX – Audit Process Step would not record Process Name – When auditing a message, the name of the current Business Process was not being captured and audited with the message. This would result in the Process Name row of the Neuron Properties within the Audited message to be blank.

FIX – Excel to XML Process Step would not process mixed datatype columns – If there were mixed datatype values within a column, some of the values may not get processed and outputted in the final xml.

Runtime

FIX – Publishing a message may through a Null Exception – Under certain circumstances, if an endpoint is publishing a message at the same time an underlying change to the configuration was detected, a null exception could occur when iterating through the Business Processes to determine what would need to be executed prior to the send. The following error may be recorded:

System.NullReferenceException: Object reference not set to an instance of an object.

at

Neuron.Esb.Internal.PipelineRuntimeHelper.<>c__DisplayClass3.<ApplyPipelines>b__1(ClientPipelineItem pipeline)

at System.Linq.Enumerable.WhereArrayIterator`1.MoveNext()

at Neuron.Esb.Internal.PipelineRuntimeHelper.ApplyPipelines(ESBClientContext context, ESBMessage& message, Boolean applyOnPublish, Boolean applyOnReceive)

Neuron 3.6.0.991

Adapters

FIX – FTP Adapter – When using the FTP Adapter in Subscribe mode and SSL is configured with a certificate, the adapter would not recognize the certificate.

MOD – Azure Service Bus Adapter - AuthenticationMode property has been removed. Authentication mode is always SAS as ACS has been deprecated.

FIX – Azure Service Bus Adapter – When processing JSON messages, the following error could occur:

Cannot access a disposed object. Object name: 'BufferedInputStream'.

FIX - Salesforce adapter – Fixed issue that was introduced with the release of Salesforce SOAP API version 43, the generation of SObject-based schemas returned the error "" when using version 43 or later.

FIX - Salesforce adapter – Metadata generation did not create XML samples for many of the core operations. This has been fixed.

NEW - Salesforce adapter – Added support for Salesforce SOAP API version 44

Installer

MOD – Bootstrap Installer .NET dependency – The Neuron ESB installer prerequisite is now .NET 4.0, rather than 4.7.1. If 4.7.1 is not installed, the setup process will prompt the user to download and install.

MOD – Event Processing Service startup – This service will no longer be automatically started as part of the Neuron Installation process.

Runtime

MOD – Reliability Policy added to SQL Operations – Retry policies have been added to all SQL Reader and Command operations to accommodate occasional network disruptions between the Neuron ESB Server and the Microsoft SQL Server

FIX – Single Instance prematurely Fails Over – If an adapter endpoint is marked as single instance, it could prematurely fail over to the secondary node, even though there is no failure, if there was a SQL server network disruption.

Business Processes

FIX – Audit Process Step – When attempting to audit a message that contains an extended HTTP Status code not natively supported by .NET, a serialization would occur. Now if Neuron detects an extended HTTP status code, it is set to NULL before the auditing process.

FIX – XSLT Transform Process Step – The xsl:output element would not be processed by the transformation step.

Workflow

FIX – Adding Workflow Assembly References – If a user added an assembly reference to the workflow using the "+" button on the workflow designer, then added a c# class or fragment and removed the reference from the code editor for the class/fragment, then tried to add another workflow reference using the "+" button, an array out of bounds exception would occur.

FIX – Adding a Variable to a Workflow – If a user added a variable that was a generic that contained a non-serializable type (e.g. having a List of a non-serializable type), when running the workflow they

would see exceptions in the log stating that a serialization exception occurred. This would happen if the generic itself was serializable, but the containing type was not.

FIX – Adding an Assembly to a Workflow from the GAC – If a user added an assembly reference from the GAC to a workflow or c# activity, the workflow designer would throw an exception stating the referenced assembly could not be found when attempting to apply the workflow changes.

FIX – Adding a reference to the Microsoft.CSharp Assembly – If a user had a workflow with a code activity that referenced the Microsoft.CSharp assembly, the compiler for the code activity would throw an exception stating the assembly could not be found when reaching the code that depends on the assembly.

FIX – Persistent Delay Activity does not Resume when used in a Correlated Type Workflow – If a user had a correlated type workflow that contained a Delay activity with a delay greater than a minute, the workflow would not resume after the delay had elapsed.

FIX – Persistent Delay before a Receive Message Activity would cause Messages to not be Queued for the Receive Message Activity – If a user had a Persistent Delay somewhere before a Receive Message Activity, any messages sent to that correlated workflow during the delay and before the Receive Message Activity was executed would not be queued for the receive.

FIX – Workflow Resuming from a Persistent Delay would not clear the Needed Database Tables – If a user had a Persistent Delay, when the workflow resumed from the delay and completed or aborted, the necessary database tables and in-memory collections would not be cleaned up.

Topics

FIX – Rabbit MQ Topic Dead Letter Queue Infinite loop – If a message was routed to Neuron's internal Rabbit MQ Dead Letter Queue and it was missing required properties, specifically the Topic name, Neuron would return a negative acknowledgment and re-queue the bad message. This could result in an infinite loop causing considerable CPU usage. Moving forward, a negative acknowledgment is still returned, however an attempt will be made to audit the message to the failed message table. Otherwise it will be written to the endpoint log rather than returned to the queue. If the Topic property is missing, the value "UNKNOWN" will be reported in its place.

FIX – Rabbit MQ Topics – If error occurred during the receipt of a message from the dead letter queue, the error count would not be incremented in Endpoint Health.

FIX – Rabbit MQ Topic Subscriber Queue Infinite loop – If a message was routed to Neuron's internal Rabbit MQ Subscriber Queue (representing the Neuron Subscriber) and it was missing required properties, Neuron would return a negative acknowledgment and re-queue the bad message. This could result in an infinite loop causing considerable CPU usage for the duration of the Time to Live period specified on the Topic (which by default is 1 day). Moving forward, a negative acknowledgment is still returned, however an attempt will be made to audit the message to the failed message table. Otherwise

it will be written to the endpoint log rather than returned to the queue. If the Topic property is missing, the value "UNKNOWN" will be reported in its place.

Neuron 3.6.0.975

Installer

MOD – WS-Discovery port disabled by default – the WS-Discovery announcements on port 9021 has been disabled by default. Previously, during installation, the Discovery Service could fail start during the install because of the WS-Discovery service port being blocked. This seems to be introduced recently on Windows 10. This can be manually enabled by setting "WSDDiscoveryEnabled" flag located in AppSettings in the DiscoveryService.exe.config file to true.

MOD – MSI package is now digitally signed.

Workflow

FIX – Namespace parsing error at design time for custom assembly references – This could occur in the designer within a C# workflow activity if referencing an external custom assembly. This bug was introduced in build 3.6.0.937.

Neuron 3.6.0.968

Logging

FIX – Client API Logging – All logging functions in Neuron ESB 3.6 were moved to Log4Net. When using the Neuron ESB Client API hosted in external .NET applications, logging can be enabled by adding the appropriate log4net configuration to the "configuration" section of the application's app.config file. The following is an example of log4net configuration that can be added to the app.config file to enable the Neuron ESB Client API to log to a text log file using the log4net Rolling File Appender:

```
<configSections>
  <section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler,
log4net"/>
</configSections>

<log4net>
  <appender name="File1Appender" type="log4net.Appender.RollingFileAppender">
    <file value="log-file-1.txt" />
    <appendToFile value="true" />
    <immediateFlush value="true" />
    <preserveLogFileNameExtension value="true" />
    <staticLogFileName value="false" />
    <rollingStyle value="Composite" />
    <datePattern value="-yyyy-MM-dd" />
    <maxSizeRollBackups value="10" />
    <maximumFileSize value="100MB" />
    <layout type="log4net.Layout.PatternLayout">
      <conversionPattern value="%date{yyyy-MM-dd HH:mm:ss.ffff} [%thread] %-5level -
%message%newline%exception"/>
    </layout>
  </appender>
</log4net>
```

```

    <level value="INFO"/>
  </root>
  <logger name="NeuronEventLog">
    <level value="ALL"/>
    <appender-ref ref="File1Appender"/>
  </logger>
</log4net>

```

FIX – Adapter Endpoint Logging – If Adapter Endpoints were configured to run in an Endpoint Host, logging levels were not honored. Changes in the logging level via the Configure Server dialog were not reflected in the Endpoint Host. This has been corrected.

Workflow

FIX – Execute Process Workflow Activity – Calling a Business Process from a Workflow using the Execute Process Workflow Activity would succeed during design time testing but would fail at runtime with the following error reported in Workflow Tracking:

The Process named “X” was not found in the ESB Configuration.

FIX - C# Code/Class Workflow Activity - If referencing a dll that wasn't in the GAC in a workflow C# code/class activity (i.e. a custom reference where user placed the dll in the Neuron Instance Folder and added the reference to the workflow), an error would be displayed anytime the code was recompiled or saved.

FIX - C# Class Workflow Activity - After initially dragging on a C# class activity, if a user opens the activity in the code designer, the generated code block would be missing. Closing and reopening the editor would display the correct generated code block.

FIX – Adapter Endpoint Activity – A key not found exception could occur when a user changes the adapter endpoint properties for an adapter selected in an adapter endpoint workflow activity while the service is running.

FIX – Renaming Referenced Entities in Workflows - If a user had an execute process, service endpoint, adapter endpoint, or publish activity in a workflow, renaming the referenced business process, service endpoint, adapter endpoint, or topic would not automatically update their name property within the workflow activity.

FIX – Renaming Business Process in Workflows - Renaming any business process or service endpoint would result in any workflow definition that had an execute process or service endpoint activity to lose the defined process or service in the workflow designer. This only happened at the designer level and did not affect the running workflow definition unless the user applied and saved the changes.

FIX - Workflow Designer closes with unsaved changes when referenced object is renamed - If a user has an adapter/service/topic or Business Process referenced in a corresponding activity (i.e. adapter endpoint activity), and the user has multiple workflow designers open that reference the same adapter, the non-active workflow designers would close if the adapter was renamed. This could cause users to lose any changes that haven't been applied in the workflows that close. The activity must be in view for

this to happen. Now we will display a prompt telling the user to manually update the references in the non-active designers with pending changes.

FIX - Workflow Designer Apply button becomes enabled when unmodified - If a user had an adapter endpoint referenced in a corresponding adapter endpoint activity in multiple workflow designers, switching between them after renaming the endpoint would cause the apply button to become enabled on unmodified workflows.

Adapters

FIX – Dynamics CRM Web API Adapter – Users may receive the following error message due to a missing Content-Type header:

The Microsoft Dynamics CRM Web API endpoint, 'MyadapterName', failed to send the message. The value cannot be null or empty.

Parameter name: mediaType

FIX – Dynamics CRM Web API Adapter – If the Adapter received a message from a Client Connector, the following warning could be written to the Neuron ESB log file:

"Bytes to be written to the stream exceed the Content-Length bytes size specified"

FIX – Dynamics CRM Web API Adapter – If the Token expired, it would not refresh as expected and would throw the following error:

The request message was already sent. Cannot send the same request message multiple times.

FIX - Dynamics CRM Online Plugin - changed target framework to 4.5.2 to match Dynamics 365 online .NET framework requirements.

MOD – Salesforce Adapter - Added support for latest versions of the Salesforce SOAP API through version 43

MOD – NetSuite Adapter - Added support for latest versions of the NetSuite API through version 2018.2

FIX - SAP Adapter – RFC and BAPI requests that contain table data as input parameters would throw the following exception:

Sequence contains more than one matching element

FIX - MQSeries Adapter - Using the adapter in Publish mode to read messages that include RFH2 headers and custom properties, the length of the RFH2 header would be appended to the the body of the message.

FIX - MQSeries Adapter - Using the adapter in Subscribe mode to send messages that include RFH2 NameValueData headers, a Memory stream is not expandable exception may occur.

Service Endpoints

MOD - BasicHttpRelay – Binding now provides appropriate security options and the ability to not require clients to send the relay an access token with each request.

FIX – Client Connector Fails to start – If the Client Connector was configured to run a Business Process rather than interact directly with the Messaging sub system and the Endpoint Host property was left blank, the Client Connector would fail to start.

MOD - Webhttprelay – Binding now provides the ability to not require clients to send the relay an access token with each request.

FIX – Webhttp and Webhttprelay – Responses were always encoded as ASCII. Now responses are encoded based on the incoming request.

FIX – HTTP Headers not case insensitive – HTTP Headers were not accounting for headers that were not properly cased. This could result in errors similar the following:

Exception: The 'transfer-encoding' header must be modified using the appropriate property or method.

Parameter name: name

Method: ThrowOnRestrictedHeader

FIX – Policy for REST Codes not firing – If a service policy was configured to fire for any selected HTTP Status code that does not result in a runtime exception being thrown (such as 500), the policy would not fire.

Neuron 3.6.0.945

Business Processes

FIX – Debugger throws ArgumentNullException – When launching the Business Process Debugger with a Business Process that does NOT have any 3rd party external .NET assemblies referenced, the debugger would throw an ArgumentNullException.

Adapters

FIX – NetSuite Adapter - production and sandbox accounts require an account-specific URL when making Web service calls. Previously, calls made with production or sandbox accounts would fail unless the correct URL was set in the NetSuite Service URL property in the adapter endpoint properties. Now the adapter will retrieve the correct URL for every account type through a call to NetSuite's data center URL rest service.

FIX – FTP Adapter – When using ImplicitSSL mode set to TLS, the following error may be reported:

System.ArgumentNullException: Value cannot be null.

Parameter name: cert

Services

FIX – Client Connector URL throws Fault when called – Under very specific circumstances, specifically if the incoming URL is resolved to a different case than the client connector URL, the client connector may throw a fault back to the calling client with an error message similar to the following: *The current service instance for the 'https://machine/myurl' url could not be found.*

Configuration

FIX – Exporting Client Connector Endpoint generates Error – When exporting a Client Connector Endpoint that has Messaging disabled, a Topic not found error would be generated.

Neuron 3.6.0.937

Services

FIX – Service Connector using Windows Credentials Authentication – When hosting a Service Connector in an Endpoint Host where the Service Connector is configured to use a Windows Credentials using either the Transport:Basic or HttpBasic Security Model, the credential information would not be forwarded to the service.

MOD – Swagger Support – Support for Swagger UI has been upgraded to version 3.0

FIX – ADP OAuth Provider – After 60 minutes the OAuth Token would expire and not renew. This was due to ADP throwing a protocol exception rather than a message security exception. The Token now automatically renews when expired.

FIX – Service Connector obfuscating true Exception – If a user received a message security exception on a SOAP service connector, they would receive the following error instead of the actual error:

Service connector X' received an exception attempting to pass on a received message. This message cannot support the operation because it has been written.

Adapters

FIX – FTP Adapter Explicit Mode and TLS - When using Explicit mode only SSL 3.1 was supported. A new property has been added so that users can select either SSL or TLS protocol when using either Explicit or Implicit mode for FTP. TLS is the default.

MOD – FTP Adapter – Remove BOM property has been added to publish mode.

MOD – FTPS Adapter – Remove BOM property has been added to publish mode.

FIX - NetSuite Adapter - Fixed issue with metadata generation in the 2018_1 version that results in "Object reference not set to instance of an object."

FIX - CRM XRM Adapter - Fix for setting objectid property to the correct entity type. One example is the regardingobjectid in activities like emails. Instead of setting the typename property to "Lookup", set it to the entity type of the lookup (i.e. "contact"). Also, users can now pass-in a reference ID as a value

instead of a lookup and have it converted to the correct type of entity. When passing the GUID, use the "reference" attribute to set the correct entity type (i.e. "contact")

Business Processes

FIX - JSON Validation Process Step - fixed issue where it would validate a JSON document if the document itself is an array.

FIX - Business Process Designer – When using the process debugger with additional assembly references registered at the process level (NOT at the code-step level), if a breakpoint was set in a C# Class step the debugger would throw an exception when starting. This issue only occurred with a C# Class step and not with a regular C# step.

Workflow

FIX – Workflow Designer crashes Neuron ESB Explorer – This could happen if we retrieved incorrect coordinates when laying out the designer within the window.

FIX – Correlated Workflows - When running two or more primary workflow hosts on separate machines with correlated type workflows, messages received on one machine that correlate to a running workflow instance on other machines would not be received by the correct machine. Rather, a new workflow instance would start as this scenario was not supported.

FIX - C# Editor within Workflow fails to identify compile errors at design time - Design Time compile errors would be reported if editing C# or C# Class workflow activities in their dedicated editor tabs (i.e. right click to display context menu and select "Edit Code"). However, if a user opens the editor within the workflow designer as an activity window, then design time compile errors would not be reported. This has been fixed so that they are always reported regardless of where they are being edited.

Neuron ESB Explorer

FIX – Neuron ESB Explorer crashes if NO SQL Servers found on local network - The Neuron ESB Explorer could crash with a null exception if the user was attempting to retrieve a list of SQL servers in one of the SQL server specific Business Process steps or in View/Manage databases.

FIX – About Form displaying incorrect version – In the Neuron ESB Explorer, the Help -> About form was showing the incorrect version (i.e. 3.5) instead of 3.6.

FIX – ImportConfig Command Line tool – When using the ImportConfig command line tool to import a Business Process by name, defined in an RSP file, the command line tool would report the error that the Process could not be found.

FIX – Connecting Online Mode Fails – Attempting to use the Neuron ESB Explorer to Connect to an Instance of Neuron ESB in “Online” mode (via the File ->Connect menu) would fail if the Neuron ESB instance was not configured to use port sharing. The following Error would be displayed showing an attempt to connect to the control service port rather than the bootstrap port:

Operation failed: Neuron ESB Explorer was unable to connect to the ESB Bootstrap Service at net.tcp://localhost:5004/ESBConfigurationService/

Installation

MOD - CreateManagementObjects.ps1 PowerShell script - Sometimes the PerformanceCounterCategory.Exists() method would return true even though the passed in performance counter didn't exist. To fix this, PerformanceCounter.CloseSharedResources() is called to flush the cache before we execute the rest of the creation script.

FIX – Rabbit MQ Management Plugin – This was not being activated automatically if a user choose to install RabbitMQ as part of the Neuron ESB installation.

Neuron 3.6.0.920

Services

FIX – WebHttpRelay Binding throws XmlReader exception – When using the WebHttpRelay binding with JSON messages, an XmlReader exception would be thrown.

Business Processes

FIX – Copying Business Processes creates reference rather than clone - When copying a Business Process, editing a step's properties in the copy would be reflected in the original. Opening and closing the Neuron ESB Explorer would previously solve the issue. This has now been corrected.

FIX – Process Step Caching at Design Time – When conducting design time testing of either the Execute Process, Service Endpoint or Adapter Endpoint process step if, after the first test, the dependency that they called was change (i.e. Adapter or Service Endpoint or child Business Process), subsequent tests would not reflect the changes as the original entity would be cached with the designer.

Runtime

FIX – Maximum Log File Size – This value was not being honored for Workflow Endpoints. Also, we removed the daily/weekly rollover option and disk size threshold for logging.

FIX - CreateManagementObject.ps1 PowerShell script – Modified to create performance counters for Neuron Endpoint Hosts.

FIX – Logging in Authentication Validators – When in verbose or info mode, logging statements were not reflected correctly in the respective endpoint log file.

FIX - RaiseAdapterError not logging custom messages - When using RaiseAdapterError from custom adapters, if custom messages were passed, they were not being logged in the endpoint log file.

MOD – WMI Failed Message Event – EndpointName and EndpointType have been added to the event properties that users can retrieve when subscribing to the Failed Message Event.

FIX – Null Exception on Shut down - On shutdown, sometimes a null exception may be thrown when attempting to unwind the events of the party, even though the party is not null.

Adapters

FIX – Azure Service Bus Adapter – Logging was added and an unhandled exception when in Publish mode has been corrected.

Configuration

FIX – Exporting Adapter Endpoint generates Error – When exporting an Adapter Endpoint that has Messaging disabled, a Party not found error would be generated.

Samples

FIX – REST to SQL - There was a process step that didn't need to be in the REST to SQL process which caused it to always go down the error path in the process. Also, when we were getting a response we were reading the wrong xml element (i.e. "EmailAddress" when it should have been "Email").

Neuron 3.6.0.900 (3.6) Features

The Neuron ESB 3.6 release represents a significant next step toward the evolution of the Neuron Platform. Much of the infrastructure has been refactored to facilitate our near-term vision of cloud hosting, as well as make independently hosted and deployable services, adapter and workflow endpoints a reality for our customers today. More organizations are seeking the agility to quickly develop, test, deploy and update business capabilities encapsulated using a Microservices architecture.

Neuron ESB 3.6 has been built on the principal of SOA and Microservices, hence it is the natural choice to build Microservice based APIs. In previous versions of Neuron ESB, organizations could easily create REST APIs or SOAP based services. However, in this release, they can now control the deployment aspect of it as well. Using Neuron ESB 3.6, organizations can now choose to run a service, (or group of services), as a fully independent service that can be scaled in or out based on consumption using platform services like Docker.

If you're using Neuron ESB 3.5 today, please download the Neuron ESB 3.6 setup to update your existing installation. Current customers can obtain either the Patch or the full installation from the [Neuron ESB Support website](#). If you are new to Neuron ESB, you can download the full installation from the [Neuron ESB website](#).

Neuron ESB 3.6 provides dozens of new features, enhancements, and connectors, some of which include:

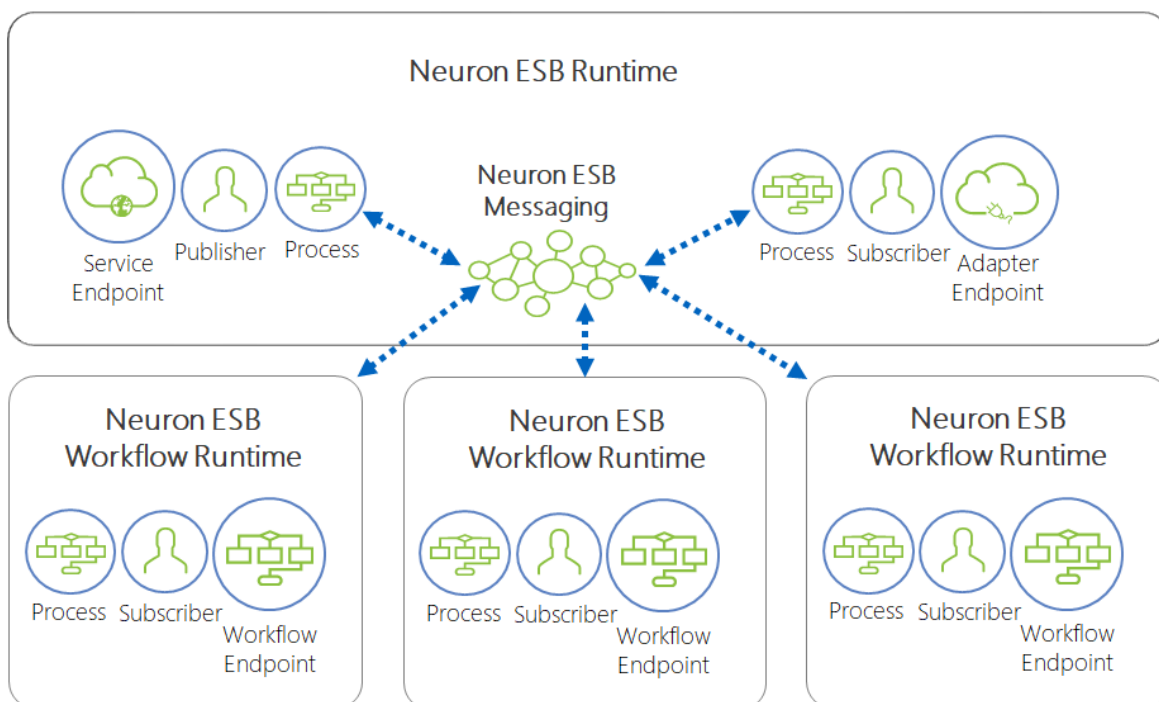
- Endpoint level Hosting
- Self-registration of Adapters/Connectors
- Messaging free Service and Adapter Endpoints
- Environment sensitive enabling/disabling of Endpoints
- ServiceNow Adapter
- New OAuth Providers
- Custom OAuth Providers
- JSON Schema Validation
- New Monitoring REST APIs
- 100+ Product Enhancements

All the changes included in the Neuron ESB 3.6 release can be found in the [Neuron ESB Change Log](#) which gets installed with Neuron ESB. In this blog post, I thought I would elaborate on some of the new enhancements and features we've added to the update.

Endpoint Level Hosting

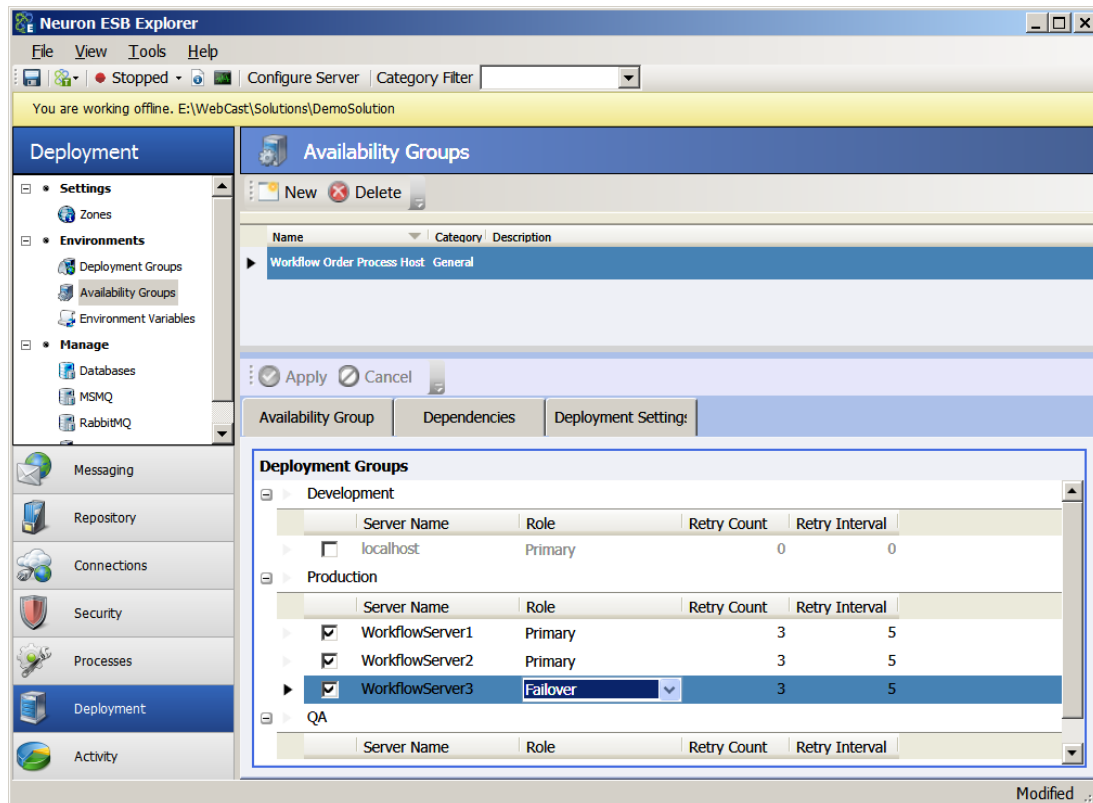
Previous Hosting Model

In previous versions of Neuron ESB, the Neuron ESB Runtime Windows Service (esbservice.exe) was used to host all Adapter and Service Endpoints, as well as all Messaging Publishing Services and all internal Services. These services were isolated in their own .NET AppDomains within the Neuron ESB Runtime service. However, a fault tolerant hosting environment was provided for the Neuron ESB Workflow Engine. The hosting was embodied by Neuron ESB Availability Groups. Availability Groups were used to load balance the execution of Workflow Instances across multiple servers in dedicated/isolated host processes.



Neuron ESB 3.5 Hosting Model – Workflow Endpoints are hosted in dedicated hosts named “Availability Groups” (i.e. esbhost.exe). Any number of Availability Groups can be created. All internal services, Messaging services, Service Endpoints and Adapters Endpoints are isolated in .NET App Domains and hosted in the Neuron ESB runtime (i.e. esbservice.exe). Parties and Business Processes run in the host process of their respective Endpoint.

These Availability Groups were designed with built in clustering to achieve high availability and fault tolerance without the need to deploy Microsoft Windows Failover Cluster Services. Servers within a Neuron ESB Deployment Group could be assigned either Primary or Failover roles which allowed failed workflows to automatically rollover onto available servers and start where they left off, providing both resiliency and reliability for mission critical functions.



Neuron ESB Availability Groups – In Neuron ESB 3.5, Availability Groups provide isolated high availability clustered hosting for Neuron ESB Workflows. Servers can be defined for load balancing Workflow execution as well as dedicated failover.

Users could create any number of Availability Groups which in turn would be assigned to any number of Workflow Endpoints. Availability Groups were isolated into their own process space and executed as child processes (hosts) of the Neuron ESB Runtime service. If a fatal error occurred that caused an Availability Group host to fail, the Neuron ESB Runtime service could restart the Availability Group on the same server, or on a different server assigned the Failover role.

Endpoint Hosts

In Neuron ESB 3.6, we've refactored and extended Availability Groups to allow them to host Adapter and Service Endpoints. As a result, Availability Groups have been renamed to Endpoint Hosts and have taken on a new and expanded role within Neuron ESB.

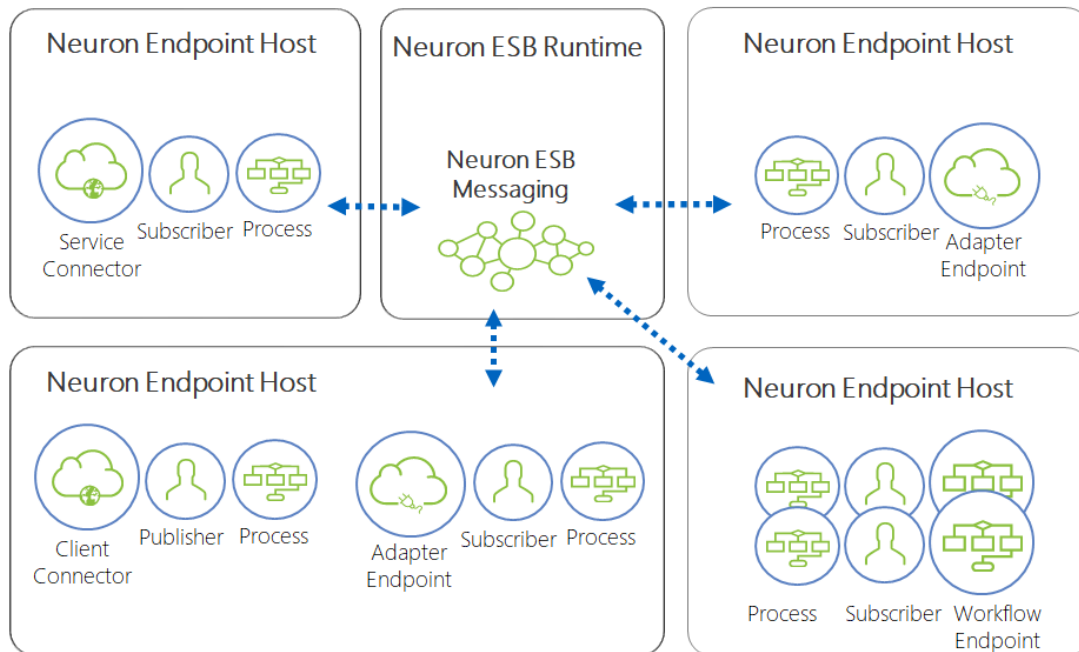
Endpoint Hosts can be used to:

- Host a single endpoint
- Facilitate Microservice deployment
- Host multiple endpoints of the same type (i.e. Adapter, Service or Workflow)
- Host multiple endpoint of different types (i.e. Adapter, Service or Workflow)
- Host endpoints under a unique service account (different from that of the Neuron ESB Runtime service)
- Isolate endpoints for system wide fault tolerance
- Reduce memory utilization

- Scale processing power

In previous versions of Neuron ESB, every Adapter and Service endpoint competed for the CPU processing of a single executable (i.e. esbservice.exe). Other competitors for those processing cycles and threads included Neuron ESB's own internal services, which would sometimes take precedent. In some cases, either an internal service or a specific endpoint could consume the majority of processing cycles, leaving other endpoints waiting for available threads to be returned to the pool before being allowed to service requests. Additionally, even though all endpoints and internal services were encapsulated in their own .NET AppDomain, it did not preclude the possibility that one misbehaving Business Process, Adapter Endpoint or, custom code writing by a customer could crash the Neuron ESB Runtime process (i.e. esbservice.exe).

Endpoint Hosts relieve these issues as each runs as an isolated process (i.e. NeuronEndpointHost.exe), with its own allocation of resources and threads. Using Endpoint Hosts, endpoints can be isolated into discreet, lightweight processes and deployed across multiple machines.



Neuron ESB 3.6 Hosting Model – All Adapter, Service and Workflow Endpoints can now be hosted in dedicated hosts named "Endpoint Host" (i.e. NeuronEndpointHost.exe). All internal services and Messaging services are isolated in .NET App Domains and hosted in the Neuron ESB runtime (i.e. esbservice.exe).

In Neuron ESB 3.6, Endpoint Hosts are more lightweight than their immediate predecessor, Availability Groups. Their executable has been renamed and the communication mechanism has been changed to use an HTTP based API (default port 51004 can be changed in appSettings). Additionally, endpoints using this hosting model are no longer encapsulated in .NET AppDomains.

NOTE: By default, all new endpoints will run in the Neuron ESB Runtime service unless they are specifically assigned to an Endpoint Host using the “Endpoint Host” property located on their respective UI configuration screens. However, if migrating a solution saved in a previous version of Neuron ESB, those endpoints will automatically be assigned to run in the new Endpoint Host environment.

Monitoring

Neuron ESB Endpoint Health has been refactored to accommodate real time activity monitoring for the Endpoint Host environment as well as the different endpoints that they host. The new user interface provides users the ability to group and sort, and now lists all machines in the selected deployment group.

The screenshot shows the Neuron ESB Explorer application window. The title bar reads "Neuron ESB Explorer". Below the title bar is a menu bar with "File", "View", "Tools", and "Help". A status bar at the top indicates "You are working offline. D:\Neuron\Testing\3.5Main\DemoSolution2". The left sidebar contains a tree view with "Activity" selected, showing sub-items like "Real-Time", "Running History", "Endpoint Health", "Database Reports", "Active Sessions", "Message History", "Failed Messages", and "Workflow Tracking". The main pane is titled "Endpoint Health" and "Dashboard Monitoring Connected". It shows a deployment group of "Development" and a timestamp of "9:24:21 AM". Below this is a table of endpoints. The table has columns: Host Name, Type, Name, State, Last Heartbeat, Message Rate, Message Processed, Warnings, and Errors. A context menu is open over the table, showing options: "Restart Service", "Restart Service On All Machines", "Stop Service", and "Stop Service On All Machines".

Host Name	Type	Name	State	Last Heartbeat	Message Rate	Message Processed	Warnings	Errors
MWASZNIQ...	Adapter	HttpEndpoint	Disabled	5/25/2018 9:18:11 AM	0	0	0	0
MWASZNIQ...	Adapter	MQTopicEndpoint	Disabled	5/25/2018 9:18:11 AM	0	0	0	0
MWASZNIQ...	Adapter	MQSeriesSolicitResponse	Started	5/25/2018 9:24:02 AM	0	0	0	48
MWASZNIQ...	Adapter	sftpPublishGMAIL	Disabled	5/25/2018 9:18:11 AM	0	0	0	0
MWASZNIQ...	TCP Publishing Service	RequestWorkflow	Started	5/25/2018 9:24:05 AM	0	0	0	0
MWASZNIQ...	TCP Publishing Service	Irvine	Started	5/25/2018 9:24:05 AM	0	0	0	0
MWASZNIQ...	TCP Publishing Service	Demo	Started	5/25/2018 9:24:05 AM	0	0	0	0
MWASZNIQ...	TCP Publishing Service	LA	Started	5/25/2018 9:24:05 AM	0	0	0	0
MWASZNIQ...	TCP Publishing Service	MSMQTest	Started	5/25/2018 9:24:05 AM	0	0	0	0

Neuron ESB Endpoint Health – Real time monitoring of both Neuron ESB Messaging and Workflow. Endpoint Hosts (child process) and their Endpoints can be stopped and started from here. Real time activity of Endpoints can be monitored.

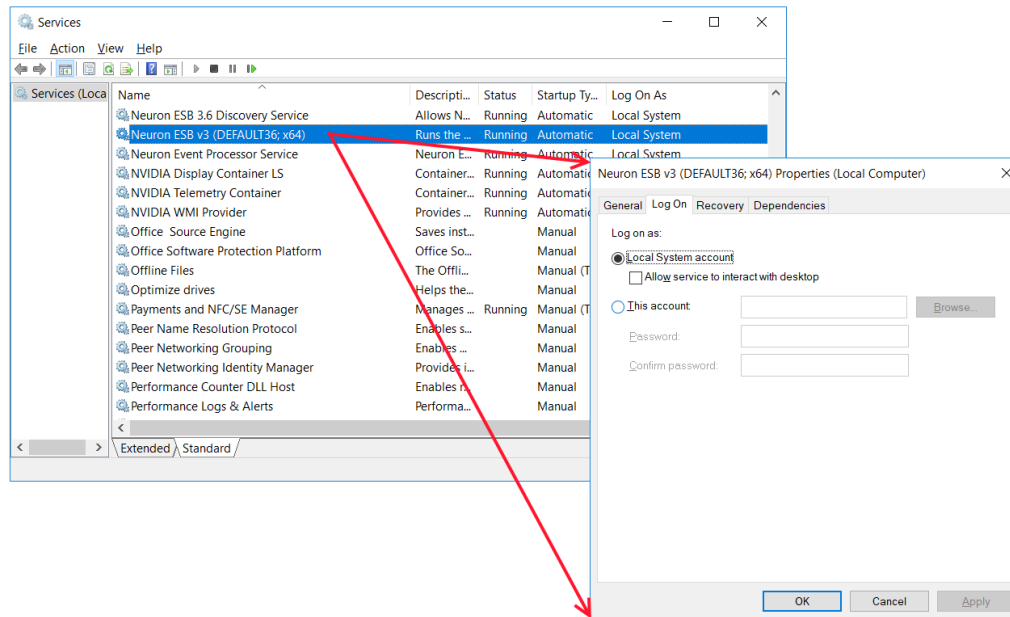
The Neuron ESB Endpoint Health interface has a scalable horizontal divider, separating Neuron ESB Messaging entities such as Topics, Adapter and Service Endpoints that are hosted in the Neuron ESB Runtime service from those hosted in the Endpoint Host environment. A context menu is exposed at the entity level that allows users to restart any selected endpoint or Endpoint Host.

Security

Neuron ESB has always provided options for managing runtime security. In previous versions of Neuron ESB, the Neuron ESB Runtime could be set to run under the context of a specific Windows local user or domain account, providing it had the necessary permissions and was granted the “Log on as a Service” right. Any service or endpoint hosted by the Neuron ESB Runtime would essentially inherit that security context during execution. If an Availability Group was created, it would inherit those same credentials. Unfortunately, this meant that there was no way to run a given Workflow Endpoint under a set of

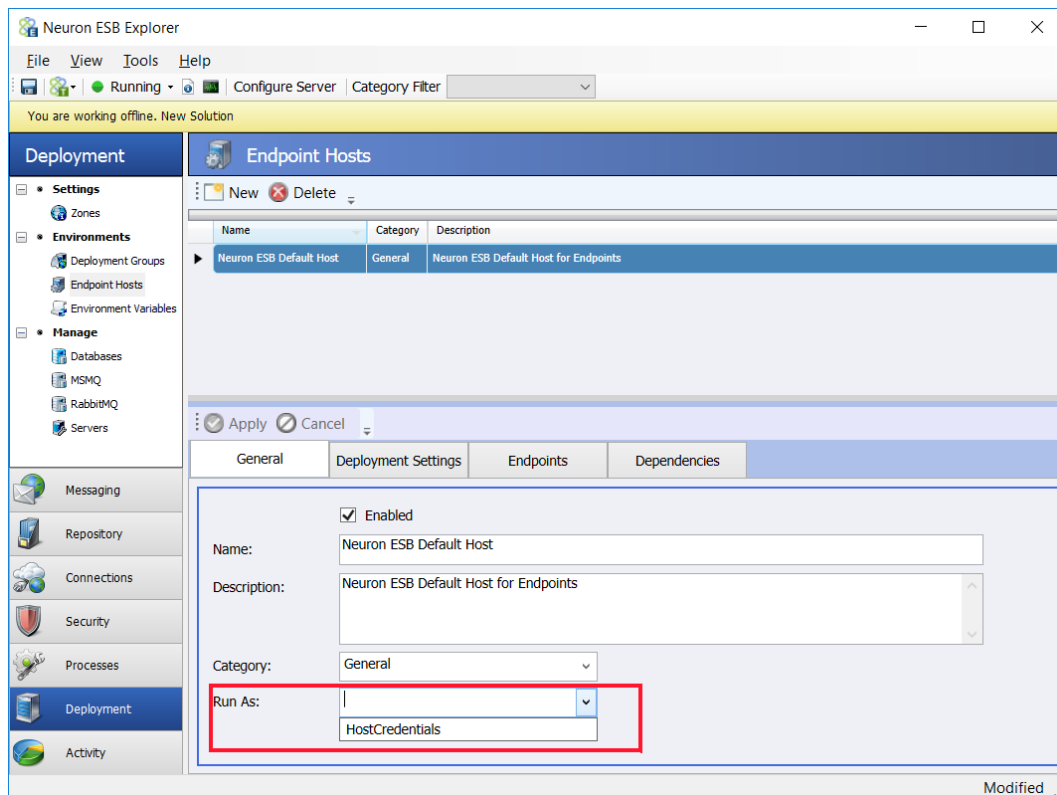
credentials other than those set at the Neuron ESB Runtime level. Only Adapter Endpoints had a “Run As” option which allowed users to specify an alternate set of credentials for an Adapter Endpoint to use at runtime.

NOTE: Requirements for changing the Neuron ESB Runtime service account can be found in the “Neuron ESB Runtime - Running under User Account” section of the Readme.html document installed in the default location of the Neuron ESB installation folder.



Microsoft Windows Service Control Manager – Displays the Neuron ESB Runtime service set to use the Local System Account credentials.

We modified this capability in Neuron ESB 3.6, introducing a Run As option at the Endpoint Host level. Unlike the service account used for the Neuron ESB Runtime service, these credentials do not require the “Log on as a Service” right. However, all other caveats that apply to the Neuron ESB runtime service account apply to the Endpoint Host credentials as well. Using Neuron ESB 3.6, users can create the appropriate Windows Credential by navigating to Security->Authentication->Credentials within the Neuron ESB Explorer. Once created, this credential will appear as an account that can be assigned to an Endpoint Host via its Run As property.



Neuron ESB Explorer – Endpoint Host – Run As property can be set with a Windows Credential stored in the Security repository of the Neuron ESB solution. The Endpoint Host will run under the credentials context at runtime.

The Run As capability for Endpoint Hosts functions much like the Run As for Adapter Endpoints. Everything executing in the Endpoint Host process space will execute under the context of the Run As credential. That means any resource that the endpoint, its underlying Party or Business Processes access will be under the context of the assigned credential. Essentially this provides the ability to change the security context at the endpoint level or for a group of endpoints, regardless of their type (i.e. Adapter, Service or Workflow).

Memory Consumption

While developing the new hosting model for Neuron ESB 3.6, we took the opportunity to reevaluate the memory utilization footprint of both the Neuron ESB Runtime and Endpoint Host. Historically, the Neuron ESB Runtime made liberal use of memory to enhance execution time. It also provided the configuration and document repositories as in memory objects that could be retrieved and manipulated at runtime, by developers as well as internal Neuron ESB services. But there was a price, especially due to our usage of .NET AppDomains. It wasn't unusual where large solutions with 100's of Business Processes, 100s of Service Endpoints and 100s of Adapter Endpoints could easily consume 10+ Gigabytes of Working Set memory on startup.

In Neuron ESB 3.6, we completely changed how we manage and share configuration objects internally as well as shared resources. Users will find that solutions which required 10+ gigabytes of memory on startup, will consume anywhere from 2X to 5X less memory.

We tested this with a large solution, containing almost 200 Adapter Endpoints. We chose Adapter Endpoints as they tend to consume more memory than Service or Workflow Endpoints due to the proprietary assemblies that must be loaded with each one. Additionally, the solution contained a couple of hundred large Business Processes, 50 Topics, several Service Endpoints and other Neuron specific entities. When this solution was loaded and started in Neuron ESB 3.5.4, it consumed over 11 gigabytes of Working Set memory and took several minutes to startup.

We migrated the solution to Neuron ESB 3.6. We assigned all the Service and Adapter Endpoints to a single Endpoint Host. The Neuron ESB Runtime was relegated to Messaging Publishing Services (i.e. Topics) and our internal services. A noted observation was the startup time. The entire solution took less than 1 minute to startup, whereas under Neuron ESB 3.5, several minutes was required.

The screenshot shows the 'Endpoint Health' dashboard in Neuron ESB Explorer. The top section, 'Dashboard Monitoring', shows the system is 'Connected' at 12:44:39 PM. Below this, there are two main tables. The first table, 'State: Started (4 items)', lists the following:

- Type : MSMQ Publishing Service (1 item)
- Type : Named Pipe Publishing Service (10 items)
- Type : RabbitMQ Publishing Service (1 item)
- Type : TCP Publishing Service (40 items)

The second table, 'State: Started (1 item)', lists the following:

- Host Name: MWASZNIQ... Endpoint Host
- Name: Endpoint Host 1
- Last Heartbeat: 5/25/2018 12:44:25 PM
- Active: 0
- Pending: 0
- Waiting: 0
- Completed: 0
- Aborted: 0
- Suspended: 0
- Rate: 0
- Warn: 0

Below the second table, there are three more sections:

- State: Disabled (13 items)
- State: FAILED (4 items)
- State: Out of Service (31 items)

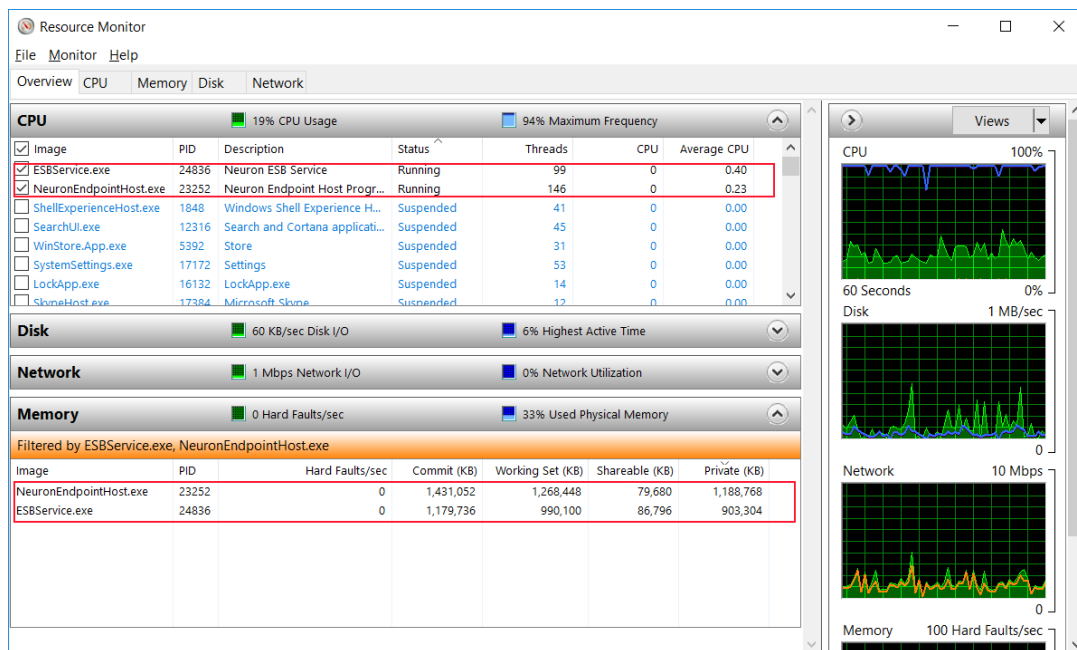
The bottom section, 'State: Started (186 items)', lists the following:

- Host Name: MWASZNIQ...
- Type: Adapter
- Name: AmexFileIn.Endpoint
- Last Heartbeat: 5/25/2018 12:44:25 PM
- Active: 0
- Pending: 0
- Waiting: 0
- Completed: 0
- Aborted: 0
- Suspended: 0
- Rate: 0

Red arrows point from the '52 Topics + 186 Endpoints' text to the 'State: Started (4 items)' and 'State: Started (186 items)' sections.

Neuron ESB Explorer – Endpoint Health in 3.6 – Endpoint Health screen showing 52 started Topics, 176 started Adapter Endpoints, 6 started Client Connectors, 3 started Service Connectors and 1 started Workflow. This solution also included 212 Business Processes. Total startup time was approximately 1 minute.

Once the solution was completely started in Neuron ESB 3.6, we noticed a significant reduction in the Working Set memory. By default, an Endpoint Host startup memory footprint is just under 200 MB of memory (no endpoints assigned). The total combined Working Set memory of the Neuron ESB Runtime and Endpoint Host was under 2.3 gigabytes. This represents about a 5X improvement in memory utilization.



Windows Resource Monitor – Displays CUP and Memory statistics of the Neuron ESB Runtime (ESBSERVICE.exe) and the Endpoint Host (NeuronEndpointHost.exe) after startup.

Adapters/Connectors

Adapters are a key piece of capability in the world of an integration broker. They serve as the bridge to and from the bus between applications, databases, transports and protocols. The completeness of “what” you ship as well as how easy it is for others to build their own adapters is critical in accelerating the development of any solution. Not only is Neuron ESB one of the easiest adapter frameworks to learn in the industry, it also has pre-built adapters for most commodity transports, queuing technologies, databases and on-premise/cloud applications. Staying true to our roots, several changes have been introduced in this release to make connectivity easier, more intuitive as well as performant.

Self-Registration

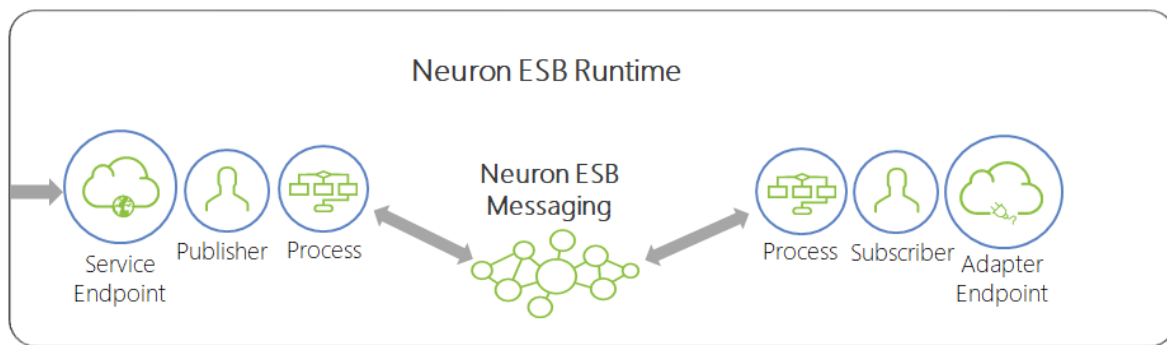
In previous versions of Neuron ESB, users had to go through multiple steps before they could use an adapter. The first step was to manually register the adapter followed by creating an Adapter Endpoint that could be used to interact with their system of choice. Starting with version 3.6 of Neuron ESB, all .NET based adapters, either shipped with or custom developed, are dynamically registered. There is no longer a need to register individual adapters before creating Adapter Endpoints. As part of this effort, we took the opportunity to rename the adapters shipped with Neuron ESB to more closely align them with their respective technologies. When migrating from previous versions, users must open their existing solution using the 3.6 version of the Neuron ESB Explorer. The Neuron ESB Explorer will automatically update the adapter names used in any existing Adapter Endpoints. Any existing registered adapters will be removed from the solution, as they are no longer needed. For reference, names of the Adapters prior to the 3.6 release, as well as their corresponding updated names are listed in the table below.

Renamed Adapters

<i>Previous Names</i>	<i>Version 3.6</i>
Active Directory Adapter	Microsoft Active Directory
AWS SQS Adapter	Amazon Simple Queue Service
ActiveMQ Adapter	Apache ActiveMQ
Apple Push Notification Service Adapter	Apple Push Notification Service
CertainSafe Adapter	CertainSafe
Client Socket Adapter	Client TCP Socket
File Adapter	File
Ftp Adapter	FTP/FTPS
WebSphere MQ Adapter	IBM MQ
Marketo Adapter	Marketo
Azure Service Bus Adapter	Microsoft Azure Service Bus
Dynamics AX Adapter	Microsoft Dynamics AX
Dynamics CRM 4.0 Adapter	Microsoft Dynamics CRM 4.0
Dynamics CRM Web API Adapter	Microsoft Dynamics CRM Web API
Dynamics CRM XRM Adapter	Microsoft Dynamics CRM XRM
Dynamics GP eConnect Adapter	Microsoft Dynamics GP
Microsoft Exchange Adapter	Microsoft Exchange
MSMQ Adapter	Microsoft MSMQ
SharePoint Subscription Adapter	Microsoft SharePoint
SQL Service Broker Adapter	Microsoft SQL Service Broker
NetSuite Adapter	NetSuite
Neuron.Crm2011.Workflow.Setup64	Neuron.Crm.Workflow.Setup64
Neuron.Crm2011.Workflow.Setup64Lite	Neuron.Crm.Workflow.Setup64Lite
Odbc Adapter	ODBC
POP3 Adapter	POP3
RabbitMQ Adapter	RabbitMQ
Redis Adapter	Redis
Salesforce.com Adapter	Salesforce
SAP Adapter	SAP
Server Socket Adapter	Server TCP Socket
SFtp Adapter	SFTP
Smtplib Adapter	SMTP
Twilio Adapter	Twilio
Zuora Adapter	Zuora

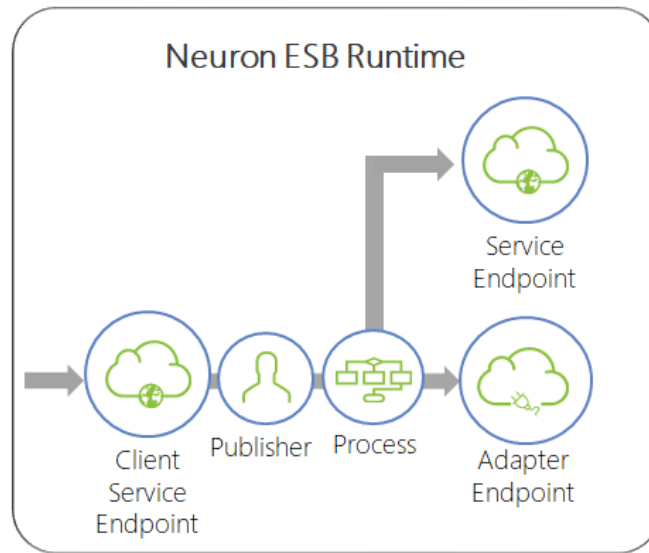
Messaging Configuration

Historically, Neuron ESB used its powerful messaging sub system to abstract and decouple service and adapter endpoints from one another. In essence, endpoints would adopt either a publisher or subscriber role, where neither would have to know about the other and, where modifying one would not affect the other. This model remains incredibly beneficial for those systems requiring a loosely coupled design. New subscribing or publishing endpoints could be added without impacting the processing of the other endpoints. However, those designs usually employ one way, multicast patterns. There are other scenarios (request/response patterns for example) though where inserting messaging as a decoupling agent makes little sense. In those scenarios it imposes unnecessary overhead, configuration and restrictions. For example, it's unusual in request/response patterns that there would ever be more than one subscribing endpoint at a time fulfilling a request.



This depicts Messaging being used to decouple Service and Adapter Endpoints. Prior to the 3.5 release all endpoint communication was brokered by using the Neuron ESB Messaging sub system.

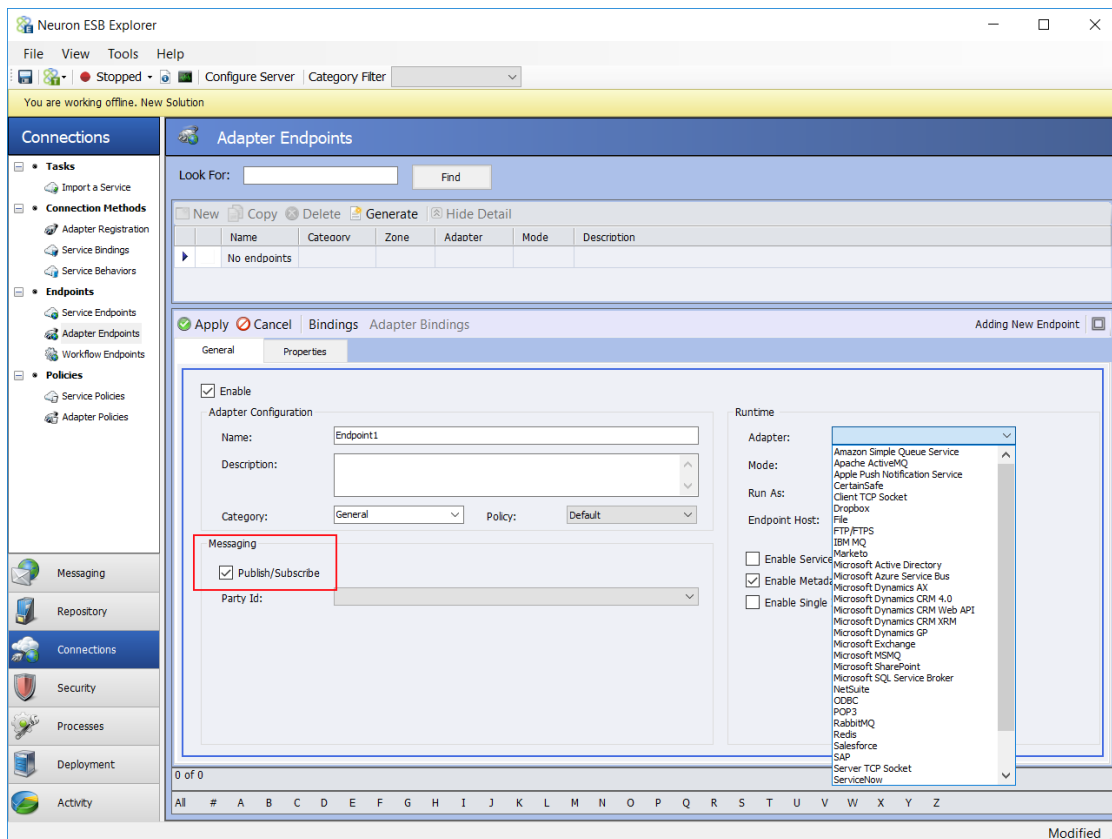
In Neuron ESB 3.5, we introduced the capability for users to call Adapter and Service Endpoints directly (without the messaging sub system) using either the Business Process engine or the Workflow engine. However, messaging elements would still need to be created, even though those elements were not used at runtime. For example, to create an Adapter Endpoint that could be used in the Business Process or Workflow Designer users were still required to first create a Topic, Publisher and Subscriber, and then associate those elements with the Adapter Endpoint using the Neuron ESB Explorer user interface.



In Neuron ESB 3.5, Adapter and Service Endpoints can be executed directly by a Neuron ESB Business Process rather than having the original request forwarded across the Neuron ESB Messaging sub system.

In this release we've made it a lot easier for developers to create Adapter Endpoints that they intend to execute directly within our Business Process or Workflow engines. Starting with Neuron ESB 3.6, users can choose to simply disable the Messaging option located on the General tab of an Adapter Endpoint for all subscribe mode type of Adapter Endpoints. This allows the developer to create an Adapter Endpoint without needing to first create a Topic or Subscriber. At runtime, the Adapter Endpoint can be called directly within the Business Process or Workflow Engine but will not be a "live" subscriber to messages published by other systems. Some immediate benefits this provides are:

- **Reduced Complexity**
 - No need to create a Topic or Subscriber before configuring an Adapter Endpoint.
- **Reduced Runtime Utilization**
 - The Adapter Endpoint does not need to be loaded in memory with its own resources.
 - The Topic Publishing Service is never created.
- **Faster Startup Time**
 - Since there is no subscribing Adapter Endpoint, the master runtime does not attempt to start it as a manageable endpoint.
 - The Topic is also not created and loaded by the master runtime.

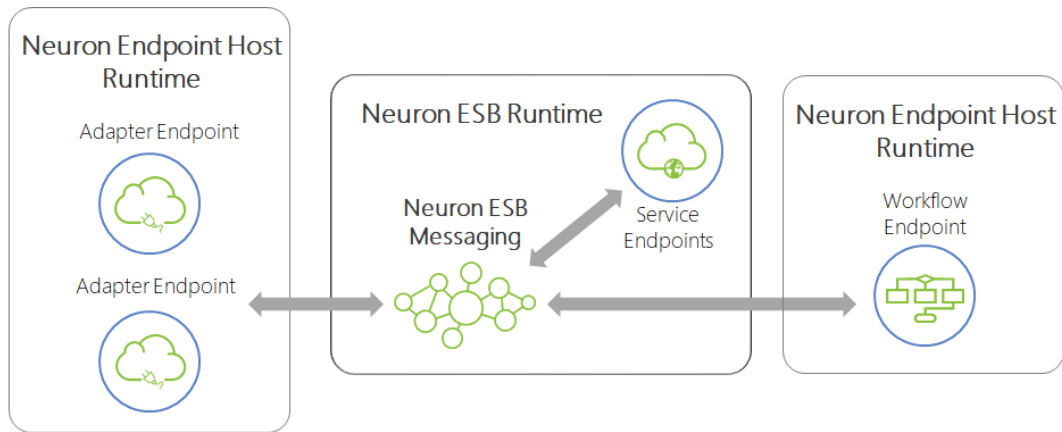


Neuron ESB Explorer – Adapter Endpoint User Interface. This has been refactored in 3.6. This now has a Message option which can be enabled or disabled by selecting the “Publish/Subscribe” checkbox encircled in Red.

Hosting Model

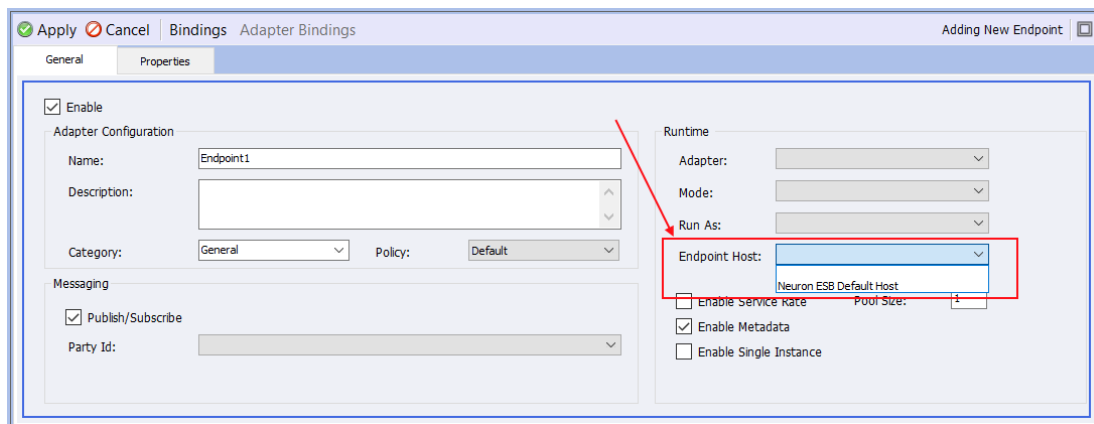
The Neuron ESB 3.6 release introduces a new hosting model that allows Adapter Endpoints (and all endpoints) to be hosted either independently or with other endpoints in a configured Endpoint Host. This is in contrast to previous releases where, although each Adapter Endpoint was isolated in their own .NET App Domain, they were hosted within the main Neuron ESB runtime process (i.e. esbservice.exe) with all other internal services, Messaging Publishing services and Service Endpoints.

Using this new model, one or more Adapter Endpoints can be assigned to a dedicated Endpoint Host to provide resiliency, failover, and scale as well as process isolation. The latter becomes important in the case of a misbehaving endpoint or, Business Processes associated with that endpoint, which have the potential to crash a process/host. If isolated to an Endpoint Host, the misbehaving endpoint or Business Processes can’t affect any other running endpoints or the core Neuron ESB runtime process.



Neuron ESB 3.6 Hosting Model – All Adapter, Service and Workflow Endpoints can now be hosted in dedicated hosts named “Endpoint Host” (i.e. NeuronEndpointHost.exe). All internal services and Messaging services are isolated in .NET App Domains and hosted in the Neuron ESB runtime (i.e. esbservice.exe).

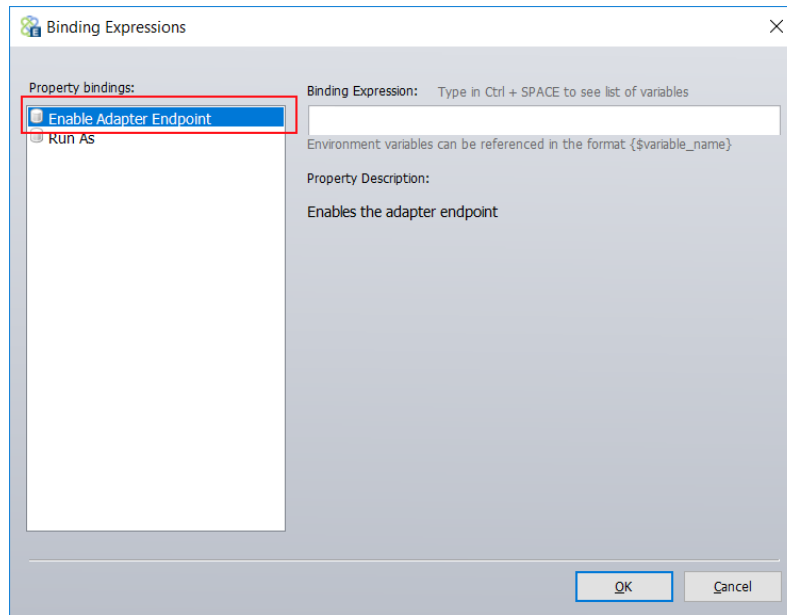
In the Neuron ESB 3.6 release, users can configure an Adapter Endpoint to run in either the existing Neuron ESB runtime host (as in previous versions) or the new Neuron Endpoint Host. The default behavior is for new Adapter Endpoints to run in the existing Neuron ESB runtime host. Users can change the hosting model by selecting a configured Neuron Endpoint Host in the Endpoint Host property dropdown box located on the General tab of the Adapter Endpoint.



Neuron ESB Explorer – Adapter Endpoint User Interface. This has been refactored in 3.6. This now has an Endpoint Host dropdown box property that lists all available Endpoint Hosts. When none is selected, the original Neuron ESB Runtime Host (i.e. esbservice.exe) will be used.

Environment Variables

To make deployments easier and more configurable at the environment level, all Adapter Endpoints now support the use of Environment Variables to disable/enable an Adapter Endpoint. The property exposed via the Bindings Expressions dialog is “Enable Adapter Endpoint”.



Binding Expressions dialog – Used to map Environment Variables to Adapter Endpoint properties. This is displayed by selecting the Bindings toolbar button on the Adapter Endpoint.

ServiceNow Adapter

Neuron ESB 3.6 includes a new ServiceNow Adapter that provides access to an instance of ServiceNow. Using this adapter, organizations can easily integrate their business applications such as SAP, NetSuite or Marketo with ServiceNow. This adapter operates in both subscribe as well as solicit/response (or query) mode. The adapter utilizes the ServiceNow OAuth Provider to manage access to ServiceNow instances. Organizations can use the adapter to query for data, create, update, retrieve and delete entities and execute ServiceNow Functions and Actions. The adapter can be used to access the entire ServiceNow API. When developing applications that utilize this adapter, developers will be able to use the Adapter Metadata Generation Wizard to generate JSON samples that describe the request and/or response documents that are exchanged with ServiceNow. The Wizard can also generate Swagger documents, describing the APIs, which can be hosted by Neuron ESB Service Endpoints. Developers can also use the new Http Utility process step in our process designer to very easily define the calls to ServiceNow.

More information about the ServiceNow adapter can be found in the article posted here:

<https://www.neuronesb.com/resources/?post>

Several other important modifications have been made to Neuron ESB Adapters that help simplify the configuration and deployment of secure solutions. The entire list of changes can be found in the change log section of this document.

Services and OAuth

Neuron ESB includes a Service Broker that enables organizations to deploy Neuron ESB as a Service Gateway, providing mediation, security, hosting and a number of other services. Service Connectors are essentially registrations within Neuron ESB that point to existing services (i.e. REST or SOAP APIs) hosted

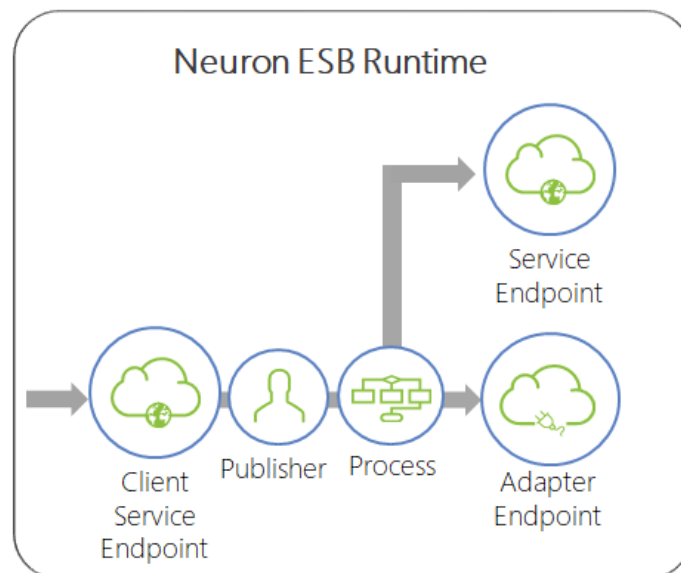
within an organization, partner or cloud domain. Client Connectors on the other hand are REST or SOAP APIs created using the Neuron ESB toolset and hosted by Neuron ESB. These are live services that can accept external requests, initiate business processing logic and/or broker to external services or applications.

Messaging Configuration

In this release, Service Endpoints have been redesigned to use the same Messaging option as introduced to Adapter Endpoints. Users can choose to simply disable the Messaging option located on the respective tab of the Client Connector (i.e. publisher role) and Service Connector (i.e. subscriber role) of the Service Endpoint. Unlike Adapter Endpoints though, this is not limited to subscribe mode Service Endpoints (i.e. Service Connectors). Messaging can be disabled on Client Connectors as well.

Service Connectors

Just as with Adapter Endpoints, subscribe side Service Endpoints (i.e. Service Connectors) can be executed directly within a Business Process or Workflow.

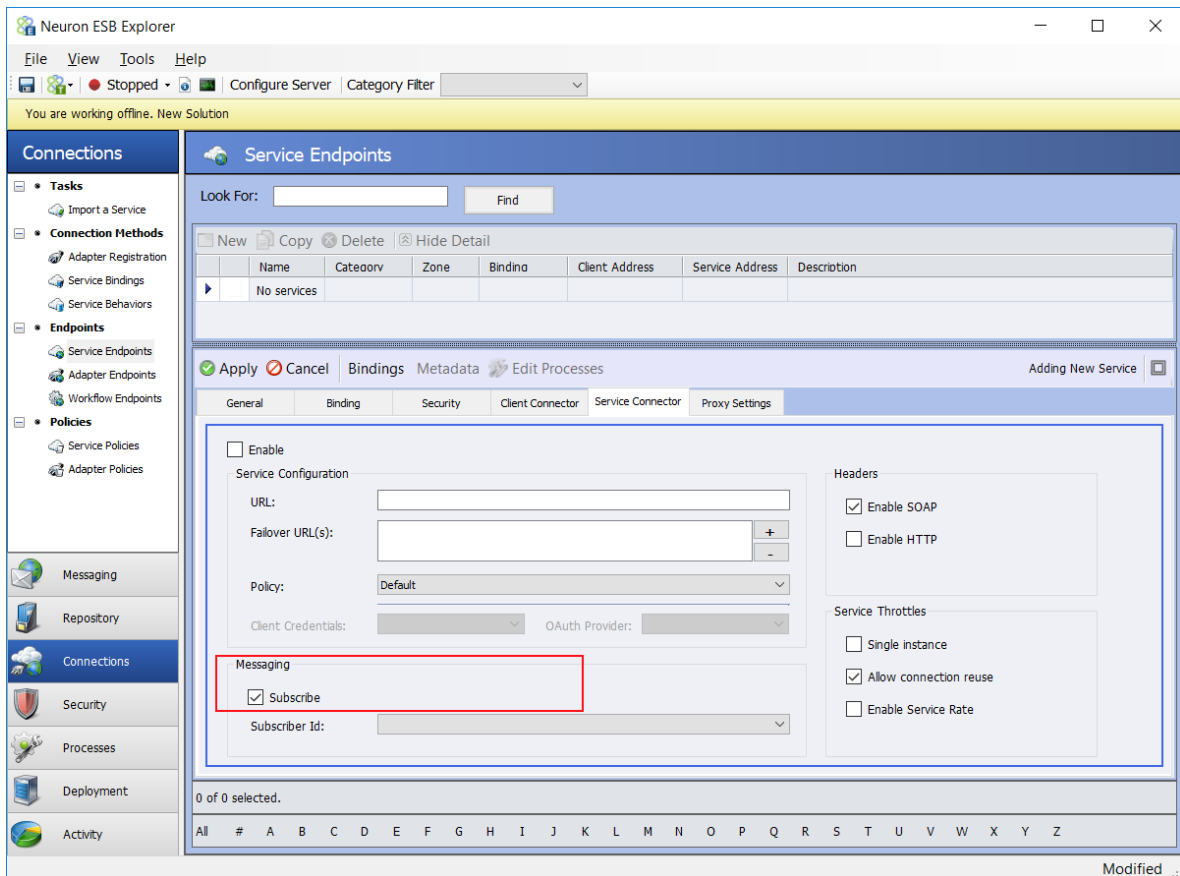


In Neuron ESB 3.5, Adapter and Service Endpoints can be executed directly by a Neuron ESB Business Process rather than having the original request forwarded across the Neuron ESB Messaging sub system.

Because of their similarity with subscribe mode Adapter Endpoints, in Neuron ESB 3.6 we made it just as easy for developers to create Service Connectors that they intend to execute directly within our Business Process or Workflow. Users can now specifically choose to disable the Messaging option located on the Service Connector tab of the Service Endpoint. This allows the developer to create a Service Connector without needing to first create a Topic or Subscriber. At runtime, the Service Endpoint can be called directly within the Business Process or Workflow but will not be a “live” subscriber to messages published by other systems. The immediate benefits to this approach are identical to Adapter Endpoints:

- **Reduced Complexity**

- No need to create a Topic or Subscriber before configuring a Service Connector.
- **Reduced Runtime Utilization**
 - The Service Connector does not need to be loaded in memory with its own resources. The Topic Publishing Service is never created.
- **Faster Startup Time**
 - Since there is no subscribing Service Connector, the master runtime does not attempt to start it as a manageable endpoint.
 - The Topic is also not created and loaded by the master runtime.



Neuron ESB Explorer – Service Connector User Interface. This has been refactored in 3.6. This now has a Messaging option which can be enabled or disabled by selecting the “Subscribe” checkbox encircled in Red.

Client Connectors

In this release, although Messaging can also be disabled for Client Connectors, it is for a different purpose than that of Service Connectors. Rather than disabling live hosting of the Service Endpoint, disabling Messaging allows the user to configure a Client Connector to call and execute Business Processes directly, without the need to configure the Client Connector with a Publisher and Topic.

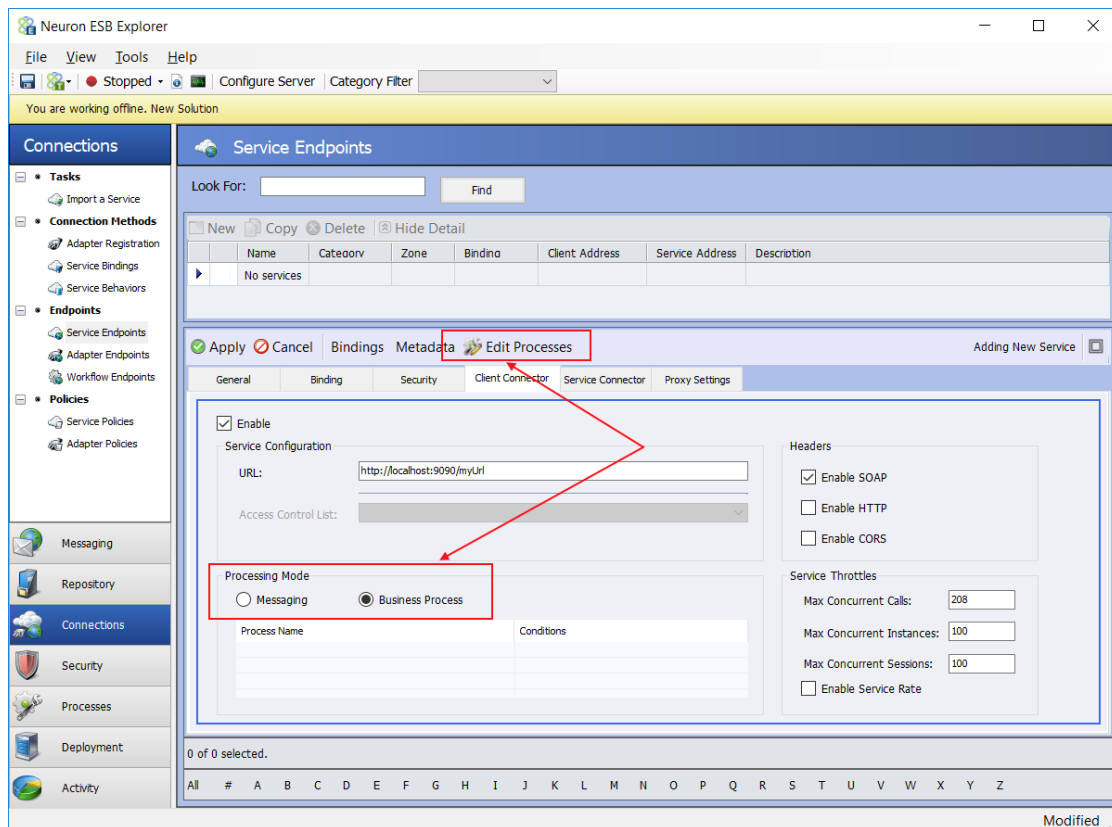
In previous releases, users could always expose a Business Process as a web service (REST or SOAP) indirectly. The steps were simple, but like everything with multiple parts, complexity introduces itself. For example:

1. Create a Topic
2. Create a Publisher
3. Create a Business Process
4. Configure the Publisher to call the Business Process
5. Create a Client Connector
6. Configure the Client Connector to send the inbound message to the Publisher, which executes the Business Process and forwards to the Topic

Using Neuron ESB 3.6, we can reduce this to a 3 step task and eliminate unnecessary dependencies i.e.

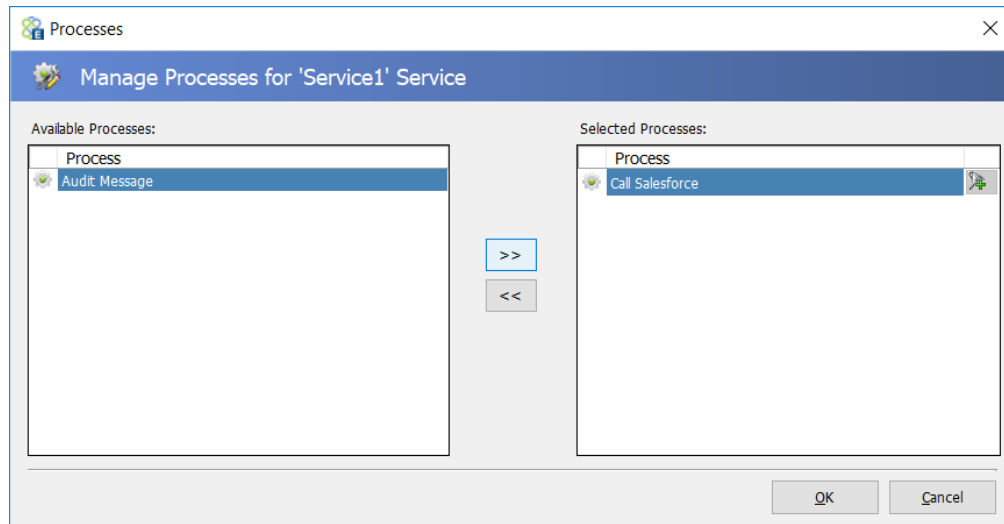
1. Create a Business Process
2. Create a Client Connector
3. Configure the Client Connector to call the Business Process

Selecting the Business Process option of the Processing Mode section of the Client Connector effectively disables the Messaging option and allows users to select one or more Business Processes to execute directly from the Service Endpoint UI.



Neuron ESB Explorer – Client Connector User Interface. This has been refactored in 3.6. This now has a Processing Mode group box which contains a Messaging and Business Process mode option. If Messaging is selected, controls appear to allow users to select a Publisher and Topic. If Business Process is selected, users can select the Edit Processes toolbar button to select Business Processes to execute (encircled in Red).

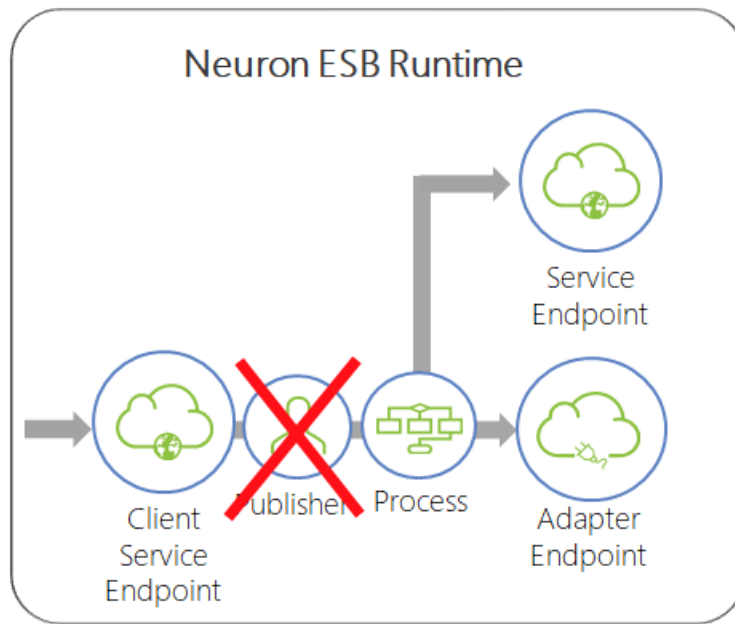
Selecting the “Edit Processes” toolbar button, displays the same Processes dialog used to select Business Processes for Publishers and Subscribers.



Processes dialog – Launched from the Edit Processes toolbar button of the Client Connector. Allows user to select one or more Business Processes for the Client Connector to directly execute.

Once a Client Connector is configured to execute Business Processes directly through the new Processing Mode option, developers will no longer need to create a Publisher or Topic as part of the configuration process. Additionally, when creating Business Processes, users can forego the placement of the Cancel Process Step as the last step in the Business Process when dealing with request/reply patterns. Client Connectors directly executing Business Processes incur other benefits as well including

- Reduced Complexity
 - No need to create a Topic or Publisher before configuring a Client Connector.
 - No mapping of Business Process to Publisher and Publisher to Client Connector
 - No need to use a Cancel Process Step in Business Processes to return the reply message to the calling client or prevent a message from being forwarded to a Topic.
- Reduced Runtime Utilization
 - The Topic Publishing Service is never created.
 - Less processing and memory consumed by the Client Connector
- Faster Startup Time
 - The Topic is not created and loaded by the master runtime.
- Better Performance
 - All requests are no longer being brokered by a Publisher object



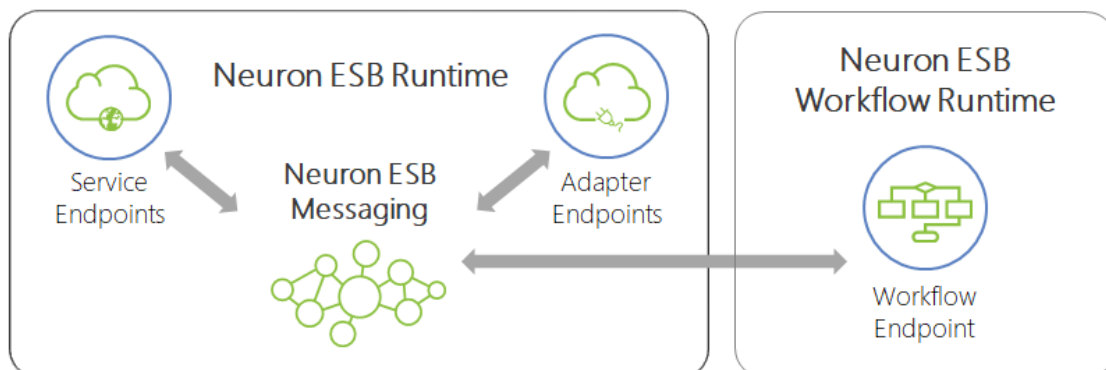
In Neuron ESB 3.6, Client Connectors can execute Business Processes directly, without an intermediate Publisher brokering messages across the Neuron ESB Messaging sub system.

Limitations

Client Connectors configured for direct execution of Business Processes should be the preferred mode of operation for users. This will generally provide simpler and more efficient execution of messages and Business Processes. Even though in this configuration there is no Publisher or Topic associated with an endpoint, messages received by the Client Connector can still be published to topics using the Publish Process Step within the Business Processes it's executing. All Topic transports are supported. The only use case that is not supported is using MSMQ based topics when the Publish Step is configured with the Request Semantic.

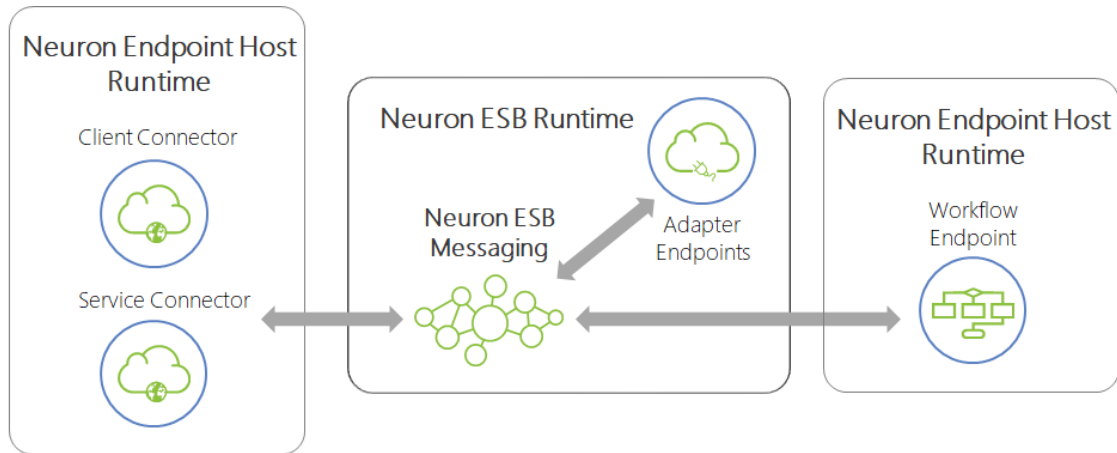
Hosting Model

Just as with Adapter Endpoints, in previous releases each Service Connector and Client Connector were isolated in their own .NET App Domain. They were hosted within the main Neuron ESB runtime process (i.e. esbservice.exe) with all other internal services, Messaging Publishing services and Adapter Endpoints.



Neuron ESB 3.5 Hosting Model – Workflow Endpoints are hosted in dedicated hosts named “Availability Groups” (i.e. esbhost.exe). All internal services, Messaging services, Service Endpoints and Adapters Endpoints are isolated in .NET App Domains and hosted in the Neuron ESB runtime (i.e. esbservice.exe).

In this release, Service Endpoints have been redesigned to use the new Neuron Endpoint Host model that Adapter Endpoints can now use. In Neuron ESB 3.6, all endpoints (Workflow, Service and Adapter) share the same hosting model. That means that both Client Connectors and Service Connectors can be configured to run using either the existing Neuron ESB runtime host (esbservice.exe) or the newer Endpoint Host for process isolation.



Neuron ESB 3.6 Hosting Model – All Adapter, Service and Workflow Endpoints can now be hosted in dedicated hosts named “Endpoint Host” (i.e. NeuronEndpointHost.exe). All internal services and Messaging services are isolated in .NET App Domains and hosted in the Neuron ESB runtime (i.e. esbservice.exe).

In the Neuron ESB 3.6 release, users can configure a Service Endpoint to run in either the existing Neuron ESB runtime host (as in previous versions) or the new Neuron Endpoint Host. The default behavior is for new Service Endpoints to run in the existing Neuron ESB runtime host. Users can change the hosting model by selecting a configured Neuron Endpoint Host in the Endpoint Host property dropdown box located on the General tab of the Service Endpoint.

Apply Cancel Bindings Metadata Edit Processes Adding New Service

General Binding Security Client Connector Service Connector Proxy Settings

☒ Enable Service Endpoints

Name: Service1

Description:

Category: General

Endpoint Host: Neuron ESB Default Host

Binding: BasicHttp

Behavior:

Neuron ESB Explorer – Service Endpoint User Interface. This has been refactored in 3.6. This now has an Endpoint Host dropdown box property that lists all available Endpoint Hosts. When none is selected, the original Neuron ESB Runtime Host (i.e. esbservice.exe) will be used.

OAuth 2.0

The Neuron ESB 3.5 CU4 release introduced OAuth 2.0 support for authorizing web service invocations using Service Connectors (i.e. HTTP/REST API service endpoints) and supported Adapters. Using OAuth, Neuron ESB can obtain access tokens that can be passed to REST services to authorize protected resource invocations. In CU4, OAuth bearer tokens based on [RFC 6750](#) are supported.

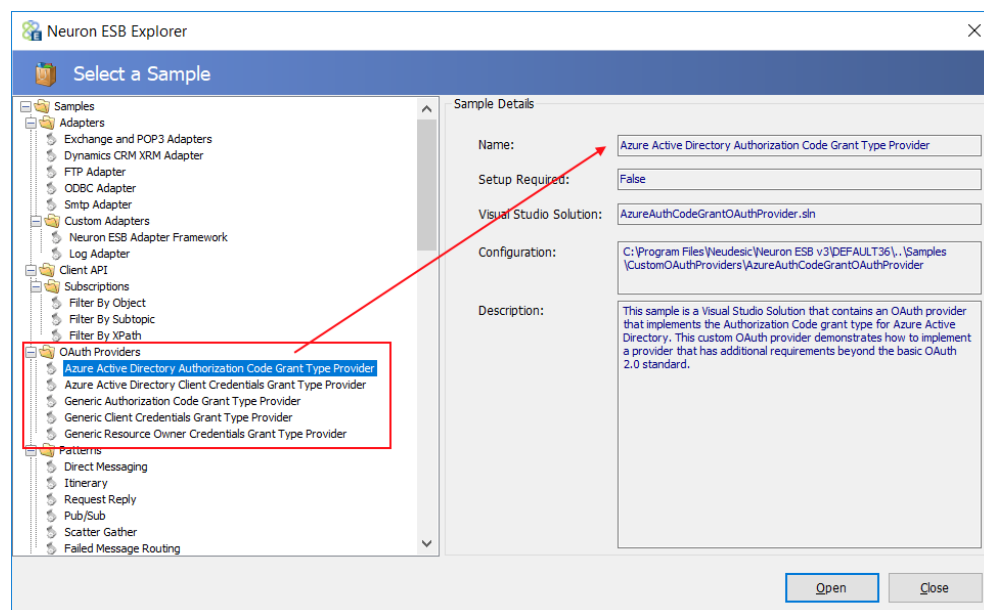
In the 3.6 release, we've extended this support, adding three new providers (ADP, ServiceNow and Thinktecture Identity Server) as well as the ability for users to create their own custom OAuth provider that can be re-used within the Neuron ESB Explorer. Below are the OAuth providers currently configurable within the Neuron ESB Explorer:



- ADP
- Amazon
- Azure Active Directory/ADFS
- Dropbox
- Facebook
- Foursquare
- GitHub
- Google
- Instagram
- LinkedIn
- Microsoft Live
- ServiceNow
- SoundCloud
- SourceForge
- Thinktecture Identity Server
- Tumblr
- Twitter
- Yahoo!

Custom OAuth Providers

Using Neuron ESB 3.6 users can develop their own OAuth providers to use with Neuron ESB endpoints. four different Grant Types are supported and several sample custom OAuth providers are now included within the Neuron ESB Sample Browser which can be modified and used as templates.



OAuth 2.0 Grant Types

When developing custom OAuth providers, the first step is to determine which grant type(s) must be supported. There are five grants for obtaining an access token:

- Authorization Code Grant
- Implicit Grant
- Client Credentials Grant
- Resource Owner Credentials Grant
- Refresh Token Grant

Each of these except for the Implicit Grant type can be used by Neuron ESB. The Implicit Grant type is intended to be used by user-agent-based clients, like single page web apps that are unable to keep a client secret. Below are descriptions of the other four grants and how Neuron ESB utilizes them.

Developing Custom OAuth Providers

To develop a custom OAuth provider, either two or three classes must be implemented, depending on which OAuth grant type required:

- A Provider class which inherits from `Neuron.Esb.OAuth.OAuthProvider`
- An OAuth Client class, which inherits from either `Nemiro.OAuth.OAuth2Client` (OAuth 2.0) or `Nemiro.OAuth.OAuthClient` (OAuth 1.0)
- An optional Login class, which inherits from `Nemiro.OAuth.LoginForms.Login`. This is only needed for the Authorization Code grant type.

These are the steps for creating an OAuth provider:

1. Determine which grant type is needed to use with the OAuth provider.
2. Determine provider-specific requirements for the grant type that must be used
3. Build the OAuth Provider solution using Microsoft Visual Studio

Compile and Deploy

Once the custom OAuth provider solution is compiled, the .NET assembly must be copied/deployed to the following folder:

`<Neuron Install Location>\<Instance>\OAuthProviders`

To configure the custom OAuth provider, close and reopen the Neuron ESB Explorer and navigate to Security->Authentication->OAuth Providers. Click on the "New" button and select the custom OAuth provider from the Provider drop-down list.

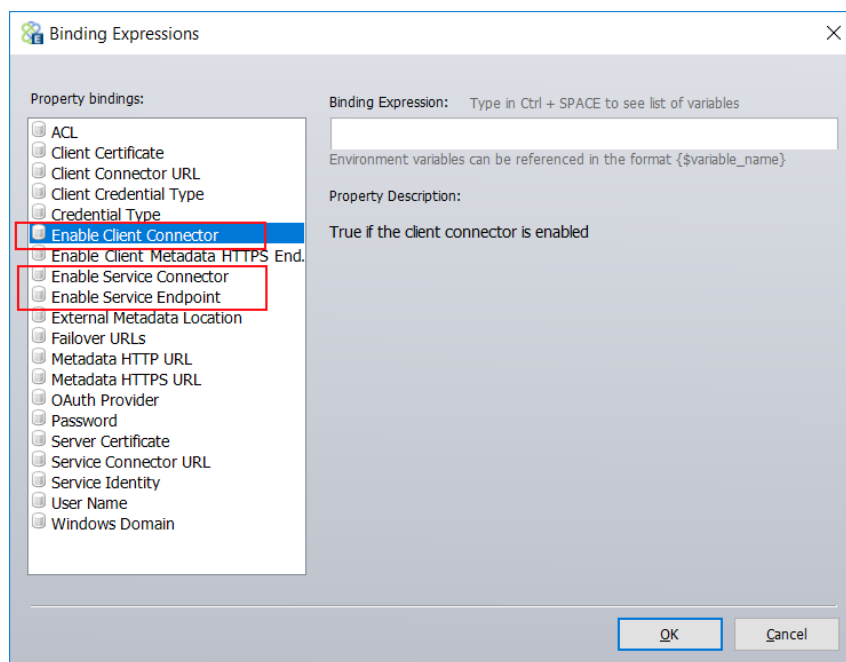
More information about creating custom OAuth providers can be found in the article posted here:

<https://www.neuronesb.com/resources/?post>

Environment Variables

To make deployments easier and more configurable at the environment level, all Service Endpoint, Client Connectors and Service Connectors now support the use of Environment Variables to disable/enable their respective function. The properties exposed via the Bindings Expressions dialog are:

- Enable Service Endpoint
- Enable Service Connector
- Enable Client Connector



Binding Expressions dialog – Used to map Environment Variables to Service Endpoint properties. This is displayed by selecting the Bindings toolbar button on the Service Endpoint.

Migration from 3.5

This release introduces significant changes in Neuron ESB's runtime environment and infrastructure. In consideration of this, users must upgrade elements of any existing solutions before attempting to run them using Neuron ESB 3.6. Other changes may be necessary if users have modified or introduced custom settings in Neuron ESB's native .config files, specifically in the AppSettings section. Overall, the migration should be straight forward for most users. One of the first things a user must do is update any existing solutions by opening them directly in the new version of the Neuron ESB Explorer.

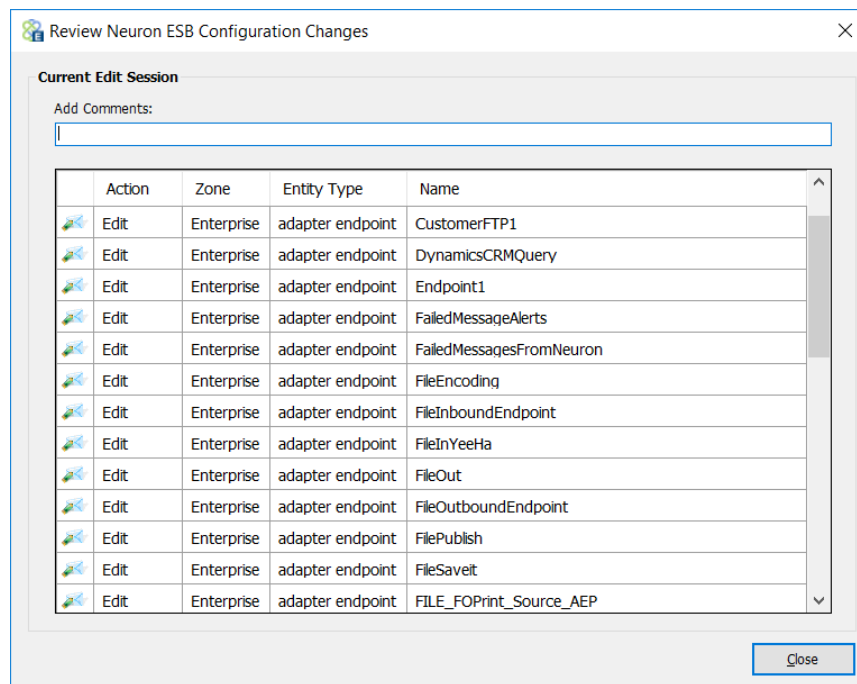
Existing Solutions

When migrating from previous versions, users must open their existing solution using the 3.6 version of the Neuron ESB Explorer. The Neuron ESB Explorer will automatically update their solution by doing the following:

- Create a new Endpoint Host named "Neuron ESB Default Host"
- Assign new Endpoint Host to all existing Deployment Groups.

- Assign all existing Adapter Endpoints to new Endpoint Host.
- Assign all existing Service Endpoints to new Endpoint Host.
- Update the internal Adapter Name property (not the Adapter Endpoint Name) of all existing Adapter Endpoints so that they properly map to all previously registered Adapters
- Remove all previously registered Adapters from the Adapter Registration section of the Neuron ESB Explorer
- Any Adapter Endpoints whose property names match either “Password”, “ConnectionString”, “key” or “passphrase” will have their respective values encrypted.

Once the updates are made, the user will be prompted with the Review Changes dialog similar to the one shown below, but specific to their solution.



The changes should be accepted and the solution should be saved for the changes to take affect and be committed.

Database

Every Neuron ESB solution requires a Microsoft SQL Server database to be associated with each Deployment Group. These databases can be created and configured by opening the solution within the Neuron ESB Explorer and navigating to Deployment->Manage->Databases.

In this release, the data type for the Sequence ID associated with the Audit tables has been changed from an int to a long. This requires users to update any existing databases used in Neuron ESB solutions. This can be done by simply clicking the “Test/Create” located on the database user interface within the Neuron ESB Explorer. Users can optionally run the SQL update scripts using the PowerShell script located in the installation folder of Neuron ESB. More information on updating the database can be found under

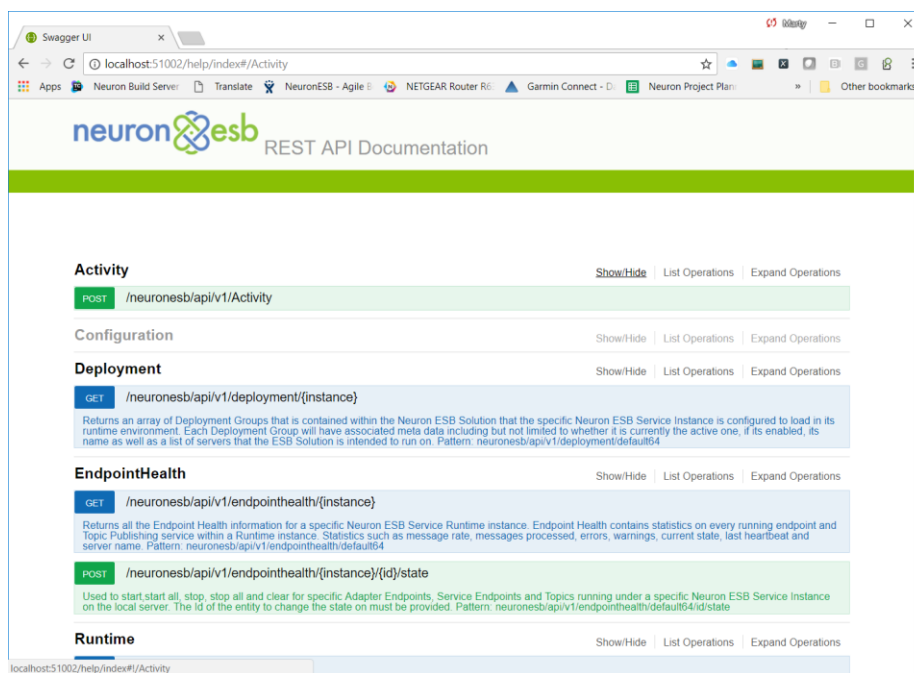
the “Upgrading an Existing Database” section within in the Readme.html file located at the root of the Neuron ESB installation folder.

If there are a large amount of records in the Audit tables, this update could take a significant amount of time. It’s strongly recommended that users first backup and purge the Audit Tables. This can be done either manually or by using the PurgeArchiveNeuron Microsoft SQL Agent Job. More information on updating the database can be found under the “Archive and Purge Neuron Database Job” section within in the Readme.html file located at the root of the Neuron ESB installation folder.

Monitoring

In Neuron ESB, many of the monitoring and server management functions were exposed through REST API based interfaces. These interfaces can be used to extend and build custom monitoring and management solutions for Neuron ESB Deployments. The Neuron ESB REST interfaces are hosted by the Neuron ESB Discovery service (DiscoveryService.exe) which is installed with the Neuron ESB Server Runtime. The default port (51002) for the REST interfaces is configurable through the Neuron ESB Discovery service’s app.config. The default installation location of the Neuron ESB Discovery service is: “C:\Program Files (x86)\Neudesic\Neuron ESB v3”.

In previous versions, the Swagger documentation for the REST API could be browsed at the default URL: <http://localhost:51002/help/ui/index.html> . In Neuron ESB 3.6 the default URL has been changed to: <http://localhost:51002/help/index> .



Neuron ESB REST Documentation Web Site – The Neuron ESB REST interface documentation is hosted on the local Neuron ESB server.

The REST interfaces are self-documenting, providing detailed instructions on how to call each specific function and are separated into several categories. In Neuron ESB 3.6 a new category, “Configuration”,

was added to support its integration with the future platforms. The following categories of REST APIs are listed below:

- Runtime
- Deployment
- Endpoint Health
- Activity
- Configuration

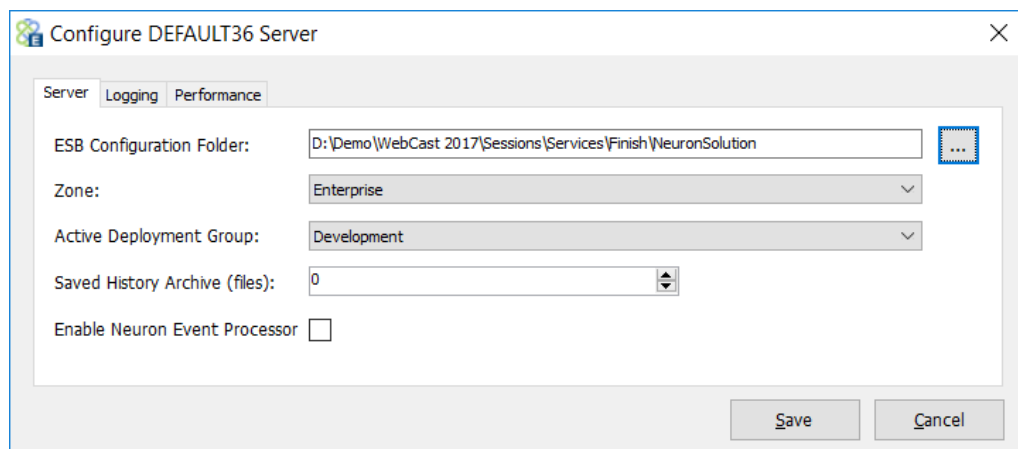
AppSettings

Neuron ESB is essentially a .NET application and as such uses its associated .config file to maintain runtime configuration settings. In previous versions, these settings were maintained in the appSettings section within the following two files

- ESBSERVICE.exe.config
- ESBHost.exe.config

These files are installed in the following default location: C:\Program Files\Neudesic\Neuron ESB v3\<Instance>.

Many of the settings are maintained and modified by using the Configure Server dialog launched from the Neuron ESB Explorer, while others can be set manually by editing the respective .config files.



Configure Server dialog – Accessible from either the Neuron ESB Explorer’s toolbar, or by navigating to Deployment->Manager->Servers within the Neuron ESB Explorer.

In the Neuron ESB 3.6 release, we’ve consolidated the appSetting entries into a dedicated .config file named “appSettings.config” located in the same default location as all the other .config files. The following .config files now contain a reference to this new file.

- ESBSERVICE.exe.config
- NeuronEndpointHost.exe.config

Additionally, several new key entries were added to facilitate integration with our future platforms. If users have added their own custom key/values to the existing Neuron ESB .config files, these will need to be migrated to the new appSettings.config file.

The table below lists all the appSetting key/values used in Neuron ESB 3.6

<i>Name</i>	<i>3.6</i>	<i>Default Value</i>	<i>UI</i>	<i>Description</i>
esbInstanceName		Name of Neuron ESB runtime Instance		This is the name of the Neuron ESB runtime instance set by the installer and visible within the Service Control Manager
esbEnvironment		Full path of solution folder	X	Set by the Configure Server dialog. Full path of solution folder
esbZone		Enterprise	X	Set by the Configure Server dialog. Should always be Enterprise.
esbDeploymentGroup		Active Deployment Group	X	Set by the Configure Server dialog. Active Deployment Group that the runtime should load on startup
overrideDotNetThreadPoolSize		False	X	Set by the Configure Server dialog. Performance tab. Whether or not to override .NET thread pool for Neuron ESB runtime.
dotNetMaxWorkerThreads		500	X	Set by the Configure Server dialog. Performance tab. Used only if Override is enabled.
dotNetMaxIOThreads		1000	X	Set by the Configure Server dialog. Performance tab. Used only if Override is enabled.
dotNetMinWorkerThreads		50	X	Set by the Configure Server dialog. Performance tab. Used only if Override is enabled.
dotNetMinIOThreads		100	X	Set by the Configure Server dialog. Performance tab. Used only if Override is enabled.
MaximumLogFileSize		104857600	X	Set by the Configure Server dialog. Logging tab. Default is 100 MB
MinimumDiskSpaceThreshold		10485760	X	Set by the Configure Server dialog. Logging tab. Default is 10 MB
LogFileSchedule		Daily	X	Set by the Configure Server dialog. Logging tab. Default is Daily. Can be set to

				Weekly or None. Determines when a new Log file will be created for an endpoint.
StartupTimeout		00:10:00		Maximum amount of time the Neuron ESB runtime is allowed to take to start up all internal services and endpoints. Timespan value
ConfigurationReloadDelay		00:01:00		The time between the detection and processing of Configuration change events. Timespan value
WSDiscoveryEnabled		true		Enables WS-Discovery on all Client Connectors, Neuron ESB Runtime as well as hosted Parties. When enabled, WS-Discovery can accept/respond to probe request, returning information and status of these endpoints.
WSDiscoveryBroadcast		true		Enables the Neuron ESB Runtime to broadcast WS-Discovery Hello/Goodbye messages. WSDiscoveryEnabled must also be set to true for broadcasting to be enabled.
WSDiscoveryAnnouncementInterval		00:00:10		Determines interval for WS-Discovery broadcasts. Timespan value.
WSDiscoveryApplicationScope		default		WS-Discovery scope used in announcements. Recommended that is should be the value of the esbInstanceName key.
DiscoverySwaggerPort		51003		Default Port used to host Swagger documents when Swagger is configured for a REST based Client Connector
DefaultHttpConnectionLimit		100		Used to set the value for the ServicePointManager.DefaultConnectionLimit property. Used for all HTTP service calls.
ClusterNetworkName		Empty String		This is set by the Neuron ESB runtime at startup. If Neuron ESB is installed on a Windows Failover Cluster as resource, this value will be the network name associated with the cluster resource. Otherwise it will be the machine name
MaxHeartbeatFailures		2		Max number of heartbeat failures against the Neuron ESB runtime master control service allowed before a Client Connector,

				Service Connector or Adapter Endpoint will shut down.
esbUseMachineEnvironmentVars		false		If set to true, allows users to use Windows Operating System Environment Variables within the Neuron ESB Explorer's Binding Expressions dialog.
XsltFilesLocation	X	Empty String		Used by the Transform-XSLT Process step and the Transform Message/XML Workflow Activities. Defines a global location to retrieve XSLT files, rather than using the Neuron ESB repository location. This is a Global override.
XsdFilesLocation	X	Empty String		Used by the Validate-Schema Process step and Validate XML Workflow Activity. Defines a global location to retrieve XSD files, rather than using the Neuron ESB repository location. This is a Global override.
LogFolderCleanupSchedule	X	10		Number of days of log files that Neuron should not delete/remove at startup. Any log file older than the value will be deleted. Must be greater than 1.
SavedHistoryCleanupSchedule	X	0		Number of history files to retain in the solution's History folder. Executes whenever a save is committed on configuration. A value of Zero means unlimited.
TempDirectory	X	Empty String		If a dedicated TEMP directory is not provided, the default TEMP directory of the service account will be used, consistent with 3.5 behavior.
DisableGlobalExceptionHandler	X	true		Enables/Disables the Neuron ESB Global Exception handler. When troubleshooting an issue, this should be set to false.
NeuronEndpointHostPort	X	51004		Communication port used by the Neuron Endpoint Host REST API. Commands between the Neuron ESB runtime and Endpoint Host are brokered across this port.
NeuronHybridFeaturesEnabled	X	false	X	Set by the Configure Server dialog. Server tab. This value represents the "Enable Neuron Event Processor" property. Can only be enabled if the Neuron ESB

				Instance has been registered within the Neuron Management Suite Portal.
NeuronHybridManagementSuiteHost	X	localhost		Location of the Neuron Management Suite Portal. Set by the portal during registration process
NeuronEventProcessorMachine	X	localhost		Location of the Neuron Event Processor. Set by the portal during registration process
NeuronEventProcessorPort	X	51005		Port configured for the Neuron Event Processor API. Set by the portal during registration process
NeuronServiceRateFlushInterval	X	5		Configuration specific to the Neuron Event Processor. Set by the portal during registration process
NeuronServiceRateCacheSize	X	10000		Configuration specific to the Neuron Event Processor. Set by the portal during registration process
NeuronEventConsumerLevel	X	1		Configuration specific to the Neuron Event Processor. Set by the portal during registration process
NeuronEventTimeToLive	X	1440		Configuration specific to the Neuron Event Processor. Set by the portal during registration process
NeuronEventPoolSizeInit	X	0		Configuration specific to the Neuron Event Processor. Set by the portal during registration process
NeuronEventPoolSizeMax	X	200		Configuration specific to the Neuron Event Processor. Set by the portal during registration process

Dependencies

Neuron ESB is entirely built on the Microsoft .NET Framework. This makes it very easy for .NET developers to learn and extend the product using common C# skills. With the release of Neuron ESB 3.6 we took the opportunity to upgrade our support for version 4.7.1 of the Microsoft .NET Framework. This allows developers to take advantage of the new features included in the framework when building any extensibility as part of their Neuron ESB solution such as adapters, workflow activities, process steps and OAuth providers. This support is also provided through our own C# editors within the Business Process and Workflow designers.

Additionally, we upgraded our support for a number of other dependent libraries, including the following:

- Log4Net from version 2.0.3 to 2.0.8
- RabbitMQ Client version 4.1.3 to 5.0.1
- Newtonsoft Json .NET version 8.0.2 to 11.0.1

Lastly, Neuron ESB 3.6 has been upgraded to support Erlang version 20.2 and Rabbit MQ version 3.7.1. Installation of RabbitMQ Server is optional and is not required to use Neuron ESB 3.6. If RabbitMQ Server is not installed, the Neuron ESB 3.6 installer will provide users the option of automatically downloading and installing Erlang and RabbitMQ Server. If a developer wants to use RabbitMQ based Topics, but the computer does not have Internet access, the following software packages can be downloaded and installed either before or after installing Neuron ESB 3.6.

- [Erlang](#)
- [RabbitMQ Server](#)

Temp Folder

One of the great things about Neuron ESB is its ability to auto detect changes and, based on those changes, re-configure itself. Part of that process involves dynamically compiling .NET assemblies from the changes detected. For instance, if a Business Process is modified, Neuron ESB will auto compile that Business Process and its respective C# code (if any) into a .NET assembly before it can be executed. This process also happens after the Neuron ESB runtime is initially started. Before the first execution of any Business Process, it is dynamically compiled. All other consecutive executions then run from the compiled assembly. Dynamic compilation only happens before the first execution of a Business Process or modified Business Process.

For this to happen, heavy use is made of the Windows TEMP directory of the service account assigned to the Neuron ESB runtime service (esbservice.exe) within the Service Control Manager. For instance, given a service account named “NeuronSvc”, the Windows TEMP directory location would be: “C:\Users\NeuronSvc\AppData\Local\Temp”. Due to our heavy use of this directory, Neuron ESB attempts to delete all the temporary files it generates during the startup of the Neuron ESB runtime service. Because Windows directories can only hold up to 65535 files, it’s important that the TEMP directory location be maintained and regularly cleared out.

However, outside processes (other than the Neuron ESB runtime) can negatively affect a running Neuron ESB solution by either placing locks on the temporary files Neuron ESB generates or, deleting those files. In other cases third party processes may also be using the same TEMP directory that Neuron ESB is configured to use (i.e. where a dedicated account is not used), generating files that exceed the maximum limit of the directory. Lastly, anti-virus programs are notorious for scanning the TEMP directories of user accounts as they are commonly exploited by viruses. Unfortunately, many tend to lock files as they scan.

To avoid these issues, Neuron ESB 3.6 can be configured to use any Windows directory as a TEMP directory. The location can be specified in a new appSettings key (i.e. TempDirectory) located within the appSettings.config file, the default location of which is: C:\Program Files\Neudesic\Neuron ESB v3\<Instance>. If a dedicated TEMP directory is not configured in the appSettings section, the default

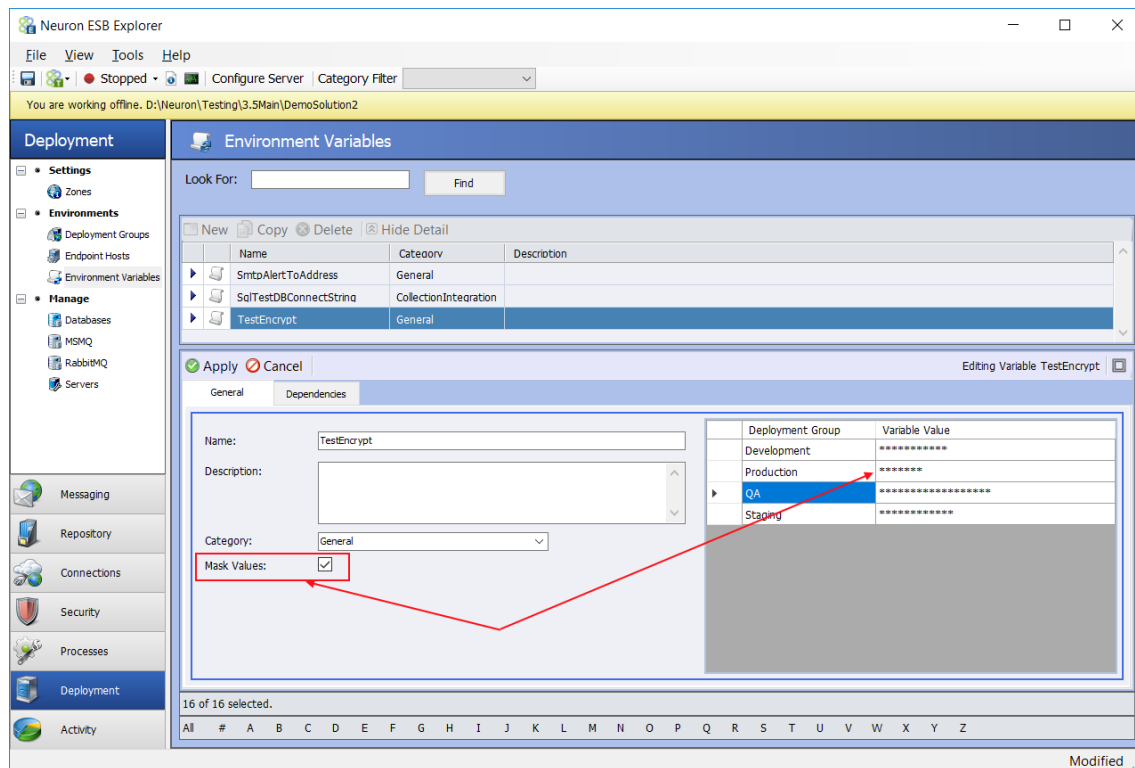
TEMP directory of the service account will be used as in previous versions of Neuron ESB. In all cases, the service account must be given full rights to the TEMP directory.

Using a dedicated TEMP directory can assist to eliminate the ability for third party processes to generate temporary files in the location that Neuron ESB uses. Also, Security protocols will usually allow the exclusion of an application specific folder from Anti-Virus scanning.

Environment Variables

Neuron ESB provides the ability to use Environment Variables for configuration of any adapter or service endpoint, database connection string or process step. By using the Bindings Expression editor, users can bind Environment Variables directly to the properties exposed by those entities. Many developers access the values of Environment Variables directly in Code Process steps within processes or workflow. An Environment Variable is defined at the Deployment Group level and can contain a unique value for each Deployment Group. On startup, the Neuron ESB service runtime replaces the Environment Variable with the value associated with the Deployment Group the service is configured for. To learn more about Environment Variables please review the [Using Environment Variables](#) section within the Neuron ESB Business Process documentation.

By default, all Environment Variable values are stored encrypted within the solution. However, those same values remain human readable in the Neuron ESB Explorer. In this release, we've introduced the ability to "mask" all the values of an Environment Variable. A check box, "Mask Values" has been added to the General Tab of the Environment Variable user interface and is disabled by default. If enabled, all the values (current and future) for that specific Environment Variable will no longer be visible. Each character will be masked by an asterisk ("*"). Once enabled, it cannot be disabled.



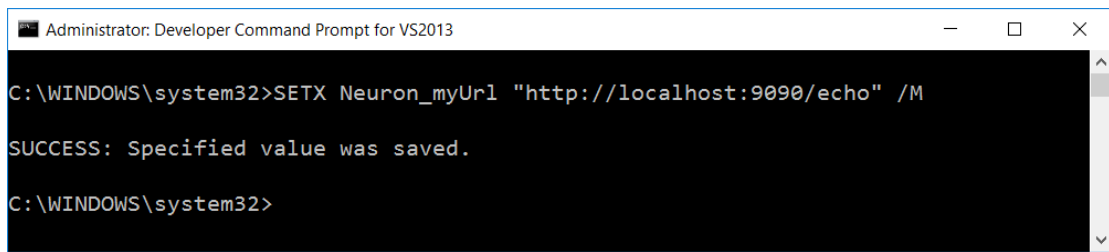
Neuron ESB Explorer – Masking Environment Variables – Once the “Mask Values” check box is checked, all values will be masked with an asterisk.

Windows OS level Environment Variables

One last thing was modified regarding the use of Windows OS level Environment Variables. In Neuron ESB 3.5, support was added for using Windows Operating System Environment Variables within the Neuron ESB Explorer’s Binding Expressions dialog. Previously, users could create Neuron ESB based Environment Variables to set most any property of most any Neuron ESB Entity, like Service Connectors, Client Connectors and Adapter Endpoints. The specific values of the Environment Variables are linked to Deployment Groups within the Neuron ESB Explorer. Hence the values used in one Deployment Group could be different than those used in another. The Neuron ESB runtime resolves these values dynamically at runtime on startup.

This new feature allowed users to enter the name of environment variables created at the machine’s Operating System level within the Binding Expressions dialog, rather than use Neuron ESB’s built in Environment Variables. There are a few requirements to use this:

1. Environment Variables had to be set at the System level. This could be done using the Windows Operating System’s UI tool, or using the Windows command line tool, “SETX” followed by the “/M” switch
2. Operating System level Environment Variables that are used within Neuron ESB must be preceded with “Neuron_”. For example, users could create an Environment Variable named “Neuron_myUrl” using the following command line:



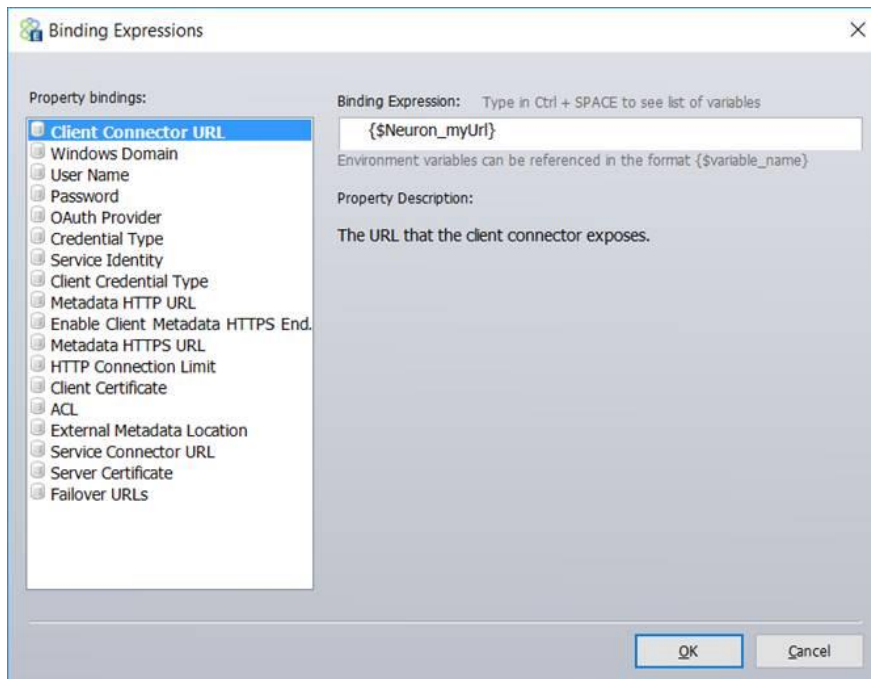
```
Administrator: Developer Command Prompt for VS2013

C:\WINDOWS\system32>SETX Neuron_myUrl "http://localhost:9090/echo" /M

SUCCESS: Specified value was saved.

C:\WINDOWS\system32>
```

3. Within the Neuron ESB Explorer's Binding Expression dialog, the Environment Variable must be enclosed in the Neuron ESB bracket notation as shown below:



4. Lastly, the following flag must be added the AppSettings group within the esbservice.exe.config file: `<add key="esbUseMachineEnvironmentVars" value="true"/>`

However, because these were at the system level, this required that the Environment Variables had to be created AFTER the process is started. This was problematic when hosting Neuron ESB within Docker containers. In Neuron ESB 3.6, this has been changed to retrieve values stored at the Machine level.

Additional Changes

Adapter Endpoints

MOD – Azure Service Bus Adapter – The Asynchronous Subscribe option has been removed

MOD – Azure Service Bus Adapter – The following dependent libraries have been updated:

WindowsAzure.ServiceBus from version 2.4.2.0 to 4.1.5

Microsoft.WindowsAzure.ConfigurationManager from version 2.0.3 to 323

MOD – Dropbox Adapter - The following dependent libraries have been updated:

Dropbox.Api from version 3.0.5 to 3.5.0

MOD – RabbitMQ Adapter - The following dependent libraries have been updated:

RabbitMQ.Client from version 4.1.3 to 5.0.1

MOD – FTP/FTPS Adapter - The following dependent libraries have been updated:

Xceed libraries updated from version 5.8 to 6.5

MOD – SFTP Adapter - The following dependent libraries have been updated:

Xceed libraries updated from version 5.8 to 6.5

NEW – Proxy Server Support – This has been added as a configurable set of properties to the following adapters:

- Microsoft Dynamics AX
- NetSuite
- Salesforce

NEW – Microsoft Dynamics AX – The Transfer Mode property has been added allowing the user to choose between buffered or streamed.

NEW – NetSuite – Support for 2018_1 SOAP Version of NetSuite API has been added

NEW – ODBC Adapter – The proper message samples and schemas can now be obtained using the Adapter Metadata Generation Wizard.

NEW – Salesforce Adapter – Support for versions 39 through 41 of the SOAP API has been added

MOD – SAP Adapter - The following dependent libraries have been updated:

ERPConnect45 from version 4.22.14.1 to 5.5.4.2

NEW – SharePoint Adapter – Site Collections and Lists can now be dynamically set at runtime using the following Neuron ESB Message Context Properties:

- sps.SiteCollection
- sps.ListName

NEW – Encrypted Storage of Sensitive Properties – The following property values will always be stored encrypted:

- Password
- ConnectionString
- Key
- Passphrase

Business Processes

MOD – Compression Process Step - The following dependent libraries have been updated:

Xceed libraries updated from version 5.8 to 6.5

NEW – Validate – JSON Process Step – This is a new Business Process Step that uses the Newtonsoft JSON Schema library for validation of JSON formatted messages. JSON schemas can be entered directly into the Process Step, or retrieved from the new JSON Schema repository (similar to the Validate–Schema Process Step). Like the Validate–Schema Process Step, it has a Valid and Invalid branch. This uses version 3.0.3 of the Newtonsoft.Json.Schema library. Additionally, like the Validate-Schema Process Step, the JSON Schema or the name of the Schema to retrieve from the JSON Schema repository can be dynamically configured at runtime by setting the following custom Neuron ESB Message properties:

- neuron.jsonSchema
- neuron.jsonSchemaName

MOD – Business Process Designer – The JavaScript Process Step is no longer supported and has been removed.

FIX – Business Process Test Client - When selecting the Load File option and an XML file is selected with a BOM character, the BOM would be loaded as well. This is now being removed.

FIX – Business Process Test Client – This only occurred when testing with a JSON body and the bodytype property is changed to something other than 'text/xml'. After the user opened the Test Client after the first test run, the text editor where the JSON message should be would be empty.

FIX – HTTP Client Utility - If a URL was not resolvable or could not be parsed, we would throw a null exception

MOD – JSON Process Step – Has been moved to the JSON folder within the Business Process Step Toolbox

MOD – Flat File Parser, Excel to Xml – These have been moved to the XML folder within the Business

MOD – Audit Process Step - If pools size equals 1, we reuse an instance of the audit proxy object, rather than retrieve from pool.

MOD - Service Endpoint Process Step – If pools size equals 1, we reuse an instance of the service proxy, rather than retrieve from pool.

NEW – Validate-Schema Process Step – A new property, “Referenced Schema Location” has been added that allow users to configure the Process Step to retrieve schemas from an external file share. If a schema contains any import or include directives, this is the location they will be searched for, if only a filename or a relative path is provided as the SchemaLocation in the schema. If this is left blank, the schema location will be resolved based on the Neuron ESB Instance that is hosting this solution as in

previous solutions. This property can be overridden globally by setting the appSetting key, “XsdFilesLocation”, to the location.

NEW – Transform XSLT Process Step – A new property, “Referenced Schema Location” has been added that allow users to configure the Process Step to retrieve XSLTs from an external file share. If this xslt contains any import directives, this is the location they will be searched for when only a filename or a relative path is provided as the SchemaLocation in the xslt. If this is left blank, the xslt location will be resolved based on the Neuron ESB Instance that is hosting this solution as in previous solutions. This property can be overridden globally by setting the appSetting key, “XsltFilesLocation”, to the location.

Workflow

MOD – Workflow Designer – The JavaScript Workflow Activity is no longer supported and has been removed.

MOD – Audit Workflow Activity – When used within the Catch Block of a Try/Catch and the Action property is set to Failure, the Audit step will automatically get the Exception object in the catch block rather than requiring it to be manually entered in the Exception property.

NEW – Validate XML Workflow Activity – A new property, “Referenced Schema Location” has been added that allow users to configure the Workflow Activity to retrieve schemas from an external file share. If a schema contains any import or include directives, this is the location they will be searched for, if only a filename or a relative path is provided as the SchemaLocation in the schema. If this is left blank, the schema location will be resolved based on the Neuron ESB Instance that is hosting this solution as in previous solutions. This property can be overridden globally by setting the appSetting key, “XsdFilesLocation”, to the location.

NEW – Transform Message/XML Workflow Activities – A new property, “Referenced Schema Location” has been added that allow users to configure the Workflow Activity to retrieve XSLTs from an external file share. If this xslt contains any import directives, this is the location they will be searched for when only a filename or a relative path is provided as the SchemaLocation in the xslt. If this is left blank, the xslt location will be resolved based on the Neuron ESB Instance that is hosting this solution as in previous solutions. This property can be overridden globally by setting the appSetting key, “XsltFilesLocation”, to the location.

Developer Tools

MOD - Import program – This was modified to work with Endpoint Hosts

FIX – Decryption Error on Load - If a decryption error occurred during a user attempt to load a solution in Neuron ESB explorer, a second attempt to load the same solution would cause the Neuron ESB Explorer to hang.

MOD – Neuron ESB Explorer making unnecessary network calls – The Neuron ESB Explorer was calling the Discovery service to retrieve status as well as the entire solution configuration twice a second. This is no longer a service call and it is no longer retrieving the solution configuration.

MOD – Certificates – When selecting and saving Certificates, users can no longer search by “find subject by name”. This could not guarantee uniqueness when the runtime must use the criteria to retrieve the correct criteria. Users should use “find by thumbprint”.

MOD – Bindings Expression Dialog – All binding properties are now displayed alpha sorted.

FIX – JSON Document Repository – If a document was modified or changed, the Apply button would not become enabled.

Neuron ESB Runtime

MOD – Environment Variables – Previously, when the Windows OS Environment Variables were enabled (by setting the esbUseMachineEnvironmentVars appSettings flag to true), Neuron would use user level Environment Variables. This has been changed to Machine level.

MOD – Global Exception Handling – This is now turned off by default using the DisableGlobalExceptionHandler appSettings flag. Users should turn this on when troubleshooting issues to ensure all exceptions are recorded.

Neuron 3.5.4.1297 Change History

Endpoint Health

FIX – REST API POST fails with Object reference not set to an instance of an object – If multiple instances are installed on a machine and a user runs a POST request against the following API, `/help/ui/index.html#!/Runtime/Runtime_Post`, for a specific instance, one may succeed while other requests for other instance names will throw an “Object reference not set to an instance of an object” error. This has been fixed.

Service Endpoints

FIX – REST Policy for Status Codes – If a user configured a Service Policy to trigger based on a specific HTTP Status Code for a REST Service Endpoint, the policy may not fire because HTTP Status Code was not being resolved. This has been corrected.

Workflow

FIX – Workflow Host startup fails with Key Not Found Error – A key not found error may occur when starting workflow hosts. We found this to be a random occurrence related to how our configuration is loaded at startup. This has been fixed.

Neuron 3.5.4.1290 Change History

Workflow

FIX – Environment Variables - The Environment variables were not being updated in the Workflow Runtime except upon restarting the runtime. This would cause an issue if a user started the Workflow Runtime then changed the environment variables as the new variables would not be sent to any new workflows. The workflows would instead get the old environment variables that the Workflow Runtime originally started with.

FIX – Adapter Endpoint Workflow Activity – If more than one Adapter Endpoint Workflow Activity is executed within the same workflow instance and each one is configured to use the same adapter endpoint, the following error could occur:

Exception: Exception Type: System.ObjectDisposedException

Exception Message: An ActivityContext can only be accessed within the scope of the function it was passed into.

Adapters

MOD – Dynamics XRM Adapter – Added support for South America, Canada, Australia, Japan, India, North America 2 and United Kingdom regions.

MOD – SharePoint Adapter - Added support for subfolders in document libraries.

Neuron 3.5.4.1285 Change History

Services

FIX - Rabbit MQ Topics with Service Policies – We were incorrectly evaluating the expires date on the request message brokered by Rabbit MQ Topics which would cause the policy to think all retries have been exhausted.

MOD – Message Duration – The total time for a Service Call to complete will now be written to the Neuron ESB log files when Info level logging is enabled. This has been added to Client Connectors, Service Connectors and the Service Endpoint Process Step.

FIX – Security Context – Attempting to access the System.ServiceModel.ServiceSecurityContext.Current.WindowsIdentity object within a Business Process attached to a Client Connector would generate a null exception as the object was previously disposed.

MOD – REST Service Policy - Previously, when a Service Policy was configured for a REST endpoint and the policy was configured to include specific HTTP Status Codes as thrown exceptions, the original request message would be logged, rather than the body of the Reply message regardless of whether or not a Reply message was received which may contain more specific endpoint error info. Moving forward, if a Reply message is received, that Reply body will be audited and forwarded rather than the original request message.

Adapters

FIX – Salesforce Adapter – Fixed issue introduced by the added support for calling Apex SOAP services. When NOT calling Apex SOAP services, if the LocalPath property was not null or empty, the adapter adjusted the URL as if it were an Apex SOAP service. When calling an Apex Service, customers can now set a custom property on the Neuron ESB Message, “sforce.apexService”, to true.

NEW – MQSeries Adapter - Support has been added for publishing and subscribing to Topics. There is also a new Durable Subscription property for supporting this feature. If set to True (default is false), the subscription will remain in MQSeries server across Neuron ESB restarts. This ensures that if messages

are published to a Topic when Neuron ESB is shut down, the adapter endpoint will get them when it starts back up again.

MOD – NetSuite Adapter – Added TLS 1.2 support.

NEW – Adapter Framework – We are now exposing a new event within the Adapter Framework. This allows customers building custom adapters to call the event within their custom code to send messages to the Neuron ESB Audit System. The examples are:

```
//Sends message to Neuron Audit system (i.e. Message History) with direction
RaiseAuditFromEndpoint(message, Internal.ESBHelper.Direction.Receive);

//Sends message to Neuron Failed Messages
RaiseReceiveFailedFromEndpoint(message, new Exception("something bad happened"));
```

FIX – SQL Service Broker Adapter – When in Publish mode the adapter may throw the following error:

System.Data.SqlClient.SqlException: The conversation handle "F61A6CC9-CE2E-E811-80FC-0050569677E4" is not found.

FIX – Dynamics AX Adapter - Some of the Dynamics AX services utilize message contracts that support any data format. Previously, the XML samples generated for these services did not reflect that <any> data format can be utilized. This have been fixed to more properly reflect the use of <any>.

FIX – Metadata Generation - When using the metadata generation wizard, if a user unchecked the option to generate XML samples, the wizard would generate them anyways. This has been fixed.

FIX – Metadata Generation - When using the metadata generation to generate XML samples for a schema containing the <any> element, the previous behavior of the wizard was to use a random schema as the format for the XML output. This has been updated so that an <any> element will be used instead.

MOD – FTP Adapter – When attempting to connect to port 990 using Implicit SSL (TLS 1.2) the adapter connect method would time out.

MOD – FTP/SFTP Adapters - Added "Detect Encoding" property, true by default. If false, the adapter will not attempt to retrieve an encoder and encode the contents of the file. It will be passed through as byte array.

MOD – FTP/SFTP Adapters - has been updated to use the Xceed 6.5 libraries.

MOD – SFTP Adapter – Connecting to some servers may through the following exception. This was corrected by updating to the Xceed 6.5 libraries. We are also now negotiating the Diffie-Hellman key exchange group size to 2048. Previously we were using 4096 as the maximum size.

Inner Exception Type: System.ArithmeticException

Inner Exception Message: Multiplication overflow.

Business Processes

MOD – Compression Process Step – has been updated to use the Xceed 6.5 libraries. Cleo's Harmony products (VL trader & VL Proxy server) version 3.6.1.0.

FIX – Business Process Designer - When testing a Business Process, if a user selected the Load File option within the Edit Test Message dialog and an XML file is selected with a BOM character, the BOM character would be loaded into the editor along with the data. This is now being removed.

FIX – Input String Error when in Debug logging mode – The following exception could be generated when using the Service Endpoint Process Step at runtime within a Business Process.

Input string was not in a correct format.

Exception Trace: at Neuron.Esb.Internal.PipelineRuntimeHelper.ApplyPipelines

Tools

FIX - Test Client - if a reply had the same message id as the original request message, a message box would be displayed indicating that it could not insert the response message into message history tab. This could occur if the Party being tested was attached to a Business Process using the OnPublish event and the last executed process step is a Cancel step.

FIX – Exporting Processes – The ExportConfig.EXE Tool would generate an error at the processes export step when exporting the whole solution. The tool would report that a process could not be found. This was due to the tool expecting the name of the process rather than the Id GUID of the process in our internal collection.

FIX – Exporting Processes – The ExportConfig.EXE Tool would generate an error on a few Neuron ESB objects (examples of Neuron ESB objects are: parties, topics, endpoints, etc.) if explicitly defining the objects to export using their name. The list of objects that were fixed are: Databases, Processes, Keys, and Adapter and Service Endpoints. The tool now looks up the correct dictionary key to pass to the export functions so that the Neuron ESB objects in question can be explicitly defined for export by name.

Workflow

FIX – Runtime Correlation Error - Under certain conditions an ESB message could be enqueued for an aborted workflow which might cause errors to occur. The error would be similar to "ProcessMessage - The ResumeCorrelatedWorkflowCommand command is NULL AND the The workflow Host Topic 'MyTopic' does NOT match the message topic 'MessageTopic'. The message will not be processed". This error would also not appear in the Neuron Log file for the respected endpoint.

Messaging

MOD – TCP and Named Pipes – Buffer size is now dynamically set to size of expected message size.

Neuron 3.5.4.1242 Change History

Service Endpoints

FIX – “Unexpected end of file” Exception - Sending an empty message body to REST endpoint would generate “unexpected end of file”. This would typically occur when sending a GET request with an empty body to a REST service. This was due to the addition of the WCF message tracing option which was added in build 1178.

Workflow

FIX – Workflow Tracking - If the workflow was in the "Waiting" state for more than the workflow command timeout, the command would execute again starting up another workflow instance. Now if we see that the command has timed out but is still active, we extend the timeout by 5 minutes. We also log an informational message informing users that the workflow it is still executing and that they should make sure that the desired behavior is that it stays in the executing state.

Neuron 3.5.4.1238 Change History

Endpoint Health

FIX – Endpoints not reflecting state of underlying topic/party – if the Party associated with a Client Connector, Service Connector or Adapter Endpoint went offline due to a problem with a Topic or, it became disabled, the endpoints would not report the error in the Endpoint Health monitor nor would it change its state from Started to Failed. This has been corrected.

Business Processes

MOD – Table Query, XML Query and Store Process Steps – The Command property of these steps can now be dynamically configured at runtime using the ‘sql.Command’ custom message property.

MOD – Disable Property of Process Steps and Environment Variables – Environment Variables can now be used to toggle the Disable property of Process Steps

Messaging

MOD – Automatic Reconnect of Parties – Previously, if there was a disruption that prevented a Party from communicating with the main Neuron Server or TCP/Named Pipes based Topics, the underlying Party would attempt to reconnect indefinitely. If left in that state for several hours, it could cause spikes in CPU utilization. This has been changed so that the Party will only attempt to reconnect for a period of 10 minutes. After that period, the party will shut down and will require manual reconnection after the disruption has been corrected.

Service Endpoints

FIX – Swagger Endpoint not reachable on remote machine – if a user attempts to browse the swagger endpoint exposed on a client connector from a remote machine it would fail. This was because of a configuration setting in the DiscoveryService.exe.config. Specifically, the appSettings key:

```
<add key="SwaggerSelfHostingUrl" value="http://localhost:51003" />
```

Has been changed to this:

```
<add key="SwaggerSelfHostingUrl" value="http://*:51003" />
```

Neuron 3.5.4.1232 Change History

Installation

FIX – Broken Rabbit MQ and Erlang download links – These installers have been moved and the URLs have been corrected in the installer and readme file.

Neuron ESB Explorer

FIX – OAUTH Test in Neuron ESB Explorer – If there were properties missing, the test would throw an unhandled exception.

FIX – Unit Testing Sample throws Error - The TestingProcessesWithVisualStudio VS solution would throw an error when compiling it. This was due to the MSMQ process step being refactored in a previous version.

Workflow

FIX – Workflow Persistence and refreshing of Environment Variables – This affects the Persist and Reload workflow activity. The EnvironmentVariableService was returning the environment variables from the workflow context object, which did not contain the newest environment variables.

Adapters

MOD - Salesforce adapter - Added support for calling custom SOAP services exposed via APEX code. The adapter will set the body of the ESB message in the same way it currently does with standard SOAP services, but instead will set the local path to the custom service, starting at /services/.

FIX – Dynamics 2013 CRM Adapter - Fixed an issue where a lookup returned an Entity Reference. Previously the value returned by the lookup in the adapter would be an empty string. Now the lookup will return the GUID of the referenced entity.

Business Processes

FIX – Audit, Service Endpoint, Adapter Endpoint, Execute Process Step – There were several lifetime issues that were addressed in these steps when pooling was enabled.

Neuron 3.5.4.1220 Change History

Samples

FIX – Excel to XML Sample – Running this would generate a key not found exception. This was due to an incorrect property being accessed at runtime. This has been corrected

Runtime

FIX – SetProperty() and Existing Key Error – Under error conditions, calling this method could generate an error indicating that the key already exists.

MOD – Clearing TEMP directory – On start up, if Neuron could not clear the temp directory, this was logged as an Error. This is now logged as a warning. At runtime (every hour), Neuron used to attempt to

clear out files in the temp directory. It no longer performs an hourly cleanup. Neuron now only attempts to delete temp files if it can obtain an exclusive lock on them.

Workflow

FIX – Workflow using incorrect Workflow Definition on Resume – If a Workflow was unloaded into the database and its definition was later changed, followed by an attempt to resume the unloaded workflow, the Workflow would attempt to resume with the new definition rather than the one it started with.

Database

FIX – Purge and Archive SQL Agent Job - This could cause deadlocking to occur when there are many workflows running and the SQL Agent Job to purge deleted workflows is executed. This is because the table could still be locked by the stored procedure despite the “NOLOCK” table hint being used. This has been resolved. The Neuron ESB Database must be updated for the fix to be applied.

Adapters

MOD – Microsoft Exchange Adapter – Added a Reuse Connection property. By default, the adapter reuses the connection to Microsoft Exchange on every poll. If set to true, a new connection will be used for each poll.

FIX - Dynamics CRM XRM Adapter - Would fail when trying to perform an update or upsert on a custom entity. Also resolved an issue when a lookup returned multiple rows and the ignoreFailedLookup = "true". Additionally, added support for a proxy server to the adapter.

FIX – Metadata Generation and Object Reference Error - In the Metadata Generation Wizard, using a credential from the repository for the adapter properties grid would result in an Object Reference error when testing a connection.

FIX – TCP Socket Adapter - In the Server Socket Adapter, if there were connected Client Socket Adapters and the connection dropped then reconnected, it could be the case where the Server Socket Adapter would not detect that the Client Socket Adapter had reconnected. This was due to a timing/threading issue on a collection that contained the list of connected Client Socket Adapters.

FIX - Dynamics CRM XRM Adapter - Lookups that referenced previous lookup results in the search criteria would fail. Also, when there are multiple SaveCrmRecords in the same XML request, if one SaveCrmCommand referenced the result of a previous SaveCrmCommand as a lookup, it would fail.

NEW – Dynamics AX Adapter – Added support for new Identity types, specifically for DNS, SPN and UPN.

NEW – SharePoint Subscription Adapter - Added support for SharePoint Online. Also added support for configuring a proxy server.

FIX – FTP Adapter - SSL connect could fail in Explicit SSL mode.

FIX – Salesforce Adapter – When using the SOAP interface, the following exception could occur:

UNSUPPORTED_CLIENT: TLS 1.0 has been disabled in this organization. Please use TLS 1.1 or higher when connecting to Salesforce using https.

Business Processes

NEW – ODBC Process Step - Added "Throw Error on Exception" property which can be used when the Semantic is set to Request.

Logging

NEW – WCF Tracing - When enabling WCF Tracing, the messages that are sent and received by endpoints were not traced. This has been added.

Neuron 3.5.4.1162 Change History

Security

FIX – Active Directory OAUTH Provider - When using the client credentials flow the following error will be generated:

Value cannot be null.

Parameter name: authorizeU

Workflow

FIX – Correlated Workflow – If a workflow is in an unloaded state due to Receive Message Workflow Activity correlated wait for an incoming event and IF the Receive Message Workflow Activity's name was changed to anything besides "ReceiveMessage", the workflow would not resume if the incoming event is received.

FIX – Workflow Tracking – Republishing a message from workflow tracking would generate the following error:

Failed to resubmit message due to the following error:

Unable to cast object of type 'Neuron.Explorer.Models.MessageItem' to type 'Neuron.Esb.ESBMessage'.

Adapters

FIX – MQSeries Adapter – A MQRC_OPTION_NOT_VALID_FOR_TYPE error can occur when in request/reply mode if the Request Queue is defined as a local Queue and the Response Queue is defined as a remote Queue.

FIX - Dynamics CRM Web API adapter – An incorrect URL was used as the Access Token Url for the Azure Active Directory.

NEW - Dynamics CRM Web API adapter – Proxy support has been added.

MOD – FTP/SFTP Adapters – File Mask – The current behavior has been changed. Now each adapter supports multiple file masks, separated by semicolons (i.e. ";"). Additionally, wildcard characters (i.e.

“*”, “?”, “#”) can be used for both filtering the name and extension of the file to monitor for. For example, the following would be valid file masks to use:

`*.txt;*.xlsx;1*W#.p?f;*.p?g`

The File Mask feature allow you to match each character in the file name against a specific character, a wildcard character, a character list, or a character range. The following table shows the characters allowed in the File Mask and what they match.

<u>Characters in File Mask</u>	<u>Matches in string</u>
?	Any single character
*	Zero or more characters
#	Any single digit (0–9)
[charlist]	Any single character in charlist
[!charlist]	Any single character not in charlist

MOD – SFTP Adapters – Now supports setting the following properties dynamically for subscribe side adapter endpoints:

- sftp.Server
- sftp.Port
- sftp.Filename
- sftp.Folder
- sftp.UserName
- sftp.Password

These properties can be set via the Set Property Process Step or through a Code Editor using the SetProperty() method.

MOD – FTP Adapters – Now supports setting the following properties dynamically for subscribe side adapter endpoints:

- ftp.Server
- ftp.Port
- ftp.SecureConnection
- ftp.SecureDataChannel
- ftp.Certificate
- ftp.PassiveFTP
- ftp.Filename

These properties can be set via the Set Property Process Step or through a Code Editor using the SetProperty() method. All values are text based. PassiveFTP and SecureDataChannel are “true” or “false” values. SecureDataChannel is either “0” for none, “1” for Explicit SSL or “2” for Implicit SSL.

FIX – MQSeries Adapter – Poison Message Handling – If monitoring NON local queues (i.e. Alias or Remote, etc), the adapter may report the following error:

WebSphere MQ Adapter - WebSphere MQ error ' MQRC_SELECTOR_NOT_FOR_TYPE : Reason Code: 2068', occurred while trying to send message to Queue

The adapter now only implements poison message handling for local queues and alias queues that resolve to local queues.

FIX – Rabbit MQ Adapter – Dynamically setting the following message properties would not override overwrite the design time properties: CorrelationId, ContentType, Exchange, Queue, RoutingKey, Expiration. This has been corrected.

NEW – SQL Service Broker Adapter – Added the ‘ConversationHandle’ custom message property. This will contain the Dialog/Conversation handle at the time that the message is retrieved and published.

FIX – MQSeries Adapter - When receiving RFH2 messages, the size of the RFH2 header would be appended to the Neuron ESB Message body as null bytes.

Runtime

FIX - WS-Discovery - Non-US Cultures could generate the following warning:

"String was not recognized as a valid DateTime"

Neuron 3.5.4.1136 Change History

Service Endpoints

FIX – CORS Anonymous support – This was accidentally broken when adding NTLM authentication support for CORS Swagger access. This is now working for all scenarios.

Adapters

MOD – FTP/SFTP/FTPS – Xceed libraries has been updated to 6.3 to support TLS 1.2. TLS version is auto negotiated with the server.

MOD – Socket Adapters - Now the client socket adapter will attempt to reconnect on any socket error.

FIX – SQL Service Broker Adapter – When in Publish Mode, the adapter was not sending END Conversation messages back to the broker after every successful retrieval of information.

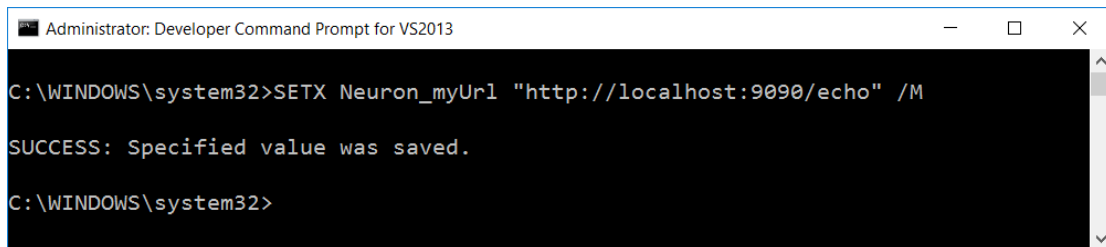
Deployment

NEW – Environment Variables at OS Level – Support has been added for using Windows Operating System Environment Variables within the Neuron ESB Explorer’s Binding Expressions dialog. Previously, users could create Neuron ESB based Environment Variables to set most any property of most any

Neuron ESB Entity, like Service Connectors, Client Connectors and Adapter Endpoints. The specific values of the Environment Variables are linked to Deployment Groups within the Neuron ESB Explorer. Hence the values used in one Deployment Group could be different than those used in another. The Neuron ESB runtime resolves these values dynamically at runtime on startup.

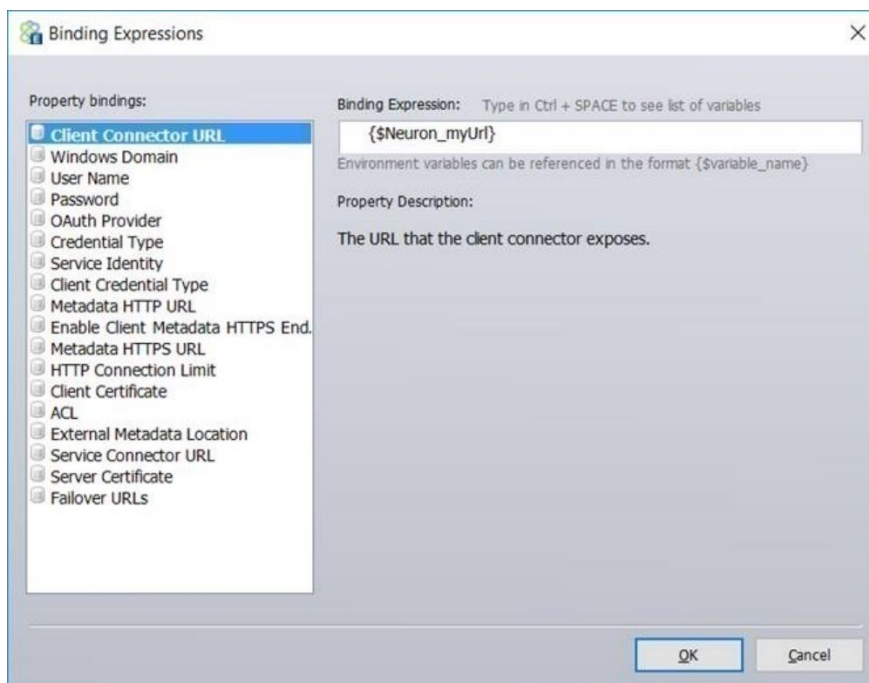
This new feature allows users to enter the name of environment variables created at the machine's Operating System level within the Binding Expressions dialog, rather than use Neuron ESB's built in Environment Variables. There are a few requirements to use this:

5. Environment Variables must be set at the System level. This can be done using the Windows Operating System's UI tool, or using the Windows command line tool, "SETX" followed by the "/M" switch
6. Operating System level Environment Variables that are used within Neuron ESB must be preceded with "Neuron_". For example, users could create an Environment Variable named "Neuron_myUrl" using the following command line:



```
Administrator: Developer Command Prompt for VS2013
C:\WINDOWS\system32>SETX Neuron_myUrl "http://localhost:9090/echo" /M
SUCCESS: Specified value was saved.
C:\WINDOWS\system32>
```

7. Within the Neuron ESB Explorer's Binding Expression dialog, the Environment Variable must be enclosed in the Neuron ESB bracket notation as shown below:



8. Lastly, the following flag must be added the AppSettings group within the esbservice.exe.config file: `<add key="esbUseMachineEnvironmentVars" value="true"/>`

Neuron 3.5.4.1130 Change History

Workflow

FIX – Correlated Workflows Not Restarting – The following error could occur when workflows with a publish and receive workflow activity would not continue execution after the workflow transitioned to an "Unloaded" state. This was because the ResumeCorrelatedWorkflowCommand would error out and not resume the workflow at the workflow activity. This required a database stored procedure fix that requires users to update the Neuron ESB database:

"**System.Data.SqlClient.SqlException (0x80131904): The COMMIT TRANSACTION request has no corresponding BEGIN TRANSACTION.**"

FIX – Workflows not restarting when ESBHost is restarted – Under some circumstances, if workflows are in the unloaded state when the ESBHost shuts down, they may not restart once the ESBHost is restarted.

FIX – Correlated Workflows/message remains in Pending State - Addresses the issue when sending an initial message to a correlated type workflow, the workflow engine would not pick up the message. This resulted in the message being stuck in the Pending Messages queue (Messages tab in workflow tracking in Neuron Explorer) and caused the workflow to never execute.

FIX – Correlated Workflows/Delay Activity - When a correlated workflow uses a persistent delay activity (delay longer than a minute so that it does persist), the workflow would not resume correctly if the service is restarted and the workflow was in the unloaded state. When the ESBHost started back up, the workflows would transition to an aborted state due to this issue instead of resuming from the persist point. This would most likely affect any correlated type workflow (as opposed to a Normal type with receive message activities) with persist activities.

FIX – Correlated Workflows/Not Restarting - There was an issue with the ESBHost.exe where we were not correctly loading the correlation sets on starting the ESBHost. This would cause workflows in the unloaded state to not start back up upon restarting the esbhost.exe. This applies to Normal type workflows with publish and receive message activities that effectively make it a correlated workflow.

Adapters

MOD – Salesforce Adapter – The adapter has been updated support versions 35-39 of Salesforce.

Installation

FIX - RegisterWmiEvent.ps1 – The powershell script did not attempt to detect the drive that Neuron ESB was running from. It was hard coded for the C:\ drive, which caused the script to fail. The script so now it detects the drive where the needed exe is located.

Messaging

FIX – CreateReplyMessage – When the CreateReplyMessage() was called, it did not correctly copy the existing LocalPath property value to the new Neuron ESB Reply Message.

Business Processes

FIX – Service Endpoint Process Step – If the service endpoint the process step calls throws exceptions, it could prevent the service proxy from being placed back into the pool. This could eventually result in warning messages being written to the log stating that the maximum number of instances have been reached. This has been corrected.

Neuron 3.5.4.1119 Change History

Service Endpoints

MOD – Swagger Support – Adding Authentication headers for Swagger to enable security models.

MOD – Credential Type – The Credential Type property if the Client Connector can now be configured using Environment Variables. The values are NOT case sensitive.

Business Processes

FIX – Retry Process Step – Setting the Retry to 1 would effectively abort any retries. This has been corrected.

Configuration

FIX – Database Environment Variables – If using an Environment Variable for the Neuron ESB Database Connection String, on startup the following error may be reported: “The environment variable <name> is not defined for the deployment group”.

Adapters

MOD – NetSuite Adapter – Now supports versions 2016.2, 2017.1, 2017.2

MOD – Socket Adapter - Connection/socket failure handling has been greatly improved in both the adapters. The Client Socket Adapter will now attempt to reconnect to the server socket on recoverable socket exceptions.

Error/Warning logging has been greatly improved to allow the user to easily identify any socket problems that may occur.

Solicit Response mode has been disabled in both adapters when the adapter property "Message Body Only?" is true. This was due to the fact that there needs to be some metadata added to the TCP message in order to facilitate correlation between requests received from the bus and the reply received back from the TCP endpoint. Adding metadata would defeat the purpose of being able to send the body only. This does not affect Request Reply mode or the non-query modes (Publish and Subscribe modes) when "Message Body Only?" is set to "True". **If either Socket Adapter attempts to start up with "Message Body Only?" set to "True" when in “Solicit Response” mode, an error will be logged in the Neuron Event Log and the adapter will not start.**

A bug was fixed where CPU usage would spike when a client socket unexpectedly disconnected from the Server Socket Adapter.

Neuron 3.5.4.1110 Change History

Monitoring

FIX – REST API – If WMI on the machine was corrupted, the Endpoint Health REST API would not be able to return results. A COM Exception could be returned because of the WMI corruption. The REST runtime controller has been modified to handle WMI corruption and resolve the underlying network name regardless.

Adapters

MOD – Sharepoint 2007 Plugin – The file has been renamed from “NeuronPublishingSP2007.wsp” to “NeuronPublishing.wsp”.

MOD – SharePoint Plugin – The current 2013 plugin has been modified to support SharePoint 2013 and SharePoint 2016. The plugin folder has been renamed from “SharePoint 2013 Neuron Publication” to “SharePoint 2013-2016 Neuron Publication”. The plugin file has been renamed to “NeuronPublishing.wsp”.

Deployment

FIX – ExportConfig Command Line - if an attempt was made to export a Business Process, the following error would be reported:

*ExportConfig Error: 1 : The process named "Order Process" was not found.
ERROR: The process named "Order Process" was not found.*

FIX – ImportConfig Command Line - Fixed issue where if a Business Process was getting overwritten due to an import, all text messages would list the GUID ID as the name of the Business Process, rather than the friendly name.

FIX – ImportConfig Command Line - Fixed issue where any import would always import Administrator roles, regardless of whether or not they were selected in the import.

Service Endpoints

MOD – Client Connector reports “An operation was attempted on a nonexistent network connection” – This may be reported as an error through the global exception handler. This would be reported due to an existing bug with WCF and rarely occurs. Although the error is reported, it did not affect the service response. However, changes have been made to report this as an Informational message instead of an error message

Documentation

NEW – Neuron.CHM – the Neuron ESB Documentation (Neuron.CHM file) has been updated with new Business Process Documentation.

Neuron 3.5.4.1100 Change History

Neuron Explorer

MOD – OAUTH Providers – The Access Token and Client Secret properties were not appropriately masked, making them visible at design time.

Adapters

FIX – MQSeries Adapter - If connection is broken or host not available, Neuron will now try to detect and recreate rather than throw an exception. In addition, on error could throw a null exception preventing the connection to the MQ Manager from being cleaned up. In some cases adapters that reported errors or warnings would throw a null exception if executed using the execute adapter process step only if an Exception object was not being passed in.

FIX - .NET Debugging – Adapter Execute process would throw an object exception when launching the .NET debugger (in contrast to just running a test) in the Process Designer.

MOD - FTP Adapter - Added new CertificateValidation property so that users could turn off Certificate Verification. When set to false, the Verification flag will be set to none and, in the receive event, the action will be set to Accept. If true (default) the previous verification process will be used.

NEW – MQSeries Adapter – The adapter now supports poison message handling when publishing messages from an MQSeries Queue to Neuron ESB. This is only in effect if the Transactional property is set to true. The following poison message handling is implemented:

1. The Backout count of the MQSeries Queue is checked, and if it is nonzero, it is compared to the Backout threshold that is defined on the message queue.
2. If the Backout count is less than the Backout threshold, the message is sent to Neuron ESB to be processed. If, for some reason, the transaction is not committed, the message is rolled back, and the MQ Manager increments the message Backout count. Additionally, the cycle continues until the message is committed successfully or until the Backout count becomes greater than the Backout threshold.
3. If the Backout count is greater than the Backout threshold, the message is written to the queue that is specified in the BOQNAME parameter (the “Backout requeue queue” property), of the queue and then the message is committed. A warning (similar to below) will be written in the Neuron ESB log files and the Neuron ESB event log:

Event Info: Adapter Endpoint 'MQSeriesSolicitResponse' reporting information.

There has been '13' attempts to read the Message with the ID of 'AMQ MWASZNICKY05^CQY!_i_' from the 'TEST' queue. This exceeds the Backout Threshold defined on the queue. The message will be moved to the 'POISONQ' backout queue.

4. If there is no Backout queue defined for the queue, then Neuron ESB will attempt to use the dead letter queue defined for the MQ Manager. If that is not defined, then Neuron ESB will log

an error to the Neuron ESB event log and the Neuron ESB log files similar to the one below. The message will continue to Neuron ESB for processing....and eventually cycle back into the queue if the transaction is rolled back:

Event Info: Adapter Endpoint 'MQSeriesSolicitResponse' reporting information.

There has been '4' attempts to read the Message with the ID of 'AMQ MWASZNICKY05^CQY!_d_' from the 'TEST' queue. This exceeds the Backout Threshold defined on the queue. However, there is neither a Backout queue defined for the queue or a Dead Letter Queue defined for the Queue Manager so the message cannot be forwarded and must be manually cleared from the source queue.

Installer

MOD - Modified the installer to accept a blank license key. Validation still occurs if there is anything in the license key field. This allows users to install Neuron ESB using the command line without the need for a license key as shown in the example below:

```
msiexec /i NeuronESB_v3_x64_Release.msi INSTALLFOLDER="C:\Program Files\Neudesic\Neuron ESB v3\" /qb /lvx* install.log
```

Workflow

FIX – The new “GetCurrentWorkflowActivityId” procedure was not assigned the appropriate permissions.

Neuron 3.5.4.1091 Change History

Workflow Tracking

FIX – Messages return null – The Messages Tab of Workflow Tracking could return a Null Exception when clicked.

Endpoint Health

FIX – Rabbit MQ - If a user stops a Rabbit MQ based Topic from endpoint health, the status would always reset back to "Started".

FIX – Sequence not found Exception – This could be reported in the Neuron Log as a Warning when an Endpoint attempts to start after a change has been recorded.

Workflow

FIX – Semaphore Timeout – There were several issues that could cause this. The symptom of which is after a certain amount of time workflows stop being processed and the following will continuously be reported in the Neuron ESB log files:

semaphore timed out...

The conditions, which caused this, were:

- If Timeout shape is used in the workflow, and Workflow completes before timeout completes, it did not release the thread which causes thread exhaustion and the workflow gets halted.
- If using multiple receive shapes in a workflow, when a correlated message is received, the workflow does not pick up the correct receive shape to resume, causing a corrupted state, which prevented the thread from being released
- If the executing workflow goes into a corrupted state, it would fail to release the processing thread.

FIX – Memory Leak with Correlated Workflows – Under certain conditions correlated workflow could cause memory usage to climb over time, preventing the runtime from releasing resources every time a workflow completes or aborts.

FIX – Request/Reply Workflows – When submitting Requests to a Neuron environment that contains more than one Neuron server, if the processor for the request is a Request/Reply type of Workflow endpoint, the user may experience random timeouts while receiving the responses. This was due to the lack of affinity when the workflow hosts query for workflow records to execute. This required a modification to the underlying stored procedure and requires users to update the Neuron ESB database using the Neuron ESB Explorer or provided scripts.

FIX – Workflow Instances Fail to Start – The Max Concurrent Workflow counter could be incremented to its maximum limit if a SQL Server failure occurs, preventing new workflow instances from starting once SQL Server connectivity is re established.

FIX –Workflow Instances Fail to Start – An orphaned record within the Neuron ESB database could occur when a Workflow instance aborts due to an exception, or when a Correlated Workflow instance completes. This will cause the Workflow runtime to believe that the Workflow instance is still running. The “Max Concurrent Workflow” property of the Workflow Endpoint controls how many concurrent instances of the Workflow the runtime will allow. The orphaned record will appear as a running instance to the Workflow runtime. If orphaned records continue to be generated, the Max Concurrent Workflow limit will be reached; no new Workflow instances will be created and messages will start accumulating in the database queue. In the Messages Tab of Workflow Tracking, users will see the message count grow. The Neuron ESB log file for the Workflow Endpoint will record entries similar to those below:

2017-05-11 23:16:57.198-07:00 [Workflow Endpoint 1 Runner] DEBUG - The semaphore timed out

2017-05-11 23:16:57.198-07:00 [Workflow Endpoint 1 Runner] DEBUG - Waiting on the semaphore for capacity to execute a command.

This condition could take several days to become visible.

FIX – Service Endpoint Activity - If a SOAP call was made with the “Enable HTTP Headers” property enabled on the Service Connector, the following warning would be written to the Neuron log file:

"captureHttpResponseProperties arguments: Method = null. This could happen if the old text properties are being used to set the Method on the ESB Message object. Please use ESBMessage.Http.Method instead."

The warning is now only raised if the call is to a REST based service and the Method is not set.

FIX – Publish Request Message – Would throw an exception if Synchronous = false. Also changed default value for Synchronous property = true.

Neuron Runtime

FIX – Failure to start on misconfigured Windows Failover Servers – Neuron ESB would throw the following errors and fail to start if the server that Neuron ESB was installed on has Windows Failover Server installed on it and it has yet to be configured or is misconfigured in some way:

System.Exception: Cluster Network Name: Neuron ESB Service is unable to determine if Failover cluster is installed. -2147023143: There are no more endpoints available from the endpoint mapper.

This would be followed by the following error:

*The configuration service failed to startup. Value cannot be null.
Parameter name: machine*

Installation

FIX - RegisterWmiEvent.ps1 – The powershell script did not check for the OS architecture (32-bit vs. 64-bit) which meant that the .Net Framework's InstallUtil.exe that ran was always the 32-bit version. This would cause the InstallUtil.exe that is ran from the powershell script to fail finding the path to the .mof file used by InstallUtil.exe in some 64-bit systems. The error will be shown in the console/powershell window that the script is ran in, giving a System.IO.DirectoryNotFoundException, even though the directory and file exist that InstallUtil.exe is looking for (which can be verified by navigating to the displayed path in Windows Explorer). The displayed error would be similar to the following:

System.IO.DirectoryNotFoundException: Could not find a part of the path 'C:\Windows\System32\wbem\Framework\root\Neudesic_ESB_v0\Neuron.Esb_SN_ed764b6eccb2f6cc_Version_3.0.0.0.mof'

Neuron Configuration

FIX – Encoding/Decoding non-ASCII Character Sets– Neuron ESB encrypts certain content (i.e. passwords, connection strings, etc.) for storage and decrypts this content at runtime using an ASCII encoder. For example, if an extended character (i.e. "Ø") is used in the name of a database within the Neuron ESB Explorer, it would end up stored and converted to "?". To resolve the issue Neuron ESB now uses an ISO-8859-1 encoder.

Service Endpoints

MOD – Message Security Exception – When using Windows:Message security on a service connector, the underlying Kerberos ticket may timeout after 10 minutes generating the following exceptions:

Exception sending request message

*Neuron.Esb.Channels.ChannelSendException: Server error. Exception Type:
System.ServiceModel.Security.SessionKeyExpiredException*

Exception Message: Cannot renew the security session key.

System.ServiceModel.Security.MessageSecurityException

Exception Message: An unsecured or incorrectly secured fault was received from the other party

Inner Exception Type: System.ServiceModel.FaultException

Inner Exception Message: The security context token is expired or is not valid. The message was not processed.

If either the SessionKeyExpiredException or FaultException occurs, the underlying proxy will be disposed and the next call will renew the Kerberos ticket. If a MessageSecurityException is generated, Neuron ESB will automatically attempt to retry the service call after recreating the underlying proxy, renewing the underlying Kerberos ticket. Service Policies can be used to ensure failed service calls automatically retry.

FIX – Service Connectors fail to preserve property – When creating the reply message the RequestHeadersToPreserve header property value was not getting copied to the new reply message.

FIX - Service Connector – If a SOAP call was made with the “Enable HTTP Headers” property enabled on the Service Connector, the following warning would be written to the Neuron log file:

“captureHttpResponseProperties arguments: Method = null. This could happen if the old text properties are being used to set the Method on the ESB Message object. Please use ESBMessage.Http.Method instead.”

The warning is now only raised if the call is to a REST based service and the Method is not set.

Adapters

FIX – MQSeries – If the Translation Option was set to None, whether or not the MQSeries Format property was populated, the message bytes were published to Neuron ESB as a binary message. This has been corrected. If the Format property is populated, the message type will be set to non binary.

FIX – MQSeries – MQMDE Format property (if present) was not being evaluated against Translation Options.

MOD – MQSeries – The Remove BOM property has been removed. If a BOM is present, it is now always removed.

MOD – MQSeries – Translation and Conversion has been modified to ensure consistency with MQSeries. If the “MQ Convert” property is set to true, the MQGMO CONVERT flag will be added to the

MQGMO options before the MQGET, directing MQSeries to do the encoding translation. If the “MQ Translation Options” property is set to NONE, the message will be converted to *CodedCharSetId* of the queue manager. If the “MQ Translation Options” property is set to UTF8, Unicode or ASCII or, where “MQ Translation Options” is set to CharacterSet and a CCSID is entered, the MQMessage’s MQMD.CharacterSet property will be set to the CCSID (or appropriate CCSID that represents UTF8, ASCII or Unicode), directing MQSeries to do the translation. If the “MQ Convert” property is set to False, the selected “MQ Translation Options” option is done by the Neuron ESB runtime.

FIX – MQSeries – If the Unicode option is selected for the "MQ Translation Options" property, the following error could be thrown:

Translation of message body to unicode failed. Attempting default translation by CCSID. Unable to read beyond the end of the stream.

NEW - SharePoint 2013 plugin – A new SharePoint 2013 plugin for event-based publishing has been added.

NEW – Dynamics AX 2012 – A new Dynamics AX 2012 Adapter has been added. This will work with AX 2012, 2012 R2 and 2012 R3.

MOD – Dynamics CRM XRM Adapter - Microsoft has issued a notice that all applications using the Microsoft Dynamics XRM SDK for connecting to Dynamics 365 must be upgraded prior to June 30, 2017. To support this requirement, Neuron ESB now ships with the necessary assemblies from version 8.2.1.1 of the Microsoft Dynamics 365 Software Development Kit. The new assemblies are backwards compatible with Dynamics CRM 2011 and later. This change requires you to install version 4.5.2 of the .NET Framework. This is only required if you are using the Dynamics CRM XRM Adapter. The previous version of the Microsoft Dynamics XRM SDK assemblies was 6.0.1.219.

NEW – SFTP Adapter – Support has been added for HTTP, SOCKS4 and SOCKS5 proxies.

Business Processes

NEW - Flat File Process Step – Supports converting flat file to JSON as well as XML.

MOD - Flat File Process Step – Users can double-click process step to launch Flat File Wizard.

MOD - Set Property Process Step – Users can double-click process step to launch Expression Editor.

MOD - Set Property Process Step – All Neuron ESB Process Step properties and all constants have been added to the drop down list.

MOD - Trace Step – Will only trace if Trace Level is set to Info.

MOD – Workflow Process Step – This has been removed.

MOD – JavaScript Process Step – This has been removed.

MOD – Rules – WF – Renamed to “Rules Engine”. \

MOD – Rules Engine - Users can double-click process step to launch Rules Editor.

MOD – Http Client Utility - Users can double-click process step to launch HTTP Form.

MOD – Xml Query, Table Query – Neither was returning a Neuron ESB Message with the Reply semantic set.

FIX - Parallel – This would throw the following exception if a Branch of a Parallel Step contained another Parallel Step:

System.ArgumentException: An item with the same key has already been added.

NEW – Flat File to Xml – The option to apply W3C XML Schema data types to fields has been added to the column definition page of the wizard. The datatypes will be reflected in the XSD Schema generated (if enabled) and will be used to validate the parsing of the flat file at runtime.

MOD – Flat File to Xml – Engine caching has been disabled when testing in the Business Process Designer. Previously, if a user made a change and retested, the change would not be reflected in the test due to caching. The only way to clear the cache was to change the name of the process step or close and re-open the Neuron ESB Explorer.

MOD - Flat File to Xml – Drop down box listing Text documents from the repository is now listed alphabetically.

FIX - Flat File to Xml – if the “First Row Contains Column Names” option was selected, the previously saved flat file definition would not properly load within the wizard. The Delimited/Comma option would always be selected instead.

FIX – Flat File to Xml – After making changes and closing the Wizard the Business Process Designer would not be notified of the change. Consequently, the Apply button would not be enabled.

NEW – Flat File to Xml – Include Schema property has been added that functions identically to the existing Include Schema feature in the Excel To Xml Process Step. When set to True, the XML output of the Flat File to Xml Process Step will be prepended with an XSD Schema.

MOD - Flat File to Xml - If a process step does not have a name assigned to it, the following warning will be written to log files:

“The Flat File Parser uses a Cache of parsing engines based on the Name property of the Process Step. The current Flat File Parser Process Step does not have a value for its name property. If there is only one Flat File Parser process Step on the <Name of Process>' Business Process then the correct parsing engine will be used. If there are more than one, and they are configured to use different Flat File Parsing definitions, the wrong engine will be used to parse incoming files. Please ensure that each Flat File Parser Process Step is assigned a unique name”

FIX - Flat File to Xml - would throw *ArgumentOutOfRangeException* exception if there were more fields in a row than there were defined columns due to missing Quoted Character. Multiple delimiters within a defined Quoted Character were not supported. Now a custom error is thrown telling the user exactly what field is affected and how to correct.

FIX - Flat File to Xml – An object null exception would occur when the User defined option was selected, followed by navigating to the page to create columns. After which if the Back button was selected followed by selecting the Next button to navigate to the page to create columns again.

MOD – Excel to Xml – Root Element Name, Row Element Name and Namespace properties are now all optional.

MOD – Excel to Xml – When generating an XSD Schema by setting the Include Schema property to True, msdata based attributes are now removed from all elements.

FIX – Execute Process Step causes runtime crash – If an endpoint referenced a Business Process that contained an Execute Process Step and the process that called, contained another Execute Process Step, if the endpoint was later recycled due to a change deployed that affected it, the Neuron ESB runtime may unexpectedly terminate.

FIX – Timeout Process Step – An additional 30 seconds would always be added to the value of the Timeout property due to an internal wait interval.

FIX - Service Endpoint Process Step – If a SOAP call was made with the “Enable HTTP Headers” property enabled on the Service Connector, the following warning would be written to the Neuron log file:

“captureHttpResponseProperties arguments: Method = null. This could happen if the old text properties are being used to set the Method on the ESB Message object. Please use ESBMessage.Http.Method instead.”

The warning is now only raised if the call is to a REST based service and the Method is not set.

MOD – Compression Step – ContentType and ContentEncoding custom Neuron ESB Message properties have been added. These have also been added to the Neuron.Esb.Zip.Attachment class.

FIX – Service Endpoint Process Step - If the service endpoint step is configured to use a service policy, and that service policy has the option "Do not apply policy if the failure is a service fault", that option will override the Service Endpoint Step property "Throw Exception on Fault". With this combination of options an Exception would never be thrown by the Service Endpoint Step when a fault occurs. This has been corrected.

Transports

FIX - Rabbit MQ - If topic is disabled at server, the parties that are connected to a Rabbit MQ based Topic, would not receive the Neuron ESB OffLine event and would remain connected to the underlying Rabbit MQ server.

FIX – Rabbit MQ – The UncompressedBodySize Neuron ESB Message Header property was not properly serialized to subscribing clients. The property would always have a value of zero.

Messaging

FIX – Party Dispose- When disposing a party from within the party’s offline event, a LockRecursionException could be thrown.

Neuron 3.5.4.1017 Change History

Transports

FIX – Rabbit MQ Dead Letter Processing – If a message expired, the Neuron ESB dead letter processor would attempt to move it to the Neuron ESB Failed Message table, generating a System.NullReferenceException and spiking CPU utilization to 100%.

Neuron 3.5.4.1013 Change History

Neuron ESB Samples

FIX – Custom Process Steps – Several resource files were missing from the samples shipped in CU4 which prevented the samples from working. These have been added back in.

Neuron ESB Solution Storage

MOD – Save Configuration – In previous releases, Neuron ESB would store the contents of the repository files as both xml files in their respective sub directories within the solution folder structure, as well as raw data in the esb_configuration.xml file, essentially duplicating the data storage. Neuron ESB was also duplicating the Zone data within the esb_configuration.xml file as well as in its respective Zone xml file within the solution folder structure. This would cause issues when editing Zone settings as sometimes they would not overwrite the settings in the esb_configuration.xml file. At runtime the Zone settings in the esb_configuration.xml file would be used, rather than those in the Zone xml file. Lastly, we were also storing the Administrator Role credentials (encrypted) in the esb_configuration.xml, rather than as encrypted XML files in a “Administrators” sub folder. With this release, once a user saves a change in an existing solution, Neuron ESB will move all the Administrator Role accounts from the esb_configuration.xml to the “Administrators” sub folder. The save process will also remove all the redundant repository document data from the esb_configuration.xml, leaving only the repository metadata. Lastly, the save process will also remove the Zone data from the esb_configuration.xml.

FIX – Save Configuration – Unhandled Exception could occur when clicking the Save button on the Neuron ESB Explorer Toolbar. This has been corrected.

FIX - XSLT Document Load Issue - When Loading the XSLT document from the Neuron ESB Repository, special Markup characters like carriage returns, etc., are not rendered in the final displayed XSLT or executed XSLT.

FIX – Swagger Document Load Issue – If a Swagger document was opened in a previous solution, it would not close correctly when opening a different solution and would remain visible if a user navigated back to the Swagger Document Repository.

Business Processes

FIX – Service Endpoint Process Step – The Service Endpoint could fail to throw an exception if a Fault Message was received and the process step’s “Throw exception on fault” property was set to true, if a previous execution of the process step threw an exception.

FIX – Execute Process Process Step – if using the Context.State dictionary within a process called by the Execute Process Process Step to maintain state between executions of the called process, the state would not be maintained between executions.

MOD – Execute Process Process Step – When declaring a value for the “ESB Solution Source” property (used to tell Neuron that the process to execute is located in a different solution other than the existing one open), Neuron was making a full copy of the remote configuration for each process instance in the pool. Now it only reads the Business Processes from the remote configuration, rather than reading and decrypting ALL Neuron ESB entities, like Topics, Endpoints, Workflows, etc.

FIX – Execute Process Process Step - When declaring a value for the “ESB Solution Source” property (used to tell Neuron that the process to execute is located in a different solution other than the existing one open), under high concurrent load, Neuron ESB would report at least one of the following exceptions:

System.NullReferenceException: Object reference not set to an instance of an object

This would be followed by many Pool Timeout exceptions. This was because we could have competing threads all trying to create copies of the remote configuration, which entailed file deadlocking. This has been corrected. Only one global remote configuration object is created now for all process instances to be created from.

FIX – Execute Process Process Step – This could throw the following exception during concurrent execution as in the case of being hosted in a Client Connector when processing many concurrent calls:

Exception Type: System.ArgumentException

Exception Message: The key already existed in the dictionary

NEW – Execute Process Process Step – This has been modified to use a Blocking pool of processes to allow better scale out for concurrency and to eliminate all calls being serialized through one process. There are now 2 new properties located in the “Pool Management” property category. Maximum Instances (defaults to 100) and Pool Timeout (defaults to 1 minute). Once the maximum number of pipeline instances have been created, the Pool Timeout determines the amount of time to wait for an existing pipeline instance to become available before throwing a timeout exception.

Properties	
✓ Bindings	
Bindings	(Collection)
✓ General	
Name	PipelineExecution
Disable	False
✓ Misc	
ESB Solution Source	
Process	Create ODBCAdapter Msg
✓ Pool Management	
Maximum Instances	50
Pool Timeout	00:01:00

NEW – Event Source access – within a Business Process users can check the direction of execution they are within. For example, users may want to do something specific if a Business Process is running on the On Publish event of a message versus the On Receive event. This can be accessed using the following C# within a Process Step:

```
if(context.EventSource == ExecuteEvent.OnReceive)
{
    // if receiving ...do something
}
```

FIX – Adapter Endpoint Process Step - A null exception could occur due to a race condition.

FIX – Referencing External Assemblies - If there is a custom assembly referenced at the process level, but that assembly is later deleted from the machine, the code step referencing that assembly will still “think” it exists and not throw any compilation errors during editing. If the assembly reference is removed from the process level assembly reference collection, the code editor will start throwing compilation errors again (users have to get the editor to recompile the code after removing the reference, which can be done by changing anything in the code for the step). This has been corrected so that the code editor will detect the missing assembly and display the appropriate compilation errors

FIX - Adapter Endpoint Process Step – this can now be tested at design time within the Business Process Designer.

MOD – We finally removed all references to the old text based properties we used to use to access SOAP and HTTP context properties before we added the HTTP and SOAP API on ESB Message. Specifically, the "MessageHeader", "MessageProperty_HttpRequest" and "MessageProperty_HttpResponse" property prefixes have all been removed. These were deprecated several years ago. Typical uses for these would be within a C# Process Step. Here is an example of how they were used:

```

context.Data.SetProperty("MessageProperty_HttpRequest", "Method", "POST")
;
context.Data.SetProperty("MessageProperty_HttpRequest", "Content-
Type", "application/xml");
context.Data.SetProperty("MessageProperty_HttpResponse", "Location",
"");
context.Data.SetProperty("MessageProperty_HttpResponse",
"StatusCode", "200");
context.Data.SetProperty("MessageProperty_HttpResponse",
"StatusDescription", "great");
context.Data.SetProperty("MessageHeader", "LocalPath", "/path");

```

Here's how this same information should be set/retrieved using the HTTP and SOAP API:

```

context.Data.Http.Method = "POST";
context.Data.Http.Headers.ContentType = "application/xml";
context.Data.Http.Headers.Location = "";
context.Data.Http.StatusCode = System.Net.HttpStatusCode.OK;
context.Data.Http.StatusDescription = "great";
context.Data.LocalPath = "/path";

```

Developers should use something like Notepad++ to search the Processes directory of their solution for all places where either the "MessageHeader", "MessageProperty_HttpRequest" or "MessageProperty_HttpResponse" text properties are being referenced and replace them with the appropriate API.

NEW - Audit Process step - This has been modified to use a Blocking pool of proxies to allow better scale out for concurrency and to eliminate all calls being serialized through one proxy. There are now 2 new properties located in the "Pool Management" property category. Maximum Instances (defaults to 1) and Pool Timeout (defaults to 1 minute). When set to 1, the Audit Process Step will operate as in previous versions, using the default Audit Proxy at the Topic level. Once the maximum number of instances have been created, the Pool Timeout determines the amount of time to wait for an existing proxy to become available before throwing a timeout exception.

Properties	
▼ Bindings	
Bindings	(Collection)
▼ General	
Disable	False
Name	Audit
▼ Misc	
Action	Receive
Audit Custom Proper	True
Audit Message Body	True
XPath	
▼ Pool Management	
Maximum Instances	48
Pool Timeout	00:01:00

MOD - Adapter Endpoint Process Step - There are now 2 new properties located in the “Pool Management” property category. Maximum Instances (defaults to 100) and Pool Timeout (defaults to 1 minute). Once the maximum number of instances have been created, the Pool Timeout determines the amount of time to wait for an existing proxy to become available before throwing a timeout exception.

Properties	
▼ General	
Name	ExecuteAdapter
▼ Misc	
Adapter Endpoint	OdbcQueryEndPoint
Timeout	00:01:00
Adapter	OdbcAdapter
Category	
Mode	Execute
Party Id	PlayaSub
Policy	Default
Run As	
▼ Pool Management	
Maximum Instances	50
Pool Timeout	00:01:00

NEW - Logging – A new logging level flag has been added to the Process Designer. This allows developers, if writing custom tracing statements, to first check the runtime logging level so that the compiler doesn’t unnecessarily create strings at runtime. For example, in the Business Process Designer within a C# activity custom logging could look like this now:

```

if (context.Instance.IsDebugEnabled)
    context.Instance.TraceInformation("only run this statement if
'Verbose' logging is enabled");
if (context.Instance.IsInfoEnabled)
    context.Instance.TraceInformation("only run this statement if 'Info'
logging is enabled");

```

```

if(context.Instance.IsWarningEnabled)
    context.Instance.TraceInformation("only run this statement if
'Warning' logging is enabled");
if(context.Instance.IsErrorEnabled)
    context.Instance.TraceInformation("only run this statement if 'Errors'
logging is enabled");

```

All logging statements will be written to the Neuron ESB log files.

MOD - Adapter Endpoint Process Step – Removed the internal retries based on policy that we previously did. This makes it consistent with the Adapter Endpoint Workflow Activity. If users require retries, this can be easily accomplished by enclosing the process step within the Retry Process Step.

NEW - Service Endpoint Process Step – This has been modified to use a Blocking pool of proxies to allow better scale out for concurrency and to eliminate all calls being serialized through one proxy. There are now 2 new properties located in the “Pool Management” property category. Maximum Instances (defaults to 100) and Pool Timeout (defaults to 1 minute). Once the maximum number of instances have been created, the Pool Timeout determines the amount of time to wait for an existing proxy to become available before throwing a timeout exception.

Properties	
Disable	True
Name	ServiceEndpoint
Misc	
Service Endpoint	PaymentService
Service Url	http://localhost:8733/De
Service Binding	BasicHttp
Message Pattern	Request-Reply
Service Policy	
Category	
Description	
Throw Exception on I	False
Pool Management	
Maximum Instances	100
Pool Timeout	00:01:00

FIX – Audit Process Step - fixed issue where audit service would report SqlDateTime overflow i.e. “Must be between 1/1/1753 12:00:00 AM and 12/31/9999 11:59:59 PM”, if testing the Audit Process Step for first time in designer. Subsequent tests would succeed.

NEW - Service Endpoint Process Step – Users can dynamically set the name of the Service Connector to execute at runtime by modifying the Service header of the ESB message like so:

```

context.Data.Header.Service = "myServiceConnector";

```

FIX - ODBC Process Step - when running in a for loop under stress the following error may occur:

Exception Type: System.NullReferenceException

Exception Message: Object reference not set to an instance of an object.

Adapters

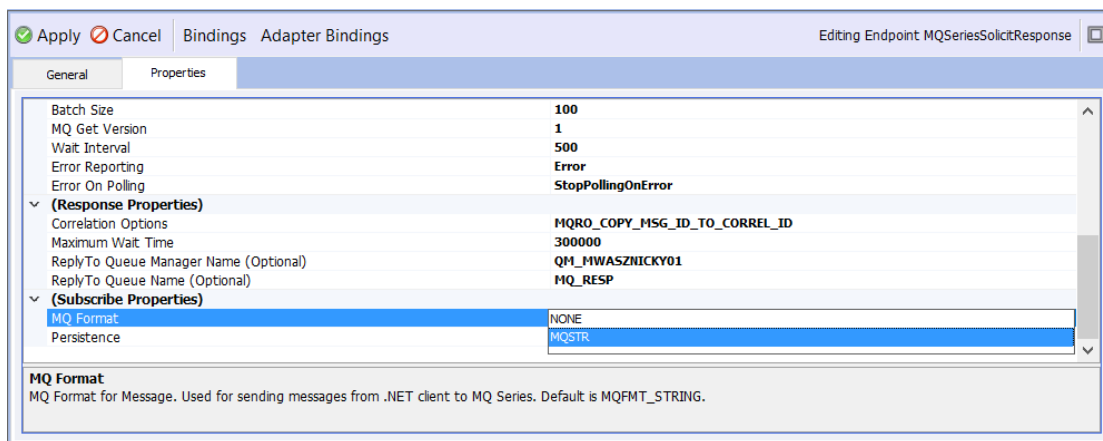
FIX – SQL Service Broker - When and “END CONVERSATION” message is being processed The following error could be thrown if multiple result sets was not enabled on the connection string:

Exception: The SQL Service Broker adapter endpoint, 'N', encountered an error while publishing to the bus. There is already an open DataReader associated with this Command which must be closed first.

Inner Exception: There is already an open DataReader associated with this Command which must be closed first.

Method: ValidateConnectionForExecute

FIX – MQSeries Adapter – When setting the MQ Format property to “NONE”, the property would show an empty string. This has been corrected.



FIX - SMTP Adapter – When sending file attachments, the name of the file in the email received would display the full file path and name of the attachment. Now it will just display the original filename minus the original path. Also, when sending non XML messages as the body, an invalid root error could be thrown when attempting to resolve custom properties via XPATH.

NEW - MQSeries Adapter – Extended logging has been added. Also, if a publish fails, the adapter will attempt to log the message to the Neuron Audit database.

NEW - Adapter Framework – Added logging levels to the Adapter Framework base class so that adapter writers can check the logging level before creating their strings i.e.

```
if (base.IsDebugEnabled)
    RaiseAdapterInfo(ErrorLevel.Verbose, "hi there");
```

FIX - NetSuite Adapter – Fixed memory leak detected under concurrent load.

FIX - Salesforce Adapter – Fixed memory leak detected under concurrent load.

FIX – ODBC Adapter – Fixed memory leak detected under concurrent load when executed from an Adapter Endpoint Process Step.

Workflow

FIX – Service Endpoint Workflow Activity – The Service Endpoint could fail to throw an exception if a Fault Message was received and the activity’s “Throw exception on fault” property was set to true, if a previous execution of the activity threw an exception.

FIX – Deserialize Workflow Activities – If a user entered a value for a property in the designer, rather than directly in the Workflow Activity’s property grid, the Workflow Activity would not pick up and use the value. This only affected the “Deserialize from DataContract” and “Deserialize from JSON” Workflow Activities.

MOD – Empty Referenced Assemblies element generates Null Exception on startup – if the *ReferencedAssemblies* element within the XML file that represents a Workflow Definition contains empty string elements (i.e. `<string></string>`), a null exception would be generated on startup. An empty string element could occur if the XML files are manually edited. If this occurs, users can open the Neuron Solution within the Neuron ESB Explorer. If empty elements are detected, they will be removed and the user will be notified of the modification via the Changes dialog box. To persist the changes, the user must accept them and save the solution using the Neuron ESB Explorer.

NEW - Service Endpoint Workflow Activity – Users can dynamically set the name of the Service Connector to execute at runtime by modifying the Service header of the ESB message like so:

```
message.Header.Service = "myServiceConnector";
```

FIX - Workflow Endpoints – if a workflow endpoint was disabled but still assigned to an Availability Group, it would still attempt to start and generate a `System.ArgumentNullException` exception.

NEW - Logging – A new logging level flag has been added to the Workflow Designer. This allows developers, if writing custom tracing statements, to first check the runtime logging level so that the compiler doesn’t unnecessarily create strings at runtime. For example, in the Workflow Designer within a C# activity custom logging could look like this now:

```
if (Log.IsDebugEnabled)
    Log.Debug("only run this statement if 'Verbose' logging is enabled");
if (Log.IsInfoEnabled)
    Log.Info("only run this statement if 'Info' logging is enabled");
if (Log.IsWarningEnabled)
    Log.Warning("only run this statement if 'Warning' logging is enabled");
if (Log.IsErrorEnabled)
    Log.Error("only run this statement if 'Errors' logging is enabled");
```

All logging statements will be written to the Neuron ESB log files.

FIX - Workflow Persistence Delay Activity – When a persistent delay workflow activity completes, Quartz.Net could throw the following exception:

“An error occurred while attempting to delete a job when the PersistentDelay activity completed: Unable to unschedule trigger”

Service Endpoints

FIX - Client Connectors – for SOAP messages, the message would not always be set to non-binary. This has been corrected.

FIX - REST Service endpoints – Buffer Size and Max Buffer Pool Size are not being set to their default properties internally. Previously these were being set to too high a value which could result in leaked memory over time. Also, REST service endpoints will now use the Maximum Message Size parameter from the UI.

NEW – Service Endpoints – Added support for users who want to use the Microsoft WCF WebHttpBinding with Neuron Client and Service Connectors. If a custom binding is created with that name, Neuron ESB will recognize it and use it. This can be used as an alternative to the built in Neuron ESB REST binding.

MOD - Client Connector - Changed from Single to PerCALL mode. Also fixed issues with service endpoints that route to SOAP where endpoint has capture http headers enabled. When they received a message that came from a REST call where HTTP headers were enabled, content type and get methods would kill the call. Lastly, faulted events with the host are captured and logged appropriately.

Neuron ESB Runtime

MOD – Zone Internal Binding Settings – These settings located on the Server tab of the Zone are used by all of the internal Neuron ESB services such as the Configuration, Control, Management and Auditing services. The default values for new solutions have been changed to:

Internal Service Binding Settings	
▼ Binding	
Close Timeout	60
Listen Backlog	128
Max Connections	1000
Open Timeout	60
Send Timeout	60
▼ Service	
Max Concurrent Calls	500
Max Concurrent Instances	200
Max Concurrent Sessions	1000
Reliable	False
Max Concurrent Calls	
The upper limit of active messages in the service. The default value is 500 but can be modified.	

MOD – Internal Session Management – In previous versions, Neuron ESB Parties and Endpoints maintained persistent connections to Neuron’s internal Control, Management and Configuration

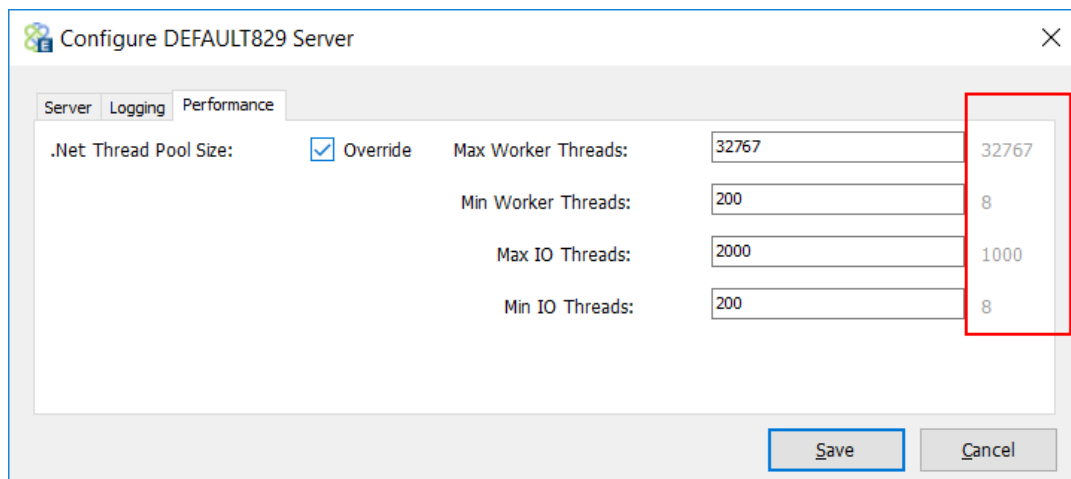
services. Increasing numbers of endpoints added to a solution could substantially increase the number of sessions required to be configured at the Zone level (i.e. Max Concurrent Sessions). In very large solutions we've seen this number approach several thousand. If this number was not configured high enough, endpoints would time out attempting to connect at startup or new endpoints could be blocked from connecting. Moving forward this has been changed so that these are no longer persistent sessions. This will result in a significant decrease in the number of sessions created against Neuron ESB internal services and should dramatically reduce the incidences of blocking or timeouts when starting up or connecting to the Neuron ESB server.

MOD – Adapter/Service Policy Retries – When using Transactional MSMQ as a Topic Transport, the following error could be reported in the Neuron ESB log file and Event log:

System.ServiceModel.FaultException: The transaction under which this method call was executing was asynchronously aborted.

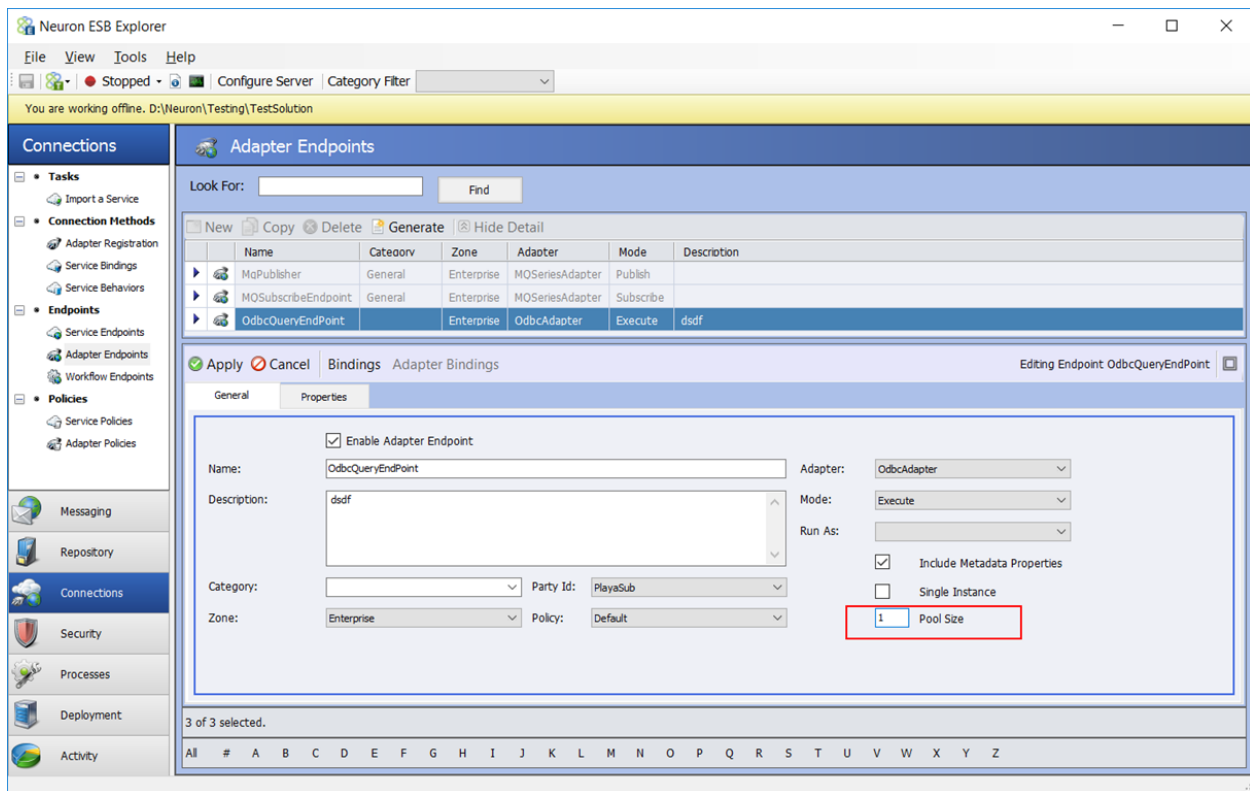
This was happening within our Global Exception Handler for WCF and occurred when Neuron ESB rolled back the underlying Transaction in the course of the Policy execution. This error was erroneous and has now been suppressed.

MOD – Configure Server - Added the ability to view the default .NET thread values when opening configure server dialog within the Neuron ESB Explorer. This dialog allows the user to override the existing .NET thread pool values to accommodate burst load scenarios. The default values are listed in greyed out text to the right of the editable text boxes as shown in the figure below:



MOD – Startup time – Solutions that have hundreds of endpoints should start up significantly faster now, anywhere from 50% or more.

MOD – Adapter Endpoint Performance Enhancements – In previous releases of Neuron ESB, message traffic received by Adapter Endpoints was always serialized to support ordered delivery. In this release, we've added a new "Pool Size" property located on the general tab of the Adapter Endpoints screen within the Neuron ESB Explorer (see below)



If this property is set to a default of 1, the behavior will be consistent with previous builds of Neuron ESB, essentially serializing all calls to the Adapter Endpoint. The pool size number indicates the maximum number of adapter instances that Neuron ESB “could” create under the covers to process messages. For example, if the Pool Size was set to 250, but Neuron ESB was only sending in a few hundred request/sec, more than likely only a handful of adapter instances would need to be created to handle the throughput. In previous builds, normal throughput for an Adapter endpoint was generally limited to a couple of hundred messages per second processed (depending on the adapter). By increasing the Pool Size property, that number can be increased to several thousand messages per second processed. If a Neuron ESB Adapter Endpoint is receiving more messages than the configured Pool Size can service, Neuron ESB will write out warnings in the Neuron Log file of the endpoint alerting users that Neuron ESB is occasionally waiting for an instance to become free to use i.e.

2017-03-26 20:29:24.822-07:00 [4] ALWAYS - ESB Trace Log - Machine: - Trace level: Error+Warning

2017-03-26 20:29:37.204-07:00 [168] WARN - Waiting to retrieve 'OdbcQueryEndPoint' adapter endpoint from collection because we've hit our pool size limit.

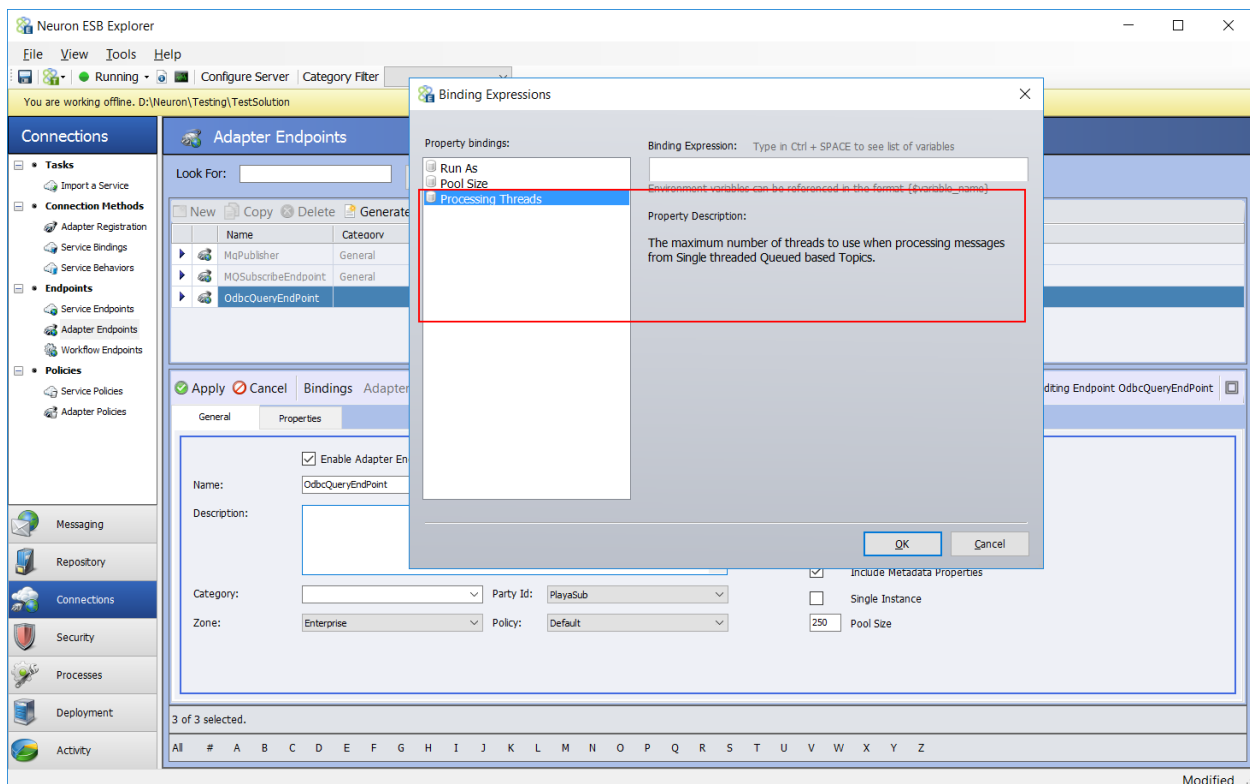
This is a good indicator that the Pool Size should be raised, or the Adapter Endpoint itself is too long running in its processing of a message.

For example, using a SOAP Client Connector, TCP Topic and an ODBC Adapter in Query mode, (basic Request/Response scenario), a pool size of 50 to 100 allowed the service endpoint to achieve 10+ times the throughput of previous releases, while decreasing the average request/response times.

In a multicast scenario, publishing to a TCP Topic to an ODBC Adapter in Execute mode (inserting a record into a table), using a pool size of 250, we saw up to a 40 times increase in throughput/inserts into the table.

An important note though. This setting will have marginal if any performance impact when messages are received by the Adapter Endpoint either over a Rabbit MQ topic or an MSMQ Topic where the MSMQ Transport is configured for Ordered Delivery.

MOD – Adapter Endpoint Performance Enhancements – A “Processing Threads” property has been added to Adapter Endpoints and accessible either by using an Environmental Variable to set the property (as shown below)



Or by manually editing the underlying XML file:

```
<?xml version="1.0"?>
<ESBEndpoint xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Id>07dfa54f-a1d1-4405-8063-3a5c17f218cb</Id>
  ...
  <PoolSize>250</PoolSize>
  <ProcessingThreads>30</ProcessingThreads>
```

</ESBEndpoint>

The “Processing Threads” value is 1 by default. This provides the behavior found in previous builds. However, where throughput could not be increased by using the Pool Size property for Ordered Delivery Queued Topics (i.e. Rabbit MQ or MSMQ set for Ordered Delivery), throughput can be substantially increased for these Topics (similar to using the Pool Size property for non-Queued Topics) by raising the value of this property. This property should NOT be used if receiving messages from non-Queued Topics like TCP, Peer or Named Pipes, or if Ordered Delivery of the messages to the adapter endpoint is a requirement. A good example where users may want to set this property to a high value is when receiving messages from a Rabbit MQ Topic as Rabbit MQ only supports ordered delivery of messages.

FIX - Auditing Service – Previously in our debug tracing, we were not logging the action date.

FIX – ESB Message Cloning – When an ESB message was cloned, the LocalPath property was not being copied to the new message

FIX - Runtime Optimizations – Several resource optimizations around string creation and memory usage have been implemented.

FIX – Management Thread – The management thread at the Party level could be aborted by an exception from an asynchronous thread. If the management thread, it could also create an orphan lock that could prevent either the Neuron ESB runtime and the Workflow engine from shutting down. The orphan lock has been corrected and management routine has been moved to a timer.

FIX – Audit Service – Added new locking mechanisms to increase throughput and concurrency and to eliminate deadlocking that could occur under high concurrency load.

FIX - Memory Leaks – While under sustained load, there were several memory leaks discovered using Service endpoints and some adapters, specifically the ODBC adapter. There were also some detected when using the Service Endpoint and Adapter Endpoint process steps. These have been fixed.

MOD – Master Service, refactored the GetMachineName() so that issues with it can be logged to neuron log files and that it raises exceptions that can be caught and logged. Added the logging of additional information to the master log on startup, regardless of the log level now. Specifically, whether or not the MSMQ and MSDTC were installed, what the service account is set to, port sharing, Neuron ESB Instance name, computer memory, # of CPUs, OS, whether neuron or OS is 64 bit and what the true NETBIOS name is.

MOD – Master Service – We no longer are writing out the whole stack trace if we can't delete a temp file. Now just writing our error message.

FIX – Unable to Assign Port when Port Sharing is enabled - Previously, users could see errors like the following even if Port Sharing was enabled:

Availability Group: Workflow Order Process Host

Message:

Workflow Endpoint id 'bffe743c-6a9f-4a9f-80c0-30969eefd2e7' could not be started.
System.Exception: Unable to assign the '50011' port to the 'Adapters' TCP based Topic. The given key was not present in the dictionary. --->
System.Collections.Generic.KeyNotFoundException: The given key was not present in the dictionary.

MOD - Configuration Service – Enabled a configuration read lock when reading the configuration.

MOD - Control Service - Removed unnecessary locks when parties are retrieving the configuration from the service.

MOD – Zone – Server Bindings - Reset the defaults for Max Concurrent Sessions and Max Concurrent Instances to 100. These settings were changed for the MSMQ, TCP, Named Pipes Publishing services as well as for the Server Bindings. We also reset the Listen Backlog default property from 1000 to 10. As a rule of thumb, Neuron ESB doesn't utilize sessions unless Reliable Sessions is set to true on the Topic Publishing services that support Reliable sessions or on the Bindings tab of Zone settings. The Binding tab of Zone settings only affects the Neuron ESB internal NON publishing services like Auditing, Control, Configuration and Management services. Out of these, ONLY the Auditing Services is affected by the Max Concurrent Instances and should not be raised above 200 without thorough load testing as this will control how many Instances remain in memory and will consume resources.

FIX – Audit Service – This was hard coded with a 5-minute command time out. Now this will use the command timeout configured for the Neuron Audit database.

MOD - Audit Service - Changed from Single to PerCall mode to better handle concurrency and scale. This effectively uses the Max Concurrent Instances setting located in the Zone settings on the Bindings tab. This setting should not be raised above 200 without thorough load testing spanning several hours.

Monitoring and Logging

FIX – Logging JSON Messages – If running with Verbose logging level enabled, some JSON formatted messages could generate the following error when Neuron ESB attempts to write the message bodies to their respective endpoint log file, as in the case of Client and Service Connectors:

System.FormatException

Exception Message: Input string was not in a correct format.

FIX - Logging – Logging has been modified. Now, All Warnings and Errors will be reported in both the Neuron ESB log files and the Neuron ESB Windows Event Log. ONLY Verbose and Info logging will be recorded in the Neuron ESB log files. Also, all logging formats have been normalized. An example of the format follows:

```
string.Format(  
CultureInfo.CurrentCulture,
```

```
"{0:yyyy-MM-dd HH:mm:ss.fffzzz} [{1}] {2} - {3}",  
DateTime.Now,  
Thread.CurrentThread.Name ??  
Thread.CurrentThread.ManagedThreadId.ToString(CultureInfo.InvariantCulture),  
InfoLevel,  
message);
```

In previously releases, this was inconsistently applied. In some areas we would be writing errors and warning to either the Event log or the Neuron ESB log files, but not to both.

FIX – Logging – Reset the ESBHost.exe.config file to only log warning and errors to the event log via log4net filter.

FIX – Adapter/Service Policy Retries – Logging has been enhanced and standardized.

FIX – Logging under Run As – When using the “Run As” feature to run an Adapter Endpoint under a specific set of credentials, if the Adapter Endpoint generated errors and warnings may fail to write to the Event Log, though they would be recorded to the underlying Neuron ESB Log file. This has been corrected.

FIX - Endpoint Health Reporting – Fixed some issues found where the WMI counter was not getting incremented properly under certain conditions when errors and warning detected. This was also happening in the Rabbit MQ publishing service.

FIX – Recycling Endpoints during Shutdown causes error – If an Endpoint was changed and saved and the ESB Service was shut down at nearly the same time, the following exception could occur:

Exception: Value cannot be null.

Parameter name: The Adapter endpoint 'OdbcQueryEndPoint' could not be found in the endpoint tracking collection.

Transports

MOD – MSMQ Transport – Previously the Neuron ESB Dead Letter and Poison Queue handlers were configured as singletons. They have been changed to Per Call.

FIX – MSMQ Transport Network Properties – If the “Use Policy for Retries” property was configured, the "Receive Error Handling" and "Enable Poison Message Processing" properties were still visible. Also, if a policy was selected, the Time to Live property could be reset to an incorrect minimal value. This has been corrected so that it will be set to 100 * the expiration period defined in the policy. This can be overridden by the user.

FIX – Rabbit MQ - If the same party instance is used across multiple client connectors, the instances could randomly receive the wrong reply destined for a different party instance. For example, if there were 2 client connectors using the same party, the user would experience a timeout on every other

request call. This is due to the fact that there is only one physical queue representing a Party. Additional correlation has been added to prevent this behavior.

FIX – Rabbit MQ – Under load, Rabbit MQ could fault with an exception reported similar to the following:

Unexpected Shutdown event from Rabbit MQ consumer for 'PlayaPub' Party. Initiating source = Library; Reply Code = 541; Reason = Unexpected Exception

This was due to a bug in the Rabbit MQ client version 3.5.6 stack. In the Rabbit MQ server's log file, the following would be noted related to heart beat failures:

ERROR REPORT==== 6-Mar-2017::17:35:43 ===
closing AMQP connection <0.16885.134> (::1):59819 -> (::1):5672):
missed heartbeats from client, timeout: 60s

Rabbit MQ fixed this in later releases of their client API. Neuron ESB has been updated to use the 4.1.3 Rabbit MQ Client Library.

FIX - Rabbit MQ Topic Transport – An error would be reported if a change was made and saved in the Neuron ESB Explorer indicating that the active deployment group could not be found. Although, harmless, this has been fixed.

MOD – TCP channel – When a party which is associated with a TCP topic starts up, it will look for an available client side port that the TCP publishing service can use to send it messages. Previously we would write out the stack trace for search when we called the FindAvailablePort() method. Now we will only write out the message in the Neuron ESB log files. Users will see this in any endpoint log when in Verbose mode.

FIX – Rabbit MQ channel – this would throw an asynchronous thread error when being disconnected, which in turn would abort the management back thread routine and cause an orphaned lock. Locking has been refactored to eliminate a deadlock that could occur when using the underlying connection object. Lastly, we were not serializing the ESB Message's Localpath property.

FIX – MSMQ Channel – Refactored this to be thread safe when used via a Publish Process Step within a Client Connector under high concurrent load. Previously, this could fail with Null exceptions and blocking. Also, several performance optimizations were made.

FIX – TCP/Named Pipes channel – Refactored locking to eliminate an Object Disposed exception that could occur when under high load.

Neuron 3.5.4.920 Change History

Neuron ESB Runtime

MOD – Clear Temp Files – On startup, the ESB Service will attempt to clear out temp files. If an error occurs, a warning will be logged indicating that a file within the temp directory could not be deleted. This has been modified to record the error message that was encountered during the delete attempt

FIX – Runtime Memory Fixes – At runtime memory could gradually increase when endpoints are recycled within Endpoint Health if those endpoints were hosting specific Process Steps.

MOD – Active Sessions/Connections to Database – Optimization have been made to reduce the number of active sessions/connections that Neuron ESB maintains against its database when connection pooling is enabled. Previously we were maintaining a separate connection pool, and hence a connection in that pool for every endpoint that was configured for Single Instance mode. This has been changed to use the connection pool of the central server. Also, we recommend that the “Load Balance Timeout” connection string setting for the Neuron ESB database be changed from the default of 0 to 120 seconds to ensure that connections that have not been used are removed from the pool.

FIX – Adapter Endpoint Process Step – Timeout Exceptions and Inconsistent reply messages received when processing concurrent requests under load – When using the Adapter Endpoint Process Step within a Process assigned to a Client Connector, Timeout Exceptions may occur as well as delivery of the wrong reply message to the calling thread if the Client Connector is processing concurrent requests under load. This was due to how the Adapter Endpoint managed the synchronization context for the reply messages it would return to the hosted Business Process. This has been corrected.

FIX – Availability Group would fail to Start – if Port Sharing was enabled, the Availability Group could fail to startup with an Instance is Null exception.

Service Endpoints

MOD – Client Connector Logging – Regardless of the logging level selected, Neuron ESB would always log the following informational message in the Neuron ESB log file for the Client Connector for every message processed:

```
ESBService.exe Information    0      SecureResponseDispatcher.BeforeSendReply - Setting  
the httpResponse header to 'Server'    2017-02-02 23:06:51Z
```

This behavior has been modified to only log when the logging level is set to Verbose.

Business Process Designer

FIX – Business Process Designer Disable Indicators – Within the Business Process Designer users can disable/enable every Process Step. When disabled, the process step would change color to a gradient grey. However, this did not previously work for the Exception, For, ForEach, While, Retry and Transaction Process Steps. If these steps were previously disabled, there would be no visual indicator. Also, disabling/enabling the individual Try/Catch/Finally blocks within the Exception block was not supported. These have been corrected.

FIX – Business Process Designer Memory Leak – if using the Business Process Designer to continually test in design mode, some of the process steps could leak memory. This has been corrected.

MOD – ODBC Process Step - Added support for setting parameters to NULL in the ODBC Process Step. This is accomplished with a new attribute on the Parameter element (query and execute) or Field element (batch) called nillable. When you set this to true and then set the value to "null", the column will be set to null. Also added support for setting DBNull or String.Empty as the default value in the parameter mapper.

FIX – Service Endpoint Process Step- Previously when an exception/fault occurred, the response message was set to the original request message. This has been modified so that the response message will contain the exception/fault information.

FIX – Service Endpoint Process Step – SOAP and REST and Policy Compliance – When using policy with the Process Step, the "Throw Exception on Fault" property will only work with SOAP-based services. REST-based services will only throw an exception when the REST tab configuration in the service policy is violated.

Adapters

FIX – FTP/STP Adapter – Timestamp Comparison feature - The following error: "Error converting data type varchar to datetime" could occur in certain localized environments because we were passing a datetime value in an ambiguous format to SQL server. Also, if more than one adapter endpoint was configured to use this feature, either one could return the others files.

FIX – Publish Mode Adapters – In publish mode, if “Error On Polling” property was set to “StopPollingOnError”, for any of the listed adapters, the adapter instance was not cleaning up its resources correctly. The affected adapters were the following:

- SQL Service Broker
- Active Directory
- Azure Service Bus
- File
- FTP/SFTP
- MqSeries
- Microsoft Exchange
- MSMQ
- ODBC
- POP3
- Salesforce

FIX - File adapter – An out of memory exception could occur if the file adapter, in publish mode, is continually called against a network share that is unavailable. Eventually an out of memory exception occurs as described in the following support article: <https://support.microsoft.com/en-us/kb/3152158>

FIX – Salesforce Adapter - The following error "adapter endpoint faulted. An invalid request URI was provided. The request URI must either be an absolute URI or BaseAddress must be set." May occur.

MOD – TCP Socket Adapter – A timer has been added to the socket adapters, configured for one-minute intervals, that will send ping messages across the socket. This is intended to make the socket appear to be alive during low-traffic periods so that the socket connection will not be dropped by intermediaries between the two servers. This is used to support Neuron ESB-to-Neuron ESB scenarios with a DMZ between the servers.

Neuron ESB Explorer

MOD – Updated Neuron ESB Help CHM and Samples – The Neuron ESB shipped samples have been updated as well as the Neuron ESB Help CHM documentation has been updated to include augmented Adapter and sample documentation.

Workflow Designer

FIX – Service Endpoint Workflow Activity- Previously when an exception/fault occurred, the response message was set to the original request message. This has been modified so that the response message will contain the exception/fault information.

FIX – Service Endpoint Workflow Activity – SOAP and REST and Policy Compliance – When using policy with the Process Step, the "Throw Exception on Fault" property will only work with SOAP-based services. REST-based services will only throw an exception when the REST tab configuration in the service policy is violated.

Neuron 3.5.4.875 Change History

Neuron ESB Service

MOD – Clearing of Temp Files – Neuron ESB generates temp files on start up. On startup, Neuron ESB would attempt to clear out the service account's TEMP directory of files previously generated by Neuron ESB. If this failed, Neuron ESB would continue to startup but would not report why the TEMP directory clean up failed. If the TEMP directory was filled, preventing Neuron ESB service from starting, there would be an indication in the Event log only. This behavior has been modified to write a warning to the Neuron logs if clearing the TEMP directory failed. Also, if the TEMP directory is filled, preventing from starting up, we'll attempt to write the reason within the Neuron ESB master log file. Lastly, Neuron ESB will attempt to clear out its temp files every hour during runtime operation.

Adapters

MOD – MQSeries Adapter – Added 3 new custom properties that are populated on Publish mode. Specifically, we now capture the inbound server name, Queue Manager name and Queue name, i.e.

```
var serverName = context.Data.GetProperty("wmq", "ServerName");  
var qManagerName = context.Data.GetProperty("wmq", "QueueManager");  
var queueName = context.Data.GetProperty("wmq", "Queue");
```

FIX – Socket Adapter – SSL would not correctly after Microsoft released the following patch:
<https://support.microsoft.com/en-us/kb/3155464> to address an existing vulnerability in the Microsoft implementation of SSL/TLS

FIX – Socket Adapter – When managing communication between internal and DMZ traffic using the Client and Server Socket adapter, the following error may be encountered:

The Server Socket adapter endpoint, 'N', encountered an error while sending message to client socket: The message cannot be sent to the TCP socket because the socket has been closed. ---> NeuronEsb.Adapters.Socket.SocketClosedException: The message cannot be sent to the TCP socket because the socket has been closed.

MOD – SMTP Adapter - When attaching files via file path, the full path and file name would be used as the visible attachment name (with non-valid chars replaced by underscores) rather than the filename minus the path.

MOD – SFTP Adapter – If the SFTP server was configured to auto delete downloaded files, the adapter would fail to publish the message. It would instead log the message to the Failed Message database and throw a file does not exist error. The behavior has been modified to accommodate servers that are configured to auto delete.

MOD – CRM 2011 Plugin – The plugin would throw an error if the FETCHXML activity returned an empty result set. This has been modified. Now the empty result set will be returned.

MOD – ODBC Adapter - Added support for setting parameters to NULL in the ODBC adapter. This is accomplished with a new attribute on the Parameter element (query and execute) or Field element (batch) called nillable. When you set this to true and then set the value to "null", the column will be set to null.

Neuron 3.5.4.860 Change History

Neuron ESB Explorer

FIX- Database Update – Neuron ESB would only present the user with the message that the database needed to be updated, it did not present and log what version it found in the database, and what version the installation required though.

Monitoring

FIX – Endpoint Health – In some circumstances if a failure occurred within the heartbeat of an endpoint, the endpoint may still be reported as Started in endpoint health, even though the heartbeat timestamp would remain unchanged.

Workflow Activities

MOD – Service Endpoint - Fixed issue for service endpoint step/activity where it wouldn't throw an exception when a SOAP fault is returned as the body of the message if the “Throw Exception on Fault” was set to true. Previously it only threw an exception if the outbound call threw a Fault Exception.

Neuron ESB Service

FIX – Single Instance Support - If an endpoint was marked as single instance, on startup Neuron ESB would not log either way whether the endpoint would be started or not started. Now, if the endpoint is eligible to start, Neuron ESB will log a verbose message. If the endpoint will not be started because another endpoint is active on another machine, a warning message will now be logged.

There could be a case where another machine could start up an Active Instance, however, the previously active machine would still register as Started in Endpoint Health. If Heartbeat failures occurred, the failing active instance may not have been stopped correctly.

If the secondary server detected that the current active instance on another server became stale, it may not take ownership and start up an active instance if the endpoint was a Service Connector.

FIX – Logging on Startup – Extra logging has been added to capture exceptions in the Neuron log file if an error occurred on startup.

FIX - Availability Group - High Availability/Failover - When Running an Availability Group on more than one machine, and using the Failover feature of Availability Group, the Availability Group could at times run on both Primary and Failover Machine both.

FIX - Availability Group - High Availability/Failover - When running an Availability Group on more than two Machines, where MachineX and MachineY are Marked as Primary and MachineZ is marked as failover, MachineZ would only monitor one of the primary machines for failover. If the other Machine which was not monitored failed, MachineZ would fail to detect the event.

FIX - Availability Group - High Availability/Failover – If after an Availability Group starts up on a Failover machine a user tried to manually restart the Availability Group on a primary server, the Availability Group would end up running on both primary and failover server.

Business Processes

FIX – Auditing – When testing the Audit Process Step within the Business Process on a machine that is NON US localized, a *SqlDateTime* overflow exception could occur, preventing the message from being audited

MOD – JSON - Modified the remove namespace function so that the JSON Newtonsoft namespaces and attributes are not removed so that they can be used to further control serialization. For example, this allows users to embed Newtonsoft directives to control the serialization of elements into arrays i.e.

```
<person xmlns:json='http://james.newtonking.com/projects/json' id='1'>
  <name>John</name>
  <url>http://www.google.com</url>
  <role json:Array='true'>Admin</role>
</person>
```

MOD – Service Endpoint - Fixed issue for service endpoint step/activity where it wouldn't throw an exception when a SOAP fault is returned as the body of the message if the “Throw Exception on Fault” was set to true. Previously it only threw an exception if the outbound call threw a Fault Exception.

FIX – Split/Join Process Step – Added right click “Edit Code” context menu for Join section which will bring up code editor.

FIX – Code Editors – If uses opened the Code Editors exposed by some Process Steps via the Ellipsis Button in the property grid, any code entered would not be saved appropriately.

FIX – Split/Join Process Step – When setting the Split section to Code and setting the Async property to True, a cast object exception would be generated. This was a bug introduced with the new parallelism features introduced in build 3.5.4.811

Topics

FIX – TCP Topics - Fix for using TCP topics with port sharing and the Client API when hosted in Windows applications. Previously, this worked when the client API was hosted in non Windows applications (i.e. console applications for example). In Windows applications, like the Neuron ESB Test Client, a delay would be incurred when connecting. The Test Client would later show it was disconnected and then would turn to a Connected state.

Neuron 3.5.4.839 Change History

Business Processes

FIX – Routing failure - when either a XSLT transform, JSON, Call service, Table or XML Query or Excel or Property Setter process steps are used on the subscribe side BEFORE the resulting message is published back to the bus, if the originating message was a request, the reply message would not get properly routed back to calling client. This was a breaking change introduced in 12/5/2015.

FIX – Parallel Step - Could throw *KeyNotFoundException* if any branch had a condition that returned false

FIX – Execute Process Step – If using the Execute Process step in a Client Connector where concurrent users could be executing and the Execute Process step is being set dynamically, the incorrect Process could be executed. Added verbose tracing as well.

FIX – Flat File Parser – When multiple definitions are used by the same party, only the first one is applied. If a publisher has one or more processes with multiple flat file parser steps defined with different column mappings, the engine would only use the first mapping executed. All other instances of the process step will try to use the first mapping, regardless of how they are defined in the UI. Now Neuron ESB will use the name of the Process Step to identify the name of the flat file mapping definition to be executed.

Neuron ESB Explorer

FIX – Rabbit MQ Management - Purge messages in the Neuron ESB Explorer's Rabbit MQ management screen would return an HTTP error if something other than the default '/' was used as the vHost.

Database

FIX – Updated Database Scripts – There were some stored procedures that were being created in the current user's schema rather than in the dbo.schema

Service Endpoints

FIX - Client Connector – The default timeout for a request/reply call would default to 397 seconds. This has been reset so that it now uses the Request/reply send timeout on the service policy multiplied by the retry factor.

FIX – Null Exception when Shutting down Client Connector – A null exception could occur if shutting down a publishing endpoint while it was still waiting for a reply message. Modified and enhanced logging. When shutting down client connector a null exception could also occur if Neuron attempted to audit the original request message. A null exception could also occur when attempting to resubmit a message directly to a service endpoint.

MOD - HTTP Basic authentication - When a Service Connector makes an initial call to Java-based REST services using Basic Authentication, the Service Connector could return a failure. This could occur because the service being called did not conform to the HTTP Basic Authentication RFC which requires the service to first return a negotiate header which tells the client what type of credentials to send. Now, Neuron ESB will send the Authorization header to the service on the initial call when the REST binding is configured with Transport:Basic.

MOD – Authentication Behaviors – Added verbose tracing output when Client Connectors are configured using either Username, NTLM or Credential based authentication options.

FIX – Swagger Support - If a user switches from SOAP (which exposed WSDL) to REST and chooses to apply Swagger the following error would result:

Could not find a base address that matches scheme http for the endpoint with binding MetadataExchangeHttpBinding. Registered base address schemes are [].

MOD – Policies - Modified the availability timer that controls activating an endpoint based on Policy, so that if an error occurs attempting to start it up based on the schedule, Neuron ESB doesn't continually attempt to start it in an endless loop. if it fails to start, the state will be set to Failed.

ESB Service

FIX – INFO level Logging – In some cases INFO level tracing would be outputting in the Neuron ESB log files whether or not INFO level logging was configured.

FIX – Memory Leaks – Fixed several memory leaks that could occur on shut down of endpoints or reconfiguration of endpoints and Parties

Adapters

FIX – Socket Adapter – Fix for a performance delay between each message being read from the socket. Also added better exception handling to report that a message could not be sent because the underlying socket was closed.

FIX – Socket Adapter - When using the Client and Server socket adapters to facilitate cross zone (i.e. DMZ to internal network) communication, any custom properties, HTTP and SOAP properties were not getting forwarded back to the original caller in a request/reply message scenario.

FIX – Metadata Generation with Salesforce Adapter – OAuth properties could be displayed after unchecking OAuth as an option.

MOD – Dynamics XRM Adapter – This now supports Bulk Upload of the SaveCrmRecord. This significantly reduces the number of round trips when sending multiple records to CRM.

FIX – SQL Adapter - An object disposed error would occur when shutting down a non-publish mode SQL Adapter Endpoint.

MOD – Policies - Modified the availability timer that controls activating an endpoint based on Policy, so that if an error occurs attempting to start it up based on the schedule, Neuron ESB doesn't continually attempt to start it in an endless loop. If it fails to start, the state will be set to Failed.

Topic Transports

FIX - Named pipes and TCP channel - If an error occurred creating the host and that error was NOT an *AddressAlreadyInUseException*, then the connect process to a TOPIC could end up in an infinite loop.

FIX – TCP Topics - Fixed a bad error message that's recorded when Neuron ESB can't assign a port when the Topic starts. The topic ID rather than the Topic name was being reported. i.e. **System.Exception: Unable to assign the '50011' port to the '8cd30384-fb30-43d9-a50d-6eaea4f050d8' TCP based Topic. The given key was not present in the dictionary.**

Neuron 3.5.4.811 Change History

Business Processes

NEW – Cancelling a Test in Business Process Designer – This feature has been added. If a user starts a test within the Business Process Designer, the test can now be stopped by clicking the “Stop Debugging” red button on the toolbar. This will work even when using the Split Process Step.

FIX – Split Process Step – Running load through a multi-threaded publisher could cause a null object exception.

FIX – Flat File Wizard – Fixed crash when setting fixed width column width value to a non-integer value.

FIX – Flat File Wizard – When fixed width sample file is smaller than the preview grid - user couldn't set width markers.

FIX – Flat File Wizard - fixed issue with optional field checkbox change on last column not being saved if Finish button clicked after changing value.

NEW - JSON Process Step - Added the ability to represent elements that have “nil = true” elements as null or empty string outputs. The JSON Process Step also now exposes the *JsonSerializerSettings* settings, *DefaultValueHandling* and *NullValueHandling*.

Services

FIX – HTTP Client Utility – Did not support HTTPS based Urls.

FIX – Service Connector Endpoints - Fixed how we resolve local address and To addresses when dynamically setting the “Addressing.To” property and Localpath property. If the hard coded service address of the service connector URL had a trailing backslash AND both the “Addressing.To” and Localpath were being set dynamically, their values would be placed together without a slash separating them to form the URL.

NEW – Multiple Instance support for dedicated IP addresses – Support has been added for running multiple instances on the same machine, while dedicated unique IP addresses to each instance. There is now a new property on the Binding tab of the Service Endpoint named, “Check HostName Comparison Mode”. Support is disabled by default.

Topic Transports

MOD – Rabbit MQ – The Rabbit MQ Topic transport has been modified to use background rather than foreground threads for IO.

FIX – Rabbit MQ – Fixed concurrency issues that could cause null object exceptions when publishing load from multi-threaded publishers.

FIX – TCP/Named Pipes – An *ObjectDisposedException* could be reported when shutting down TCP/Named Pipes based topics.

Neuron ESB Workflow

FIX – Port Sharing - If port sharing was enabled and the instance name for Neuron ESB runtime was not “Default”, any workflow endpoints would fail to start up with the following error:

Workflow Endpoint '<nameOfEndpoint>' configured using the '<nameOfWorkflow>' Workflow Definition could not be started. Neuron.Esb.ESBException: The Party object failed to connect to the ESB service. The "<nameOfEndpoint>" service could not be started. ---> System.ServiceModel.CommunicationException: Unable to connect to the Neuron Configuration Service at localhost:50000. There was no endpoint listening at net.tcp://localhost:50000/DEFAULT/ESBConfigurationService/ that could accept the message

FIX – Logging Error - Fixed a bad format string error when logging that the workflow host is queuing a workflow commands to the database queue.

FIX – Logging Issues - If a workflow was terminated using the Terminate Workflow Activity, only the reason property would be reported in Neuron logs. Added logging to the Retry step so that the specific failure is written to the Neuron log file. Previously this was only logging a generic message that a child process failed or faulted. Logging for the Publish, Publish Request and Execute Process Workflow Activities were not logging to the Neuron log. If a workflow was aborted the error information was not being logged the Neuron log files.

FIX – Adapter Endpoint Workflow Activity – This was updated to output log messages from the adapter.

NEW – Workflow Tracking - implemented Select All for context menu.

FIX – Workflow Tracking – When querying workflow records and if an error occurred due to database contention, a random error dialog may appear after the primary error dialog gets displayed.

Neuron ESB Service

FIX – Failure to Shutdown – Under certain circumstances, specifically heavy load, when the Neuron ESB Service is being shut down, the Neuron ESB Service and Workflow runtime service may appear to fail to shutdown.

FIX – Memory Leak – If “Use Remote TCP Clients” was enabled in the active Deployment group, SQL Connections could be leaked, causing memory for the Neuron ESB Service to escalate.

FIX – Memory Leak – Fixed a possible memory leak when Auditing Service is used.

FIX – Activity Sessions Reporting - Under heavy load the following error could be reported:

"Failed to restore session activity state" - SqlError in - 233 Severity 20 - A transport-level error has occurred when receiving results from the server"

Neuron ESB Explorer

FIX – Rabbit MQ Management – A null object exception could be reported in the header of the screen.

MOD – Message Viewer – When viewing the individual messages within the Message History or Failed Message reports using the Message Viewer, the messages will be automatically formatted as the user navigates from one message to the next.

Adapters

NEW – Rabbit MQ Adapter - Added correlation id and content type as configurable properties in subscribe mode.

FIX – Socket Adapter – If invalid data was sent to the adapter by an outside program that maintain a connection to the socket, the adapter would stay in an endless loop waiting for more data. Now, if data stops arriving over the socket, the receive loop will terminate and close the client socket

FIX – Salesforce Adapter - Fixed error that could occur if multiple threads are attempting to authenticate and return a token at the same time. When this happens occasionally the following error could be reported:

Exception: Exception Type: System.Exception

Exception Message: There was an error while obtaining an OAuth token from Salesforce.com - {"error":"invalid_grant","error_description":"authentication failure"}

MOD – Salesforce Adapter – modified logging

FIX – Salesforce Adapter Timeouts – When using the Salesforce adapter from the Adapter Endpoint Workflow Activity moderate load, Timeout exceptions could occur due to internal deadlocks. Also, if an error did occur, the ESB Message's HTTP StatusCode, StatusDescription, Content-Length and Content-Type properties were not being properly set.

FIX – Dynamics XRM Adapter - When executing a SetCrmState command, a **KeyNotFound** error would occur. This is because the underlying response did not contain the ID of the changed entity. Regression error introduced in 3.5.4.

Neuron 3.5.4.777 Change History

Workflow

FIX – Availability Group Failover – If an Availability Group has a primary and failover node configured and the primary node fails, the workflow endpoints would start up and continue processing on the failover node as expected. However, when the primary node was restored, workflow endpoints would begin executing on both nodes, rather than only on the primary node. This has been corrected.

Reporting

FIX – Using Neuron ESB Auditing causes *SqlDateTime overflow* exception – When attempting to store a date time value in Australian format (e.g. 13/09/2016), the Neuron ESB Auditing service could throw the following exception:

SqlDateTime overflow. Must be between 1/1/1753 12:00:00 AM and 12/31/9999 11:59:59 PM.

Neuron ESB Explorer

MOD – ODBC Connection Manager – Users can now select an individual ODBC driver from a dropdown list. When selected, this will enable a load button which can be used to query and load the specific connection string keywords from the selected driver.

Services

FIX - HTTP Client utility in Workflow and Business Process Designers. – If the base path was included in the address entered into the Address bar of the HTTP Client utility, the service endpoint would report an endpoint not found exception, effectively it would be appending the localpath value to the entire

service address. Also, spaces either before or after the addresses would cause the service endpoint to be set with an address that appeared to be duplicate appended to one another.

FIX – Datagram Client Connectors – A *System.NullReferenceException* exception could occur from our dispatcher.

Adapters

FIX – MQSeries Adapter – If an exception occurred during the initial connection to MQSeries, the errors were not being properly propagated to the monitoring system. Also, there could be circumstances where the original invalid state of the MQSeries connection would not be properly reset, causing the same connection error to be reported even though a successful connection could be made.

FIX – NetSuite Adapter – In some cases, if multiple NetSuite adapter endpoint steps are used within the same Business Process all subsequent NetSuite adapter endpoints would inherit the properties of the first adapter endpoint.

FIX – SMTP Adapter – When attempting to use a code step to attach 4 or more files, the SMTP adapter could throw the following exception:

System.ObjectDisposedException: Cannot access a closed file

Topic Transports

FIX - Rabbit MQ monitoring – Under the Deployment- Manage - Rabbitmq section of the Neuron ESB Explorer, the message stats for publishers were not being reported. Also, reporting always assumed the default vhost was configured. If a different vhost was configured this could result in a duplicate key exception being reported. Lastly, incorrect metrics were being reported for subscribers.

FIX - Rabbit MQ monitoring – Under the Endpoint Health monitoring section of the Neuron ESB Explorer, reporting always assumed the default vhost was configured. If a different vhost was configured and used, only the metrics for the default would appear.

Business Processes

NEW – Flat File Wizard – Users can now mark a column as optional within the flat file wizard.

NEW – Flat File Wizard – A new feature has been added that allows users to create flat file definitions without requiring a sample flat file. This feature is useful when creating definitions for flat files with optional fields or when creating definitions for delimited flat files with quoted fields and the quoted field will contain the delimiter. For example, using a comma-separated flat file with these three fields: *1234,Neuron,"Neudesic, LLC"*. If you use a sample file with this data, the parser will read that as four fields. You will instead need to select the new “User Defined” option and manually add the fields:

Neuron ESB Flat File Parser

Fixed-Width Flat File
Select a sample fixed-width file to guide you through the wizard.

Select sample test file: ☒ **User Defined**

☐ Repository

☐ File System ...

☐ First Row Contains Column Names

Ignore first Lines

Ignore last Lines

10 20 30 40 50 60 70

Cancel < Back Next > Finish

After clicking Next, you can right-click in the Columns box and select “Add” to add the first column:

Neuron ESB Flat File Parser

Set the column details
Verify the name and width of each column

Columns

Add

Column Properties

Name:

Start Index:

Width:

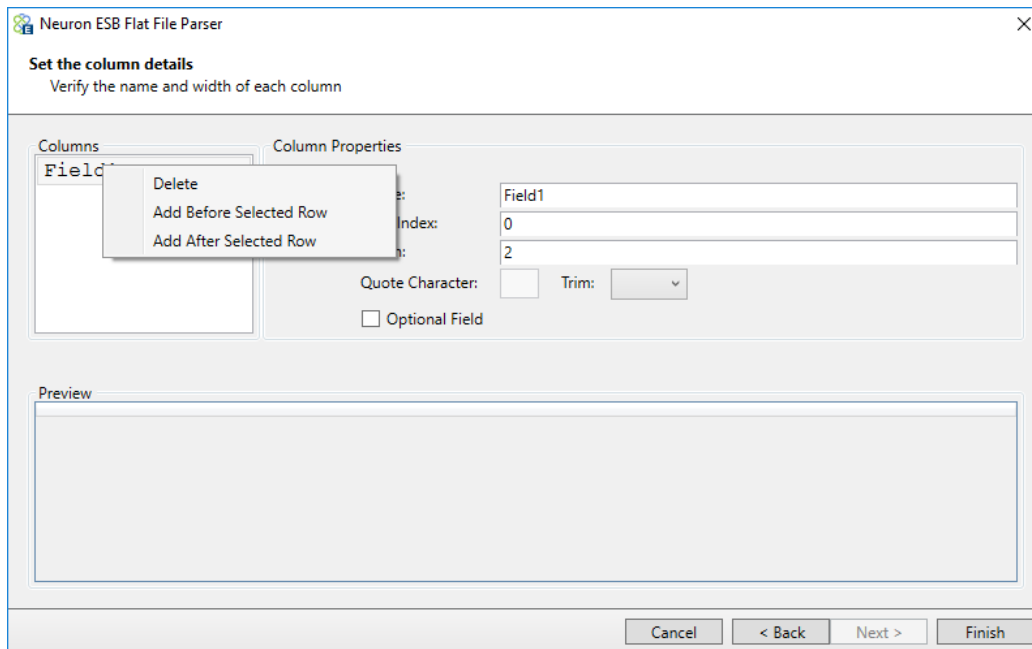
Quote Character: Trim:

☐ Optional Field

Preview

Cancel < Back Next > Finish

After adding the first column, you can right-click and “Delete”, “Add Before Selected Row” or “Add After Selected Row”:



FIX – TableQuery Process Step returns XmlException – When querying a table that has column names that include spaces, the TableQuery Process step could generate the following exception when attempting to serialize the results. Neuron will now replace any spaces in column names with underscores.

System.Xml.XmlException: '>' is an unexpected token. The expected token is '='.

Neuron 3.5.4.762 Change History

Adapters

MOD – SFTP Adapter – Using a relative path such as “../../another/home/input” was not previously supported when the Adapter Endpoint was placed in Publish mode. This is now supported for both the Archive folder and SFTP folder paths.

MOD – ODBC Adapter – Previously, when setting the FORXML property to true in publish mode, the ODBC driver used had to be the SQL Server Native Client 11.0. This feature has been extended to now support any ODBC driver for SQL Server.

MOD – ODBC Adapter – ODBC Database Connection Dialog – Drop down box would take a very long time to populate the list of ODBC drivers because we were previously querying each driver for their connection string keyword values that they supported. We no longer do that, hence populating the list of drivers to use is instantaneous now.

Business Processes

NEW – Flat File Wizard – Support to mark columns/fields as optional has been added

Neuron 3.5.4.757 Change History

Adapters

FIX – SFTP Adapter – When configured to poll against some non-compliant SFTP servers, the adapter endpoint may report an error similar to the one below in the Event Log which would stop the adapter:

The current app domain for the '<Name of Endpoint>' endpoint threw an Unhandled Exception.
Exception: Thread was being aborted. Method: ReadThread Stack: at
Xceed.SSH.Core.SSHIOManager.ReadThread()

To correct this, the Xceed libraries have been updated from 5.8 to 6.0.

Neuron 3.5.4.756 Change History

Business Processes

FIX - Flat File Parser Wizard – The flat file parser could not parse a semicolon-delimited file.

Furthermore, if the file had headers in the first row and there were repeating header name values, they were not supported. Support has been added for semicolon-delimited files. Also, repeating header names will be processed by taking the subsequent occurrences of headers and appending an underscore and a number. i.e. name, name_1, name_2, etc.

FIX - Flat File Parser Wizard – Column names were not being populated correctly if the user navigated backwards in the wizard, checked the "First Row Contains Column Name" box, and then navigated forward to the next screen the preview pane would list the correct column names, but the actual names of the columns would be "Field1", "Field2", etc.

FIX - Flat File Parser Wizard – The column names sometimes do not match up to the actual name of the column if underscores are present in the column name. In "Set the column details" page of the Flat File Parser Wizard, sometimes the name column name in the preview pane does not match up with the actual column name that the user enters in the "Name:" text box. The wizard would remove the first underscore in a column name unless it's at the end of the name

FIX - Flat File Parser Wizard – An expected exception could occur at design time, causing the Neuron ESB Explorer to crash if both the Vertical Bar delimiter for a flat file type and "First Row Contains Column Name" check box was selected before loading a sample file to parse.

FIX – Flat File Parser Wizard - The flat file parser was recreating the engine with every call. To improve the performance while using the parser within a split step we now reuse the engine during execution.

FIX – HTTP Client Utility – The value provided in the “Request URL” text box was not being used correctly to set the Full URL address of the Service Connector that would eventually be called. If only a local path value was given, that would correctly. Now both work.

FIX – HTTP Client Utility – Attempting to use an Environment Variable to set the value of the “Request URL” text box would result in a “Key not found” exception. This has been fixed

FIX - When Using ellipses to edit the code in Process Steps such as the For, For Each, While, Split/Join, Decision, Publish, or Parallel, the code entered would always revert to the previous saved version.

FIX – JSON Process Step - When converting from JSON to XML, if a root was specified and there was only one element in the JSON, the root would not be applied. Also, if an xml declaration was set to be added on the conversion (JSON to XML) it would not be applied.

Repository

FIX – Text Documents - When importing a text document with extended ascii characters, the characters display as invalid in the UI. However, if copying and pasting the same characters from Notepad++ into the text repository they would display correctly.

Services

FIX – Swagger – When using the meta data dialog of the Client Connector to assign a Swagger document to the Client Connector endpoint, if the user later disables it, the setting would remain enabled.

FIX – Swagger – When moving a Neuron Solution from one environment to another (e.g. QA to Production), the Swagger documentation that was previously accessible from the Client Connector url in the first environment would be inaccessible in the new environment.

FIX – Swagger – if a Swagger enabled Client Connector URL was configured to use a port, the Swagger documentation would be inaccessible.

FIX – TLS Security Support - When attempting to resubmit a message to a service endpoint that only supports TLS 1.1 or higher, or to a Salesforce adapter endpoint, or when testing within the Neuron ESB Explorer or Test Client, the user could receive the following exception:

System.ServiceModel.FaultException: Unsupported_Client:TLS1.0 has been disabled in this organization

Monitoring

MOD - When availability groups or workflow endpoints are stopped, no messages were written to the Neuron ESB log file indicating that the availability group or endpoint was being stopped or has stopped.

Workflow

MOD – Code Editor - When editing a workflow code activity, the "SourceCode" property was displayed in the property grid with an edit button. Clicking on the edit button would open the editor tab, but the source code would be erased. This property has been removed. Users can use either the inline workflow editing or the right-click context menu's "Edit Code" menu option to open the Code Editor for editing.

FIX – C# Class Workflow Activity - Customers upgrading existing workflows that use the C# Class workflow activity to CU4 will encounter “Parameter mismatch” exceptions at runtime. In CU4, we added an additional parameter to the template to allow for logging which broke backwards compatibility. This has been corrected.

FIX – Workflow Endpoints Not Starting after Database failure - During the execution of workflows, if the Neuron ESB Database became unavailable due to a SQL Server or Network issue, the workflows would

remain in a reported Executing state. Once the Neuron ESB database became available again, the workflows would continue to remain in an Executing state within the Workflow Tracking system, appearing as if they were "stuck" and never completing. New messages would not be processed. Instead they would build up in the pending messages queue. This occurred because the system believed that the max concurrent workflows limit had been reached because the terminated workflows were left in an Executing state. The only way to resolve this would be to restart the Neuron ESB Service. This has been resolved. The Neuron ESB service no longer needs to be restarted for the new messages to be processed.

Neuron Explorer

FIX – Generation of MSMQ PowerShell Scripts - When this script was generated, ALL the lines were commented out. Now the script will check for the existence of the Queue before creating it and assigning permissions. Also, verbose and warnings are logged to the console.

Adapters

FIX – Salesforce Adapter – Once an Oauth token expired, the adapter would not attempt to renew the token and instead would log the following error:

The received status code, "Unauthorized", is not a success status code. The reason phrase is "Unauthorized". The response message is:

```
[{"message":"Session expired or invalid","errorCode":"INVALID_SESSION_ID"}]
```

This was a regression bug introduced by CU4 and has been corrected.

FIX – Salesforce Adapter – Any HTTP non-success code would result in an exception getting thrown back from the adapter. This was a regression bug introduced by CU4 and has been corrected.

FIX – Salesforce Adapter – If the REST method was not set, the adapter would only report a Null Exception rather than reporting what was missing and must be fixed.

FIX – Salesforce Adapter – Users could experience the following exception if non-compliant HTTP Headers were forwarded as part of the request message to the Salesforce Adapter Endpoint:

The underlying connection was closed: Could not establish trust relationship for the SSL/TLS secure channel.

Other exceptions, such as "Misused header name" could also occur. This could happen if the request message sent to the Salesforce adapter originated from a REST client sending a request to a Neuron ESB Client Connector and that request contained non-compliant HTTP headers. The Salesforce adapter has been updated to automatically remove these headers before forwarding the request:

- Accept-Encoding
- Accept-Language

- Accept
- Host
- Cache-Control
- Origin
- User-Agent
- Content-Length
- Location
- Sforce-Limit-Info

Neuron 3.5.4.730 Change History

Workflow

FIX – WMI Performance Counter memory leak – When starting up Workflow endpoints, the following error may be reported:

“Workflow Definition could not be started. System.InvalidOperationException: Custom counters file view is out of memory.”

Business Processes

FIX – Audit Process Step – When running in Design mode, user may encounter the following error:

Unable to cast object of type 'Neuron.Esb.EsbService.MessageAuditProxy' to type 'System.ServiceModel.ICommunicationObject'

This was a bug introduced in CU4

FIX – JSON Process Step – When attempting to convert an Array into an XML document, the following error may occur:

XmlNodeConverter can only convert JSON that begins with an object.

This was a bug introduced in CU4

Neuron 3.5.4.727 Change History

Business Processes

MOD – Transform, Set Property, Call Service, XML Query, Table Query, Excel to Xml and Json Process Steps – Previously, these steps would maintain the current Message ID on the message that resulted from their execution. This has been changed so that they will now generate a new Message ID, while setting the Parent Message ID property with the value of the original Message ID.

FIX – Transaction ID – When creating an ESB Message, the Transaction ID would only be sent once a message was published. If a user audited the message before it was published, this value would be blank. The property is now being initialized during the creation of the message.

FIX – Adapter Endpoint Process Step - If an adapter endpoint called by the Process Step is in subscribe mode and has its adapter policy configured to "<None>", the message would not get processed by the adapter. This has been corrected

FIX – HTTP Utility - The Clear Message Body check box was not getting saved if checked.

Service Endpoints

FIX – REST based Client Connector – When receiving messages published through a REST based Client Connector, the uncompressed message size header property would not be set.

Adapters

FIX – Azure Service Bus Adapter - The Azure SB adapter restricted the message size to be greater than 0 (no empty messages) and less than 256 KB. These were the original limits from Azure Service Bus. Now Azure SB Adapter supports empty messages and has different size allowances based on subscription tier.

Neuron 3.5.4.719 Change History

Business Processes

FIX – Audit Process Step – When testing a process within the Neuron ESB Explorer Process Designer, if the Audit Process Step generated a fault to the Audit Service, it would not be recreated.

FIX – Compression Process Step – When attempting to use this the following error would be generated:

Aborted: Could not load file or assembly 'Xceed.Zip.v5.8, Version=5.8.16152.0, Culture=neutral, PublicKeyToken=ba83ff368b7563c6' or one of its dependencies.

Service Broker

FIX – Swagger Support – If a Client Connector was enabled to expose Swagger, the Swagger URL hosted by Neuron ESB would not be available from a remote machine. For example, if the Neuron ESB Client Connector URL was <http://mymachinename:9192/Users>, then the <http://mymachinename:9192/Users/Help> URL that would normally redirect users to the Swagger documentation, would generate a 404 error.

FIX – HTTP.Query – When receiving inbound REST calls, the HTTP.Query object would not contain query elements in the inbound URL if present. This was a bug introduced with 3.5.4.700 (i.e. CU4).

Workflow

FIX – Cluster Failover Support – When using Availability Groups, the server designated as a “Failover” server would host and run workflow endpoints even if the designated “Primary” servers have not failed.

FIX – Cluster Failover Support – When a primary server would fail, any dehydrated workflows would not automatically startup on the designated “Failover” machine.

FIX – Referenced Assemblies – if Assemblies were added as a reference at the Workflow designer level, they would not be inherited and used by the Code Workflow Activities such as C# or C# Class. The user could receive a compile error when trying to use them. This bug was introduced with CU4.

Neuron Explorer

FIX – Neuron Explorer – File Open Dialog - If either the Configuration Server dialog was set with a folder (either local or UNC path) that was no longer available, or the last solution opened pointed the folder that was no longer available, a user would see the error dialog indicating that the "Network Name Cannot be Found" error. However, the open file dialog box would never be displayed, functionally preventing the user from either opening a new solution using the File Open menu, or changing the runtime configuration by using the Configure Server dialog box. If the folder is no longer available, the open folder dialog box will now be displayed using the My Documents folder location as the default. Also, upon closing the Explorer, the old path will be removed from the MRU list.

FIX – Endpoint Health – Workflow - If a user had a primary server and a failover server defined in the Availability Group, the endpoint health would say that both the primary server and failover server were started. The primary server column for the failover server would also wrongly state that the failover server is the primary server for the failover server (i.e. the failover server listed itself as the primary server).

Adapters

NEW – ActiveMQ Adapter – Added the “Time to Live” property for Subscribe mode endpoints to control when a message sent by Neuron ESB to an ActiveMQ queue will be moved to the ActiveMQ dead letter queue.

FIX – SFTP/FTP Adapter – Warnings could be generated stating that the log file could not be deleted when adapter starts up. This can be ignored.

FIX – Adapter Policies – If an Availability Window was activated for an adapter while the adapter was in the process of processing a message, the message maybe discarded rather than rolled back.

Database

FIX – Updating database causes error – When attempting to update an existing database to use Neuron ESB 3.5.4 (CU4) the following error could be generated:

Event Info: Message Audit Service

Exception: Violation of PRIMARY KEY constraint 'PK_MessageAudit_AllTopics'. Cannot insert duplicate key in object 'dbo.MessageAudit_AllTopics'. The duplicate key value is (835380).

The statement has been terminated.

This was due to a bug introduced into the PurgeAndBackUpNeuronDatabase database sql agent job.

Neuron 3.5.4.700 (CU4) Features

Adapters/Connectors

The Neuron ESB CU4 includes a number of new adapters/connectors. Adapters are key piece of capability in the world of an integration broker. They serve as the bridge to and from the bus between

applications, databases, transports and protocols. The completeness of “what” you ship as well as how easy it is for others to build their own adapters is critical in accelerating the development of any solution. Not only is Neuron ESB hands down one of the easiest adapter frameworks to learn in the industry, it also has pre built adapters for most commodity transports, queuing technologies, databases and on premise/cloud applications. However, we continue to deliver a more complete and richer set of connector with every release. This update is no exception.

Certainsafe

Neuron ESB’s CertainSafe Adapter is the culmination of a partnership between the two companies. The adapter enhances your efforts to securely store and transmit encrypted secret data while leaving mundane information in clear text form. It does so by simplifying the steps you employ to interact with your CertainSafe Digital Vault from your Neuron ESB based integration infrastructure. Together Neuron ESB and CertainSafe enable automated data security for databases, files, processes, messages and API communications.

The Neuron ESB CertainSafe adapter functions in a Solicit/Response pattern in that the adapter listens for messages published to the bus based on its Topic. Those messages can contain data bound for secure storage and encryption by the CertainSafe service or tokens that map to previously secured data. The response from the CertainSafe Digital Vault service is a new token in the former case and the returned data in the latter. For more information about CertainSafe can be found here:

<https://certainsafe.com/>.

More information can be found here: <http://www.neuronesb.com/resources/?article>

Zuora

Zuora is an enterprise software company that designs and sells SaaS applications for companies with a subscription business model. The new Neuron ESB Zuora adapter enables you to easily embed any or all of Zuora’s commerce, billing and finance Software as a Service products into your own subscription-based services. The adapter turns your Neuron ESB integration infrastructure and all of your connected services into relationship business management dynamos that provide you the information you need to design, iterate and understand your subscription revenue streams.

The Neuron Zuora adapter functions as either a publish or subscribe endpoint to the bus. Using Publisher or Subscriber model, the Zuora adapter enables connectivity with the Zuora Rest API, through which businesses can perform operations such as managing subscriptions from a web storefront, retrieving user information, and synchronizing catalogs with ERP systems. Using the Zuora adapter users can easily configure integration with popular third party applications and systems such as NetSuite, Dynamics CRM and Salesforce.

More information can be found here: <http://www.neuronesb.com/resources/?article>

Marketo

Marketing in the brave new services world is different. Marketo excels at Lead Management, email, consumer, customer base and mobile marketing processes for your service based products. The Neuron

ESB Marketo adapter enables you to connect all of Marketo's Software as a Service product through your Neuron ESB integration infrastructure to your own services and backend systems. Modernizing your own marketing efforts couldn't be easier than with Marketo and the Neuron ESB Marketo adapter.

The Neuron ESB Marketo adapter can function as either a publishing or subscribing endpoint to the bus. Using the subscriber model, the Marketo adapter enables connectivity with the Marketo API, allowing users to interface with marketing data through API calls to add/update leads, manage campaigns, run analytics, and track opportunities. Marketo can also interact directly with the Neuron ESB Marketo Adapter via its WebHook feature (i.e. Publish mode). Using the WebHook feature, Marketo can send a Notification to Neuron ESB. Examples would be Customer Inactivity, Duplicate prevention and Sending Reminders for meetings prior to demo.

The Neuron ESB Marketo adapter allows your businesses to integrate Marketo with existing ERP, eCommerce, and CRM applications such as NetSuite, Salesforce and Dynamics CRM.

More information can be found here: <http://www.neuronesb.com/resources/?article>

Twilio

Twilio's popularity as a communications Infrastructure as a Service provider makes it an easy choice for any customer interaction scenario that includes in-app VOIP/Telephony or text/picture messaging. The new Neuron ESB Twilio adapter and your Neuron ESB integration infrastructure enables you to quickly add these feature sets to your own services and products.

The Adapter provides easy accessibility to Twilio's REST API and it supports both publish and subscribe modes. Solicit/response and request/reply patterns are also supported. This enables you to build customer communication scenarios that include your own internal processes in the communications patterns that your customers expect. For example, you can fire off processes that are initiated from customer phone calls or text messages or respond to any event with direct phone calls or texts out to your customers. Neuron ESB and Twilio turn your business systems into customer retention tools.

More information can be found here: <http://www.neuronesb.com/resources/?article>

Apple Push Notification Service

The Apple Push Notification Service adapter allows your Neuron ESB solutions to interact with iOS mobile devices (iPhone, iPad). Using the adapter, Neuron ESB solutions can send alerts and notifications to iOS devices, set application icon badges, and play sounds. The adapter can also be used to receive notifications from the Apple Feedback Service when messages cannot be delivered to iOS devices. This is typically a result of the user unsubscribing from receiving notifications for an application.

More information can be found here: <http://www.neuronesb.com/resources/?article>

Amazon Web Services Simple Queueing Service

Amazon Web Services is the leading cloud hosting provider, and the SQS service lets Neuron ESB integrate with solutions running in the AWS cloud. The adapter supports publishing messages to an SQS queue or receiving messages from an SQS queue. The adapter can be used to pass work to batch

processes running in EC2 or Lambda, or to publish events and notifications from your AWS applications to other services that are integrated using Neuron ESB.

More information can be found here: <http://www.neuronesb.com/resources/?article>

Dropbox

Dropbox is a leader in cloud storage and now you can use Neuron ESB to upload files to Dropbox or download files from Dropbox. The Neuron ESB Dropbox adapter wraps the Dropbox REST API and exposes several operations around file management in Dropbox accounts. You can use the Dropbox adapter to archive messages safely to Dropbox for backup, syndicate documents to subscriber accounts, or import documents from subscriber accounts. You can also use the Dropbox adapter to copy files, move or rename files, delete files, or create folders in user Dropbox accounts.

More information can be found here: <http://www.neuronesb.com/resources/?article>

Rabbit MQ

The Neuron ESB Rabbit MQ Adapter allows Neuron ESB to interact with Rabbit MQ in a variety of ways, with configurable levels of Transaction support and Quality of Service. Organizations can use the adapter to monitor Rabbit MQ Topics or Queues for new messages, forwarding those (i.e. publishing) to a Neuron ESB Topic where other subscribing parties (e.g. Adapter, Service or Workflow Endpoints) can receive the information. Conversely, the adapter can also be used to forward information published to or generated by Neuron ESB to either a Rabbit MQ Queue or Topic, essentially making Rabbit MQ a Neuron ESB subscriber. Both types of Rabbit MQ Transactions are supported (e.g. Transaction and Publish Confirms) as are Virtual Hosts, SSL and all the latest features of Rabbit MQ.

More information can be found here: <http://www.neuronesb.com/resources/?article>

Active Directory

Neuron ESB's Active Directory (AD) Adapter simplifies efforts to make an organization's AD Server(s) part of a Neuron ESB-based integration infrastructure. Using the Neuron ESB Active Directory adapter, organizations can incorporate AD into their Business Process automation and management solutions to automate tasks involved in new hire and user/group provisioning scenarios. The Neuron ESB Active Directory adapter can function as either a publishing or subscribing endpoint to the bus. In Publish mode, the adapter can monitor an organization's AD server for specific changes/events (i.e. password change, deleted user, etc.), publish those changes to bus where they would be routed to all interested subscribing users or Business Processes/Workflows. In Query mode, the AD server can be queried for information, or existing users and groups can be updated, added or deleted. Users and Groups can also be enabled/disabled and passwords can be reset.

More information can be found here: <http://www.neuronesb.com/resources/?article>

Neuron ESB Microsoft Project Connector for NetSuite

With NetSuite's cloud-based project management solution, stakeholders can see, monitor and manage project status anytime, anywhere. By delivering complete visibility and control, NetSuite's project

management solution gives you real-time access to all of your project information. The result is a significantly improved project completion record, more satisfied clients and reduced non-billable work.

Microsoft Project makes project management easy. With tools designed specifically for team members, project managers, executives and the PMO, it enables collaboration from virtually anywhere with the right tools for any role.

The Neuron ESB Microsoft Project Connector for NetSuite enables you to sync projects and resources across both toolsets through your Neuron ESB integration infrastructure. When you've got NetSuite as your project management system of record and Microsoft Project's role focused editions in the hands of your stakeholders, synchronized with Neuron ESB, your business will provide instant access to critical project data anytime, anywhere for anyone.

More information can be found here: <http://www.neuronesb.com/why-neuron-esb/netsuite-integration/>

Dynamics CRM WEB API

The Neuron ESB Dynamics CRM Web API Adapter provides direct access to Microsoft Dynamics CRM 2016 Web API. Using this adapter, organizations can easily integrate their business applications such as SAP, NetSuite or Marketo with Dynamics CRM 2016, both on-premises and online. The adapter utilizes the new Azure Active Directory/ADFS OAuth Provider to manage access to Dynamics CRM. This adapter operates in both subscribe as well as solicit/response (or query) mode. Organizations can use the adapter to query for data, create, update, retrieve and delete entities and execute CRM Functions and Actions. The adapter can be used to access the entire Dynamics CRM Web API. When developing applications that utilize this adapter, developers will be able to use the Adapter Metadata Generation Wizard to generate JSON samples that describe the request and/or response documents that are exchanged with Dynamics CRM. Developers can also use the new Http Utility process step in our process designer to very easily define the calls to Dynamics CRM.

More information can be found here: <http://www.neuronesb.com/resources/?article>

OAuth Providers

The Neuron ESB 3.5 CU4 release introduces OAuth support for authorizing web service invocations using Service Connectors (i.e. HTTP/REST service endpoints) and supported Adapters. Using OAuth, Neuron ESB can obtain access tokens that can be passed to REST services to authorize protected resource invocations. In CU4, OAuth bearer tokens based on [RFC 6750](#) are supported.

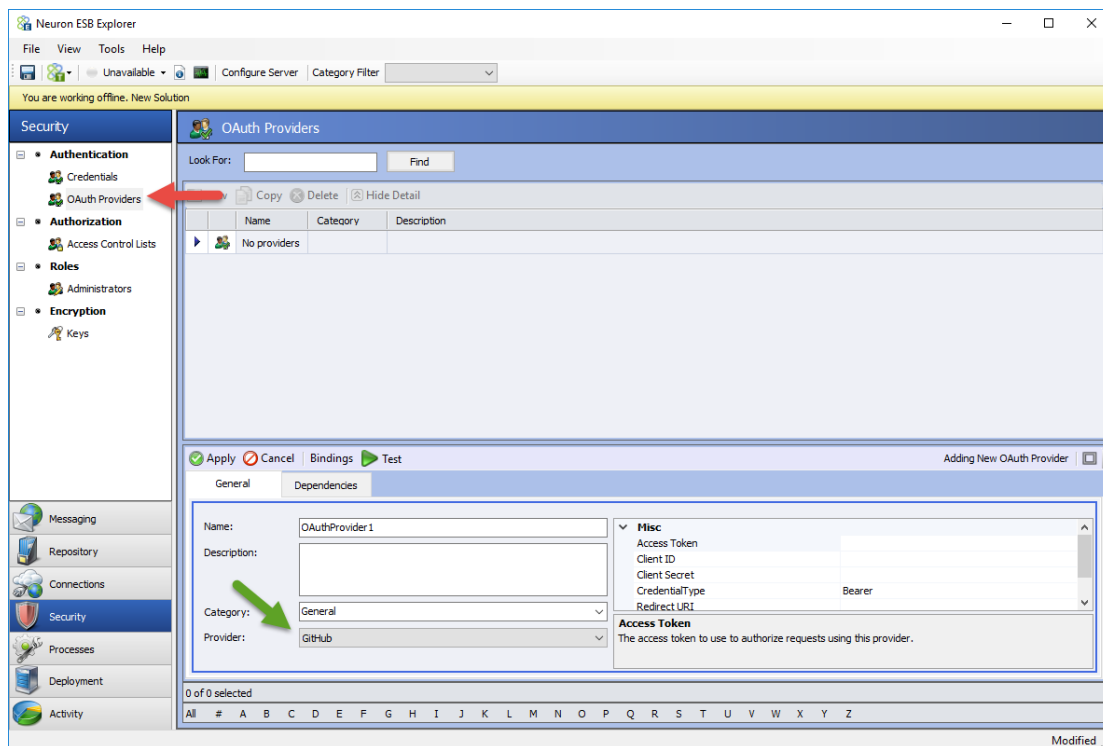
Neuron ESB supports obtaining OAuth access tokens using the following authorization services:

- Amazon
- Azure Active Directory/ADFS
- Dropbox
- Facebook
- Foursquare
- GitHub

- Google
- Instagram
- LinkedIn
- Microsoft Live
- SoundCloud
- SourceForge
- Tumblr
- Twitter
- Yahoo!

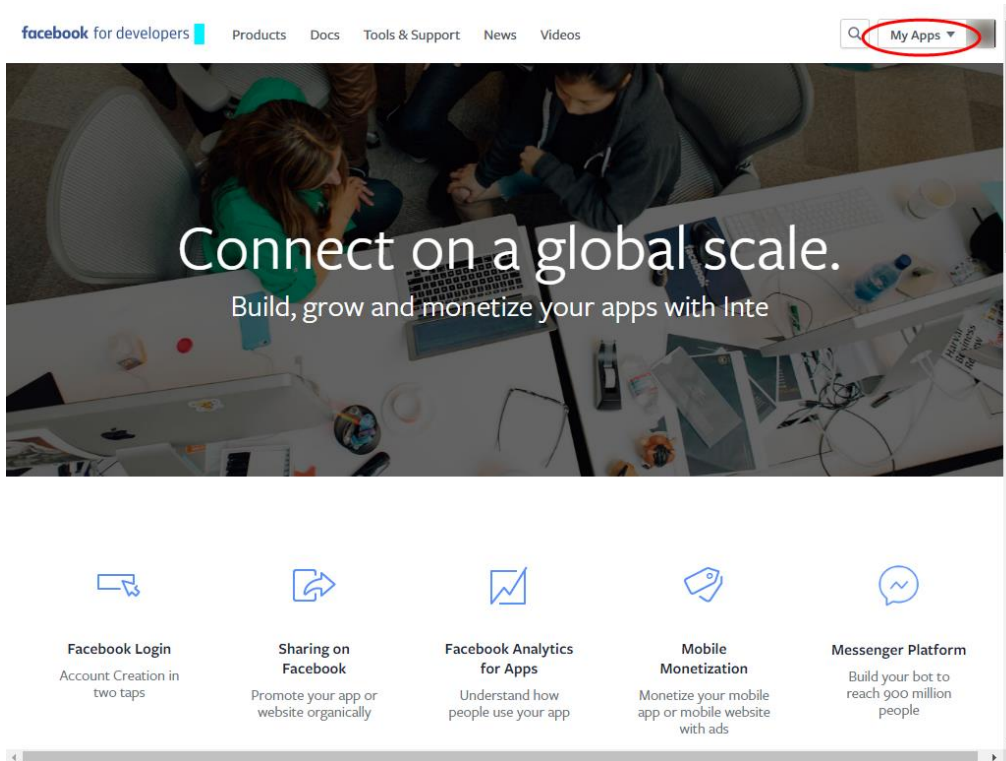
Registering an OAuth Provider

Configuring an OAuth provider for web services is very simple, but will require you to coordinate with the authorization service. Most authorization services will require you to register your application with them in order to obtain an identifier and secret value that is used to identify your application and authorize your application to act on behalf of you or your application's users.

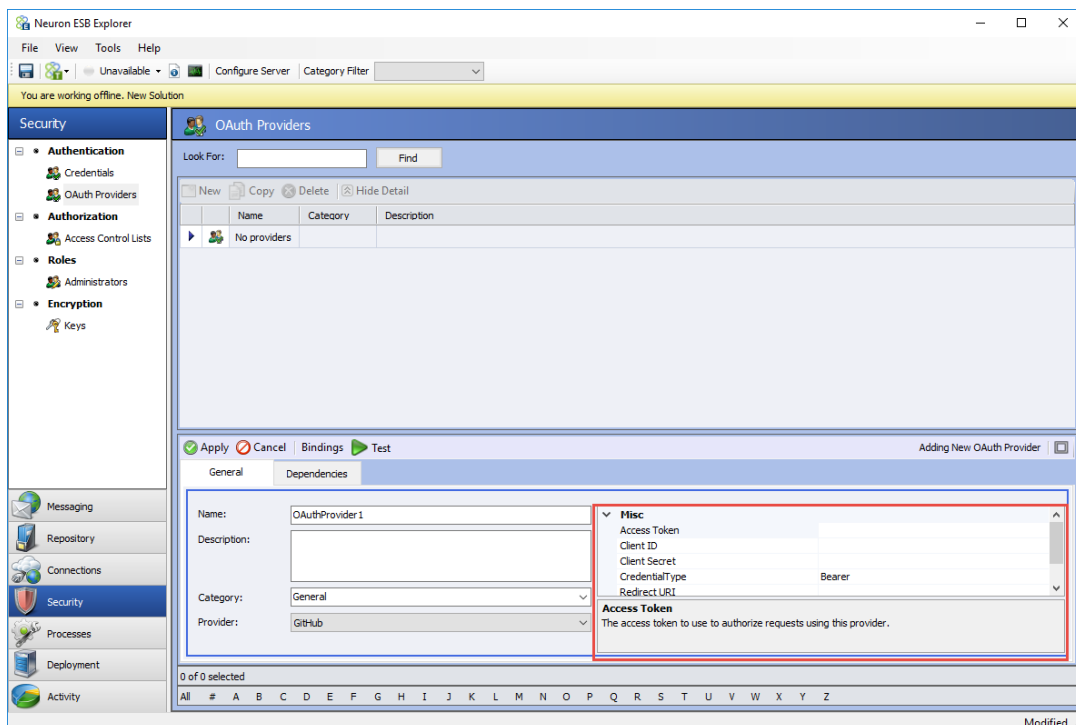


To begin, navigate to the Security tab in Neuron ESB Explorer and select the OAuth Providers node in the tree. Create a new OAuth provider and enter in a name. At the bottom of the details view is a list of providers. Select the authorization service provider that you want to use to obtain an access token for your application to use.

You will next need to register your application with the authorization service. For example, to use Facebook to obtain an access token to authorize access to remote web services, you will need to create an application at the [Facebook Developer](#) website.



After creating your application, you will be able to access the client identifier and client secret. These values are used to identify your application to the authorization service and to authenticate the access tokens and codes from the authorization service. When you have these values, you should register them in the OAuth provider's properties in Neuron Explorer:

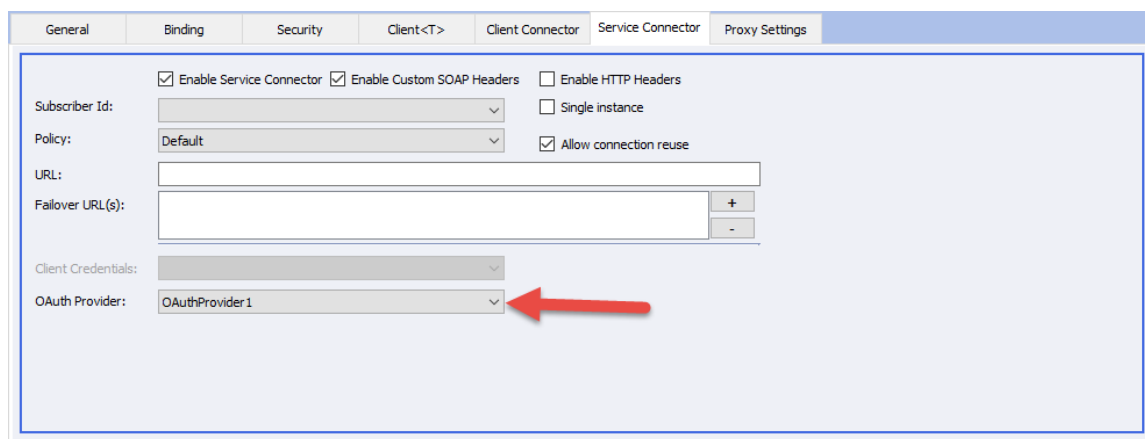


You will typically also need to register a callback URI with the authorization service when you register your application. The callback URI is used as part of the OAuth 2.0 protocol to redirect the user back to your application after signing into the authorization service. You can use any URL. During the OAuth authentication process, Neuron ESB's OAuth providers will use the URI to determine when the authorization process has completed successfully in order to obtain the access token from the authorization service for the OAuth provider.

After your OAuth provider is configured, you can test the provider within Neuron ESB Explorer using the web-based flows. By clicking on the Test button in the toolbar for the detail view, a web browser will be opened and you will be able to validate that the client identifier, secret value, and redirect URI are correct by authenticating with the authorization service and obtaining an access token.

Using an OAuth Provider in a Service Connector

Once an OAuth provider has been configured, it can be used by a RESTful service connector to invoke web services from a remote web service. All that is necessary is to associate your OAuth provider with the service connector using a new drop-down list that has been added to the service connector detail tab:



The screenshot shows the 'Service Connector' configuration tab in Neuron ESB Explorer. The tab is divided into several sections. The 'General' section includes checkboxes for 'Enable Service Connector' (checked), 'Enable Custom SOAP Headers' (checked), and 'Enable HTTP Headers' (unchecked). Below these are dropdowns for 'Subscriber Id' and 'Policy' (set to 'Default'), and a text field for 'URL'. The 'Fallover URL(s)' section has a text field and '+'/'-' buttons. The 'Client Credentials' section has a dropdown. The 'OAuth Provider' dropdown is highlighted with a red arrow and shows 'OAuthProvider1' selected. Other options include 'Single instance' (unchecked) and 'Allow connection reuse' (checked).

Once the service connector has been configured with an OAuth provider, the service connector will use the OAuth provider's access token to invoke the remote web service. The access token will be passed to the remote web service as a bearer token using RFC 6750.

Flat File Parser

In Neuron ESB CU4 there is a flat file parser that can be used to convert single record format flat files into XML. The supported flat files must have the following attributes:

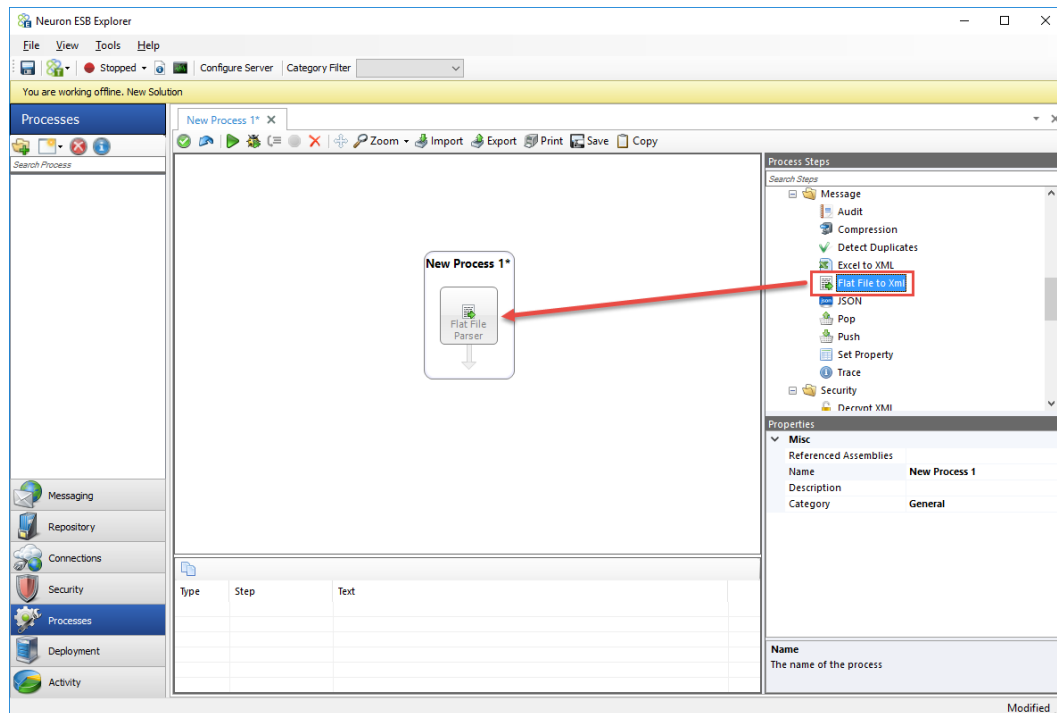
- Fixed-width or delimited. The following delimiters are supported:
 - Comma
 - Tab
 - Vertical Bar (otherwise known as "Pipe-Delimited")
 - Semicolon
 - Custom string (any combination of characters)

- All records in the flat file must conform to the same definition. The only exception to this are leading records and trailing records which can be ignored.

Using the Flat File Parser

Process Definition

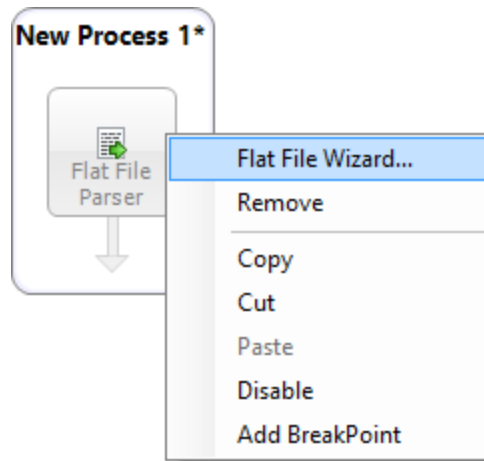
The flat file parser is included as a Business Process step in Neuron ESB Explorer. To use the flat file parser, create a new Business Process (or use an existing one) and drag the Flat File Process step from the Process Steps toolbox onto the Business Process Designer:



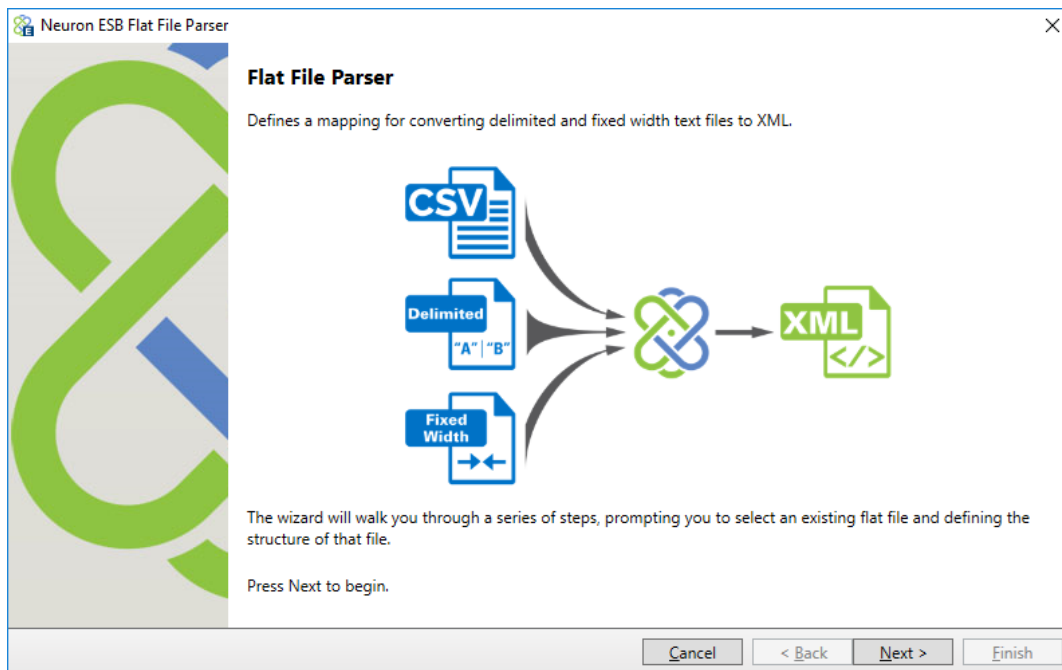
Creating a Flat File Definition

The flat file parser includes a wizard that will assist users in creating the flat file definition. The flat file definition cannot be created without using the wizard. The Flat File Wizard requires a sample flat file in order to create the definition. This sample flat file can either be imported into the new Neuron ESB Text Repository located by navigating to Repository->Documents->Text Documents within the Neuron ESB Explorer or, it can be retrieved directly from the file system.

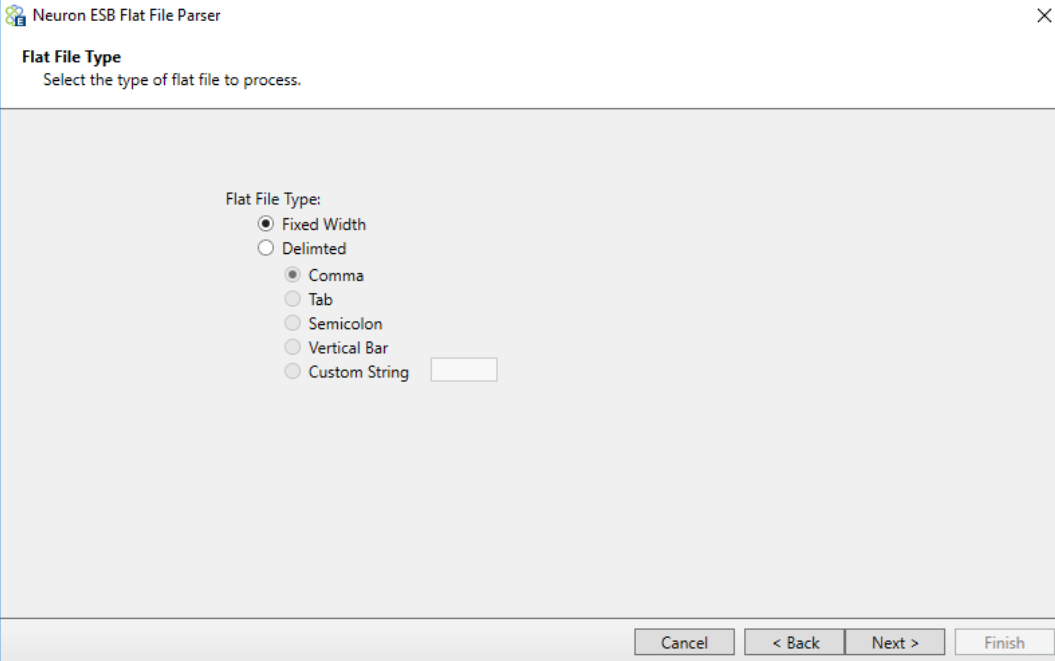
1. To start the wizard, right-click on the Flat File Parser step in the process designer and select **Flat File Wizard...**



2. On the **Welcome** page, click **Next**:



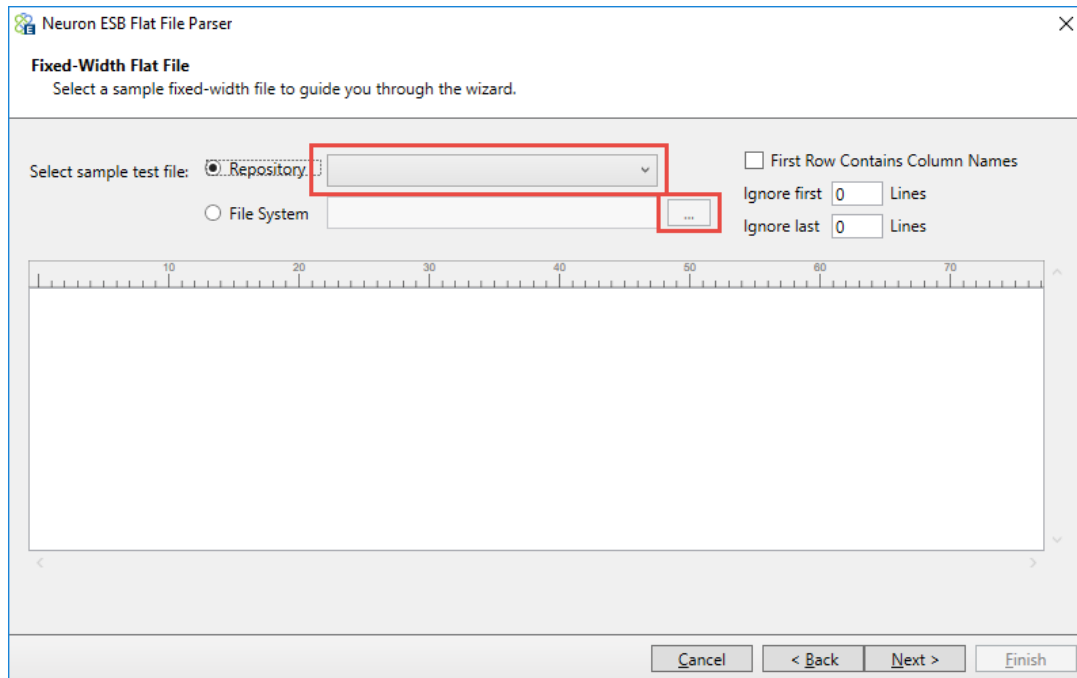
3. On the **File Type** page, select the type of file to process – *Fixed Width* or *Delimited*. If *Delimited* is selected, the appropriate delimiter must also be supplied. If something other than the provided options are needed, select *Custom String* and enter the character(s) to be used as a delimiter. Click **Next**:



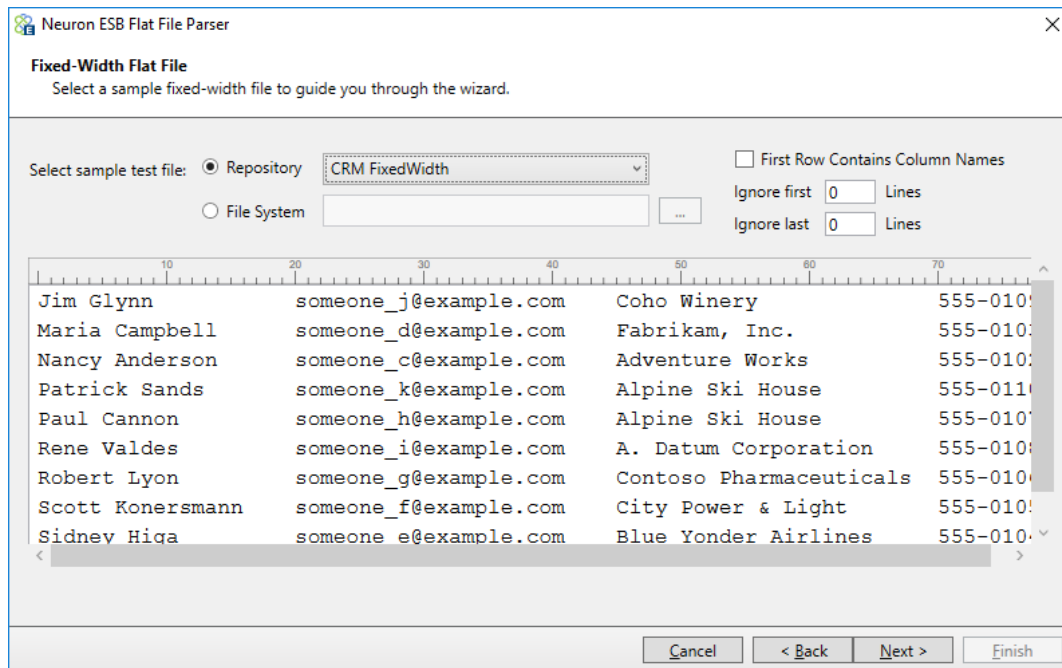
The image shows a Windows-style dialog box titled "Neuron ESB Flat File Parser". Inside the dialog, the title "Flat File Type" is followed by the instruction "Select the type of flat file to process." Below this, under the heading "Flat File Type:", there are two main radio button options: "Fixed Width" (which is selected) and "Delimited". Under the "Delimited" option, there are four sub-radio button options: "Comma" (selected), "Tab", "Semicolon", and "Vertical Bar". Below these is a "Custom String" option followed by a small text input field. At the bottom of the dialog, there are four buttons: "Cancel", "< Back", "Next >", and "Finish".

If *Fixed Width* is selected, continue with step 4. If *Delimited* is selected, skip to step 8.

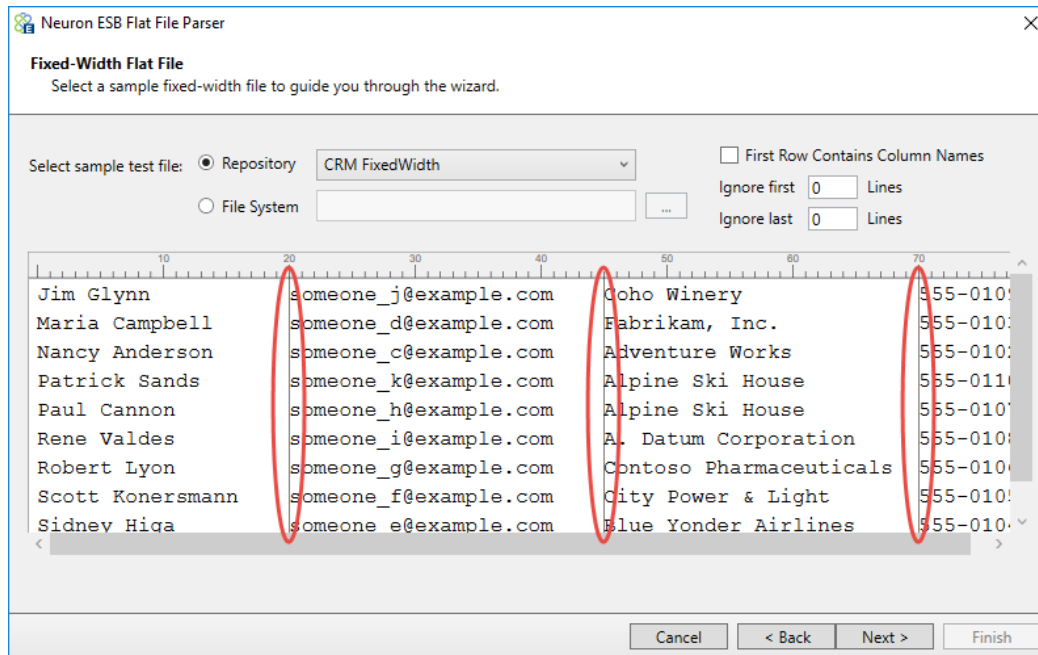
4. On the **Fixed-Width Flat File** page, either select a sample flat file from the repository or the file system. To select a sample from the repository, select the *Repository* radio button and select from one of the files listed. To select a file from the repository it must be imported before running the wizard. To select a sample from the file system, select the *File System* radio button and click the ellipsis button to browse for a file from the file system. Some flat files include column header names in the first line. If your flat file includes column headers, check the box for *First Row Contains Column Names*. Also, if you want the Flat File Parser to ignore any of the leading or trailing rows, indicate how many of each row to ignore:



- Once you have selected a fixed-width document the preview pane will display the first 10 rows:



- When parsing a fixed-width file you need to set the column breaks. To set a column break, click anywhere in the preview pane and a vertical line will appear. Click the line again and it will disappear. Note – you don't have to click on the ruler above the preview pane, just click inside the preview pane:



7. After selecting all the column breaks, click **Next**. Skip to step 10.
8. If you selected *Delimited* in step 3, the **Comma-Delimited Flat File** page will appear. On this page, either select a sample flat file from the repository or the file system. To select a sample from the repository, select the *Repository* radio button and select from one of the files listed. To select a file from the repository it must be imported before running the wizard. To select a sample from the file system, select the *File System* radio button and click the ellipsis button to browse for a file from the file system. Some flat files include column header names in the first line. If your flat file includes column headers, check the box for *First Row Contains Column Names*. Also, if you want the Flat File Parser to ignore any of the leading or trailing rows, indicate how many of each row to ignore:

Neuron ESB Flat File Parser

Comma-Delimited Flat File
Select a sample comma-delimited text file to guide you through the wizard.

Select sample test file: ☒ Repository ☐ First Row Contains Column Names

☐ File System ... Ignore first Lines

Ignore last Lines

Cancel < Back Next > Finish

9. Unlike the preview pane for fixed-width files you do not need to set column breaks. This file includes headers in the first row. Click **Next**:

Neuron ESB Flat File Parser

Comma-Delimited Flat File
Select a sample comma-delimited text file to guide you through the wizard.

Select sample test file: ☒ Repository ☒ First Row Contains Column Names

☐ File System ... Ignore first Lines

Ignore last Lines

Name,Email,Company,Phone
 Jim Glynn,someone_j@example.com,Coho Winery,555-0109
 Maria Campbell,someone_d@example.com,Fabrikam Inc.,555-0103
 Nancy Anderson,someone_c@example.com,Adventure Works,555-0102
 Patrick Sands,someone_k@example.com,Alpine Ski House,555-0110
 Paul Cannon,someone_h@example.com,Alpine Ski House,555-0107
 Rene Valdes,someone_i@example.com,A. Datum Corporation,555-0108
 Robert Lyon,someone_g@example.com,Contoso Pharmaceuticals,555-0106
 Scott Konersmann,someone_f@example.com,City Power & Light,555-0105

Cancel < Back Next > Finish

1. When you check the “First Row Contains Column Names” checkbox, the column names are automatically populated. However, you can still change these names on the Column Details page. After confirming the column details, click **Finish** to close the wizard.

Neuron ESB Flat File Parser

Set the column details
Verify the name and width of each column

Columns

- Name
- Email
- Company
- Phone

Column Properties

Name:

Start Index:

Width:

Preview

Name	Email	Company	Phone
Jim Glynn	someone_j@example.com	Coho Winery	555-0109
Maria Campbell	someone_d@example.com	Fabrikam Inc.	555-0103
Nancy Anderson	someone_c@example.com	Adventure Works	555-0102
Patrick Sands	someone_k@example.com	Alpine Ski House	555-0110
Paul Cannon	someone_h@example.com	Alpine Ski House	555-0107
Rene Valdes	someone_i@example.com	A. Datum Corporation	555-0108
Robert Lyon	someone_g@example.com	Contoso Pharmaceuticals	555-0106

Cancel < Back Next > Finish

After finishing the wizard, the flat file definition is stored as a process property. You can re-run the wizard and see the same values you selected the previous time you ran it.

To test the Flat File Parser, use the Process Tester and use the same sample data you used when running the Wizard.

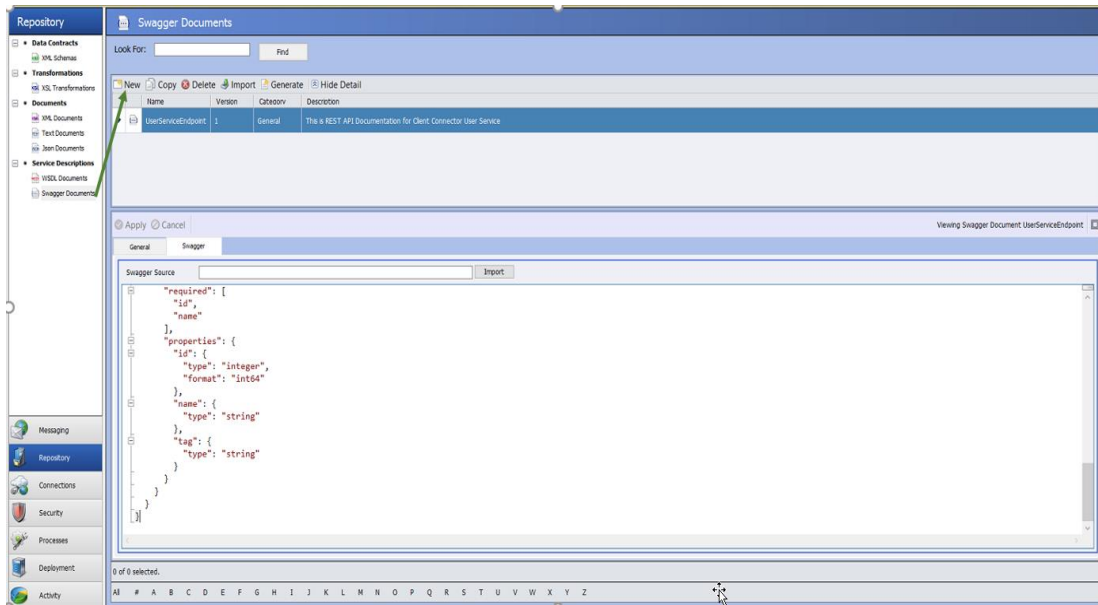
Swagger Support for REST APIs

Swagger is a simple yet powerful representation of any RESTful API. With one of the largest ecosystems of API tooling, thousands of developers are supporting Swagger in almost every modern programming language and deployment environment. With a Swagger-enabled API, organizations get interactive documentation, client SDK generation and discoverability.

It's natural that the Neuron ESB Client Connector (i.e. SOAP/REST API services hosted by Neuron ESB) should also create Swagger-enabled APIs. Starting with Neuron ESB 3.5 CU4 the Client Connector does exactly that.

Creating Swagger Documentation

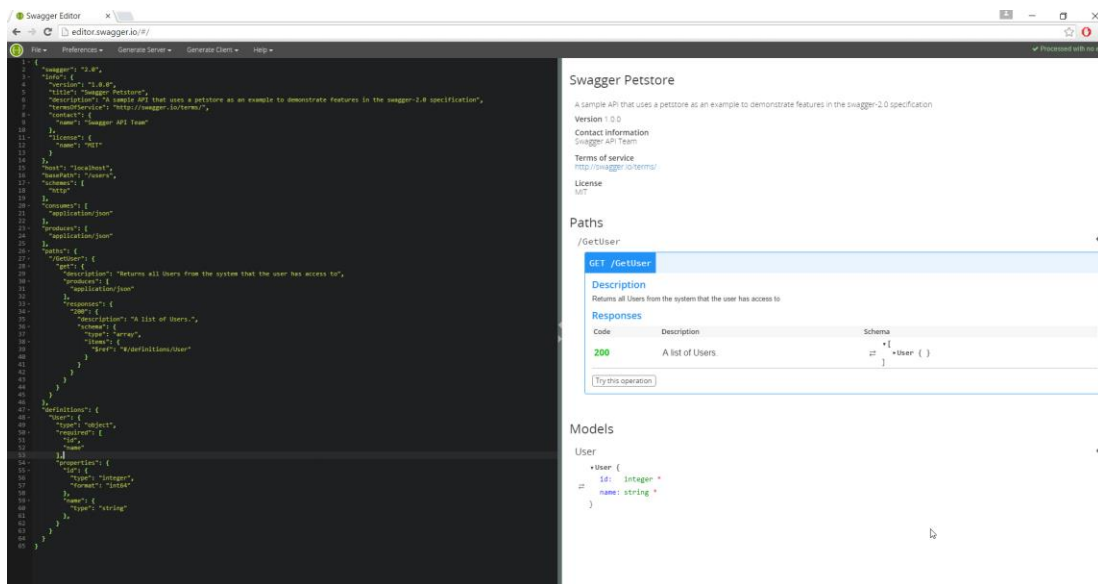
The CU4 release includes a new section in the Neuron ESB Explorer Repository called Swagger Documents (see figure 1). This is where Swagger API documentation can be stored and managed. Users can create a new Swagger document in the Repository by either copying or entering the content of an existing Swagger document or by importing an existing Swagger document from any External URI.



(Figure 1)

Once a Swagger document is stored in the Repository, the document will automatically be hosted by Neuron ESB at <http://localhost:51003/docs/<newlycreatedswaggerdoc>>. All Neuron ESB managed Swagger Repository documents are hosted on Port 51003. However, the port can be changed by editing the value of the “*SwaggerSelfHostingUrl*” key in the *appSettings* section of the *discoveryservice.exe.config* configuration file located in the “*C:\Program Files (x86)\Neudesic\Neuron ESB v3*” folder.

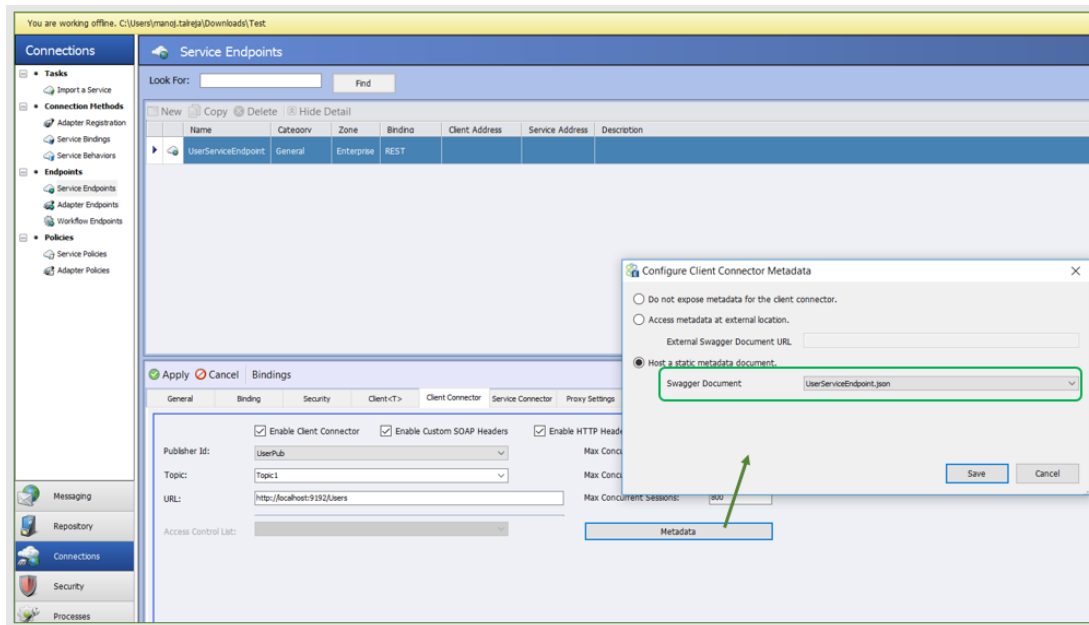
To create Swagger documents from scratch, the Swagger Editor (<http://editor.swagger.io>), an online tool, can be used as shown in Figure 2 below.



(Figure 2)

Using Swagger Documents

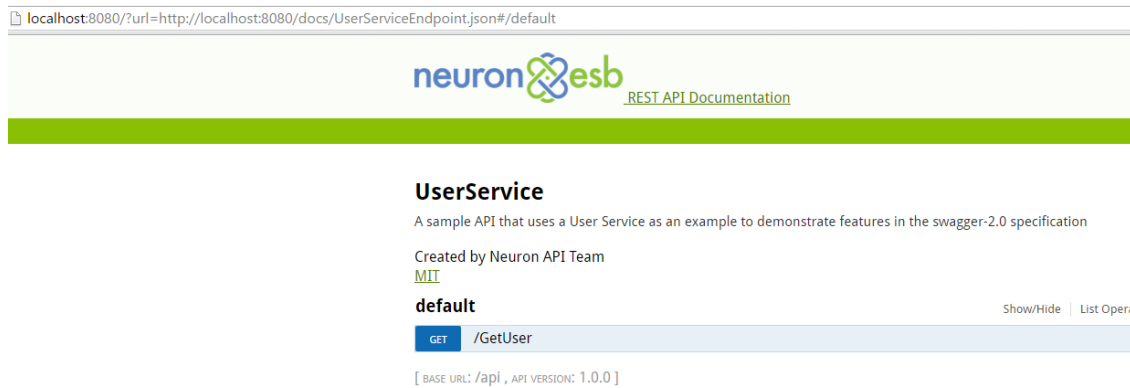
In order to assign a Swagger document to the Neuron ESB Client Connector, navigate to the Client Connector Tab then select the Metadata Button, which will present you a “Configure Client Connector Metadata” dialog box. Here either an external Swagger document URI can be provided or a static Swagger document located in the “Swagger Documents” section of the Neuron Explorer ESB Repository can be chosen as shown in Figure 3.



(Figure 3)

Once a Swagger document is selected from the dropdown selection, or an external Document URI is provided, the Client connector will have Swagger documentation associated with its endpoint. Pointing a web browser to the Client Connector URL along with **/help** will expose the Swagger documentation.

For example, if the Neuron ESB Client Connector URL was <http://localhost:9192/Users>, then the <http://localhost:9192/Users/Help> URL will redirect users to the Swagger documentation (see Figure 4).

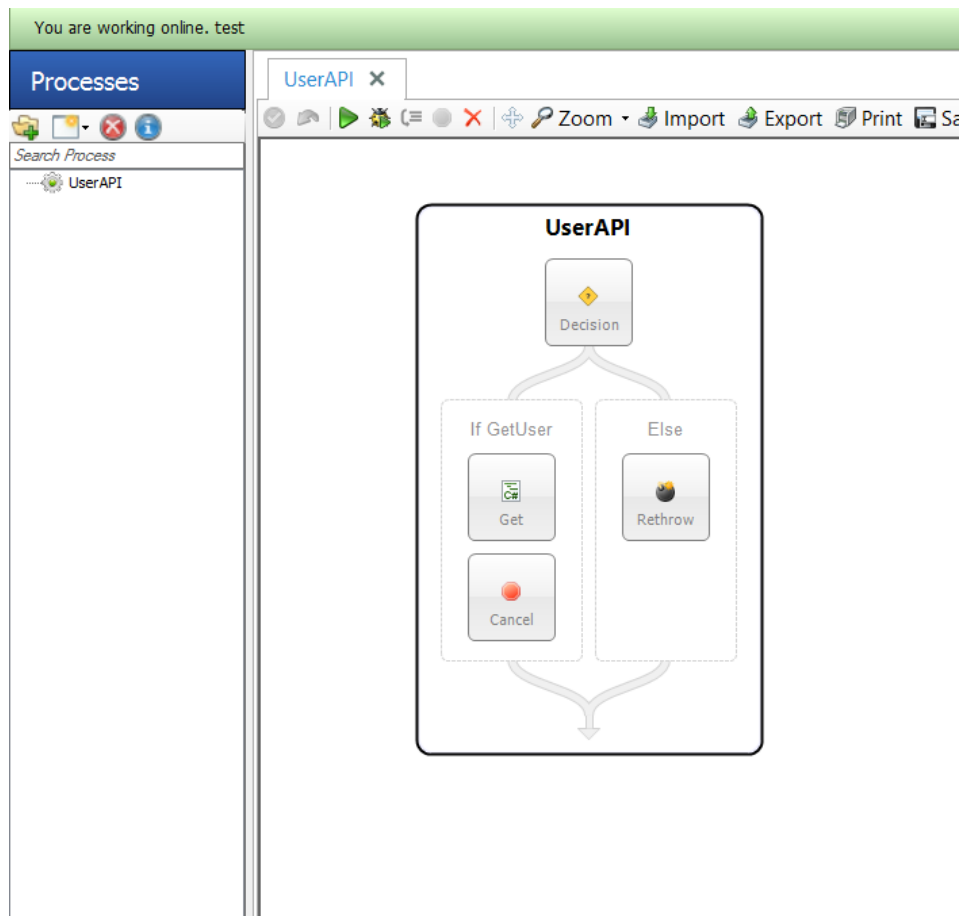


(Figure 4)

Implementing and Testing

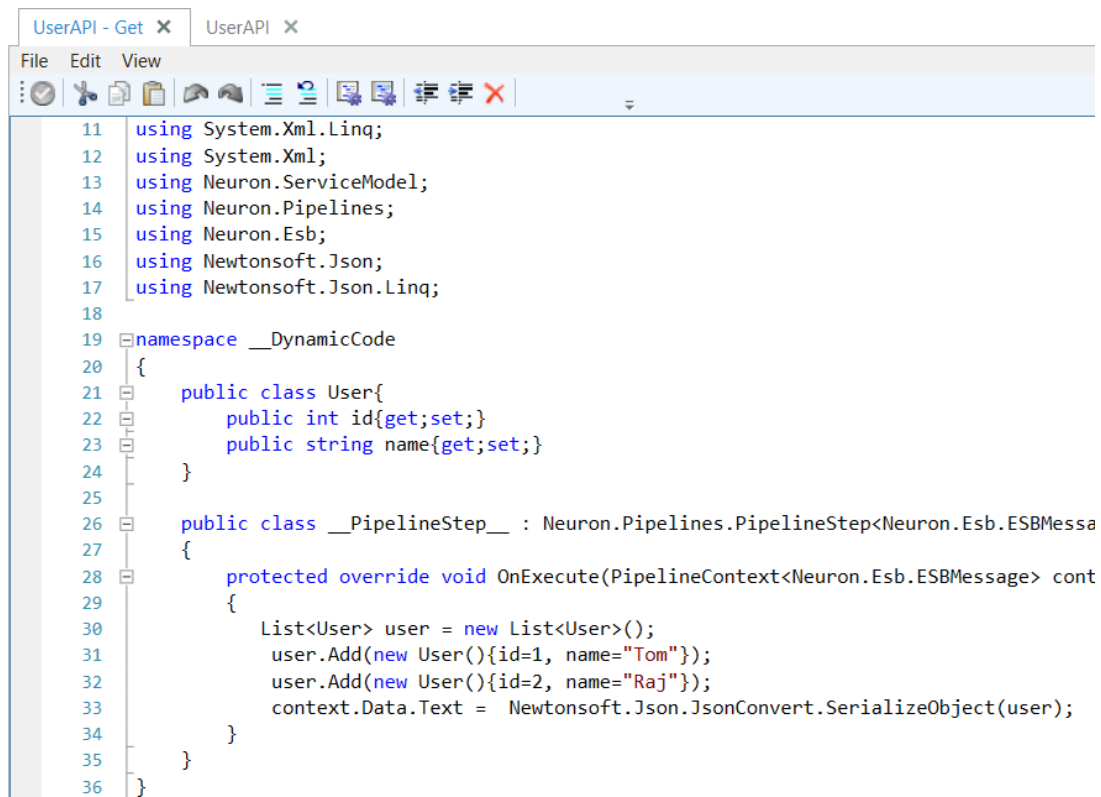
Once a Swagger document (like the one created and shown above) is assigned to the Neuron ESB Client Connector, the implementation for the call either has to already exist or be created. The following example demonstrates how we can create a simple implementation that can be tested by users browsing to the Client Connector's Swagger document.

Using the Neuron ESB Business Process Designer a simple Process can be created that contains the implementation of the “*GetUser*” API call as shown in the Figure 5 below. The Business Process is then attached to the OnPublish event of the Neuron ESB Publisher assigned to the Client Connector.



(Figure 5)

The actual implementation for “*GetUser*” is simple and encapsulated within the “*Get*” C# Process Step shown in Figure 5. The “*Get*” Process Step expands into the Neuron ESB C# Code Editor as shown in Figure 6. The Code Editor contains C# code to mock users in a list, convert the list to a JSON object, setting the Neuron ESB Message body.



```
11 using System.Xml.Linq;  
12 using System.Xml;  
13 using Neuron.ServiceModel;  
14 using Neuron.Pipelines;  
15 using Neuron.Esb;  
16 using Newtonsoft.Json;  
17 using Newtonsoft.Json.Linq;  
18  
19 namespace __DynamicCode  
20 {  
21     public class User{  
22         public int id{get;set;}  
23         public string name{get;set;}  
24     }  
25  
26     public class __PipelineStep__ : Neuron.Pipelines.PipelineStep<Neuron.Esb.ESBMessage>  
27     {  
28         protected override void OnExecute(PipelineContext<Neuron.Esb.ESBMessage> context)  
29         {  
30             List<User> user = new List<User>();  
31             user.Add(new User(){id=1, name="Tom"});  
32             user.Add(new User(){id=2, name="Raj"});  
33             context.Data.Text = Newtonsoft.Json.JsonConvert.SerializeObject(user);  
34         }  
35     }  
36 }
```

(Figure 6)

Once implemented, this Neuron ESB Client Connector API can be tested directly from the Swagger document as can be seen below in Figure 7. Clicking the “Try it out!” button will execute the *GetUser* API call, returning the Response message.

default Show/Hide List Operations Expand Operations

GET /GetUser

Implementation Notes
Returns all Users from the system that the user has access to

Response Class (Status 200)
A list of Users.

Model Model Schema

```
[
  {
    "id": 0,
    "name": "string"
  }
]
```

Response Content Type application/json ▼

[Try it out!](#) [Hide Response](#)

Curl

```
curl -X GET --header 'Accept: application/xml' 'http://localhost:8181/users/GetUser'
```

Request URL

```
http://localhost:8181/users/GetUser
```

Response Body

```
[{"id":1,"name":"Tom"}, {"id":2,"name":"Raj"}]
```

Response Code

```
200
```

Response Headers

```
{
  "content-type": "application/xml; charset=utf-8"
}
```

(Figure 7)

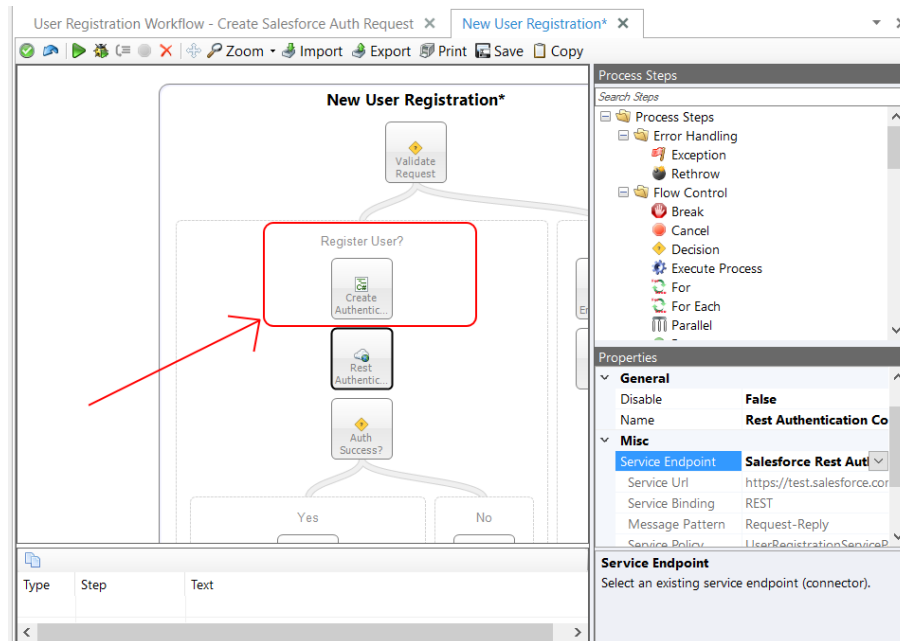
HTTP Utility, JSON and Swagger

Neuron ESB includes a Service Broker that enables organizations to deploy Neuron ESB as a Service Gateway, providing mediation, security, hosting and a number of other services. Service Connectors are essentially registrations within Neuron ESB that point to existing services hosted within an organization, by a partner or in a cloud domain. These services, which can be either SOAP or REST (HTTP) based, can be called either through the Neuron ESB messaging system via a Topic subscription or by using a *Service Endpoint* Workflow Activity and Process Step. The latter two options can be used with either the Neuron ESB Workflow Designer or the existing Business Process Designer and allows a user to directly call any Service Connector without the need to publish a request to a Topic, eliminating all pub/sub overhead. These are commonly used to create service aggregation and composition solutions.

For calling REST (HTTP) based Service Connectors (endpoints), it's not uncommon that various pieces of information need to be provided at runtime according to the REST specification. For example, the

Method name (e.g. GET, POST, PUT, PATCH, etc.) must be provided. HTTP Headers usually need to be provided (e.g. Content Type, Authorization, Accept, etc.) as well as Query string or URL parameters.

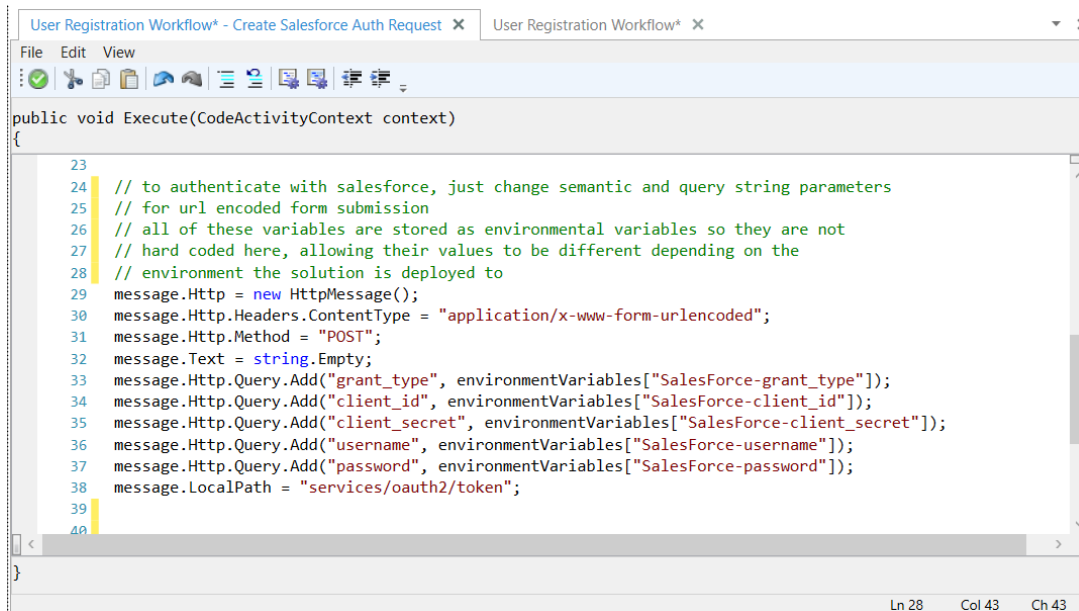
In previous versions of Neuron ESB, this information could be provided by using a C# Code Editor within either the Business Process Designer or Workflow Designer directly preceding the Service Endpoint activity as depicted below:



Opening the C# Code Editor allowed developers to provide the information they needed to initialize the service call at runtime by primarily using the HTTP object of the Neuron ESB Message as shown below.

```
void OnExecute(PipelineContext<Neuron.Esb.ESBMessage> context)
{
    1 // Retrieve the auth token from the incoming message
    2 dynamic jo = context.Data.ToJsonObject();
    3
    4 // to register, just change semantic and add headers
    5 context.Data.Http = new Neuron.Esb.HttpMessage();
    6 context.Data.Http.Headers.ContentType = "application/json";
    7 context.Data.Http.Method = "POST";
    8 context.Data.Http.Headers.Add("Authorization", "Bearer " + jo.data);
    9
    10 // set the local path so that we're calling register now
    11 // set the outgoing body with the original request
    12 context.Data.LocalPath = "/api/user/register";
    13 context.Data.Text = context.Properties["originalRequestBody"] as string;
    14
}
```

The information used to initialize these HTTP elements could come from the existing Neuron ESB Message, Neuron ESB Message or context properties or even Neuron ESB Environmental Variables that may have different values at runtime depending on the runtime environment they are deployed to. The example below shows how Query string parameters could be set in a Code Editor in the Workflow Designer.



```
public void Execute(CodeActivityContext context)
{
    23
    24 // to authenticate with salesforce, just change semantic and query string parameters
    25 // for url encoded form submission
    26 // all of these variables are stored as environmental variables so they are not
    27 // hard coded here, allowing their values to be different depending on the
    28 // environment the solution is deployed to
    29 message.Http = new HttpResponseMessage();
    30 message.Http.Headers.ContentType = "application/x-www-form-urlencoded";
    31 message.Http.Method = "POST";
    32 message.Text = string.Empty;
    33 message.Http.Query.Add("grant_type", environmentVariables["SalesForce-grant_type"]);
    34 message.Http.Query.Add("client_id", environmentVariables["SalesForce-client_id"]);
    35 message.Http.Query.Add("client_secret", environmentVariables["SalesForce-client_secret"]);
    36 message.Http.Query.Add("username", environmentVariables["SalesForce-username"]);
    37 message.Http.Query.Add("password", environmentVariables["SalesForce-password"]);
    38 message.LocalPath = "services/oauth2/token";
    39
    40
}
```

Ln 28 Col 43 Ch 43

HTTP (REST) Endpoint Configuration

Rather than requiring that developers use C# to set HTTP properties, CU4 introduces a new tool, the “HTTP Utility”. This utility can be used within the Business Process or Workflow Designer to do more advanced levels of HTTP configuration. The HTTP Utility is exposed as both a Process Step and Workflow Activity that can be dragged onto either designer. Users can select “Configure” from the context menu to display the main configuration screen of the HTTP Utility as shown below:

Http Client Utility

Use Swagger document

Enter Request URL and/or local path i.e. '/api/user/register/'

Headers

Body

Url Parameters

Please enter Key/Value pairs. Press '+' button to add more. Type in CTRL + SPACE to see list of available Environmental Variables and properties that can be used to dynamically set values.

Key

Value

Save

Cancel

For users familiar with tools like “Postman” and the REST client of Chrome, the HTTP Utility will look very similar and they will find it just as easy to configure. The HTTP Utility has built in intelligence and allows users to use Neuron ESB Environmental Variables, Neuron ESB Message properties, Context properties and literal values to configure any Value, Message body or URL as shown below:

Http Client Utility

Use Swagger document

POST /api/user/register

Headers

Body

Url Parameters

Please enter Key/Value pairs. Press '+' button to add more. Type in CTRL + SPACE to see list of available Environmental Variables and properties that can be used to dynamically set values.

Authorization

Bearer {#JObject.data}

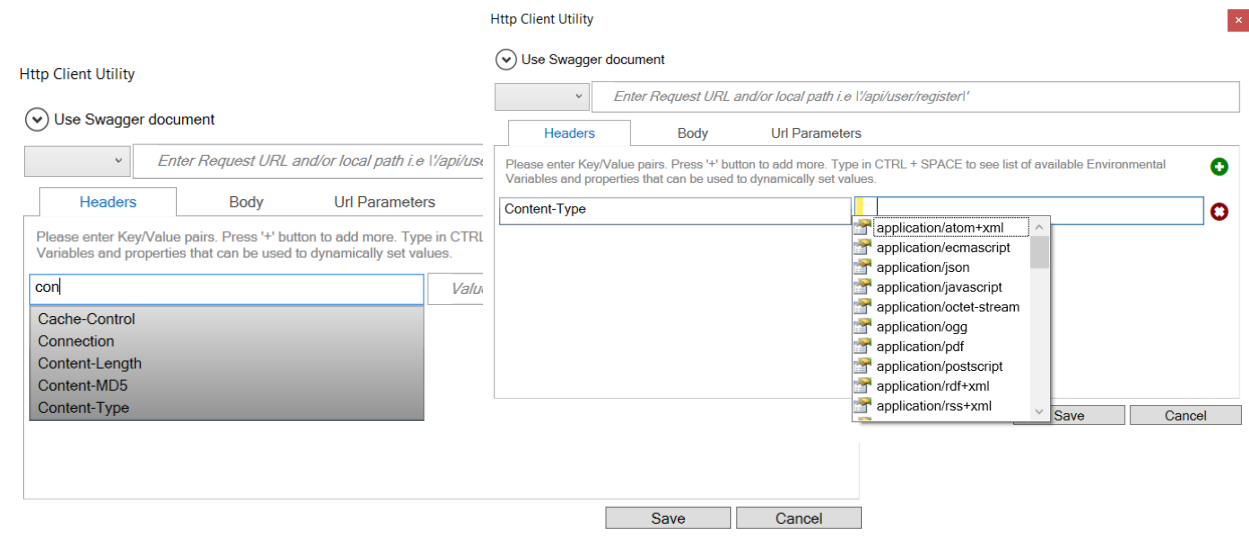
Content-Type

application/json

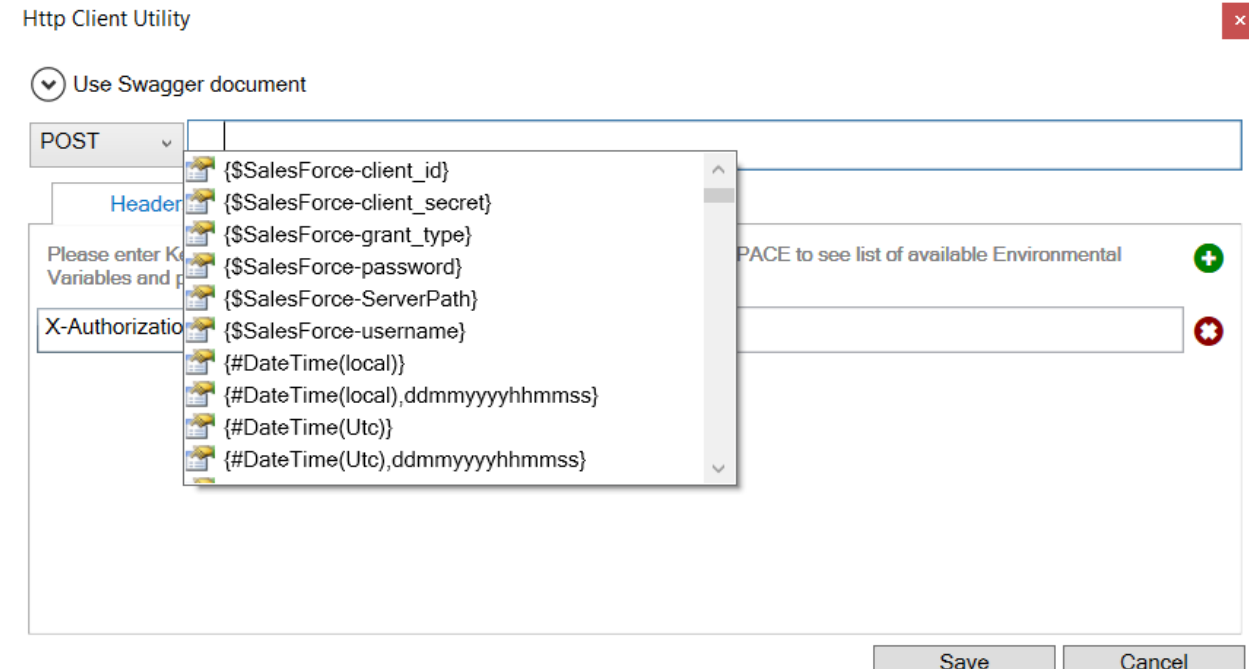
Save

Cancel

Valid HTTP Header options and methods are displayed as drop down boxes, while possible values for the selected HTTP Header “key” also appear as context sensitive drop downs.



Alternatively, Neuron ESB Environmental Variables, properties (context, message, etc.), message body and literals can be used by selecting from the available list. Environmental variables are preceded by the “\$”, whereas all others are preceded by “#” as shown below.



The literal values that appear in the list (i.e. by pressing CTRL + SPACE) that can be used are:

- {#DateTime(local)} - Adds the local date time value with following format: "yyyyMMddHHmmss"

- {#DateTime(local), ddmmyyyyhhmmss } - Adds the local date time value using the provided format specifier
- {#DateTime(Utc)} - Adds the UTC date time value with following format: "yyyyMMddHHmmss"
- {#DateTime(Utc),ddmmyyyyhhmmss} - Adds the UTC date time value using the provided format specifier
- {#IMF-fixdate} – Adds the RFC 7231 Date/Time Format e.g. 'Tue, 15 Nov 1994 08:12:31 GMT'
- {#GUID} – Adds a unique GUID string value (e.g. 00000000-0000-0000-0000-000000000000)

Values that can be used to access either some part of the body or a custom message property of the ESB Message are:

- {#<custom>.<property>} – Returns custom message property i.e. context.data.GetProperty("myPrefix", "MyProperty")
- {#JsonObject.<property>} – Returns the property from the JSON dynamic object property from the current Neuron ESB Message.

Besides the Method drop down box and the URL text box, the HTTP Utility has 3 primary tabs: Headers, Body and URL Parameters. The Headers tab allows users to specify any HTTP or custom key/value pairs that will appear as HTTP Headers in the final REST service call. A good example of an HTTP Header could be either the Content Type or Authorization header. The Body tab allows users to specify how the body should be encoded and sent to the service as shown below:

Http Client Utility ✕

☒ Use Swagger document

POST /api/user/register

Headers Body Url Parameters

☐ Form Body
 ☐ x-www-form-urlencoded
 ☒ Raw
 ☐ Clear ESB Message Body

Type in CTRL + SPACE to see list of available Environmental Variables and properties that can be used to dynamically set values.

```
{#context(originalReqeustBody)}
```

Save
Cancel

Using the settings, Neuron ESB will ensure the proper HTTP Content Type and data are appropriately set and encoded if necessary. Users have additional options like clearing out the underlying Neuron ESB Message body if either Form Body or Form Url Encoded options are chosen.

The URL Parameters tab allows users to enter key/value pairs that will be serialized as URL parameters for the service call. The HTTP Utility removes the need for developers to use C# to set the required HTTP properties for any service call. In the example below, a Salesforce authentication call is made where all the values are using Neuron ESB Environmental Variables, allowing the values to be specific to the actual runtime environment the solution is deployed to:

The screenshot shows the 'Http Client Utility' dialog box. At the top, there is a dropdown menu set to 'POST' and a text field containing the URL '{SalesForce-ServerPath}/services/oauth2/token'. Below this, there are three tabs: 'Headers', 'Body', and 'Url Parameters'. The 'Body' tab is selected. Under the 'Body' tab, there are three radio buttons: 'Form Body', 'x-www-form-urlencoded' (which is selected), and 'Raw'. To the right of these radio buttons is a checkbox labeled 'Clear ESB Message Body' which is checked. Below the radio buttons, there is a text instruction: 'Please enter Key/Value pairs. Press '+' button to add more. Type in CTRL + SPACE to see list of available Environmental Variables and properties that can be used to dynamically set values.' To the right of this instruction is a green '+' button. Below the instruction, there is a table with five rows, each representing a key-value pair. The keys are 'grant_type', 'client_id', 'client_secret', 'username', and 'password'. The values are all Neuron ESB Environmental Variables: '{SalesForce-grant_type}', '{SalesForce-client_id}', '{SalesForce-client_secret}', '{SalesForce-username}', and '{SalesForce-password}'. To the right of each value field is a red 'x' button. At the bottom right of the dialog box, there are 'Save' and 'Cancel' buttons.

Key	Value
grant_type	{SalesForce-grant_type}
client_id	{SalesForce-client_id}
client_secret	{SalesForce-client_secret}
username	{SalesForce-username}
password	{SalesForce-password}

Swagger Integration

Over the years, Swagger has become a common documentation format for REST (HTTP) based services. As such, the CU4 release introduces a number of new JSON/Swagger features, one of which is the ability to configure the HTTP Utility using Swagger.

For example, CU4 ships a Marketo adapter which is accompanied by its respective Swagger documentation. The Marketo Swagger documentation, as well as any Swagger document registered within the new Neuron ESB Swagger Repository, can be accessed directly within the “Use Swagger document” section of the HTTP Utility.

Swagger documents can be imported in the Neuron ESB Swagger Repository by navigating to Repository->Service Descriptions->Swagger Documents within the Neuron Explorer.

By expanding the “Use Swagger document” section, the Document and Operations dropdown fields will be visible and populated using existing Swagger documents in the Repository. These can be selected to auto configure the HTTP Utility. If a Swagger document doesn’t exist, one can be imported directly within the UI by selecting the “Import Swagger...” option from the Document dropdown and providing the URL of the Swagger document.

⬆ Use Swagger document

Document:

Operation:

GET

Headers Body **Url Parameters**

Please enter Key/Value pairs. Press '+' button to add more. Type in CTRL + SPACE to see list of available Environmental Variables and properties that can be used to dynamically set values. +

name	Value	✖
status	Value	✖
folder	Value	✖

JSON Templates and Data Mapping

JSON has become more prominently used by developers and integrators as organizations move to using more REST (HTTP) based services. CU4 provides new repositories and access methods that make using JSON much easier than in previous releases of Neuron ESB.

In CU4, the *ToJsonObject()* has been added to the Neuron ESB Message object. This will return the Neuron ESB Message body as a *Newtonsoft.Json.Linq.JObject*. This is accessible within either a Workflow or Business Process. It allows users to use .NET's dynamic object keyword to access data correctly using a hierarchical syntax. For example, a sample json document like the following:

```
{
  "message": "Success. Lead Created",
  "data": {
    "SalesOwnerAccountUserId": "118527",
    "AccountUserId": "89027"
  }
}
```

Could then be accessed directly in a Code Editor using C# as shown below:

```
dynamic jo = context.Data.ToJsonObject();
var ownerID = jo.data.SalesOwnerAccountUserId;
```

This makes it much easier for developers to access and modify json data within a Business Process or Workflow.

To complement this, a new JSON Repository for storing JSON documents and templates has been introduced which can be located by navigating to Repository->Documents->Json Documents within the Neuron ESB Explorer. Sample JSON documents or templates can be stored and accessed at runtime within any of the Business Process or Workflow environments. Integrators can use JSON templates as a quick, simple way to facilitate data mapping/transformations. Json templates are nothing more than Json messages that have format specifiers inserted for their values. For example, if the following template was stored in the repository and named, "AccountUpdateRequest":

```
{
  "salesRepAccountUserID":{0},
  "auditAccountUserID":{1}
}
```

It could then be retrieved in any Business Process or Workflow by using the "JsonFormatString" property of the repository document:

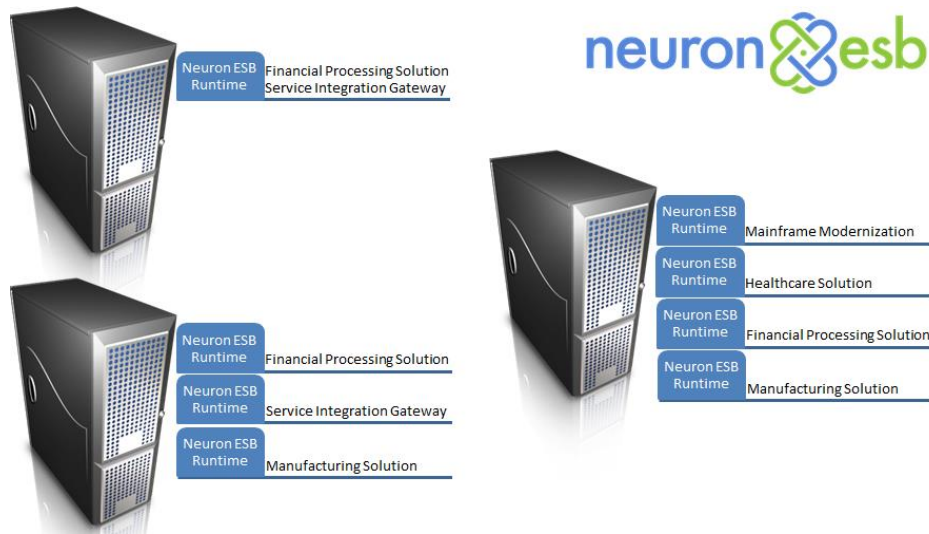
```
var accountUpdateTemplate =
  configuration.JsonDocs["AccountUpdateRequest"].JsonFormatString;
```

Once the template is retrieved, data mapping can be as easy as this:

```
Context.Data.Text = string.Format(accountUpdateTemplate,
  (string)jo.data.SalesOwnerAccountUserId??string.Empty,
  (string)jo.data.accountUserID??string.Empty);
```

Port Sharing

One of Neuron ESB's scalability features is the ability to install multiple instances of the Neuron ESB Runtime on a single server. Each runtime instance can be configured as either a 32 or 64-bit process, capable of running side by side. Each instance of the runtime can load an independent Neuron ESB Configuration store (solution). This allows organizations to easily partition business solutions to run on a single server and scale across multiple servers.



However, one of the configuration challenges has always been TCP port configuration at the solution level. Neuron ESB uses TCP ports to communicate between its internal subsystems e.g. Auditing, Control, Configuration, Master and TCP Publishing Services. Remote Neuron ESB Parties also use TCP to connect to the Neuron ESB Server; to retrieve their respective configuration, receive updates and to regularly send reporting information. The TCP Port configuration for a solution can be found on the Ports tab of the Enterprise Zone by navigating to Deployment->Settings->Zones within the Neuron Explorer.

The screenshot shows the 'Ports' tab in the Neuron Explorer for the 'Enterprise' zone. The interface includes a 'Hide Detail' button, a table with columns for 'Zone', 'Location', and 'Description', and a 'Viewing Zone Enterprise' indicator. Below the table, there are tabs for 'General', 'Ports', 'Transports', and 'Server'. The 'Ports' tab is active, displaying a list of services and their configured TCP ports.

Service	Port	Service	Port
Bootstrap Service	50000	Peer Resolver Service	50005
Management Service	50002	Audit Service	50200
Control Service	50004	TCP Service	50010 - 50100

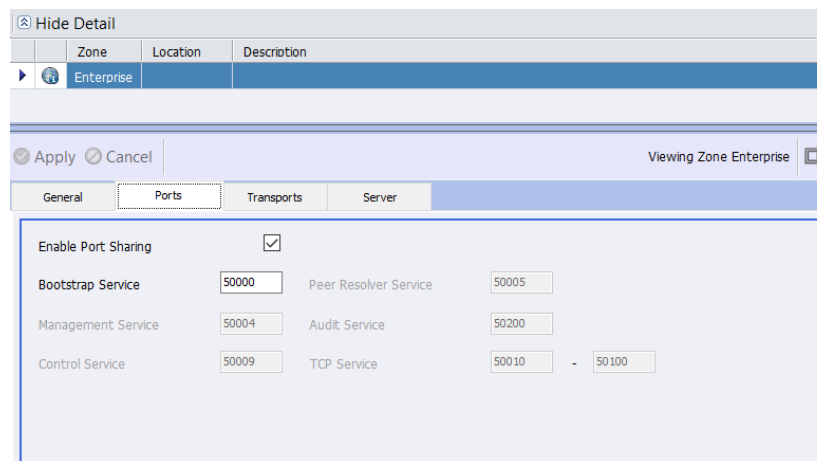
Prior to CU4, if multiple instances were installed on a single server, users would need to ensure that the ports configured for each solution assigned to those instances were unique. For example, if the Bootstrap Service was configured in one solution (instance A) to run on port 50000, then the other solution (instance B) would need to be configured to use a different port. If they weren't, then one of the solutions would fail to start, reporting a Port in use exception.

The unique port requirement made it more challenging, operationally, to manage the solutions as more runtime instances were installed on a single server. To resolve the need to change and manage port conflicts between solutions we are introducing Port Sharing in the 3.5 CU4 release. Port Sharing can

significantly reduce the operational and management overhead when installing and running multiple instances of the Neuron ESB Runtime on the same machine.

Port Sharing is easily enabled by doing 2 things

1. Enable and then set for automatic startup the “Net.TCP Port Sharing Service” in the service control manager. Start the service. For more information: [https://msdn.microsoft.com/en-us/library/ms733925\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms733925(v=vs.110).aspx)
2. Enable the new Port Sharing option located on the Port tab of the Enterprise Zone within the Neuron ESB Explorer as shown below:



Once the Port Sharing option is enabled, the solution must be saved and the runtime instance assigned to it restarted.

Organizations using the Neuron ESB Party API directly in their .NET applications will be required to append the name of the runtime instance to the service address url. For example, previously the service address would look similar to following if the runtime instance name was “default64”:

```
<configuration>
  <appSettings>
    <add key="esbZone" value="Enterprise"/>
    <add key="esbServiceAddress" value="net.tcp://localhost:50000"/>
    <add key="esbServiceIdentity" value=""/>
  </appSettings>
</configuration>
```

If Port Sharing is enabled, the service address would now look like this:

```
<configuration>
  <appSettings>
    <add key="esbZone" value="Enterprise"/>
    <add key="esbServiceAddress" value="net.tcp://localhost:50000/default64"/>
    <add key="esbServiceIdentity" value=""/>
  </appSettings>
</configuration>
```

Lastly, the other implication to consider is performance tuning. When Port Sharing is disabled, performance tuning for the Neuron ESB internal subsystems is controlled by modifying the “Internal

Service Binding Settings” located on the Server tab of the Enterprise Zone as shown below. These parameters are typically modified to reflect the number of CPUs/Cores balanced against the load being placed on the services.

Hide Detail

Zone	Location	Description
Enterprise		

Apply Cancel Viewing Zone Enterprise

General Ports Transports **Server**

Client Reporting

Client Polling Interval: 15

Cleanup Interval: 60

Service Reporting

Service Polling Interval: 15

Performance Counters

☒ Enable Neuron Parties

☒ Enable Neuron Topics & Endpoints

Internal Service Binding Settings

Binding

Close Timeout	30
Listen Backlog	1000
Max Connections	200
Open Timeout	60
Send Timeout	60

Service

Max Concurrent Calls	208
Max Concurrent Instances	1008
Max Concurrent Sessions	800

However, when Port Sharing is enabled, the SMSvcHost.exe, (which hosts the Net.TCP Port Sharing Service) manages the TCP sockets on Neuron ESB’s behalf. This means that performance tuning will be controlled by the binding settings within the SMSvcHost.exe.config file. For more information regarding the configuration of tuning parameters and the creation of the config file:

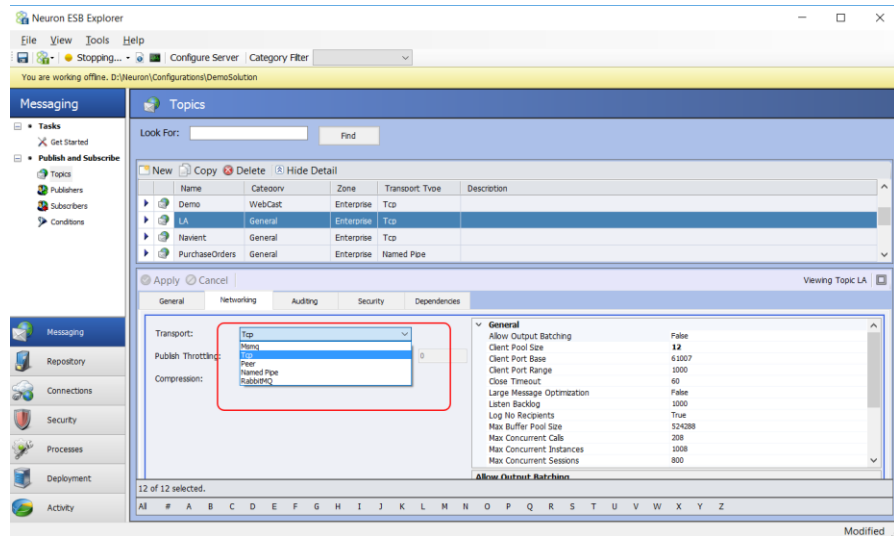
[https://msdn.microsoft.com/en-us/library/aa702669\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/aa702669(v=vs.110).aspx) .

The new Port Sharing option will greatly simplify the creation, deployment and management of multiple Neuron ESB runtime instances and solutions on individual servers!

Rabbit MQ Topics

Neuron ESB provides a hierarchical, Topic-based publish and subscribe model to mediate the routing of messages between Parties (Publishers and Subscribers that are hostable in .NET applications), Adapter, Service and Workflow Endpoints. Topic based messaging can be a great way of abstracting endpoints from one another. In essence, all endpoints become publishers and subscribers with one never having to have knowledge of the other. This allows organizations to easily change and modify existing solutions without interrupting the processing of other endpoints.

Neuron ESB is unique in that the Quality of Service and underlying Transport of Topics can be changed to better accommodate and more closely align with the business requirements of specific use cases. For example, durable, guaranteed, and reliable messaging as well as in-memory routing of messages, can be provided by simply changing the Transport property of a Neuron ESB Topic as show below:



When changes are made, the underlying Party API is self-aware of changes made at the server and auto configures itself to accommodate the Topic configuration changes, even when deployed and hosted on remote machines. This means that developers using the Party API do not need to know, nor do they directly have to work with or individually program the underlying transport configuration, transaction or QOS requirements. This is all controlled and managed at the server level. For the user who needs to publish messages, the code is as simple as this:

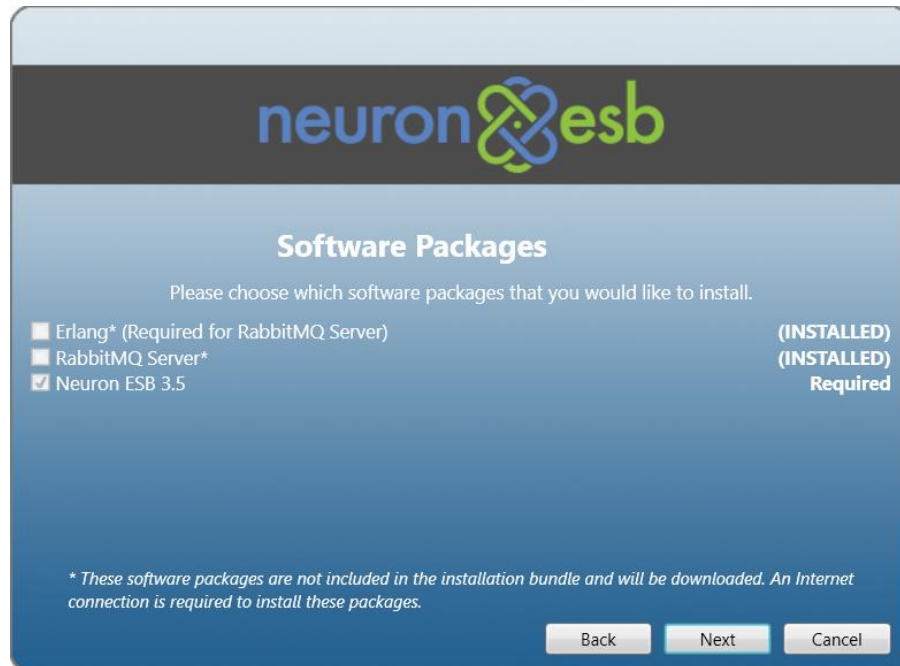
```
// Create an instance of a publisher
using (Publisher publisher = new Publisher("MyPublisher"))
{
    // catch any exceptions that may occur while connecting
    // to each individual topic
    PartyConnectExceptions exceptions = publisher.Connect();
    if (exceptions.Count < 1)
    {
        publisher.Send("MyTopic", "<Test>My Request</Test>");
    }
    else
    {
        // log the errors
    }
}
```

Neuron ESB 3.5 provides a number of Transports that users can select for Topics including Named Pipes, TCP, Peer, Rabbit MQ and MSMQ. Some provide durable, guaranteed, and reliable messaging such as MSMQ and Rabbit MQ. In the CU4 release, the Rabbit MQ Transport for Topics has been completely refactored to provide not only better performance, new manageability and more flexibility, but also better fault tolerance, reliability and security.

Updated Version Support

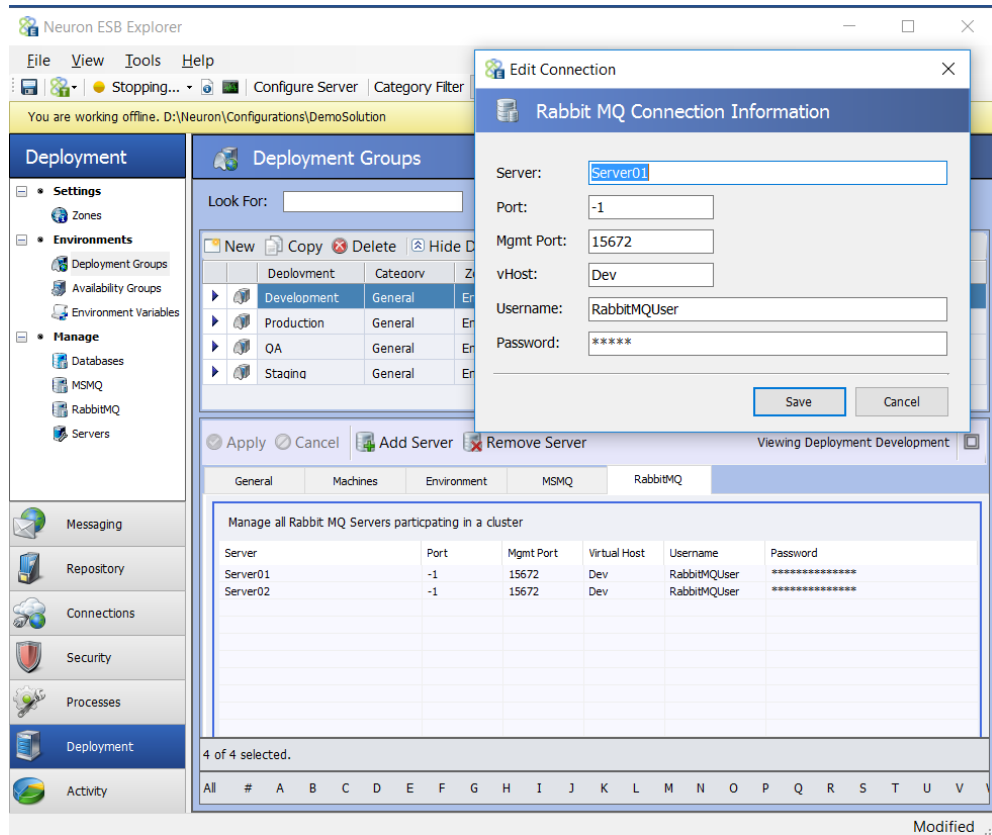
In the CU4 release, the first notable change is that our support for Rabbit MQ has been upgraded to support Rabbit MQ version 3.5.6 and Erlang 18.1. Using the Neuron ESB installer, users can optionally

select to install these during the setup process, or can download these from the locations list in our readme.html file or directly from the Rabbit MQ web site.



Configuration

The Rabbit MQ configuration for a Deployment Group can be found on the RabbitMQ tab of a Deployment Group by navigating to Deployment->Environments->Deployment Groups within the Neuron Explorer.



When registering Rabbit MQ servers for a Neuron ESB Deployment Group the following information is required:

<u>Property</u>	<u>Description</u>
<i>Server</i>	Name of the Rabbit MQ Server
<i>Port</i>	Port of the Rabbit MQ Server. -1 will use the default Rabbit MQ port of 5672
<i>Mgmt Port</i>	Port of the Rabbit MQ Management Portal. Default is 15672
<i>vHost</i>	The configured vHost of the Rabbit MQ Server. Default value is "/"
<i>Username</i>	Username to access Rabbit MQ.
<i>Password</i>	Password to access Rabbit MQ

Although some properties may seem fairly straight forward, others may not. For instance, the “*Mgmt Port*” represents the port of the Rabbit MQ Management Plugin. Neuron ESB requires this to be installed and configured for every Rabbit MQ server registered as it uses it to query for the health and message throughput rates which appear in both Endpoint Health and the Rabbit MQ Message Management console in the Neuron ESB Explorer. More information regarding the Rabbit MQ Management Plugin and how to install it can be found here: <https://www.rabbitmq.com/management.html>

As of Rabbit MQ version 3.3.3, the default username “Guest” cannot be used against the Rabbit MQ management plugin if the server name is anything other than “Localhost”. It is always recommended to create a dedicated username and password in Rabbit MQ that can be used by Neuron ESB.

High Availability and Failover

Neuron ESB allows users to enter multiple instances of Rabbit MQ servers to support the mirroring of the underlying queues that Neuron ESB will use for the Publishers and Subscribers created within the Neuron ESB Explorer. Rabbit MQ uses mirroring, rather than Windows Failover Clustering, to achieve High Availability of messages. More information regarding Rabbit MQ HA and its configuration can be found here: <https://www.rabbitmq.com/ha.html>

When multiple servers are configured, at runtime Neuron will use the first server it can connect to as the primary message server. If for any reason that server becomes unavailable, Neuron ESB will automatically failover over the next server in the list until it can find one that it can establish a connection against. If a connection can be established, publish and subscribe activities remain undisrupted at runtime; messages will not be lost and the failover will be invisible to the processes and users of Neuron ESB. Internally we detect connection issues and then cache and resend the messages when we fail to receive the original acks/nacks from the Rabbit MQ servers.

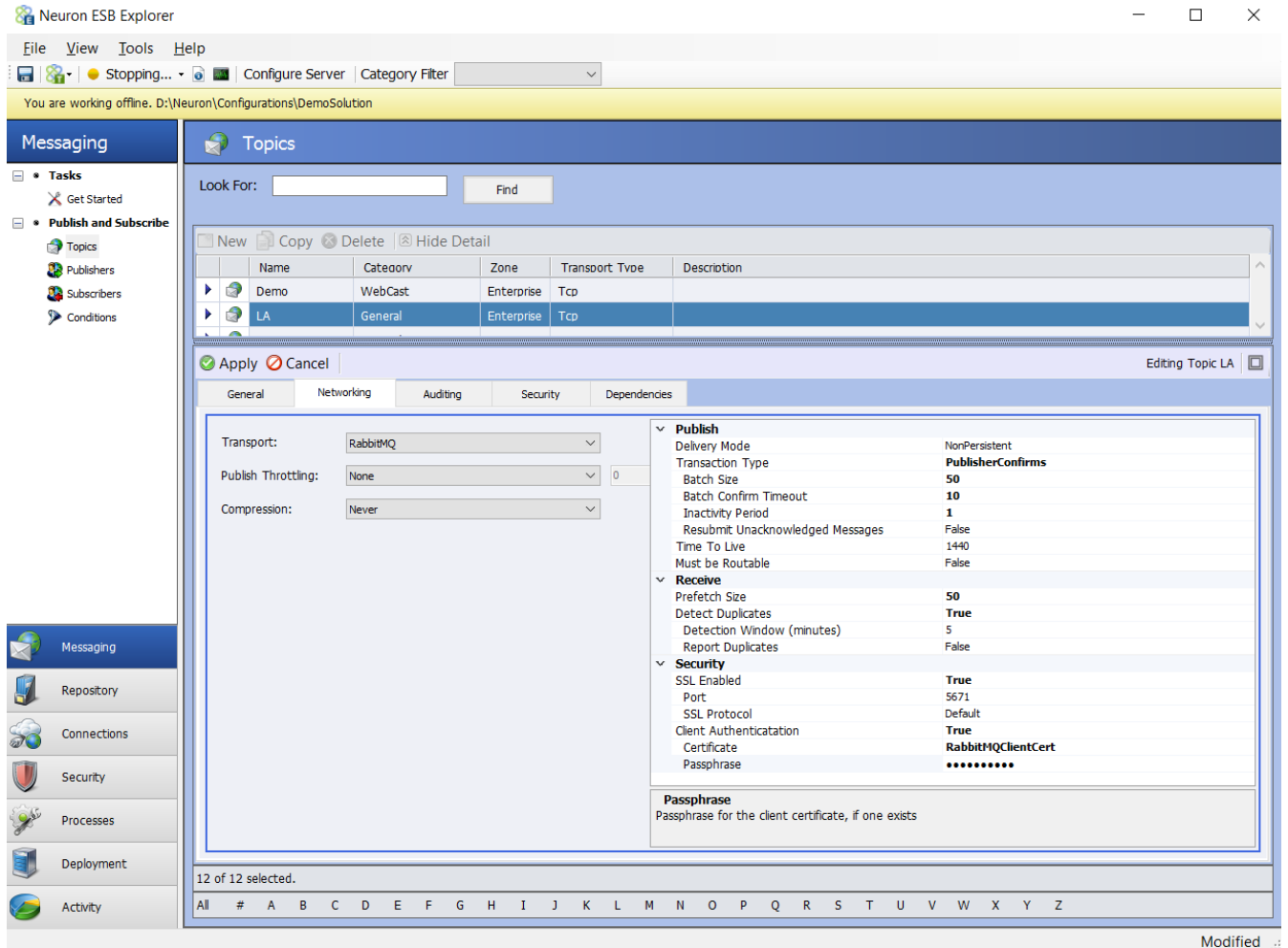
Multiple Environments and VHosts

New to CU4 is the inclusion of the vHost property. A common practice by some Rabbit MQ users is to establish specific vHosts to mirror their deployment environments. For example, there may be a vHost named “Development”, “QA” and “Production”. By supporting Rabbit MQ vHost environments, Neuron ESB users can easily map Neuron ESB deployment groups to their respective vHost environments. This will functionally isolate the underlying queues and exchanges that Neuron ESB creates for one deployment group from another on the same Rabbit MQ server instance.

Once the Rabbit MQ servers are registered with a Deployment Group, Topics and Parties can be configured to use the Rabbit MQ Transport.

Neuron ESB Topic Configuration

The Rabbit MQ Transport configuration is located on the Networking tab of the Topic by navigating to Messaging->Publish and Subscribe->Topics within the Neuron Explorer.



In the CU4 release a number of new properties have been added, some changed (*), others removed

<u>Property</u>	<u>Category</u>	<u>CU4</u>	<u>Description</u>
<i>Delivery Mode</i>	Publish	*	Persistent (i.e. Durable) or NonPersistent. Persistence ensures message is written to disk. If using Transactions, Persistence SHOULD be used.
<i>Transaction Type</i>	Publish	*	Controls the level of reliability for messages. Either 'None', 'PublisherConfirms' or 'Transactional' can be selected. 'PublisherConfirms' uses an asynchronous Ack/Nack protocol while 'Transactional' forces a commit/rollback on each message published.
<i>-Batch Size</i>	Publish		<i>Only for use with PublishConfirms type of transactions.</i> The number of messages that will be published to Rabbit MQ in a Publish Confirm transaction
<i>-Batch Confirm Timeout</i>	Publish	NEW	<i>Only for use with PublishConfirms type of transactions.</i> The number of seconds to wait after the Batch of messages have been published to receive all ACKS/NACKs from Rabbit MQ. Should be a value between 1 and 60.

<i>-Inactivity Timeout</i>	Publish	NEW	<i>Only for use with PublishConfirms type of transactions.</i> The number of minutes to wait after the last message sent before checking to determine if all ACKs/NACKs from Rabbit MQ have been received. Should be a value between 1 and 5.
<i>-Resubmit UnAck'd Messages</i>	Publish		<i>Only for use with PublishConfirms type of transactions.</i> If true and 'Detect Duplicates' are selected, messages that were published but were not acknowledged will be republished. This should only occur when dealing with clustered/mirrored Rabbit MQ instances. The republish would occur when a connection is reestablished with another server in the cluster.
<i>Time To Live</i>	Publish		A value in minutes that specifies how long messages are valid for delivery before they are expired (dead letter) and transferred as a failed message into the Neuron ESB database.
<i>Must be Routable</i>	Publish	NEW	If set to true, the message must be routable by Rabbit MQ. If the message cannot be routed to a destination queue, it will be stored as a failed message into the Neuron ESB database. **This will incur a significant performance penalty. This should NEVER be needed as Neuron ESB will dynamically create any missing queue or exchange on startup.
<i>Prefetch Size</i>	Receive	NEW	The number of messages that will be prefetched from the queue to transport layer. A value of 0 means unlimited.
<i>-Detect Duplicates</i>	Receive		<i>Only for use with PublishConfirms type of transactions.</i> If true, duplicate messages received by the Neuron ESB Party will be discarded. This can be used to provide once only delivery.
<i>-Detection Window</i>	Receive		<i>Only for use with PublishConfirms type of transactions.</i> The amount of time (in minutes) in which previously received message meta data will be maintained in memory to search against for duplicate messages.
<i>-Report Duplicates</i>	Receive		<i>Only for use with PublishConfirms type of transactions.</i> If true, if a duplicate message is discovered it will be logged as a Warning in the Neuron ESB Windows Event log.
<i>SSL Enabled</i>	Security	NEW	Connect to Rabbit MQ Server using only SSL
<i>-Port</i>	Security	NEW	SSL Port for all Rabbit MQ server connections
<i>-SSL Protocol</i>	Security	NEW	SSL Protocol to use for all Rabbit MQ server connections.
<i>-Client Authentication</i>	Security	NEW	Require Neuron ESB to provide Rabbit MQ Server a client certificate to authenticate against.
<i>-Certificate</i>	Security	NEW	Select a client certificate configured in the Security section of the Neuron ESB Explorer to authenticate against the Rabbit MQ Server.
<i>-Passphrase</i>	Security	NEW	Passphrase for the client certificate, if one exists

The values of these properties control how the Rabbit MQ transport functions for each Publisher or Subscriber that has a subscription to the configured Topic. In the CU4 release the following properties were removed because they were no longer relevant:

- Failed Message Path
- Auto Acknowledge

For greater clarification of their respective capabilities, the *Recoverable* property was renamed to *Delivery Mode* and *Reliability Mode* was renamed to *Transaction Type*. Also, in previous versions a user could only select to use Transactions if the persistent messaging was also selected. In CU4, that is no longer a requirement.

In most cases though then internal implementation of most of these properties have changed to provide, better performance, scalability reliability and fault tolerance.

Management

In previous versions, the Neuron ESB Rabbit MQ Transport channel would dynamically create the necessary underlying Rabbit MQ based infrastructure (e.g. Queues, Exchanges, Bindings, etc.) anytime a Neuron ESB Party connected to the bus or, when the Neuron ESB Topic started up. This eliminated the need for administrators to manually create and maintain the Rabbit MQ infrastructure Neuron ESB required. Consequently, this made the underlying Rabbit MQ Transport virtually invisible to anyone using the Neuron ESB Party API, and alleviated the need for administrators to manage additional infrastructure requirements.

However, what Neuron ESB did not do was keep the infrastructure in sync with changes made within the Neuron ESB Explorer. For example, if Topics/Parties were either renamed or deleted, their respective Rabbit MQ Queues and Exchanges would remain, unchanged and orphaned. If renamed, Neuron ESB would simply create a new set of Queues and Exchanges to use. If there were messages in the old Queues, they would remain but would have to be dealt manually by administrators.

Neuron ESB also created an unnecessary extra sets of Queues and Exchanges if the Recoverability property was set to True. All of this has been changed in the CU4 release.

Naming Conventions

The first notable change that we made in the CU4 was renaming of the underlying Rabbit MQ Queues and Exchanges. In previous versions of Neuron ESB, Queues and Exchanges used the following naming conventions:

Rabbit MQ Exchanges:

<InstanceName>.<Zone>.<Topic>.d.ad

Rabbit MQ Queues:

<InstanceName>.<Zone>.<Topic>.<Party>.d.ad

Where:

- *InstanceName* = Name of the Neuron ESB runtime Instance running the solution
- *Zone* = The name of the Neuron ESB Zone within the solution
- *Topic* = The name of the Neuron ESB Topic within the solution
- *Party* = The name of the Neuron ESB Party within the solution
- *d* = The “d” is appended if the Neuron ESB Topic was configured to be persistent
- *ad* = The “ad” is appended if the Neuron ESB Topic was configured to be deleted when Parties are no longer connected to the Topic

In the CU4 Release the naming conventions have been simplified to the following, eliminating the unnecessary extra sets of Queues and Exchanges to be managed:

Rabbit MQ Exchanges:

NEURON.<InstanceName>.<Topic>

Rabbit MQ Queues:

NEURON.<InstanceName>.<Topic>.<Party>

Synchronization

New in the CU4 release is the full management of the Rabbit MQ infrastructure Queues and Exchanges used by Neuron ESB. Previously, Neuron ESB would only create their respective dependent Rabbit MQ Queues and Exchanges on startup as well as when any Party connected to the bus. Neuron ESB now additionally handles renaming and deleting these if their respective counter parts within the Neuron ESB Explorer (e.g. Topics, Publishers and Subscribers) are renamed or deleted. For deletions, Neuron ESB will delete its respective underlying Rabbit MQ Queues only if there are no messages that remaining in the Queue.

Performance and Reliability

A number of internal changes have been made to the Rabbit MQ Transport to enhance reliability, error reporting, and functionality while at the same time significantly increase the performance of certain operations and message throughput.

Serialization Format

One of the first changes made was to the serialization method that Neuron ESB uses to interact with the underlying Rabbit MQ infrastructure. Previously, Neuron ESB serialized the entire Neuron ESB Message using Binary Serialization. With the introduction of CU4, this has been changed to a custom serialization method where only the body of the Neuron ESB Message is published, while the necessary internal Neuron ESB Message headers are custom serialized as Rabbit MQ custom header properties. This has several advantages; reduces CPU utilization, allows for the message body to be accessed without proprietary methods and reduces the overall payload size that the underlying Rabbit MQ infrastructure has to work with.

Publish Confirm Transactions

The Neuron ESB Rabbit MQ Transport supports both Transaction types that Rabbit MQ offers; their channel based Transaction model as well as their batched style Transaction model e.g. Publish Confirms. Both are Acknowledge, Negative Acknowledgement (ack/nack) based models. Users can learn more about Publish Confirms here: <https://www.rabbitmq.com/confirms.html> , as well as why Rabbit MQ introduced them: <http://www.rabbitmq.com/blog/2011/02/10/introducing-publisher-confirms> .

In CU4 a number of enhancements were made to the Publish Confirm model to make it more reliable as well as more performant. Neuron ESB now handles the ack/nack reconciliation process more efficiently, especially where multiple acks/nacks are received on a single event, alleviating unnecessary locking on the collection of messages we have to maintain internally. The CU4 release also exposes additional properties that can be used to finely tune the performance and throughput as well the reliability of the batch transaction such as *“Batch Confirm Timeout”* and *“Inactivity Timeout”*. These properties force Neuron ESB to call into Rabbit MQ for it to finish sending any pending acks/nacks, and only then do we resubmit the messages that we’ve neither received acks or nacks for. Messages we receive nacks for or where we’re notified by Rabbit MQ that that message is undeliverable, are automatically moved into the Neuron ESB Failed database table.

Receiving messages

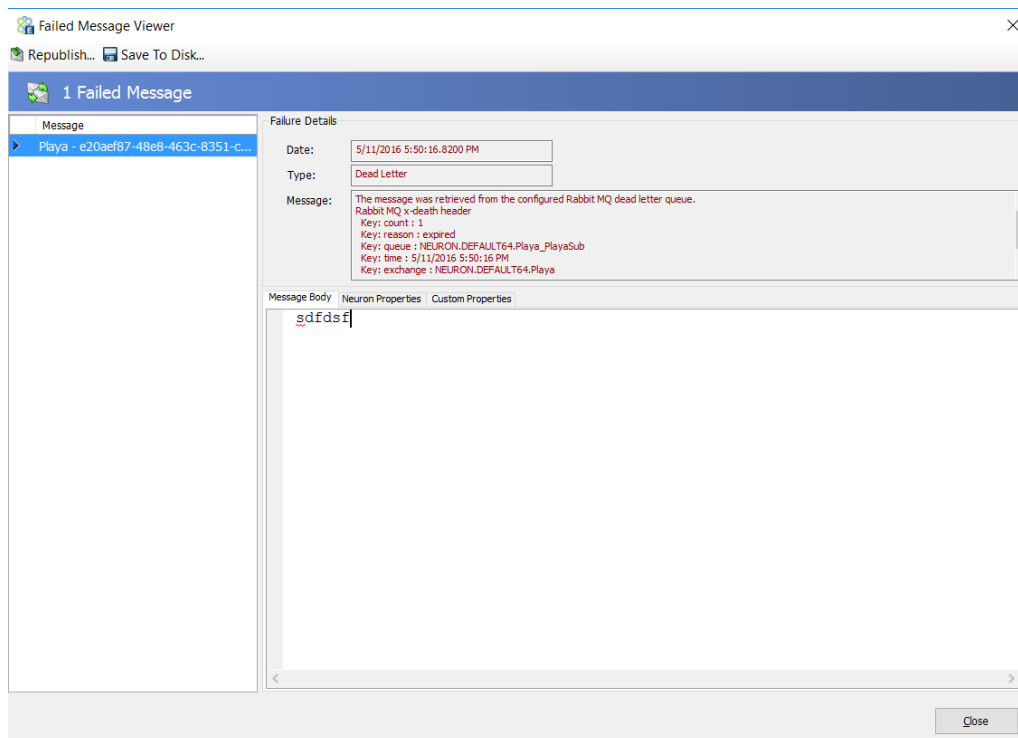
In previous versions the underlying Neuron ESB Rabbit MQ Transport would receive messages for its respective Neuron ESB Parties (or Dead Letter processing) by creating an underlying consumer and continually polling the queue. In CU4 this has been changed to use Rabbit MQ’s new event based consumer.

Dead Letter Processing

The CU4 Rabbit MQ Topic transport implementation includes a custom Dead Letter processor, Exchange and Queue to handle those messages delivered to Subscribers that exceed the configured Time to Live property. These messages are automatically detected and moved into the Neuron ESB Failed Message database table.

These messages will be available to be queried, viewed, modified and resubmitted by using the “Failed Message Viewer” window launched from the Failed Messages report located by navigating to Activity->Database Reports->Failed Messages within the Neuron ESB Explorer.

The messages recorded will have an Exception Type of “Dead Letter”. The failure message will contain failure details such as the name of the underlying Queue and the associated Exchange that the message expired in as well as the date time stamp indicating when the message expired.



CU4 Performance Comparison

The latest Neuron ESB public release was tested against the CU4 release using a relatively underpowered non server grade machine configured with 16GB RAM, Windows 7 64 bit, Dual CPU (quad core). The results of the testing can be found in the table below:

<u>Configuration</u>	<u>Messages Sent/Received</u>	<u>Message Size (Bytes)</u>	<u>CU3 (msg/sec)</u>	<u>CU4 (msg/sec)</u>
No Persistence/No Transactions	500,000	100	4,550	12,000
No Persistence/No Transactions	500,000	1024	4,200	9,400
Persistence/No Transactions	500,000	100	4,400	8,500
Persistence/No Transactions	500,000	1024	4,075	7,900
Persistence/Publish Confirm Transactions/Batch Size = 1000	500,000	100	1,750	3,000
Persistence/Publish Confirm Transactions/Batch Size = 1000	500,000	1024	1,300	3,000

Basic Improvements

A variety of improvements for the Neuron ESB Rabbit MQ Transport have been included in the CU4 release. For example, Request/Response type message communication when using Neuron ESB Sub Topics was not supported in previous releases. Neuron ESB now monitors more events from Rabbit MQ

to better respond to error conditions for everything from resource constrictions to unanticipated shutdown events. Failed message auditing has been refactored to decrease the opportunity for faults to occur when attempting to store failed or dead letter messages to the Neuron ESB failed database table

Security

With CU4's support for Rabbit MQ 3.5.6 comes support for SSL. SSL can be enabled in the Transport Properties located on the Networking tab of the Topic by navigating to Messaging->Publish and Subscribe->Topics within the Neuron Explorer.

Once SSL has been enabled, the Rabbit MQ SSL port needs to be provided and the SSL Protocol to use must be selected. Although Neuron ESB and Rabbit MQ supports both SSL2, SSL3, TLS, TLS 1.1 and TLS 1.2 by default SSL3 support is disabled by Rabbit MQ to avoid POODLE attacks. More about Rabbit MQ and its SSL support can be found here: <http://www.rabbitmq.com/ssl.html>

Client Authentication can also be enabled by providing a Certificate (registered within the Security section of the Neuron ESB Explorer).

Configuring SSL support is done by modifying the Rabbit MQ configuration file as well as registering several important Environment Variables for the machine. More information can be found here: <http://www.rabbitmq.com/configure.html> .

Neuron 3.5.4.700 (CU4) Change History

Runtime Services

NEW – Added support for the “*legacyUnhandledExceptionPolicy*” flag within the Workflow and Neuron ESB runtime hosts. This is enabled now by default. This will prevent unhandled exceptions from terminating the host environments. For pre CU4 behavior users can set the *Enabled* attribute to 0. The “*legacyUnhandledExceptionPolicy*” element is located in the runtime section of the host's .config file.

```
<runtime>
  <!-- the following setting prevents the host from closing when an unhandled exception is
        thrown -->
  <legacyUnhandledExceptionPolicy enabled="1"/>
```

NEW – The Neuron ESB runtimes depend on the operating system's temp directory for healthy runtime performance. Many of the Neuron ESB artifacts are dynamically compiled at runtime when first used, or when a modification is detected. Unfortunately, the Windows operating system has a limit to the number of temp files that can be created in the temp directory and it has no facility to periodically clean the directory out. On startup, Neuron ESB will now attempt to delete the contents of the temp directory to prevent startup or runtime failures due to too many files being present in the directory.

MOD – Neuron ESB uses Xceed libraries for various functions like FTP, SFTP, Compression, etc. These libraries have been updated to version 5.8

MOD – Neuron ESB provides both a Rabbit MQ Topic transport and a Rabbit MQ Adapter/Connector. All Rabbit MQ functionality has been updated to support Rabbit MQ version 3.5.6 and Erlang 18.1

MOD – The name of the installer package has been changed from *Setup.exe* to *InstallNeuronESB.exe*. This was done to address a security vulnerability on Windows based on a recommendation from the WiX team.

MOD - If the Neuron ESB Master Control service faults within a child service, the service name of the child service will be logged with the information in the Neuron ESB Event log

FIX – The auditing service could fail to audit a failed message if the Topic was missing. This would be recorded as a Topic is null exception. However, this would prevent the original exception information and message from being logged. If this condition is now detected, we will automatically the topic property to “TopicsMissing” so that the original message and error information gets logged successfully.

FIX – Neuron ESB TCP Ports could remain open if remote (cross machine) clients disconnected due to a network failure or a dirty disconnect. This could lead to leaking sockets, which eventually would cause Neuron ESB to run out of ports. To address this issue, the *Reliable Session* and *Inactivity Timeout* properties have been added to the Internal Service Bindings Settings located on the Server tab of the Zone within the Neuron ESB Explorer. By setting *Reliable Sessions* to true and accepting the default value for *Inactivity Timeout* (10 minutes), the TCP ports will be closed after 10 minutes. If the remote party re-connects within the 10-minute window, it will connect to the previously connected open ports.

FIX – Within each Neuron ESB Party is a management thread routing that periodically checks for version changes against the Neuron ESB server. This could sometimes generate the following warning message in the event log:

Event Info: A problem occurred while trying to check the server configuration version This request operation sent to net.tcp://localhost:50000/DEFAULT/ESBConfigurationService/ did not receive a reply within the configured timeout (00:01:00). The time allotted to this operation may have been a portion of a longer timeout. This may be because the service is still processing the operation or because the service was unable to send a reply message. Please consider increasing the operation timeout (by casting the channel/proxy to IContextChannel and setting the OperationTimeout property) and ensure that the service is able to connect to the client.

Once this was generated, it would sometimes generate the same warning on every version check. Also, between checks we would sleep the thread for the amount of time specified for the Client Refresh Interval. This could unnecessarily lock the party during shutdown and clean up procedures.

The former has been resolved by checking the state and recreating the proxy for the configuration service as necessary. The latter has been resolved by implementing a manual reset event.

FIX - When Neuron ESB was starting up, adapters could be started prematurely and could begin processing messages before other subsystems were ready. This could cause various issues like message loss.

FIX - On startup, if the amount of time to start the Neuron ESB service exceeded the configured limit in the esbservice.exe.config file (default is 10 minutes) a timeout exception would be thrown in the subscribing endpoints, but would not be properly logged.

FIX - If a custom Business Process step was being used in a process and the assembly was not present at runtime, the error would be reported across all services that were running in the Neuron ESB Runtime.

FIX - When configuration changes are detected we now perform a full garbage collection and reclaim heap from the generation 2 heap and release unused memory pages back to the operating system. This previously could appear as a memory leak.

FIX - An *ObjectDisposedException* could occur when sending pings or heartbeats from Neuron ESB child services.

FIX – Machine names longer than 15 characters are now supported for Neuron ESB produce License key generation and validation during the installation and usage of Neuron ESB. In previous editions license keys that were generated for machine names vs domain names were limited to the NetBios naming limitation of 15 characters.

FIX – The Neuron ESB Master Control service could throw timeout exceptions when a configuration change is made.

FIX - If the ESB Configuration (solution) path configured for the Neuron ESB runtime included a trailing slash, all Availability Groups (ESBHost.exe) would fail to start.

Database

NEW – Added the ability to Add Groups to the security tab of a Neuron ESB database within the Neuron ESB Explorer.

FIX - If users were removed from the security tab of a Neuron ESB database within the Neuron ESB Explorer, they would not be removed from the *NeuronUsers* database role on the target SQL Server.

FIX – The Neuron ESB Purge and Archive SQL agent jobs were not resetting the ID of the identity columns.

FIX – Configuring a deployment group to use a database may cause the following exception to be raised in the Neuron ESB Explorer:

System.Exception: The configuration cannot be saved due to the following issue(s):

1. Deployment <deployment group name> references state database myserver.net.cn.dx\<sqlinstancename>,24851.<databaseName> which could not be found. The database must exist in the same zone as the deployment zone or in (All Zones)

The above issue(s) must be resolved before the configuration can be saved.

This was due to incorrect parsing of the machine name.

Messaging

NEW – Extended JSON support – The *ToJsonObject()* has been added to the Neuron ESB Message object. This will return the Neuron ESB Message body as a *Newtonsoft.Json.Linq.JObject*. This is accessible within either a Workflow or Business Process. It will allow users to use .NET's dynamic object and write code similar to this in a Code editor window:

```
dynamic jo = context.Data.ToJsonObject();  
var myToken = "Bearer " + jo.token;  
var userName = jo.username;
```

MOD – TCP/Named Pipes Topic – Connected Neuron ESB Parties are now pinged in parallel to determine if they are alive.

MOD – TCP/Named Pipes Topic – Client expiration interval (the amount of time before cached client is removed from the connected client pool) has been set to twice the amount of the Send Timeout.

MOD - MSMQ and Rabbit MQ Topics - The underlying MSMQ and Rabbit MQ channels for Neuron ESB Parties will read messages from their respective queues ONLY IF the *OnReceive* event is wired up in the Neuron ESB Party. This only works with the Party object. This Will NOT work when using a Publisher or Subscriber object. This prevents message loss if a user connects the Party before wiring in the *OnReceive* event.

MOD – Rabbit MQ Topic Transport – This transport has been completely refactored and enhanced, over both significant performance and feature capabilities.

FIX – When using the Party API on the same machine as the Neuron ESB Runtime, users could send messages before the Neuron ESB runtime was fully initialized. This has been fixed.

FIX – Editing or creating a TCP topic would sometimes cause all TCP topics to have their ports change and recycle.

FIX – When changing the transport property of a Topic, Neuron ESB would incorrectly report as a warning that it cannot start workflow endpoint X because it was already in the collection of running workflow endpoints. This is expected and has been changed to an informational logging event.

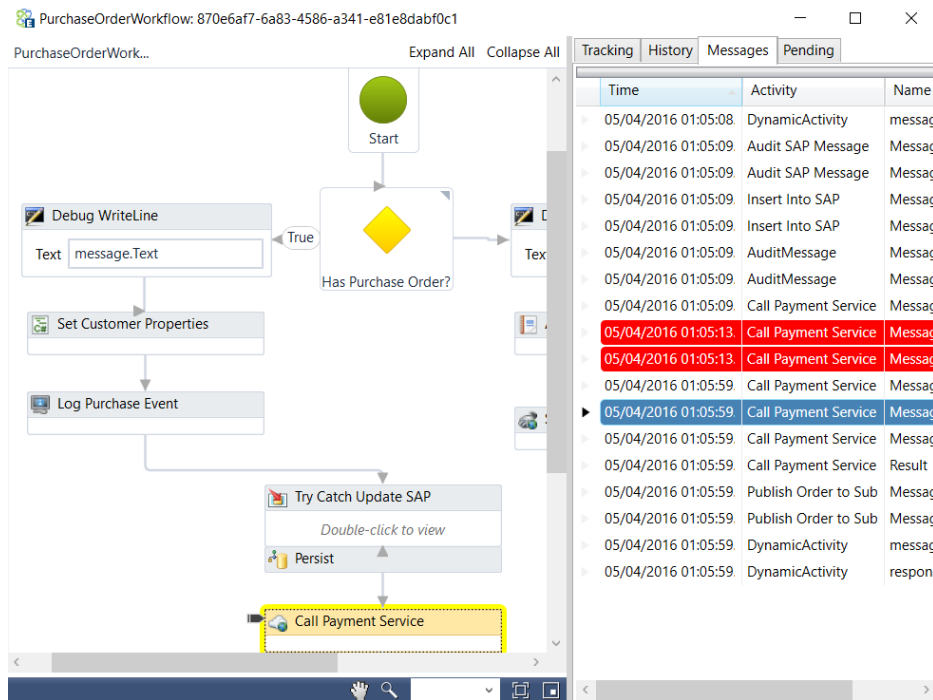
FIX – MSMQ Topic – When new sub topics are added to an existing MSMQ based Topic and the subscriptions of their dependent Parties AND there are Neuron ESB Parties (Publishers/Subscribers) currently connected to the topic, the currently connected Parties would not recognize the new sub topic subscriptions.

FIX – TCP/Named Pipes Topic – The Neuron ESB TCP or Named Pipes Publishing service would sometimes throw errors on Disconnect or Shutdown.

FIX – TCP Topic – TCP properties could sometimes get out of sync with Neuron ESB Parties trying to connect. For example, if reliable was set to true on the server, Parties would try to connect with the reliable bit set to false, causing a mismatch error.

Workflow

NEW – A new Messages tab has been added to the Workflow Tracking Detail page which lists all of the message variables and arguments generated from the Workflow instance. This can be used to view the message values at various points in the workflow as well as edit and resubmit them back to endpoints or topics.



NEW – A new “*Throw with Message*” workflow activity has been added that will throw an exception and cause an associated message to be written to the Failed Message database table when the workflow aborts. This was created to replace the existing *Throw* workflow activity which did not log the last message recorded to the Failed Message database table.

NEW – The existing Persist Workflow Activity has been replaced. The new Persist Workflow Activity will now refresh the Environment Variables argument if the workflow aborts and is later restarted at the persistence point. This means that if the Workflow is using environmental variables as parameters and their incorrect value caused the Workflow to abort, the user could easily correct the value of the Environmental Variable and then restart the Workflow instance which would now get the refreshed values.

MOD – Transform Message and Transform XML Workflow Activities – support for passing in variables has been added. Variables can be accessed by using `{variable:var_name}` in a property value. Also, the Parameters editor was not visible to edit the activities.

MOD - The Adapter and Service Endpoint workflow activities no longer attempt internal retries if sending a message fails.

MOD - The *IsDesigner* property has been added to the C# and VB.NET code workflow activities so that at runtime users can determine whether or not the activity is executing inside of Neuron ESB Explorer.

MOD – Extended JSON Support. All Code editors for Business Processes and Workflows reference the following assemblies and include their relevant using statements to make it easier to work with JSON messages:

Modified all code editors to include the following libraries:

```
"Newtonsoft.Json, Version=6.0.0.0, Culture=neutral, PublicKeyToken=30ad4fe6b2a6aeed",  
"Microsoft.CSharp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
```

Added following using statements to all the code editors:

```
using Newtonsoft.Json;  
using Newtonsoft.Json.Linq;
```

FIX - When attempting to restart an aborted workflow, if the "*error processing*" message was displayed there was no useful error information logged or made available to the user.

FIX – Workflow Tracking Detail – Date time based columns were not always sorted properly, making the events appear to be out of order.

FIX - Workflows were being reported as Unloaded instead of Canceled or Completed in the Workflow Tracking. The Unload event was sometimes getting processed after the Canceled or Completed states were reported, and instead of being Canceled or Completed, the workflows appeared as Unloaded.

FIX - If the Party configured for a Workflow Endpoint threw an error at runtime, or an error occurred during connection (except for connecting to primary topic) the error would not be recorded in the Neuron ESB logs or in the Neuron ESB event log. The Workflow Endpoint would go into a failed state, but nothing to indicate why.

FIX - When a Workflow Endpoint started up (in the constructor) and if any condition was present preventing Neuron ESB from loading any of the workflow assemblies in the workflow directory, Neuron ESB would ignore the exception and allow the workflow endpoint to start. We now report the error for the workflow endpoint and prevent it from starting.

FIX - Previously, we loaded all assemblies located in the workflow directory for every workflow endpoint, regardless if the workflow endpoint referenced the particular assembly. This meant if there was an issue loading any one assembly it would cause all workflow endpoints to fail.

FIX - If there were issues encountered when the Workflow Endpoint started the GUID of the workflow endpoint, rather than the name of the endpoint would be logged.

FIX - Running workflow commands have a timeout associated with them. By default, the timeout is an hour. Periodically, that timeout would be extended for workflows that are actively executing. Due to a logic issue, the default timeout could be lost resulting in re-queuing and re-executed the command every 5 minutes. The following describes the behavior users would have seen:

User submit a message to a topic that executes workflow A. Workflow A has an activity in it that takes 10 minutes to complete. 5 minutes later, workflow B gets executed, yet the user only published one message to the topic. The user looks it up in workflow tracking and sees that workflow B has the same initiating message as workflow A. 5 minutes later, workflow C will get kicked off with the same message.

FIX - At runtime the Service Endpoint workflow activity could report a *NullReferenceException*

FIX - The *environmentVariables* argument to a workflow was *IReadOnlyDictionary<string, string>* and the *EnvironmentVariables* property of the *Execute Process* workflow activity was *Dictionary<string, string>*. This meant that the *environmentVariables* argument could not be used directly by the *Execute Process* workflow activity. This has been corrected.

FIX - The *<assemblyBinding>/<probing>* path did not include the Workflows subdirectory where custom workflow activity DLLs are stored. If a custom workflow activity was created and the DLL was added to the Workflows directory it could be successfully used in the workflow designer. However, at runtime the workflow engine would report an error because it was unable to resolve the custom activity type.

FIX - In some cases the workflow host would not log when a workflow definition failed to load correctly.

FIX – The Service Endpoint Workflow Activity never set either the HTTP Status Code or Status Description properties of the Neuron ESB Message. If the call failed, any check of the status code would return incorrectly return a 200;

FIX - If a user had a previous solution opened attempted to start an aborted workflow a key not found in collection error: *System.Collections.Generic.KeyNotFoundException* could be generated because we could not find the workflow endpoint in the current solution

FIX - Workflow Code Editors - Apply does not persist changes if user navigates away from editor window and then navigates back.

FIX – When testing a Workflow in the Workflow Designer that uses either the Audit Message, Publish Message or Service Connector workflow activities (live activities), a *SourceID* and *Topic* must always be provided through the “Edit Test Message” dialog box. However, other workflow activities that create messages may not have the Source ID set. If one of the “live activities” follows it could throw an exception because the Source ID is missing, even though it was set in the “Edit Test Message” dialog. This was only an issue when testing in the Workflow Designer.

FIX – Using the *Terminate* Workflow Activity failed to log the message to the Failed Message database table and would instead log a “*Object reference not set to an instance of an object*” exception message.

FIX – When a message was logged to the Failed Message database table due to the Workflow aborting, the Workflow Name, Workflow Endpoint Name and Workflow Instance ID properties on the failed message were not being set.

FIX – When shutting down the ESB Service, the following errors could be logged:

Can't perform operation STOP# on Workflow Endpoint <workflow name>, running under <availability group name>, because Cannot write to a closed StreamWriter.

Can't perform operation START# on Workflow Endpoint <workflow name>, running under <availability group name>, because Cannot write to a closed StreamWriter.

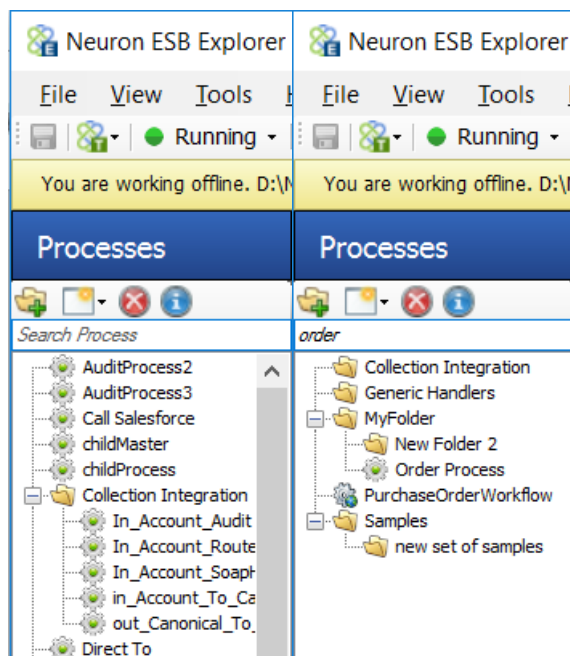
FIX – If a Workflow instance aborted followed by the Workflow definition being modified by a user and then an attempt to restart the Workflow instance was made after the modification, the following error would be logged:

System.Activities.WorkflowApplicationAbortedException: The WorkflowApplication has been aborted because a Load or LoadRunnableInstance operation threw an exception. Create a new WorkflowApplication object to try loading another workflow instance.

This was because we were trying load the aborted workflow definition from disk rather than the definition originally stored with the workflow instance.

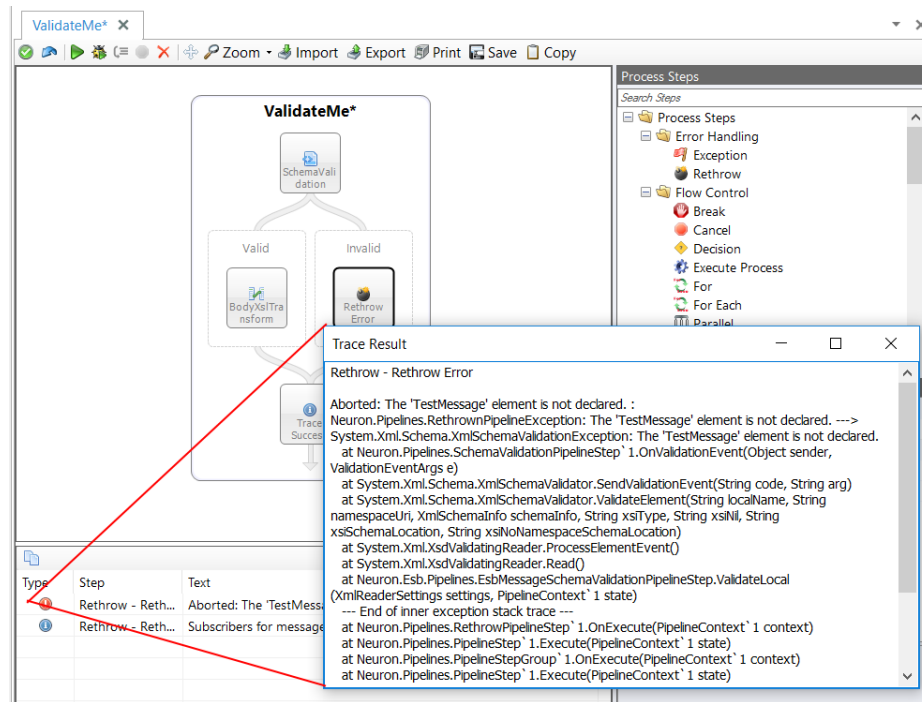
Business Processes

NEW - A search textbox feature has been added to the toolbar of the Business Process library.



NEW - Users can now double-click or right click (select "Display" from the context menu) on the row within the Trace window within the Business Process Designer display the full text, including its source

process step in its own Window. Additionally, the “Step” column of the trace window now displays the type of step followed by its label/name.



MOD - Modified how we write error information to the trace window during testing of Business Processes. Previously, we just wrote out the exception message. Now we write out the exception message, followed by the stack trace.

MOD - ODBC Process Step - To make the ODBC process step easier to use we added the same parameter mapping UI that is used in the SQL Store and SQL Query steps. This allows users to make database calls without using our proprietary message format.

MOD – Extended JSON Support. All Code editors for Business Processes and Workflows reference the following assemblies and include their relevant using statements to make it easier to work with JSON messages:

Modified all code editors to include the following libraries:

"Newtonsoft.Json, Version=6.0.0.0, Culture=neutral, PublicKeyToken=30ad4fe6b2a6aeed",
 "Microsoft.CSharp, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"

Added following using statements to all the code editors:

using Newtonsoft.Json;
 using Newtonsoft.Json.Linq;

FIX – Debugger - If a user launched the Debugger to test a Business Process within the Designer and stopped debugging part way through the process and closed the process, the last Process Step that the debugger was stopped on could still retain the debugging state and Yellow color.

FIX – Debugger – Attempting to access the Neuron ESB configuration variable while debugging would result in a null object.

FIX – The Trace Process Step now exposes the Name and Disabled properties.

FIX - Double clicking on a Business Process would not open the Business Process in the designer.

FIX - Changed several Business Process steps that used the Clone(false)method internally which did not preserve the existing Message ID. If these Business Process steps (e.g. json, xml query, table query, set property or call service) were used in the middle of a request/response flow (e.g. client connector calling a service connector or query/solicit response adapter), the calling client would time out waiting for the response message because the original Message ID would've been changed.

FIX - When testing the Service Endpoint Process Step within the Business Process designer, if a user tested with a Source ID of a Party that the Service Connector endpoint was configured to use, an "*Object reference not set to an instance of an object*" error would be thrown.

FIX – When testing a Process in the Business Process Designer that uses either the Audit Message, Publish Message or Service Connector workflow activities (live activities), a *SourceID* and *Topic* must always be provided through the “Edit Test Message” dialog box. However, other Process Steps that create messages may not have the Source ID set. If one of the “live process steps” follows it could throw an exception because the Source ID is missing, even though it was set in the “Edit Test Message” dialog. This was only an issue when testing in the Business Process Designer.

FIX – Processes from one solution may be opened in a new solution if the processes were not closed prior to opening the new solution

FIX - If a user attempts to delete a process but clicks "no" on the confirmation message dialog box, the Neuron ESB Explorer would keep the configuration state as modified.

FIX – Debugger – When using the debugger in the Business Process designer, pressing the Stop button could generate a Null Reference Exception

FIX – Within either a Transform – XSLT or Validate – Schema process step, a user opens the xsl/xsd editor and attempts to make any edit that in the process of the edit would result in an invalid xml document, an unhandled exception would be thrown.

FIX - When using the Set Property process step, and more than one item is added to it, the Apply button may remain disabled.

FIX - Excel to XML process step: target namespace was getting assigned to the rows returned instead of the root node.

FIX - When testing the Transform - XSLT process step, by setting the "XSLT Name" property, if a user changed the xslt in the repository between tests, the updated xslt would not be used until the Neuron ESB explorer was closed and reopened.

Monitoring and Reporting

MOD – Endpoint Health - Changed default refresh interval from 5 to 10 seconds.

MOD - MSMQ Management – PowerShell MSMQ Create Scripts generated by the Neuron ESB Explorer now includes commented lines that can be uncommented to create queues successfully on a Windows Failover Cluster

FIX - Neuron ESB text based logging could throw and log an *"Object reference not set to an instance of an object."* Exception.

FIX - When navigating to Endpoint Health, the Neuron ESB Explorer was not detecting the current deployment group of the existing local online service, or if offline mode, detecting what the local service is configured for and auto selecting it.

FIX - When in the Message Viewer (via Message History or Failed Message reports) and multiple messages are displayed, when a user clicks back and forth on the properties tab of each message the expires property value would consistently decrement.

FIX – Endpoint Health - Clicking the refresh button would cause a refresh, but further monitoring would stop.

FIX – Endpoint Health - starting endpoint health monitoring by pressing *"start monitoring"* button would toggle the button text to *"stop monitoring"* as expected. However, after the first refresh interval, the button text would toggle back to *"start monitoring"*.

FIX – Endpoint Health - Restarting an endpoint in Endpoint Health could result in an unhandled *System.NullReferenceException: Object reference not set to an instance of an object.*

FIX - Performance counters for Workflow Endpoints were not being created properly by including their instance name. This would prevent statistics of the workflow endpoints from being reported and correlated back to the correct instance of the Neuron ESB runtime. The issue only arose if there were multiple instances of the Neuron ESB runtime installed on a machine.

FIX – Endpoint Health - There could be occurrences where an error may occur in a Publish mode adapter, functionally disabling it, yet it would still be reported as Started without errors reported within Endpoint Health.

FIX – Endpoint Health - Users could encounter error conditions with workflow endpoints that would not be logged appropriately, preventing the workflow endpoint from being updated to a failed state in Endpoint Health

FIX – Management REST API - Specifically calling a GET on <http://localhost:51002/neuronesb/api/v1/Runtime>, would only return the information for one Neuron ESB runtime instance even if multiple instances were installed on the box.

FIX – Workflow Tracking – When running a workflow tracking report filtering on the State field and Date Range, the date range settings were ignored.

FIX - Message Viewer for Message History and Failed Message reports - The message body window did not resize correctly when displaying binary messages.

FIX - MSMQ Management - PowerShell MSMQ Create Scripts generated by the Neuron ESB Explorer includes all queues, not just missing queues, causing errors if the script is run multiple times.

Adapters

NEW - SAP Adapter – Added support for custom IDOC segment versions

NEW - SAP Adapter – A new feature has been added to allow users publish the raw IDOC to the bus. Now there are two options - "Typed" will publish a strongly-typed XML document representing the IDOC. "String" will publish the raw IDOC data.

NEW - SAP Adapter – Added Unicode support for connections.

NEW - File Adapter - added ability to disable lazy writes.

NEW – File Adapter – On Publish mode, users can now use the new “Encoding Override” property to overwrite the character set encoding of the document that is received and published to the bus. For example, if the native encoding of the file on disk that Neuron ESB is reading is UTF8 and there are western European characters (i.e. umlaut, etc.) they will not be properly encoded when published to the bus. But if the new “Encoding Override” property is set to 1252, then the data will be encoded as readable as expected.

NEW - File adapter - added the error reporting and error on publish properties.

NEW - MQSeries Adapter – Added new translation options, “None”, “CharacterSet”, “UTF8” and “Unicode”.

NEW - MQSeries Adapter – Added the error reporting and error on publish properties.

NEW - MQSeries Adapter – Added a new property, “Remove BOM”.

NEW - MQSeries Adapter - added new translation option which is the default (acts as the old “none” option). New Translation options are “None”, “CharacterSet”, “UTF8” and “Unicode”.

NEW - Dynamics CRM Adapter – Can be configured to use specific user credentials based on deployment group and environment variables. Username/password and windows credentials that have been created in the Neuron ESB security section now appear as a drop down box in the adapter’s property grid.

NEW - SharePoint Adapter - Added Security selector to assist in cross-domain integrations. User can select to use the service (or Run As) account or to type-in username, password and domain.

NEW - SharePoint Adapter - Added Request-Response mode for returning the document or list ID and the URL as a response

NEW - SharePoint Adapter - Added support for setting the library column values to update in code

NEW - SharePoint Adapter - Added support for setting the content type for uploaded documents or added list item.

NEW - NetSuite Adapter - Added support for SOAP API version 2015_2.

NEW - NetSuite Adapter - Support async calls for list operations - getList, addList, updateList, upsertList, deleteList, initializeList and search. Async operations have also been added to the Metadata Generation wizard.

NEW - NetSuite Adapter - token-based authentication and wsdl 2016.1 support has been added. For wsdl versions 2015.1 and earlier, users can still use username/password authentication. For wsdl versions 2015.2 and later, users can use consumer key and secret or token id and secret.

NEW – SMTP Adapter - Two new properties were added to allow users to better manage throughput. 1.) *UnlimitedMessagesPerSession* - set to false by default and 2.) *MaxMessages* - set to 1 by default. This will now allow users to send N number of messages on the same session.

MOD – Renaming of Microsoft Dynamic adapters. In the CU4 release we’ve renamed several Dynamics adapters so that the names provide better clarity of feature and function.

<u>Original Name</u>	<u>New Name</u>
CRM Subscription Adapter	Dynamics CRM 4.0 Adapter
Dynamics CRM 2013 Adapter	Dynamics CRM XRM Adapter
eConnect Adapter	Dynamics GP eConnect Adapter

MOD - Removed/retired Microsoft Dynamics CRM4 Adapters from active release. These are now only available on request.

MOD – Added the *RaiseReceiveFailedFromEndpoint()* event to the Adapter Framework. There is no longer a need to manually create and call the audit proxy when wanting to send a failed message to the Failed Message database table if a custom publishing activity within a custom adapter fails.

MOD – Removed the “Audit On Failure” property for the following publish mode adapters. We now internally always audit the failure to the Failure database table.

- FTP/FTPS
- SFTP
- ActiveMQ
- Azure Service Bus
- POP3

- Microsoft Exchange
- SAP
- SalesForce
- SQL Service Broker
- MQSeries **
- File **

** NOTE: in 3.5.4, audit on publish failure and the error on publish properties were added.

MOD - Socket Adapter – Certificate selection is now a drop down populated through the Neuron ESB Repository.

MOD - File Adapter – now changed to a warning - "File adapter is not processing file '<filename>' because exclusive read-write access could not be attained after 10 attempts"

MOD - File Adapter - In publish mode, if the directory didn't exist we were creating it. Now we will not create the directory.

MOD - MQSeries Adapter – Now calls the RaiseOfflineEvent on Disconnect()

MOD - MQSeries Adapter – Changed how we handle replies. Previously we were manually setting correlation properties. Now this uses the standard CreateReplyMessage() method from the ESB Message.

MOD - MQSeries Adapter – Disconnect() was being called twice on shutdown when in Publish mode.

MOD - MQSeries Adapter – Removed the "Maximum Get Retry Failures" property. This has been made obsolete by the "Error on Polling" and "Error Reporting" properties

MOD - NetSuite Adapter - Updated XML Sample generation to properly handle recordref fields. If there is a "record" element for request messages, we remove the "type" attribute from any element that is an xsi:type="RecordRef". If there is a "get" or "getList" element, we modify the type attribute to be equal to the element that was selected in the browser.

FIX - Socket Adapter – the Solicit Response and Request Reply modes were operating differently than the other adapters. For Solicit Response, the adapter will now receive a request from Neuron ESB and send the request over the socket. For Request Reply, the adapter will receive a request from the socket, publish the request to the bus, and return the reply.

FIX – SQL Service Broker Adapter – users could encounter the following error while publishing to the bus.

System.Data.SqlClient.SqlException: A transport-level error has occurred when receiving results from the server. (provider: Session Provider, error: 19 - Physical connection is not usable)

This could be due to the SQL server becoming temporarily unavailable on the network. We now will detect the error and clear the connection related connection pool.

FIX - SAP Adapter - Custom IDOCs were not validating against the adapter's generated schemas due to an extra element being nested

FIX - SAP Adapter - At runtime, when the SAP adapter receives an IDOC it will connect to SAP to retrieve metadata about that IDOC. The adapter will search that metadata for the IDOC segments. Custom IDOC segments were not being retrieved correctly.

FIX - File Adapter – When polling remote directories using UNC paths and the paths become unavailable for extended period of times, a resource leak could occur because the underlying SMB sessions and ports were not being released.

FIX - File adapter - if we attempted to sort the list and that failed we logged the error but published unsorted list. Moving forward, we'll throw the error so files don't get published out of order.

FIX – SFTP Adapter - With archive mode enabled, if the folder path had a leading slash i.e. “\” or “/”, it would not get resolved correctly.

FIX - SFTP Adapter – some SFTP servers that were non-compliant with the SFTP specification could cause the adapter to disconnect in the middle of its operations.

FIX - SFTP Adapter – When in Publish mode, the underlying SSH Disconnect could cause a thread Abort Exception which in turn would terminate the polling of the adapter.

FIX - SFTP Adapter – Login user paths were not getting resolved correctly, causing the errors to occur when the adapter had to navigate to different folders on the SFTP server. This was problematic when the login credentials used were mapped to a specific directory other than the root folder of the SFTP server.

FIX - MQSeries Adapter – The adapter could throw a " Memory stream is not expandable." when trying to set the RFH2, CIH, MDE or IIH headers. This occurred because the default structure length was not getting properly set.

FIX - MQSeries Adapter – Adapter was not honoring Include Metadata checkbox. Even if unchecked, the existing wmq properties in the incoming message would be used.

FIX - MQSeries Adapter – If users had 7.0 or greater of MQSeries installed they would receive an error when attempting to populate the list of queue managers installed using the property grid of the adapter

FIX - MQSeries Adapter – The adapter was not polling at the configured interval in Publish mode because of a missing multiplier.

FIX - MQSeries Adapter – If the MQMD format is none, we will always use a byte array for a body. If the MQMD format points to different header, we'll use that header's format value to determine encoding. Previously, we were not converting appropriately using the correct encoding and character set values specified in the headers.

FIX - Dynamics CRM Adapter - the CRM Adapter was not able to generate XML samples when multiple entities/operations were selected.

FIX - Dynamics CRM Adapter - FetchXML allows users to retrieve values from linked entities, and assign an "alias" to the linked fields. The adapter handles aliased fields in the result set in a special manner, but did not handle aliased fields that are also entity references (i.e. a field in a linked entity that is actually a look up to a third entity).

FIX - NetSuite Adapter - if a user selected more than one entity/operation, the metadata generation only correctly created schemas and XML documents for the first one

FIX - NetSuite Adapter – A duplicate passport soap header could be applied causing an exception to be generated by NetSuite.

FIX - NetSuite Adapter – Modified to accommodate invalid Schema element names

FIX - Adapter Metadata Generation Wizard - If user changed the properties of a selected Adapter Endpoint, the property would be modified permanently on the endpoint.

FIX - Adapter Metadata Generation Wizard – Was not logging out and cleaning up connections appropriately on exit.

FIX - Salesforce Adapter – On startup, the validation routine was not being called appropriately. Also, once the connection was made successful and the session was created, it was persisted. If the user changed the login details to salesforce, those changes would not be used.

FIX - SMTP Adapter – Users could experience the following error when trying to send 100's of messages through the SMTP adapter if the target SMTP server was restricted in the number of messages allowed per session.

System.Net.Mail.SmtpException

Service not available, closing transmission channel. The server response was: Error: too many messages in one session

The error occurred because Neuron ESB would create an SMTP connection on every send, but was not disposing the SMTP client after the send.

FIX – Adapter Framework - If tryReceive() returns false because the Error On Polling was set to StopPollingOnError, we now call disconnect so resources are cleaned up and the RaiseOfflineEvent is called, updating Endpoint Health state. This ensures that users who create publish/polling based adapters with the framework will record their proper state when polling discontinues.

Service Broker

MOD – Client Connector – Neuron ESB no longer will return the Server HTTP header in the response.

MOD – Service Connector - Sending JSON encoded messages to service connector required users to enable the raw flag on the binding tab when using REST, regardless if the http Content-Type was set to application/json or text/json.

FIX – Client and Service Connectors - Verify to make sure the certificate exists in the configuration before attempting to retrieve it.

FIX - When attempting to override the HTTP StatusCode and Description from within Neuron ESB, the HTTP Response would be set correctly but the HTTP StatusCode and Description property were not being set in the HTTP Headers.

FIX – Client and Service Connectors - If a REST call is made, we now check to make sure the HTTP method has been set. Previously we just throw the following error:

System.NullReferenceException: Object reference not set to an instance of an object.

FIX – Service Connector - When windows client credentials were configured for the service connector, they were not being passed through to the service, unless binding environmental variables are used.

FIX – Client and Service Connectors - *ContractFilter mismatch* exceptions would be reported from the Neuron ESB global exception handler.

FIX – Client Connector - A *key not found* exception could occur when we attempted to retrieve the remote IP address and port of the inbound caller to the client connector endpoint.

FIX – Service Connector - Could not connect to an SSL server using TLS 1.2. Neuron ESB is now enabled to support TLS 1.1 and 1.2.

FIX – Client Connector - *Index out of range* exception would be returned if a query string with missing values were sent to a client connector e.g.

<http://localhost:8989/rest/?pickuplocationid=0011900000A&assessortype&status=assessment>

Neuron Explorer and Test Client

FIX – Neuron ESB Explorer - If an error occurred during the loading of saved user settings, the exception dialog did not contain the inner exception details.

FIX – Neuron ESB Explorer and Test Client - When trying to connect via WS-Discovery, the entries disappear before a user is able to click "connect."

FIX – Neuron ESB Explorer - Business Process selection for a Party - If the dropdown box for the direction is expanded and user navigates away and clicks anywhere on the "Selected Processes" list box, the current process in the window is removed from the Party.

FIX – Neuron ESB Test Client – The message body window did not resize correctly when displaying binary messages

FIX – Neuron ESB Test Client – Selecting clear or clear all did not effectively clear the message from the hex editor if the message was binary.

FIX – Neuron ESB Explorer and Test Client - Changed query time for WS-Discovery to 2 seconds from 5. Also the Connect window would continue to make WS-Discovery queries after being closed.

FIX – Neuron ESB Explorer - When a solution is locked by TFS and user attempts to open the solution using the Most Recently Used List, an unhandled Null Reference Exception would occur

FIX - Importing Business Processes in the Neuron ESB Explorer would not overwrite existing processes with the same name.

FIX – Neuron ESB Explorer - When a user saved any configuration change, the Neuron ESB Explorer would iterate through all Business Processes and deserialize them.

FIX – Neuron ESB Explorer - The Business Process's Version GUID was not being changed when the process was changed and saved within the Business Process Designer.

FIX – Neuron ESB Explorer - The Topic's version GUID was not being changed appropriately when a change was made to the Topic within the Neuron ESB Explorer.

FIX – Neuron ESB Explorer - *Object reference* error when trying to create a new solution could occur if business processes were not completely closed before attempting to create a new solution.

FIX – Neuron ESB Explorer - if a user navigates to workflow endpoint and during editing there was an error condition e.g. missing topic, etc., and the user presses the cancel button, the user would not be able to navigate away from the screen.

FIX – Neuron ESB Explorer - if a user clicked on subscribers or publishers, then without clicking/choosing an item (a subscriber or publisher entry) and clicked on show details, then clicked on edit subscriptions, a UI error would be thrown. The same error would occur with adapter endpoints when clicking on adapter bindings after clicking on show details without choosing an item.

FIX – Neuron ESB Explorer - in the XSLT document repository, if a user clicked on show details without choosing an item, the test button would still be enabled.

FIX – Neuron ESB Explorer - if a user selected file->export and a topic had an encryption key associated with it, the user would immediately get an error message and the export dialog would never be displayed. The error message would be: "*topic N has a dependency on key X but key X was not found*"

Neuron 3.5.3.409 Change History

Adapters

NEW – MQSeries Adapter – MQMDE header support has been fully added. These headers are now parsed and preserved as Neuron ESB message custom properties which can be accessed anywhere within Neuron ESB

MOD – Dynamics CRM Subscription Adapter - Adjusted response messages to include the external Id of the entity added or updated.

FIX – NetSuite Adapter – When using a NetSuite sandbox instance our call to resolve data center URLs returned the URL for the production site. Also made the use of the Role property optional as users can define a default web services role inside NetSuite directly.

FIX – NetSuite Adapter – Fixed issue when generating the Employee schemas. It seems that NetSuite moved the Employee element from one schema to another between different releases. We now handle potential differences across releases.

FIX – NetSuite Adapter – Fixed exception when creating schemas for timeentry and timesheet entities

NEW – NetSuite Adapter – Added support for dynamically sending user credentials to the adapter endpoint

FIX – SFTP Adapter - string.format index error could occur when publishing fails for sftp adapter endpoint.

FIX – SFTP Adapter – Operations timeout was being calculated in milliseconds rather than seconds.

Business Processes

FIX - Adapter Endpoint process step - if an adapter name or endpoint name was not found at runtime, a generic KeyNotFoundException would be thrown.

Workflow

FIX – Adding assembly reference to Workflow can cause serialization error when saved. When adding an assembly like System.Data.dll which contain C++ code and complimentary C++ namespaces, a serialization can occur when a user attempts to save the workflow within the Neuron ESB Explorer.

FIX – Referencing a custom DLL in a Workflow may cause the workflow endpoint to fail at runtime. When referencing a custom DLL in a workflow, the user would be able to successfully test the workflow and load the dll within the Neuron Explorer, but the workflow endpoint would fail to start because the runtime would not be able to find the dll. This was because the probing path did not include the workflow subdirectory.

Neuron 3.5.3.398 Change History

Adapters

NEW – Salesforce Adapter – Provide support for sending the following content types for Salesforce.com via a call to an Apex REST service – ByteArrayContent, FormUrlEncodedContent, MutipartContent and MultipartFormDataContent. These different content types are created in a C# step in a business process or workflow, and then saved using code similar to this:

```
context.Data.Bytes = content.ReadAsByteArrayAsync().Result;
context.Data.Http.Headers.ContentType =
    content.Headers.ContentType.ToString();
context.Data.SetProperty("sforce", "restMethod", "PATCH");
```

Where content is one of these types - ByteArrayContent, FormUrlEncodedContent, MutipartContent and MultipartFormDataContent. There is a corresponding property for the Salesforce adapter endpoint that also needs to be set to the same value. Note – the restMethod property needs to be either POST or PATCH.

MOD – WebSphere MQSeries Adapter – Added ability to set the Message Get Option Version for retrieving messages from MQSeries to support Version 2 MQMD extensions.

FIX – WebSphere MQSeries Adapter – Setting Encoding or CharacterSet MQMD header properties using Neuron ESB custom message headers as shown below did not set their respective MQMD properties of the MQSeries message sent by the adapter.

```
context.Data.SetProperty("wmq", "MQMD.Encoding", "546");  
context.Data.SetProperty("wmq", "MQMD.CodedCharacterSetId", "1200");
```

Service Endpoints

FIX – Using NTLM or Windows Security with Client connector would cause neuron log file to fill with errors due to WS-Discovery hello calls on the endpoint. The Neuron authorization manager has now been configured to ignore our internal WS-Discovery calls. Also, if authorization failed, we were returning the default 500 status code rather than specifically setting the REST HTTP status code to 401.

FIX – Client Connector and REST errors - If a client connector used a REST binding and there was any error during the sending of the message to Neuron's pub/sub infrastructure the REST status code returned to sender would be 200 i.e. OK, rather than a non-success code. Lastly, if specific exceptions did occur (i.e. timeout or fault exception) while using the REST binding, Neuron was not attempting to audit as a failed message.

Neuron 3.5.3.389 Change History

Adapters

FIX – Socket Adapters - The socket adapter would fail with errors when SSL was enabled. There was a ReceiveTimeout set to 2 seconds on the socket that was causing the SSL session to become invalidated.

FIX – Socket Adapters - When a socket adapter is reused and changed from processing a custom message to a standard ESBMessage (MessageBodyOnly is set to false after being true), the socket adapter continues to use the settings for the custom message header and not the default settings for an ESBMessage object (length at offset 0, size of length field as 4 bytes).

NEW – SharePoint Adapter - Added Query support to SharePoint lists and libraries.

FIX – Salesforce Adapter - "Custom" data format option that allows users to set the ESBMessage.Http.Headers properties and pass them along to Salesforce via REST calls was not being propagated properly.

FIX – File Adapter - The archive file after download property of the file adapter could not be set to true unless the Publish Metadata Only property was set to true

FIX – SFTP Adapter - sftp timeout property (in seconds) was being incorrectly interpreted as milliseconds. Effectively setting the timeout to 30 seconds, meant the actual timeout would be set to 30,000 seconds.

FIX – SAP Adapter – When using the SAP adapter to listen to incoming documents an exception in the listener could cause the ESB Service runtime to crash. This has been fixed.

FIX – MSMQ Adapter – Publish Mode - If user had “Connect on Startup” property set to false, the queue to monitor would never get created and in turn the adapter endpoint would generate an "Object reference not set to an instance of an object" error.

Business Processes

FIX – Service Process Step – Datagram i.e. one way service calls were not being executed properly. The SOAP Action (if present) was not being set properly if being passed in dynamically or configured within the Service process step’s property grid. The Session and Transaction properties were also not be properly set.

Neuron Explorer

MOD – A manifest has been added so that users no longer have to explicitly open the application as an “Administrator”

Neuron Test Client

MOD – A manifest has been added so that users no longer have to explicitly open the application as an “Administrator”

Neuron ESB Database

FIX - The 0009_UpgradeTo3_5_0.sql script was missing the 'dbo' schema name in several CREATE or ALTER statements. If a user ran the script manually in the course of an upgrade, errors would be generated because several of the objects could not be found.

Neuron 3.5.3.373 Change History

Neuron ESB Runtime

FIX - A condition in GlobalExceptionHandler.HandleError prevented errors from being reported properly. If the trace level was not Warning, then the error would not be output to the log. We changed it to log if the trace level is either Warning or Error.

FIX - "The given key was not present in the dictionary." error could occur when stopping an endpoint.

FIX – “The given key was not present in the dictionary” error could occur when validating an endpoint during shut down.

FIX – Extraneous errors were being logged when shutting down the Neuron ESB service.

MOD – The maximum number of heartbeat failures are now configurable via the esbservice.exe.config file (“MaxHeartbeatFailures” property). The current default is 2. This should be expanded in dev environments when changes are being made to management service. Once an endpoint (Adapter or service connector) reaches the maximum heartbeat failure level it will stop allow another server to take control (providing single instance is configured on the endpoint).

Endpoint Health

FIX – Attempting to stop an endpoint on all machines in a deployment group could fail if the hostname in the deployment group did match the environmental machine name.

Workflow

FIX - Invalid condition at Availability Group (AG) level causes an infinite loop of Warnings messages. If the AG heartbeat attempt failed, rather than stop the AG, the system would continually log the heartbeat failures in an infinite loop

MOD - Quartz.NET implementation for internal workflow scheduling has been updated to use our implementation of reliable database connection to overcome transient network issues.

FIX - Adding/removing an assembly reference for a workflow definition could cause validation errors to happen.

FIX – A fix was applied to the PublishRequestMessage activity. Replies when sent via a bookmark were not being routed correctly back to the parent workflow.

FIX – If the database could not be connected to during startup, Neuron ESB would continually spin up more instances of ESBHost.exe (Availability Groups) in an attempt to connect. This behavior has been modified. If an AG cannot connect to the database on startup, it will shut down and will have to be manually restarted by an administrator when the database is access.

NEW - The "Event Timeout" field on the workflow endpoint settings tab was hidden. We changed the UI settings to make the field visible on the form so that the workflow command timeout period can be customized for a workflow endpoint.

Business Processes

FIX - When using the Execute Process step, a user could experience a duplicate key or key not found exception when manipulating or working with Global Properties due to a race condition that could occur.

FIX - Updated the execute adapter process step to log to the neuron log files when RaiseAdapterInfo().

MOD – ODBC Process Step - Allows users to insert binary data into databases. If the binary, image or varbinary datatype is passed, we'll use the text within the xml element, assume its base64 encoded and attempt to convert to byte[] and pass to odbc parameter. If user passes a "file:/" url as the content, we'll use that to read in the byte[] array and pass that as the parameter to the stored procedure.

Neuron Explorer

MOD - When scanning for adapters, the ListNetAdapters method would fail if an adapter assembly contained an abstract class derived from ESBAAdapterBase.

FIX - Fixed the Export form so that selecting a category selected the processes that belonged to the category

FIX - Modified the logic to use the Category property on availability groups, workflow endpoints, and workflow definitions when exporting objects by category.

Adapters

FIX – MQSeries Adapter – Thread abort exception could be recorded during Disconnect.

FIX – MQSeries Adapter – The adapter was not honoring the Wait Interval property when in Publish mode.

NEW – Salesforce Adapter - Added a "Custom" data format option that allows users to set the ESBMessage.Http.Headers properties and pass them along to Salesforce via REST calls. This is useful for sending multi-part messages.

FIX – Socket Adapter - If a client socket closes the connection, the server socket adapter wouldn't know it until it tries to write to the socket and would log an error.

MOD – ODBC Adapter - Allows users to insert binary data into databases. If the binary, image or varbinary datatype is passed, we'll use the text within the xml element, assume its base64 encoded and attempt to convert to byte[] and pass to odbc parameter. If user passes a "file:/" url as the content, we'll use that to read in the byte[] array and pass that as the parameter to the stored procedure.

FIX - SharePoint 2007 plugin - fails on install with a message that it cannot find the file Neuron.dll.

FIX - CRM 2013 Adapter - The connection properties do not display in the metadata generation wizard.

MOD - Salesforce Adapter – Added PUT support for APEX web services that accept the Http PUT method.

FIX – Salesforce Adapter - SAML Assertion flow was using the wrong url returned from the login.

FIX – Salesforce Adapter - Set the max received message size to Int32.MaxValue

Neuron 3.5.3.330 Change History

Adapters

MOD – The Neuron ESB Audit Proxy has been removed from the following adapters: FTP, SFTP, Apache ActiveMQ, Azure Service Bus, POP3, Microsoft Exchange, SAP, SalesForce and Microsoft SQL Service Broker. For those adapters in Publish mode, if there is a failure they will automatically log the failed message to the Neuron ESB Audit database through an event mechanism. Also, the "Audit on Failure" property has been removed as this is now automatic.

MOD – A "Connect on Startup" property has been added to the IBM WebSphere MQSeries, POP3, Microsoft Exchange and MSMQ adapters.

NEW – File Adapter - A Publish Metadata Only property has been added so that the file data does not get published. Requires that the Archive File property is set to true and an archive file path is provided.

The encoding body name and archive folder are now exposed as adapter context properties with the message which are accessible within a Business Process and Workflow

FIX – TCP Socket Adapter - Fixed logic for serializing the message when Message Body Only is set to true. The standard header data was incorrectly being added to the outbound message. The logic has been changed to only output the raw bytes in the message body.

MOD – TCP Socket Adapter - Modified the socket adapter to support a generic header. We added properties to the socket adapter for the length of the header, offset of the length in the header, size of the length field, and whether or not the header is included in the message length. Modified the adapter to support the generic message in "message body only" mode.

MOD – Neuron ESB Adapter Framework – Modified to include new event to Audit failed publish attempts.

NEW – Salesforce Adapter - Added Single Sign-on support for the SAML Assertion Flow. The SAML assertion flow is an alternative for organizations that are currently using SAML to access Salesforce, and want to access the Web services API the same way. The SAML assertion flow can only be used inside a single organization. You do not have to create a connected app to use this assertion flow. Clients can use this to federate with the API using a SAML assertion, in much the same way as they would federate with Salesforce for Web single sign-on.

NEW – Salesforce Adapter - Added support for the Salesforce Web API version 34.

NEW – Dynamics CRM Subscription Adapter - Added support for claims-based authorization with an Internet-Facing Deployment (IFD) installation of Dynamics CRM.

FIX – SAP Adapter - SAP Adapter schema generation wizard would fail to generate schemas for BAPI and RFC calls which contain the same SAP structure multiple times.

Test Client

FIX – The Neuron ESB Test client was unable to send a binary body as a Reply message.

Neuron ESB Explorer

FIX - If solution has lots of business processes, in this case hundreds, and a user clicks on an Environmental Variable in the management screen, it can take several seconds for the values of that variable to display. Sometimes the explorer appears as if it is hung up.

Monitoring/Logging

FIX – Client Connectors – Message metrics were not being reported in Endpoint Health. This was a bug introduced in 3.5 CU3

FIX – The Logging Schedule setting for the Neuron ESB Service was not recognizing the Weekly setting. Only Daily would be enforced.

Neuron ESB Runtime

FIX – Neuron ESB service could fail to start up if the esb service did not have the appropriate permissions to query WMI management objects. This would result in an error being reported in the Application event log. The Neuron ESB would record an error stating the machine name could not be null.

Repository

NEW - In previous versions of Neuron ESB, XSL documents that contained “include” references (i.e. `<xsl:include href="inlines"/>`) would be automatically resolved from a file based location. This has been modified to be consistent with how we support Xml Schemas and their respective “schemaLocation” attribute. Support has been added to resolve xsl include locations to the ESB Repository, like this:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
xmlns="http://www.w3.org/1999/xhtml">
  <xsl:include href="esb:inlines"/>

  <xsl:template match="chapter">
    <html><xsl:apply-templates/></html>
  </xsl:template>
</xsl:stylesheet>
```

At runtime, if the href attribute starts with esb: Neuron ESB will look for the XSL in the configuration’s repository. The name following the esb: is the name of the XSL document in the repository.

Service Endpoints

FIX - When a service endpoint threw a soap fault, it would get propagated back through the bus (i.e. from service connector to client connector) but the fault details were missing. This was a bug introduced in 3.5 CU3

FIX - When ‘Enable remote TCP clients’ option is enabled and more than 1 Neuron ESB Server is configured to run the same solution and database, Client Connectors will fail to connect to TCP based topics, except for the first node. This will result in the following exception.

Exception: The default RetryManager has not been set. Set it by invoking the RetryManager.SetDefault static method, or if you are using declarative configuration, you can invoke the RetryPolicyFactory.CreateDefault() method to automatically create the retry manager from the configuration file.

This was a bug introduced in 3.5 CU3.

Business Processes

FIX – Attempting to copy an exception step and its contents, and then paste the contents into a new business process would generate an unhandled null reference exception.

Neuron 3.5.3.308 Change History

Service Endpoints

FIX – Service Connectors throw generic *ArgumentNullException* when trying to pass an invalid Query string parameter whose value is null to an external endpoint. Neuron ESB will now report an

InvalidOperationException with a message that specifically tells the user that a Query parameter with the name of X has a null value being passed to it.

FIX – REST Service Connectors – when using the deprecated “*MessageProperty_HttpRequest*” context property for setting the http “Method” property, the following error would be thrown:

System.NullReferenceException

Object reference not set to an instance of an object.

Workflow

FIX – Transform Message Workflow Activity – Would output a UTF-8-encoded string that included a byte order mark.

MOD – Default HTTP Connections Limit is now set to 100 and can be controlled by modifying the esbhost.exe.config file. Previously, workflows and any HTTP calls from the workflows would use the default set by .NET which was 2.

FIX – Availability Groups – When verbose logging is enabled, the database connection string would be written to the Neuron Log file, possibly exposing user credentials

Adapters

FIX – Azure Service Bus Adapter - When publishing a test message from Visual Studio's Azure Explorer, the adapter would fail with a null reference exception because Visual Studio would publish an empty message.

FIX – TCP Socket Adapter - When using *MessageBodyOnly* mode, the *SocketAdapter* class would not let you use *RequestReply* or *SolicitResponse* modes.

FIX – Adapter Wizard - When generating XML in the Adapter Metadata Wizard, the name as specified in the grid was not respected.

Neuron ESB Runtime

FIX – The Neuron ESB Service fails to start – When the temp directory is full, preventing new temp files from be created, the Neuron ESB Service would fail to start and not log any error indicating why it failed. This has now been corrected. The error condition will now be logged in the Neuron ESB Event log.

FIX – Activity Session State management – At runtime the following error information may be logged, even with few endpoints:

Failed to store session activity state for session Id 3c0b780a-7ae6-4681-86f0-d6c93762eb18

Exception: Timeout expired. The timeout period elapsed prior to obtaining a connection from the pool. This may have occurred because all pooled connections were in use and max pool size was reached.

This was due to the inclusion of the new transient exception handling introduced in 3.5 and has been corrected.

FIX – Database Transient Network Exception Handling – Additional transient exception handling to handle the unreliable network connections of SQL Azure has been added to support using the SQL Azure as the Neuron ESB Audit database.

Business Processes

FIX – Execute Process Process Step – If a user runs a simulation test in the Business Process Designer against the Execute Process step, calling a child process, the expected result is returned. If the user changes the child process and then runs the simulation a second time they will see the result from the original child process and not the newly changed child process. This is because the Execute Process step is designed to cache child processes that it executes. This has been changed so that child processes are only cached at runtime.

FIX – When a user simply clicked on a Process Step or Business Process it forced the Business Process into a Modified state. Now it will only happen when something is either edited or a Process Step is moved.

Neuron 3.5.3.300 (CU3) Change History

Business Processes

NEW - A new attribute, `ProcessStepAttribute`, was added for developing custom process steps to allow for self-registration. Previously, users would be required to register their custom process steps manually by editing the `neuronpipelines.config` located in the pipelines directory of the instance install directory. Users no longer need to edit the config file. Users can now decorate their custom process step class with the `ProcessStepAttribute` as shown in the pseudo code sample below:

```
namespace NewCustomProcessStep
{
    [DataContract]
    [DisplayName("Sample Process Step")]
    [ProcessStep(typeof(MyStep), typeof(MyResource), "myname", "mydescription",
        "CustomProcessStep", "", Path="Training/My New Steps")]
    public class MyStep:CustomPipelineStep
    {}
}
```

Samples that demonstrates using the `ProcessStepAttribute` can be downloaded from here:

<http://support.neuronesb.com/downloads/neuron35/SampleProcessSteps.zip>

NEW - A search box has been added to the top of the Process Step library within the Business Process designer to allow users to easily search for Process Steps.

MOD - When a name is not specified for a business process step, a default named based on the activity type will be used so that any resulting error messages will have meaningful name of the process step within it.

MOD - Modified the drop down box property of the Adapter Endpoint Process Step and Workflow activity so that they no longer display Publish mode adapter endpoints.

MOD - The `System.Collections.Generic` namespace was added to the C# code templates for the process code steps and workflow activities. This will allow types like `List<T>` to be accessed without using the fully-qualified namespace name.

MOD - Transform Process Step - added option to include or exclude the xml declaration on output.

MOD - A user will now be warned if the ESBSERVICE.exe is running a different configuration than what the user is attempting to test at design time within the Business Process Designer.

MOD - Service Endpoint Process Step - A fault or exception did not cause an exception to occur. A Boolean property has now been added to the process step that, if set to true, will cause the exception to be fired allowing the user to handle the exception within the Business Process. Default value is false.

FIX - Adapter Endpoint Process Step - would send out duplicate messages, the number of which would equal the number of retries associated with the Adapter Policy assigned to the underlying adapter endpoint.

FIX - Changes made within the Neuron ESB Business Process Designer would sometimes not be persisted when the user selected to save all changes. This could happen if the user was previously prompted to reload their configuration due to a change that the Neuron ESB detected in the underlying configuration.

FIX - Publish Process Step - In the Neuron ESB Business Process Designer, the code topic selector dialog would not save edited code when closed. The Publish step would still display the tooltip *"The code for the selector must be set"*. When the code editor was reopened, any previously entered code would be lost.

FIX - Attempting to access the `context.Configuration` object from a Business Process running remotely using the Client API or hosted by the Neuron ESB Test Client would fail with the following exception:

The type initializer for 'Neuron.Configuration.NeuronConfiguration' threw an exception

FIX - Code Process Step - When a user adds an assembly reference to a Code Process Step, then follows that by removing the same assembly reference, the user would be unable to re add the same assembly reference.

FIX - ODBC Process Step - having a '?' in data that's sent to either the ODBC adapter or Process Step would generate a parameter error. For example, if an ODBC adapter endpoint was configured for execute mode, the following statement would generate a `System.IO.InvalidDataException` exception. The adapter would misinterpret the '?' that appends the data in the value clause as a parameter:

```
string sqlStatement = "<Statement type=\"Text\" sql=\"INSERT INTO Customer(Name)
VALUES('Joe the guy?')\"></Statement>";
```

FIX - Code Process Step - If a user deleted a code process step from a Business Process while the code Process Step Editor was also opened in another tab, the tab remained open. It will now be automatically removed.

FIX - Service Endpoint Process Step - In an associated service policy, when electing to republish to a new topic, custom properties would be lost.

FIX - Audit Process Step - After several exceptions, an Audit Process Step may throw a *System.ServiceModel.CommunicationObjectAbortedException* exception.

FIX - Code Process Step - Debugging C# Code Process steps in the Business Process Designer causes duplicate window to appear— During the debugging process if a user hit F10 a new duplicate window of the current C# Code editor would open and the user would receive prompts during debugging that changes to the underlying configuration were detected, prompting the user to save them.

FIX - Adapter Endpoint Process Step – *base.name* and *base.PartyID* were not being populated which prevented adapters from reporting those values when errors occurred.

FIX - When using the *Execute Process* process step and dynamically setting the value of the child process to run at design time, users could experience high CPU utilization at runtime when running under load. This was caused because Neuron was not caching the child process that would be called. Every execution would cause the child process to be recreated/recompiled, which could also generate unnecessary temp files.

FIX - If a user moved an existing business process step to a different location on an existing business process, the Apply button would not be activated.

FIX - An unhandled exception would occur when trying to load an existing business process or workflow that referenced a custom process step that had yet to be registered.

FIX - If a user deleted an existing Business Process, upon saving the changes they would be presented with the Review Change List dialog. This would contain the change, as well as a number of “ADD” changes for existing publishers and subscribers that were not affected by the changed Business Process.

Workflow Designer

NEW - Retry Workflow Activity - A new Retry workflow activity has been introduced that can contain other workflow activities and be configured to retry their execution N (user defined) number of times if an exception is thrown.

MOD - Service Endpoint Workflow Activity – A fault or exception did not cause an exception to occur. A Boolean property has now been added to the activity and process step that, if set to true, will cause the exception to be fired allowing the user to handle the exception within the Workflow. Default value is false.

MOD - Transform Workflow Activities - added option to include or exclude the xml declaration on output.

MOD - The `System.Collections.Generic` namespace was added to the C# code templates for the process code steps and workflow activities. This will allow types like `List<T>` to be accessed without using the fully-qualified namespace name.

FIX - Within the Neuron ESB workflow designer, a Workflow with Execute Process activity would always enter a "modified" state (i.e. Apply button would be enabled and there would be an asterisks in the tab) when a person navigates away and then back to the workflow designer.

FIX - Code Workflow Activity - When using the C# workflow activity within the Neuron ESB Workflow designer, compile errors would still be reported, even after the compile error had been corrected in the code.

FIX - Code Workflow Activity - When a user adds an assembly reference to a Workflow Activity, then follows that by removing the same assembly reference, the user would be unable to re add the same assembly reference.

FIX - A user will now be warned if the ESBSERVICE.exe is running a different configuration then what the user is attempting to test at design time within the Workflow Designer.

FIX - Service Endpoint Workflow Activity - In an associated service policy, when electing to republish to a new topic, custom properties would be lost.

FIX - C# Class Workflow Activity - The message variable was not available in the Workflow C# Class Activity. In order to retrieve the value of the argument or variable, a user must write a public property with a getter/setter, and at runtime the activity will invoke the setter to set the value of the property. For example:

```
public ESBMessage message { get; set; }
```

FIX - Adapter Endpoint Workflow Activity – *base.name* and *base.PartyID* were not being populated which prevented adapters from reporting those values when errors occurred.

FIX - Publish Workflow Activity – if an error occurred publishing to the Topic, 2 errors would be generated rather than one and both were incorrect.

FIX - Workflow Designer – any design time change to a C# Workflow Activity would not be reflected at runtime when in online mode and the user saves the changes.

FIX - Publish Workflow Activity – When the Publish Workflow Activity would fail to publish to the assigned Topic at runtime it would not throw an error back to the running Workflow. This would cause the Workflow to appear in Workflow Tracking with a Completed state rather than an Aborted state, even though the error was correctly logged in the Neuron ESB Event log.

FIX - Users may experience the following error when referencing a custom assembly within the Workflow Designer during simulation after the Neuron ESB Explorer has been closed and reopened:

Compiler error(s) encountered processing expression

FIX - When Assembly References were added or removed in Add Assembly reference dialog within the Workflow Designer, the Apply button would not become enabled.

FIX - When browsing for types within the Workflow Designer, no types would appear in the type browser dialog.

FIX - Adding an assembly reference to an existing workflow could generate a Null reference exception.

FIX - When adding assembly references to a workflow, duplicate references could be added if one was added at the code workflow activity as well.

FIX - Adding a reference at the workflow activity would work at runtime, however a compile time error would occur when testing the workflow within the designer.

Workflow Runtime

FIX - Adapter Endpoint Workflow Activity - would send out duplicate messages, the number of which would equal the number of retries associated with the Adapter Policy assigned to the underlying adapter endpoint.

FIX - When Neuron ESB starts up, if the ESB Management objects feature were not installed, the Availability Groups would not start up and there would be no warning as to why they would not startup and how to correct.

FIX - Stopping ESB Service with active Availability Groups could generate the following errors –

1. *Can't perform operation STOP# on Workflow Endpoint 668ba063-6f99-4175-a37b-39e25cca6a8e, running under Workflow Order Process Host, because cannot write to a closed TextWriter.*
2. *Can't perform operation Stop Operation on Availability Group Workflow Order Process Host, because Object reference not set to an instance of an object.*

FIX - Availability Group would not automatically recover/restart if the SQL Server suddenly went down (i.e. network failure) and came back on line.

FIX - In environments where more than one Neuron ESB server is running the same solution and sharing the Neuron ESB database, users may experience the following error associated with Availability Groups:

Control of the Availability group Availability Group X has been transferred to another server in the farm. Shutting down heartbeat thread process.

FIX - Availability Groups were not applying the Environment Variable value applied as the `ConnectionString` property on the Neuron ESB Database.

FIX - When a persistence exception occurred in a Workflow the original message was not written to the Failed Message table within the Neuron ESB database.

FIX - Race condition in correlated workflows could occur and prevent subsequent workflows from starting - users could experience one of the following:

- a) The user submits messages to the workflow endpoint, but no workflows start running. The user would see messages queueing up in the pending messages table.
- b) The user submits messages to the workflow endpoint. One workflow may start, but will move into an Unloaded state and will not resume; no other workflows would start. Again, you'd see messages queueing up in the pending messages table.

FIX - Error when attempting to start a Workflow Endpoint - When either a user or the Neuron ESB runtime attempts to start a Workflow Endpoint, the following error could be reported if the Topic assigned to it is unavailable or disabled:

System.AggregateException: One or more errors occurred. --->

System.Collections.Generic.KeyNotFoundException: The given key was not present in the dictionary.

The Workflow Endpoint would also appear in a Started state within Endpoint Health. This has been corrected.

FIX - Assigning an Availability Group to multiple machines could cause the ESBHost.exe to constantly recycle itself.

FIX - When using the Persistent Delay Workflow Activity, the Neuron ESB scheduler for deserializing the workflow may fire multiple times. This could result in the following entry in the Neuron ESB log file:

System.Runtime.DurableInstancing.InstanceNotReadyException: The execution of an InstancePersistenceCommand was interrupted because the instance '831bc3b6-39e9-4890-b776-e026d975e012' has not yet been persisted to the instance store.

FIX - Within the workflow, if a publish activity publishes a message to a Topic that is stopped, failed or in any kind of inaccessible state, it would not throw an exception back to the workflow instance to abort it. Instead, Workflow tracking would show that the workflow completed successfully.

FIX - Users could experience a memory leak for the Neuron ESB Workflow host when Workflow Tracking was enabled.

FIX - A “Synchronous” property has been added to the [PublishRequestMessage](#) Workflow Activity. Its default value is false. Previously, the [PublishRequestMessage](#) activity would send a request then persist the workflow, even if the response was expected to be returned in several seconds. However, if there were a great many instances of workflows running, it could take much longer for the workflows to receive the responses and continue the workflow instances since all the response messages would be effectively queued. By setting Synchronous to true, Neuron will no longer let the workflow go idle and

persist, instead the [PublishRequestMessage](#) activity will now wait for the reply to be returned and continue, keeping the workflow alive.

FIX - If a persistence error occurred when trying to resume a workflow, semaphore locks could be released.

FIX – During runtime the operating system could return a null reference for a performance counter which could cause the Workflow host environment to crash.

FIX – SQL Server could experience very high CPU usage (>80%) after all workflows in a batch had completed.

Service Endpoints

NEW - CORS header support has been added as a checkbox option to the Neuron ESB Client Connector to allow cross domain scripting for REST and HTTP based endpoints. Previously customers would need to create a custom behavior and add that to the service endpoint.

NEW - Added support for adding/retrieving http headers from non-REST (http based bindings) service endpoints. A new *“Enable HTTP Headers”* checkbox has been added to both the Client and Service connector tabs of the Service Endpoint entity. Regardless of whether or not this is enabled, HTTP headers are always processed for REST endpoints. Additionally, the Capture and Restore Headers options have been renamed to *“Enable Custom SOAP Headers”* on both the Client and Service connector tabs of the Service Endpoint entity. This also modifies the previous behavior of the Service endpoints when dealing with custom SOAP Headers. Previously, if a user created 2 service endpoints (i.e. one a client connector, the other a service connector) and wanted to have custom SOAP header support end to end, the user could have to enable both the *“Capture Headers”* and *“Restore Headers”* on each Client connector and Service connector tab of both endpoints, even though one tab on each would effectively be disabled in all other respects. Now the user just need to check *“Enable custom SOAP Headers”* on either the enabled Client or Service Connector tab.

NEW - Support has been added for using Username and Password with the [TransportWithMessage](#) security setting for Service Endpoints. Previously, only Certificates were supported.

MOD - Service Endpoints called through either the Service Endpoint Process Step or Workflow activity no longer need to be enabled to be used at runtime.

FIX - If user called into a Neuron ESB Client Connector endpoint by first adding a Proxy to the HTTP header on the Request message that is the same proxy as the one that the message will be forwarded to (i.e. a configured Service Connector), it would generate a multi hop cycle error from the target endpoint.

FIX - Deleting a running Client or Service connector from the Neuron ESB Explorer could generate the following exception when the changes were saved:

“Could not find service endpoint with Id”

Monitoring/Reporting

NEW - Added WMI *EndpointStateChangeEvent* event to Topics, Availability Groups and Workflow Endpoints – Users can now use WMI to monitor all endpoints within Endpoint Health for state changes as well as failures. Users can use WMI's query language to limit the monitoring to any combination of properties exposed by the *EndpointStateChangeEvent*. For example, the “*Type*” property can return any of the following values:

- TcpPublishingService
- AmqpEndpoint
- MsmqPublishingService
- NamedPipePublishingService
- ServiceConnector
- AdapterEndpoint
- AvailabilityGroup
- WorkflowEndpoint
- ClientConnector

The WMI code that monitors only Adapter Endpoints may look like this:

```
using System;
using System.Globalization;
using System.Management;

namespace WmiFailedMessageSample
{
    class Program
    {
        static void Main()
        {
            try
            {
                var managementScope = new ManagementScope("\\\\.\\root\\Neudesic_ESB_v0");
                managementScope.Connect();
                var eventEndpointQuery = new WqlEventQuery("EndpointStateChangeEvent");
                eventEndpointQuery.Condition = "Type = 'AdapterEndpoint'";
                var watcher1 = new ManagementEventWatcher(managementScope, eventEndpointQuery);
                watcher1.EventArrived += EndpointChanged;
                watcher1.Start();

                Console.WriteLine("Listening for events. Press Enter to exit.");
                Console.ReadLine();
                watcher1.Stop();
            }
            catch (Exception ex)
            {
                Console.Error.WriteLine(ex);
                Console.WriteLine("\nPress Enter to exit.");
                Console.ReadLine();
            }
        }

        static void EndpointChanged(object sender, EventArrivedEventArgs e)
        {
            string zone = e.NewEvent["Zone"] as string;
```

```

DateTime eventDate =
    ManagementDateTimeConverter.ToDateTime(e.NewEvent["Datetime"] as string);
string type = e.NewEvent["Type"] as string;
string name = e.NewEvent["Name"] as string;
string state = e.NewEvent["State"] as string;
string hostName = e.NewEvent["Hostname"] as string;
string instanceName = e.NewEvent["EsbInstanceName"] as string;
string application = e.NewEvent["Application"] as string;
string deploymentGroup = e.NewEvent["DeploymentGroup"] as string;
string endpointId = e.NewEvent["Id"] as string;
string info = e.NewEvent["Message"] as string;

var message = string.Format(
    CultureInfo.InvariantCulture,
    "Name={0}, Endpoint Type={1}, State={2}, Instance={3}, Zone={4}, Machine={5},
    Application={6} , DeploymentGroup={7}, Info={8}, DateTime={9}",
    name,
    type,
    state,
    instanceName,
    zone,
    hostName,
    application,
    deploymentGroup,
    info,
    eventDate.ToString());

Console.WriteLine(message);
    }
}
}

```

NEW – Resubmission count for messages resubmitted using the Message Resubmit dialog. When users resubmit a message, the row in the Neuron ESB Audit database for the message will be updated with an incremented submission count value. That value is now displayed as the “Submission Count” column in the Message History and Failed Message reports. Once a message has been resubmitted, users can re-query the reports and view the number of times a message has been resubmitted.

MOD - Endpoint Health refresh interval is no longer linked to service report interval. There is a new Refresh Interval property located on the Tools → Options menu of the Neuron ESB Explorer.

FIX - Within Endpoint Health, the rate being reported would be inaccurate if the screen was refreshed.

FIX - Within the Message History and Failed Message reports, the *Related Messages* context menu would throw exception indicating a *Missing Instance column*

FIX - If an exception occurred when starting a Workflow Endpoint, Endpoint Point would reflect a “Starting...” state rather than a failed state.

FIX - If an exception occurred during the connection of the underlying Party associated with a Workflow Endpoint, Endpoint Health would incorrectly show that the Workflow Endpoint was started rather than failed.

FIX - Within the Failed Message and Message History reports, setting the Topic Filter could sometimes throw an *Index Out of Range* Exception. This would occur if audited records contained sub topic references. Users could also experience the following exception:

System.Data.SqlClient.SqlException (0x80131904): Ambiguous column name 'TopicRoot'

FIX - Sorting Issues with Message History and Failed Message reports - [Message Id](#) and [Transaction Id](#) column sorting were not returning results in the correct order.

FIX - Endpoint Health – Running Workflow Endpoints do not appear in Endpoint Health if a different solution is running than that which is loaded in the Neuron ESB Explorer.

FIX - Users could not delete a workflow entry in the Workflow Tracking UI if the Workflow Definition file was missing. This could be because a user deleted it, or a different solution was loaded in the Neuron ESB Explorer that referenced the same Neuron ESB database.

FIX - Starting an aborted workflow generates a generic error processing message – When attempting to start an aborted workflow that by design could not be started (i.e. there was not a persistent point defined within the workflow) the following generic exception would be thrown:

System.Runtime.DurableInstancing.InstanceNotReadyException

Neuron will now display an exception letting the user know why it couldn't start i.e. a persistent point did not exist in the workflow.

FIX - PendingEvents counter on Endpoint Health could erroneously display 2 billion records. Restarting the Availability Group/Workflow endpoint could cause this when there were active workflows in the database to process.

FIX - Workflow Tracking – On the Messages tab of Workflow Tracking, the [Move](#) Context Menu did not always work.

FIX - Endpoint Health – Previously, if errors occurred on an endpoint, the icon representing that endpoint would turn red. However, if the error conditions for the endpoint were resolved and successful messages were sent, the icon would remain red. This has been resolved. The icon will turn back to Green once a subsequent successful message is detected sent over endpoint.

FIX - Cluster Support - Workflow Endpoints – When running in a Windows Cluster environment the metrics for Workflow Endpoints within Endpoint Health were not being incremented.

FIX - Workflow Endpoint and associated Party dependencies would fail to remain in sync. The dependent Workflow endpoint could enter a state where it would be unaware if the dependent Party became disabled, disconnected or failed to successfully connect to the dependent Topic. If any of these conditions occurred, within Endpoint Health, the Workflow Endpoint would appear in a healthy, Started state rather than move into a Failed state.

FIX - App Domains were not unloading from memory when endpoints are reconfigured or stopped within the Endpoint Health. This could result in errors reconnecting to Topics.

FIX - When using a Named Pipes based Topic and attempting to restart an ODBC Adapter Endpoint multiple times from Endpoint Health, the user may experience the following exception:

Cannot listen on pipe

FIX - Client connector would be displayed in a Stopped state in Endpoint Health instead of disabled when its underlying party is disabled - If the party assigned to a service endpoint (i.e. client connector/service connector) or adapter endpoint or workflow endpoint is disabled and if a user attempts to start the endpoint in Endpoint Health, it shows up RED, in a stopped state, and NO errors are indicated in endpoint health even though an error is recorded in the Neuron ESB event log. Moving forward, if any party assigned to an endpoint is disabled, the endpoint will be displayed as disabled within Endpoint Health and a warning will be logged to the Neuron ESB event log indicating the reason why its disabled.

FIX - Endpoint Health can generate an unhandled exception if “while displayed”, a user restarts the Neuron ESB service from the Neuron ESB Explorer toolbar.

FIX - Restarting a Neuron ESB Topic within Endpoint Health when its assigned port is currently in use will cause the Topic to be set to a failed state which is expected. However, it will immediately thereafter be switched to a Stopped state. Once that happens, even if the port condition is corrected, the Topic can no longer be restarted in Endpoint Health.

FIX - When a polling based adapter reports that it will be disconnected, its state in Endpoint Health would remain in a Started state. This affected the following adapters:

- Azure Service Bus
- File
- FTP/SFTP/FTPS
- IBM MQseries
- Microsoft Exchange
- POP3
- ODBC

FIX - Unable to stop polling based adapter in Endpoint Health. When a user attempts to stop a polling based adapter within Endpoint Health, its state will briefly be set to Stopped and then immediately reset to Started.

FIX - If a Service Connector or Client Connector failed to start up, specifically open a proxy for the target URL, Neuron ESB would increment of the error count in Endpoint Health, but the endpoint would still register as Started in Endpoint Health.

FIX - If an error occurs when starting or reloading a workflow, the active performance counter was not being decremented. This would make it appear in Endpoint Health that there were more active workflows than there really were.

FIX - When republishing a large number of messages using the Message Resubmit dialog to an MSMQ based topics, users may experience the UI hang and high memory utilization. This only occurred when users selected to republish using a Party. This was due to Neuron ESB waiting for .NET garbage collection to finish. This has been fixed.

FIX - When republishing a messages users may see an error message box displayed in the background with a **System.AggregateException** message. This only occurred when users selected to republish using a Party. This has been fixed.

FIX - When republishing messages using the Message Resubmit dialog, if there was an existing Action value on the original message, it would not be republished with the message. This only occurred when users selected to republish using a Party and has been fixed.

FIX - Modified the Message Resubmit dialog so that it only displays Subscribe side adapter endpoints. Previously this would display all adapter endpoints regardless of mode (i.e. Publish mode, etc.).

FIX - Modified the Message Resubmit dialog so that messages can be resubmitted directly to Endpoints without the need for the Neuron ESB Service to be running.

FIX - When attempting to conduct an operation on a workflow record in Workflow tracking a user could experience the following erroneous error message:

"One or more selected workflows do not belong to current instance 'N'. These workflows will be skipped. Do you want to continue?"

This message could be displayed even though the selected Workflow item DID belong to the current instance. This has been fixed.

FIX – User could experience a **"The SELECT permission was denied on the object"** exception when running either the Message History, Failed Message or Workflow tracking reports. This would occur if the specific user was added to the **NeuronUsers** database role.

FIX - When using Endpoint Health for monitoring workflow endpoints, they could appear under the incorrect Availability Group.

Neuron Database

NEW - SQL Azure as the Neuron ESB database can cause Network Connection and Timeout Exceptions. Transient fault handling for components that interact with a SQL Azure database has been introduced. We found in testing that customers that used SQL Azure as their SQL data store were experiencing frequent faults that are common to SQL Azure. The transient fault handling code allows Neuron to intercept and respond to these events by applying a strategy for retrying the database operation multiple times before failing.

Configuration Section

The configuration settings for the transient fault handling are currently configured in the following configuration files:

- ESBSERVICE.exe.config
- ESBHost.exe.config

The configuration section is named `neuron.transientFaultHandling`. The configuration section needs to be registered in the `<configSections>` header element of the configuration file:

```
<configuration>
  <configSections>
    <section name="neuron.transientFaultHandling"

type="Neuron.Esb.TransientFaultHandling.Configuration.RetryPolicyConfigurationSettings, Neuron.Esb.T
ransientFaultHandling"/>

    ...
  </configSections>
</configuration>
```

The `neuron.transientFaultHandling` configuration section is used to define retry strategies that will be executed to retry database operations when database operations fail. The specific retry strategies are documented in the next section. In the Neuron ESB 3.5.3 release, we currently only use the default strategy as configured in the configuration section. The configuration section will look like this:

```
<configuration>
  <neuron.transientFaultHandling defaultRetryStrategy="incremental">
    <incremental name="incremental"
      firstFastRetry="true"
      maxRetryCount="15"
      retryIncrement="00:00:01"
      initialInterval="00:00:01"/>
  </neuron.transientFaultHandling>
</configuration>
```

This configuration section indicates that the **incremental** retry strategy will be used. The **incremental** retry strategy is configured to retry the database operation a maximum of 15 times before failing. The retry strategy will initially wait for 1 second and will pause for 1 second between successive retries.

Retry Strategies

Incremental Strategy

The incremental retry strategy attempts to retry a failing operation for a specified maximum number of attempts and uses an incremental time interval between retries. The incremental strategy is defined in the configuration using the incremental configuration element and has the following attributes:

```
<incremental name="..."
  firstFastRetry="true"
```

```
maxRetryCount="10"
retryIncrement="00:00:01"
initialInterval="00:00:01"/>
```

Attribute	Type	Description
name	String	A unique identifier for the retry strategy.
firstFastRetry	Boolean	If true, the first retry attempt occurs immediately if a failure occurs. If false, the first retry will remain subject to the retry interval.
maxRetryCount	Integer	The maximum number of retry attempts to perform before the operation fails.
initialInterval	TimeSpan	The initial interval that will apply for the first retry.
retryIncrement	TimeSpan	The incremental time value that will be used to calculate the progressive delay between retries.

Fixed Incremental Strategy

The fixed incremental retry strategy attempts to retry a failing operation for a specified maximum number of attempts and uses a fixed time interval between retries. The fixed incremental strategy is defined in the configuration using the `fixedIncremental` configuration element and has the following attributes:

```
<fixedInterval name="..."
  firstFastRetry="true"
  maxRetryCount="10"
  retryInterval="00:00:01"/>
```

Attribute	Type	Description
name	String	A unique identifier for the retry strategy.
firstFastRetry	Boolean	If true, the first retry attempt occurs immediately if a failure occurs. If false, the first retry will remain subject to the retry interval.
maxRetryCount	Integer	The maximum number of retry attempts to perform before the operation fails.
retryInterval	TimeSpan	The time interval between retries.

Exponential Backoff Strategy

The exponential backoff retry uses an exponential delay between retries to attempt to complete the failing operation. The exponential backoff strategy is defined in the configuration using the exponentialBackoff configuration element and has the following attributes:

```
<exponentialBackoff name="..."  
  firstFastRetry="true"  
  maxRetryCount="10"  
  minBackoff="00:00:01"  
  maxBackoff="00:00:30"  
  deltaBackoff="00:00:10"/>
```

Attribute	Type	Description
name	String	A unique identifier for the retry strategy.
firstFastRetry	Boolean	If true, the first retry attempt occurs immediately if a failure occurs. If false, the first retry will remain subject to the retry interval.
maxRetryCount	Integer	The maximum number of retry attempts to perform before the operation fails.
minBackoff	TimeSpan	The minimum backoff time.
maxBackoff	TimeSpan	The maximum backoff time.
deltaBackoff	TimeSpan	The value that will be used to calculate a random delta in the exponential delay between retries.

MOD - The Update Purge and Archive Sql Agent job was modified so that the archive portion of job could be optional.

FIX - Workflow and using SQL Azure – Transaction deadlocks could occur against the *WorkflowCorrelatedWorkflow* table could occur when using SQL Azure.

Logging

FIX - Availability Group logging has been corrected so that issues are now reported in the Neuron ESB Event log. Previously they were only logged in the Neuron ESB log files. Error messages have also been augmented with more actionable information.

FIX - Cluster support - Event Logging - When running in a Windows Cluster environment Host name was being recorded rather than cluster network name

FIX - Cluster Support - When running in a Windows Cluster environment MSFT Failover cluster events were being captured in the Availability Group Neuron log files

FIX - The Neuron ESB Event log contains several errors on startup when completed workflow instances existed in the Neuron ESB database – specifically:

System.Runtime.DurableInstancing.InstanceNotReadyException

FIX - If the Neuron logging location was changed by setting the following registry keys:

HKEY_LOCAL_MACHINE\SOFTWARE\Neudesic\Neuron ESB v3\ESB\SystemLogFolder
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Neudesic\Neuron ESB
v3\ESB\SystemLogFolder

Users would experience an exception (**System.ServiceModel.FaultException**) when they attempt to “Connect” to the running instance using the Neuron ESB Explorer. The exception would occur when attempting to select the Instance to connect to within the drop down box. To correct this, the registry key has been moved. If the Neuron logging location needs to be changed, users should now use the following registry keys, where *SystemLogFolder* is a string value:

HKEY_LOCAL_MACHINE\SOFTWARE\Neudesic\Neuron ESB v3\SystemLogFolder
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Neudesic\Neuron ESB v3\SystemLogFolder

Neuron Explorer/Test Client

NEW - Added the format context menu to the xml editors located on the Neuron ESB Test Client so that users can just right-click and select “Format”.

FIX - The Neuron ESB Web Service Import would fail with WSDLs that had schemas which included ##any elements. Although these can now be imported correctly, these WSDLs are still invalid to use as a metadata setting for client connectors

FIX - When closing Neuron ESB Explorer, saving a new configuration did not add the configuration to the MRU list of the Neuron ESB Explorer

FIX - In the Neuron Explorer, within the Binding Expression dialog, the Binding Expression field would only allow a user to scroll to the right in order to check first 75 characters of the url. If the url was longer than 75 characters, the full length would be stored, but only 75 characters could be viewed by scrolling to the right in the field.

FIX – Within the Neuron ESB Explorer the Delete button/icon for the Process/Workflow Library did not work for workflow objects.

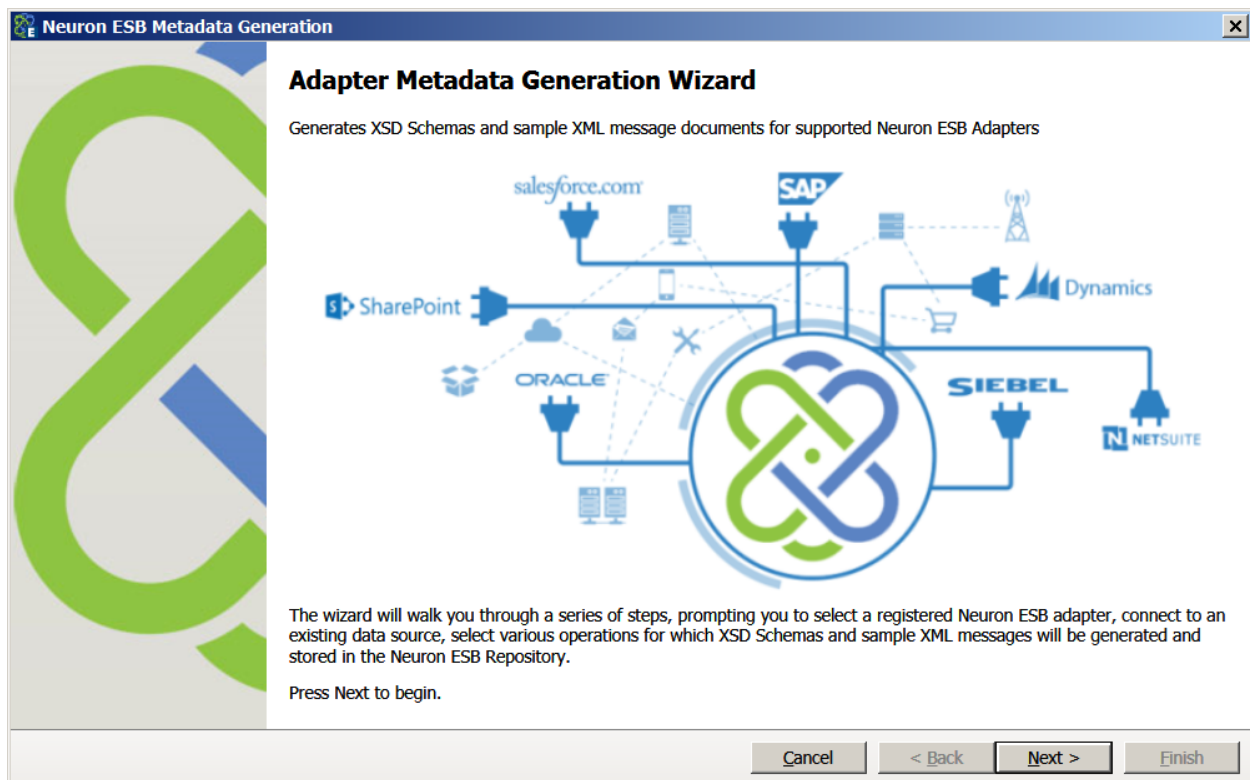
FIX - The Endpoint Health toolbar button should only be enabled IF there is a solution opened in the Neuron ESB Explorer.

FIX - When in offline mode, users could not configure the “Endpoint Policy” property of the MSMQ Transport for a Topic.

FIX - The Neuron ESB Log files for Availability Groups and Workflow Endpoint logs were being misnamed when new daily logs were created

Adapters

NEW – A new Adapter Wizard for generating XML Schemas and Sample XML messages has been added. This wizard is accessible through the tool bar “Generate” button located on Adapter Endpoints and XML and Schema Repository within the Neuron ESB Explorer. Using this wizard, users can select a registered adapter that supports meta data generation, choose the operations and select to generate both the schemas and sample messages that the adapter endpoint requires. Currently, the SAP, NetSuite, Dynamics CRM and Salesforce adapter are all supported. Additionally, developers can use the Neuron ESB Adapter framework to develop their own adapters that support our new meta data API. Those adapters will automatically be supported by the Adapter Wizard.



NEW - TCP/IP Socket Client and Server Adapters - The TCP/IP Socket adapters establish a bi-directional channel for sending messages between two processes using just the TCP/IP protocol. The primary use case for the TCP/IP Socket adapters is to link two Neuron adapters that exist in separate zones such as a DMZ and a secure intranet. Using the adapters, the DMZ ESB service can expose WCF client connectors to the unsecure Internet, and forward requests to be processed by the linked ESB service that is running in a secure intranet. The TCP/IP Socket adapters have been designed to support overcoming firewall connection issues that typically exist when establishing a DMZ. The TCP/IP Socket adapter can also be

used to send messages to or receive messages from external programs that can communicate using TCP/IP Sockets.

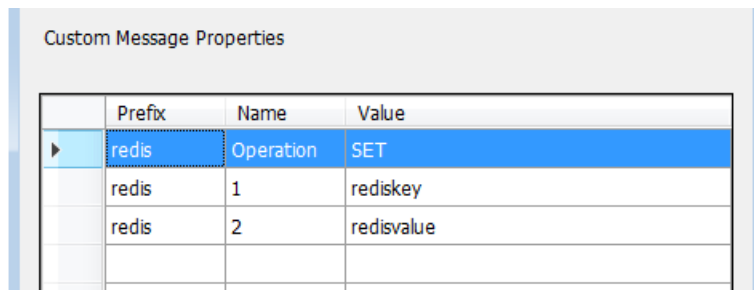
NEW – Redis Adapter - This adapter supports one way subscribe and two-way, solicit response modes. Users can send updates, inserts keys/values, or make all get Query requests against a Redis Server. The Neuron ESB Redis adapter supports all operations provided by Redis.

In order to use Redis adapter you need to configure connection properties:

- Server Name
- PrimaryPort
- SecurePort
- SlavePort

The Neuron ESB Redis adapter expects the developer to pass custom properties as part of the ESB message header. Below is the example of setting custom properties (via the Neuron ESB Test Client) to the adapter end point in order to perform a **SET** operation.

SET command

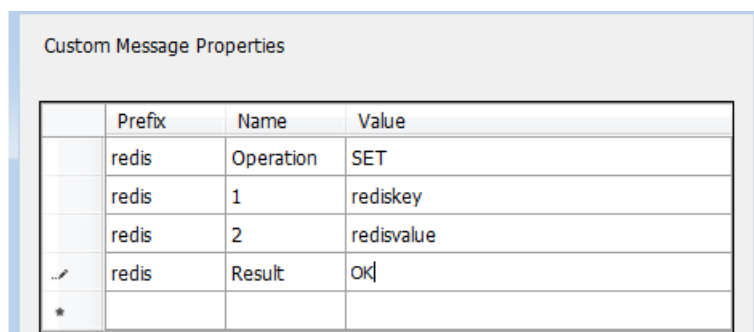


The screenshot shows a window titled "Custom Message Properties" containing a table with four columns: Prefix, Name, and Value. The first row is highlighted in blue and contains the values "redis", "Operation", and "SET". The second row contains "redis", "1", and "rediskey". The third row contains "redis", "2", and "redisvalue".

	Prefix	Name	Value
▶	redis	Operation	SET
	redis	1	rediskey
	redis	2	redisvalue

As shown above, in order to use the Neuron ESB Redis adapter you need to pass the custom properties on the ESB message header. The adapter requires you to pass **“what”** operation you want to do and its respective parameters. One need to pass the exact **Operation** name which is used in Redis. You can find the list of commands here: <http://redis.io/commands>

When you send the message to an adapter endpoint, the **Result** custom property will be returned which will contain the status of operation.

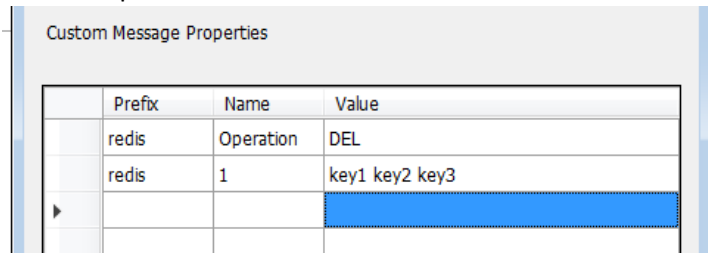


The screenshot shows a window titled "Custom Message Properties" containing a table with four columns: Prefix, Name, and Value. The first row contains "redis", "Operation", and "SET". The second row contains "redis", "1", and "rediskey". The third row contains "redis", "2", and "redisvalue". The fourth row contains "redis", "Result", and "OK".

	Prefix	Name	Value
	redis	Operation	SET
	redis	1	rediskey
	redis	2	redisvalue
✎	redis	Result	OK
★			

Note

- a) In order to pass a parameter which is of type **Timespan** as expected by the **EXPIRE** command, the developer needs to only pass in the value as the number of seconds.
- b) In Order to pass array in the parameter the developer needs to pass it by space value. For example, if the developer wants to perform the **DEL** command for multiple keys, the parameter would be passed as follows:



	Prefix	Name	Value
	redis	Operation	DEL
	redis	1	key1 key2 key3

- c) If you do not pass a parameter which is supposed to have a specific **value**, Neuron will pass the body of the **ESBMessage** as a value.

NEW – An SAP adapter has been added that will allow users to easily update and query existing SAP systems using IDOC, BAPI and RFC interfaces as well as receiving IDOC messages to be published to the bus. This adapter also supports meta data generation in the form of XML Schemas and sample XML messages using the new Adapter Wizard.

NEW - A new Microsoft SQL Service Broker adapter has been added. This can be used to either monitor Microsoft SQL Service Broker queues or send messages to those queues.

NEW – NetSuite adapter - Added support for multiple versions of NetSuite's SOAP API. The adapter currently supports 2013_2, 2014_1, 2014_2 and 2015_1.

MOD - Adapter Endpoints called through either the Adapter Endpoint Process Step or Workflow activity no longer need to be enabled to be used at runtime.

MOD - Dynamics CRM 2013 Subscription Adapter - Added a send timeout property.

MOD - Salesforce Adapter – When generating XML and XSD schemas from the Meta generation tool, those documents stored in the Neuron ESB Repository will include the Salesforce version number configured for that endpoint.

MOD - Salesforce Adapter – Now supports latest version(s) of the Salesforce SOAP API - Added support for changing the Salesforce SOAP API version. This includes a new adapter property, along with adding an annotation to updated schemas for which version of the SOAP API they were generated with. Also modified the Neuron ESB Schema version number to match the SOAP API version. Current support is for versions 28.0 - 33.0.

FIX – The SharePoint 2010 Plugin – referenced incorrect versions of Neuron ESB assemblies – the correct versions are compiled against .NET 3.5

FIX – SharePoint 2010 P – users with limited access security could not publish events to Neuron ESB.

FIX - ODBC Adapter - having a '?' in data that's sent to either the ODBC adapter or Process Step would generate a parameter error. For example, if an ODBC adapter endpoint was configured for execute mode, the following statement would generate a *System.IO.InvalidDataException* exception. The adapter would misinterpret the "?" that appends the data in the value clause as a parameter:

```
string sqlStatement = "<Statement type='Text' sql='INSERT INTO Customer(Name)
VALUES('Joe the guy?')'></Statement>";
```

FIX - If an Adapter Endpoint was marked as Single Instance in solution running on more than 1 server, it could go into a stopping state.

FIX - Dynamics CRM 2011 plugin - The Neuron EB Dynamics CRM 2011 plugin would generate an error during installation due to incorrect XRM SDK assembly references.

FIX - Salesforce adapter – Exceptions could arise from concurrency issues related to how the adapter was internally managing the Salesforce session ids.

FIX - Salesforce adapter – When using the REST capabilities of the adapter, if an exception occurred, the aggregate exception rather than the true exception could be recorded/logged.

FIX - Dynamics CRM Adapter – The x64 Neuron ESB installer only installed the x86 CRM SDK assemblies.

FIX - ODBC Adapter Endpoint – If the endpoint was marked as single instance and deployed to more than 1 machine, the endpoint would continue to flip back to a Started state if a user stopped it from Endpoint Health.

FIX - Salesforce adapter – when attempting to launch the Meta generation dialog from the adapter properties grid with an expired password, the dialog would launch but when a user attempted to select/expand the Salesforce objects, no objects would appear. Moving forward they will be alerted to the expired password condition.

FIX - A *threadabortexception* could occur during the shutdown of a File adapter endpoint configured for publish mode. This was due to a race condition which has been fixed.

FIX - Added a new property to the Azure Service Bus adapter to allow a developer to choose between ACS and SAS (default) for authentication. The adapter will create an ACS URI for the service bus if ACS authentication is requested.

FIX - The Dynamics CRM 2013 adapter would throw an instance of an object exception if a CRM entity didn't have a description.

FIX - When sending a message to a custom adapter using either the Adapter Endpoint Process Step or Workflow activity, the Neuron ESB Configuration object was not being set. This could prevent specific

functions from working, for example, auditing the message on a publish mode adapter when the publish activity failed.

FIX - SMTP Adapter – Previously we did not allow context properties to be used for the Filename parameter in property grid UI. If used, a Filename validation exception would be thrown.

FIX - File Adapter – When using an Adapter policy to control retries, a retry would write both a warning and an error to the Neuron ESB event log. Only warnings should be written for each retry with 1 failure message when the retries exhaust themselves.

Installation and Setup

MOD - Updated Xceed libraries to version 5.7

FIX - Neuron ESB Installer: Installing only the Development Tools option would not install the Pipelines directory. If using the Neuron ESB Test Client, the following error would result:

Unable to deserialize pipeline from configuration file

FIX - The "*Neuron Availability Groups*" performance counter category had a typo and was previously named "*Neuron Availabilty Groups*" in the installer.

Messaging

NEW - A global application setting in `ESBService.exe.config` has been added to force TCP topics to use the IP address of the client instead of the host name when establishing the connection to the client's receiver service. This circumvents the need for DNS lookups. The new application setting is "`UseClientIPAddress`". This can be set to True or False. If not present, false is assumed.

NEW - Added the new "`RemoveProperties`" method to `ESBMessage` class as a way to clear all the custom properties of like prefix.

MOD - The `EnvironmentVariables` collection object accessible within a Business Process or Workflow has been marked as readonly to prevent users from attempting to modify the contents.

FIX - `ESBMessage.Clone()` was sharing `binaryBody` between the original and cloned object - `ESBMessage.Clone` copied the reference to `binaryBody` from the original `ESBMessage` object to the new `ESBMessage` object. Any change to the byte array of either object would also affect the other, and potentially any number of clones that were created from the original message or clones. A copy of the byte array for each clone will now be made.

FIX - Party may not reconnect to a Topic in some circumstances if the underlying Transport for the topic was changed or if the Topic's publishing service failed and later came back on line. Also, in some cases the Party may believe its connected when it is not which could cause the following exception when sending a message:

can't send..not connected error

FIX - When using RabbitMQ based topics and connecting to the Neuron ESB server using the remote Client API, there will be 2 different queues created in RabbitMQ with similar names but different casing.

FIX - When using either TCP or Named Pipes based Topics, users may experience the following exception:

System.InvalidOperationException: Cannot send - party is not connected

This could be followed by an inner exception of:

System.ObjectDisposedException: Safe handle has been closed

This could occur when the party calls a Send() method to publish to the bus. Effectively a race could occur where the pool of connections for the party would be disposed before the Send() call completed.

FIX – MSMQ Topics - If a policy was associated with the “Endpoint Policy” property of the MSMQ transport for a Topic, its expiration value would override the configured Time to Live property of the transport. If the expiration was set to 0, this could cause all messages delivered to the subscriber to almost immediately be rerouted to the System Dead Letter queue as they would expire within moments of delivery.

FIX - When attempting to publish a message using the Neuron ESB Client API within a service hosted by an IIS app pool running under the network service account, the following error could occur:

System.UnauthorizedAccessException "Access to the path is denied"

Neuron 3.5.2.33 Change History

Adapters

FIX – Salesforce.com – When configured with the Salesforce.com REST url, consecutive calls could result in a *System.InvalidOperationException* exception due to the new timeout property introduced in 3.5.2.

NEW – File – Two new capabilities have been added to the File Adapter. 1) The ability to Poll sub folders. When the *File Detection* property is set to *Polling*, the *Include Sub Folders* will be become visible. When set to true, the root folder as well as all its sub folders will be polled for files which will be published to the specified Topic. 2.) The ability to archive files to an alternate folder location after they have been published to the specified Topic. If the *Archive File After Download* is set to true, the *Archive Folder* and *Create Matching Sub Folders* properties will become visible to set.

Neuron 3.5.2.31 Change History

Several critical enhancements and fixes were added to Neuron ESB 3.5 CU2 to augment support for running the Neuron ESB Service runtime and Neuron ESB Discovery Service in Microsoft Failover Cluster Server environments as Generic Applications. These included the following:

- Endpoint Health – The ability to stop and restart Topics and Endpoint services did not work if the Neuron ESB runtime was configured as a clustered Generic Application.

- Endpoint Health – Host Name column did not reflect the clustered network name (Client Access Point) of the resource group (Role) in which the Neuron ESB Service runtime belongs to.
- Workflow – Workflow Availability Groups did not recognize the clustered network name in the machines tab of the Deployment Group and startup as expected.
- RESTful Operational Services – REST calls to `get /neuronsb/api/v1/runtime` would return the host name rather than the clustered network name.
- Auditing Reporting – Message History and Failed Message reports would reflect the host name in the Machine column rather than the clustered network name.
- Active Sessions Reporting - The host name would be reflected in the Machine column rather than the clustered network name.
- WS-Discovery – The host name rather than the clustered network name would be announced and displayed in the Connect dialogs of the Neuron ESB Explorer and Neuron ESB Test Clients.
- MSMQ Management – The message count function of the Neuron ESB Explorer's MSMQ Management screen would always return zero for clustered MSMQ
- MSMQ Management – Editing and saving messages back to their underlying queue would fail for clustered MSMQ
- Workflow Tracking - The host name would be reflected in the Machine column rather than the clustered network name.
- WMI EndpointStateChangeEvent – This event was not firing for the clustered network name after manually registering the event.
- Endpoint Health – Rate and Items columns for workflow endpoints may not increment at runtime.
- Neuron Logging – Microsoft Failover Cluster Server events were being captured in the Availability Group neuron log files
- Neuron Logging - The host name would be reflected in the log files rather than the clustered network name.

Several other fixes have also been included:

Business Processes

FIX – Service Process Step - Calling Service Process step in 3.5 with Restore Headers = true would cause an object reference exception due to the introduction of the new SOAP Header message collection. We thought we had fixed this in CU2, but found another area where this could happen.

FIX – Adapter Endpoint Process Step – Would send out duplicate messages the number of which would coincide with the number of retries associated with the Adapter Policy associated with the selected adapter.

FIX – Custom Process Steps – Would not load at runtime. This was due to a missing Neuron pipeline redirect in the `ESBService.exe.config` and `ESBHost.exe.config`.

Workflow

FIX – Audit Workflow Activity – If the Action property is set to 'Failure' but the Exception property is not set, at runtime this will fail with a *System.ObjectDisposedException* and

CommunicationObjectAbortedException exception. Validation has been added for both design time and runtime.

FIX – Workflow Availability Group Logging – Events were not getting logged to the Neuron ESB Event log, only to the Neuron text logs.

FIX – Transform Workflow - Concurrently executing workflows with Transform activities could cause the workflow to abort.

Neuron ESB Explorer

FIX - Certificate selection dialog – This could throw an unhandled exception when clicking on the list box when no certificates were displayed

FIX - Certificate selection dialog – Under certain resolutions, the Permissions command button would appear misaligned and overlap the list box of certificates.

FIX - Certificate selection dialog – Permissions command button could become enabled if no certificates are in list. This could also throw an unhandled exception if selected.

FIX - Certificate selection dialog – An unhandled exception could be thrown when selecting a non supported find type.

Neuron ESB WS-Discovery

FIX – Discovery could fail with a Null exception object in either the Neuron ESB Explorer’s or Test Client’s Connect form

Topic Transports

FIX – MSMQ Poison Message Handler – Under certain circumstances, this could throw an error message continuously alerting that the service has faulted. This has been corrected.

Adapters

FIX - FTP Adapter - Implicit SSL throws Required Certificate exception. Also fixed issue where if certificate was selected, user could not unselect it.

FIX -SFTP and FTP Adapters - Adding closing trace file after every send. Only happens when verbose logging is enabled.

Neuron 3.5.2.1 Change History

Business Processes

NEW – Business Process Designer - There is a new “Undo” button on the toolbar which will undo all changes to the current Business Process. A Business process will store all changes even after the tab is closed. The “Undo” button will allow users to remove those changes.

NEW – Business Process Designer – When a change is detected in the Business Process Designer, an asterisk (“*”) will appear on the workflow’s tab to indicate that there have been edits made to it.

NEW – Business Process Designer – Users can now make a copy of the entire process. A “Copy” option has been added to the right click context menu of the Processes library.

MOD – Set Property Process Step – If a custom value was entered but not enclosed in double quotes, the property would not be set. Now Neuron will treat values that are not string literals, dates, or a special name in the existing list as if they were a string literal. So saving "myValue" and myValue as the Value in the Expression Collection Editor will display differently but will be treated as a string literal.

MOD – Split Process Step – Previously, when the Synchronous property was set true, if an exception occurred during individual message process, the exception would never propagate beyond the Steps block. This would require a user to use an enclosing Exception process step within the Steps block to report on any errors that would occur. This has been changed so that the exception will be thrown. However, if the Synchronous property is set to false, then an Exception Process Step should be added to catch the exceptions that may occur at the individual message processing level.

FIX – Adapter Endpoint Process Step – Would throw the following exception when used within a Split Process Step (Steps Block):

System.NullReferenceException

FIX – Adapter Endpoint Process Step – When using the process step in a process hosted by an application other than the Neuron ESB runtime (i.e. Party hosted in a .NET application), a null reference exception could occur.

FIX – Business Process Designer – When working on a process within the Business Process designer, if there is a validation error reported when hitting the apply button (specifically with Transform Process Step), the apply button becomes disabled but the save button is not enabled.

FIX – Business Process Designer – When using a mouse wheel to navigate within the Business Process Designer, the following exception may occur:

System.OverflowException

FIX – Audit Process Step – Will now throw an error if the Source ID property is missing from the message header.

FIX – Transform Process Step - When deleting the XSLT and saving, the XSLT is not removed from the step.

FIX – Exporting out a Process to its .npx format would cause a *SerializationException*

FIX – Apply button would not be enabled if code was changed in a Decision Process Step.

FIX – ODBC Process Step - removed the persistence of connection string, prefix, namespace, row and root name as returned message context properties when a reply message is returned. This could cause

the wrong database from being called or results returned if executing multiple times sequentially within a process.

FIX – Service Process Step - Calling Service Process step in 3.5 with Restore Headers = true would cause an object reference exception due to the introduction of the new SOAP Header message collection.

FIX – Query Process Step – If users opened a Query Process step that was previously configured with Parameters, it would be empty.

Workflow

NEW – Workflow Activity – a new “**Timeout**” workflow activity has been added. This activity can contain other workflow activities and be configured with a specific period of delay. The configured period of delay is the amount of time the user wants to allocate for all the workflow activities contained within the Timeout activity to complete their execution. The Timeout activity can be used to “Timebox” one or more activities. The Timeout activity will cause the existing workflow instance to unload into the database. Once the timer expires, the workflow instance will be reloaded and continue running where it left off. Users can inspect the “Result” argument after the Timeout activity has fired to determine its state. A good example of using this is the following. User sends out document to be approved by a manager. However, the manager only has 3 days to approve before it needs to be escalated to a director.

NEW – Users can now make a copy of the entire Workflow definition. A “Copy” option has been added to the right click context menu of the Processes library.

NEW – Workflow Correlation Dialog – This has been enhanced to auto expand selections once the dialog opens. Also, a description of the current correlation properties are now summarized within the status bar of the dialog.

NEW – Workflow Designer - The “Apply” button for the Workflow Designer is now disabled until a change is detected.

NEW – Workflow Designer – When a change is detected in the Workflow designer, an asterisk (“*”) will appear on the workflow’s tab to indicate that there have been edits made to it.

NEW – Workflow Designer – Error messages can now be copied from the Errors tab of the Workflow Designer.

NEW - Workflow Activity - a new “**MessageCorrelationScope**” activity has been introduced. This activity will register a correlation set for the workflow so that the workflow can use the “**ReceiveMessage**” activity to receive messages without having to use a “**PublishMessage**” activity to first publish a message and register the correlation scope. Very useful in non-correlated workflows.

MOD – Workflow Activity – The **PublishRequestMessage** now will clone requests before sending the request and assign a new message identifier to each request.

MOD – Workflow Activity – The Delay activity is now a persistent Delay. This means that after 60 seconds of delay, the workflow will be serialized into the database. Once the delay period expires, Neuron ESB will reload the workflow into memory and complete execution of the workflow.

MOD – Workflow Exceptions – Reporting has been augmented on capturing and logging exceptions.

MOD – Workflow Activity - **Transform** workflow activities were outputting XML encoded as UTF-16. This has been changed to UTF-8.

FIX – Workflow Activity - **PublishRequestMessage** did not work correctly, causing a *System.ArgumentException* exception, when used in multiple branches of a Parallel activity.

FIX – Workflow Activity - The **ExecuteProcess** workflow activity would report that it could not find and load the referenced process at runtime. It would also report a *System.NullReferenceException* during design time testing.

FIX – Workflow Testing/Simulation – When testing workflow using the Edit Test Message dialog, the message and header properties submitted would reflect the outcome of the previous test run. Now all messages are cloned before submitting them for testing.

FIX – C# Code Workflow Activities would not recognize Assembly References added to the parent workflow.

FIX – Inline C# code edit window – Apply button would not be enabled even after edit.

Workflow Tracking

NEW – Workflow Tracking – There is now a new tab under the Workflow Tracking called “Messages”. This will display all messages pending workflow instance activation. This allows users to manage the messages that will start workflow instances. Users can suspend, resume, delete or move to message history and move to failed messages.

MOD – Workflow Tracking context menu commands – The context menu commands (i.e. Cancel, Suspend) have been changed to a synchronous model and set to update the UI automatically.

FIX - Aborted or cancelled workflows sometimes were not being removed from the list of actively running workflows.

FIX - The Neuron ESB Explorer window was not in focus when Workflow Tracking Detail modeless window is closed.

FIX – The Start, Suspend, Cancel and Resume context menu commands could fail under specific circumstances.

Workflow Runtime

FIX - Workflow Availability Groups and Endpoints - All Workflow Availability Groups and Endpoints will restart several time if a save is made to just one definition that one workflow endpoint is assigned to.

FIX - If there are completed workflow instances in the database, when the Neuron ESB service is restarted the following error may be logged to the Neuron ESB Event Log: you will see several errors in the event log like the below error:

An exception was reported on the workflow runner thread for workflow endpoint "<Endpoint Name>".

FIX – Stopping a workflow endpoint may result in the following error being logged:

Collection was modified; enumeration operation may not execute.

FIX – Editing and Saving a Workflow definition when the Availability Group it's assigned under is running may cause the following exception:

System.NullReferenceException

FIX – Availability Group throws error on Restart after SQL Server comes back online after previously going offline.

FIX – Cannot write to a closed TextWriter – When shutting down the Availability Group's underlying host, the following exception may occur:

Exception: Cannot write to a closed TextWriter

FIX – Attempting to stop an Availability Group may cause the underlying Availability Group host exe to crash. This could happen if there were currently executing workflows that were taking too much time to unload. This has been corrected.

FIX – Editing and saving changes to an Availability Group while closely followed by the shutting down of the runtime could cause the following errors to be reported:

Exception: Object reference not set to an instance of an object.

Exception: The given key was not present in the dictionary.

FIX – Duplicate workflow instances could be started for a single message under specific scenarios.

FIX – When shutting down the Neuron ESB runtime and an Availability Group was running, the following exception may be reported:

Exception: Collection was modified; enumeration operation may not execute.

Service Endpoints

FIX – REST based Service Connectors - Setting the url dynamically for REST based service connectors did not work using:

```
context.Data.SetProperty("Addressing","To","http://some url");
```

Discovery

MOD - If we can't set the firewall rules on startup, we will log an informative message.

Neuron ESB Runtime

NEW – REST API – The current Neuron ESB deployed version has been added to the REST API call:

/neuronesb/api/v1/runtime

MOD – MSMQ Channel – On startup, if the runtime cannot set the registry key to allow cross domain msmq access, an informational message will be logged rather than a warning.

MOD - MSMQ Poison Queue handler which automatically moves all messages found in the Poison Queue to the Neuron ESB Failed Message table would only start if the Topic's MSMQ network property, Direct, was set to true.

MOD – MSMQ Receive Service was removed from the runtime. This was deprecated in 3.0. This was necessary in previous releases to support remote transactions using MSMQ 3.0. MSMQ 4.0 (not 3.0) is supported as a Topic transport in Neuron ESB version 3.x.

FIX – When running the Neuron ESB runtime under a user account and setting an ACL to a Party associated with Endpoint, the following exception may occur:

Exception: Unable to cast object of type 'System.Security.Principal.GenericPrincipal' to type 'System..Security.Principal.WindowsPrincipal'.

FIX – When the Neuron ESB service starts up and a Solution is not assigned to the runtime, Neuron ESB will now log and present an informational message letting the user know how to fix.

FIX – Management Service – if an endpoint was unable to update its statistics, an error would be written but the endpoint name was excluded from the message

FIX – Management Service – The following exception may be reported in rare circumstances:

Exception: Violation of PRIMARY KEY constraint 'PK_ActivitySession'. Cannot insert duplicate key in object 'dbo.ActivitySession'.

FIX - Some error messages appear may appear in the Neuron logs indicating that a party was not connected to a bus or could not publish to a topic, but there was no indicator which party could not connect because it was not included in the error message.

FIX – Named Pipes – Parties may fail to connect to a Named Pipe topic if they're app.config file contains or is missing a Neuron ESB runtime Instance name.

FIX – Modifying Service Endpoint during runtime may log the following exception:

Error while unloading appdomain. (Exception from HRESULT: 0x80131015)

Neuron ESB Explorer

NEW – Users can now delete 1 or more messages from Message History and Failed Messages Reports as well as from the Message Viewer when called from Message History or Failed Messages. This is exposed as right click context menu.

MOD – Users could not query against the same database if a different instance of the Neuron ESB Explorer is used, versus the runtime instance that created the database entry. Previously when installing a Neuron ESB Instance, instance aware client tools (Neuron ESB Explorer) were also installed. Being instance aware prevented someone from querying any database type report where the database entries were created by a different runtime instance. Now the Neuron ESB Explorer is instance agnostic in respect to the database.

FIX – Message Viewer – When deleting a message, the underlying list would be refreshed with a message from the underlying report.

FIX – User Property Cache Corruption – If the user property cache (used to store a user's preferences) becomes corrupt do to moving back and forth between installed versions of Neuron, or for any other reason, the following unhandled exception may occur when the Neuron ESB Explorer is opened, closed or when a solution is being saved:

System.ArgumentException: The parameter 'sectionGroupName' is invalid.

This has been changed so a dialog is presented with instructions on how to manually clear the user cache.

FIX – Creating Neuron ESB Database – If the Sql Server Agent service was stopped at the time when a user attempts to create a database through the Neuron ESB Explorer, the following exception may result:

System.FormatException: Input string was not in a correct format.

This has been changed to show a dialog directing the user how to correct.

FIX – Neuron ESB Explorer may not close when a user attempts to close through the control box function due to a background process that was running on the UI thread.

FIX – Saving an XSLT with a function can generate an error. On save, Neuron ESB would validate the literal string that's being passed in. Now the Neuron ESB Explorer will display a dialog alerting the user that the XSLT could not be validated but save the document to the repository.

FIX – Message History and Failed Message Reports – Clicking either the "Reply to Session ID" or "Workflow Instance ID" column to sort would generate the following unhandled exception:

System.IndexOutOfRangeException

FIX – Message History and Failed Message Reports – Clicking either the "To" or "From" column to sort would generate the following unhandled exception:

System.Data.SqlClient.SqlException (0x80131904): Incorrect syntax near the keyword "N"

FIX – Loading Solution with non-Neuron XML files in solution structure would throw the following exception:

System.InvalidOperationException: There is an error in XML document (1, 2)

Neuron will now report a specific error with steps to correct as well as filter out all non *.xml files in the solution folder.

Endpoint Health

FIX – Workflow Endpoints – Endpoint Health would list the correct number of Workflow Endpoints started, but all would be the same name.

FIX – Unhandled Exception could occur while viewing the Endpoint Health monitor while shutting down the Neuron ESB runtime.

FIX - Workflow Endpoints could get locked into a "Starting..." state after an Availability Group was assigned to the running deployment group in the Deployment Settings tab of the Availability Group.

FIX – Availability Group reverts to Stopped State – After the first time the Neuron ESB Service is started, if an Availability Group is restarted in endpoint health, the workflow endpoints will eventually restart and go green. The Availability Group will also change to a Started state and go green however, it will then revert to a Stopped state and RED.

SETUP

FIX – Using setup.exe to install an x86 runtime instance AFTER an x64 runtime instance was installed would fail.

Adapters

MOD – Salesforce - Added a timeout property to the Salesforce.com subscription adapter to allow to the retrieval of very large data sets or images.

MOD – Salesforce – Max Number of Connections property has been added. Enhancements to underlying proxy management has been added with additional verbose logging.

FIX – ActiveMQ - Fixed an issue where the adapter does not reconnect to ActiveMQ once there was a disconnect (i.e. the ActiveMQ server stops).

FIX – ODBC - removed the persistence of connection string, prefix, namespace, row and root name as returned message context properties when a reply message is returned. This could cause the wrong database from being called or results returned if executing multiple times sequentially within a process.

Neuron Database

FIX – Purge an archive could fail due to an inverted comma and lack of a database name.

FIX – Azure Service Bus database support has been updated for workflow support. A clustered index to the dbo.EndpointStatus table was recreated and several new tables and stored procedures that were added to support the new Workflow features and enhancements included in CU2.

Neuron 3.5.1.1 Change History

Business Processes

FIX – ODBC Process Step – When setting the dynamic connection string property and running multiple steps in sequence or within a loop, the first setting would take effect, but any other modifications to the connection string property would be ignored. Also, in a prior fix, to use dynamic properties the user was required to clear the associated property grid property. This requirement has been removed.

NEW – JSON Process Step – A new property, “Exclude Xml Root”, has been added. This property is available when serializing an XML document to JSON. When set to True, this will exclude the root element of the XML document from being serialized within outputted JSON.

FIX – Schema Validation Process Step – When a user attempted to delete all schemas entries from the Schema collection dialog, close the dialog and reopen it, all the previous deleted schemas would reappear. The UI was not handling empty collections correctly. This has been fixed.

FIX – Adapter Endpoint Process Step – When attempting to run in a Business Process that is remotely hosted, an exception would occur because the Neuron ESB configuration was inaccessible. This has been corrected.

Service and Adapter Policies

NEW - Republish to Topic option has been changed to preserve all header of custom properties of original message. The policies will also publish 3 additional custom properties that users can access to retrieve the specific error information related to the message. The custom property prefix is "neuron". The properties are "*FailureDetail*", "*FailureType*" and "*FailureDateTime*"

Adapters

FIX – Salesforce.com Adapter – A exception may occur with the following message: The request message was already sent. Cannot send the same request message multiple times.

MOD – Microsoft Exchange Adapter - modified the Microsoft Exchange adapter so if an error occurs during the poll process (Publish mode), the following poll attempt will recreate the reference to the Microsoft Exchange service.

NEW – Microsoft Exchange Adapter – A KeepAlive property has been added to the adapter. Default value is false. This controls whether a request to Microsoft Exchange should contain a Keep-alive header. This property is only applicable for endpoints that target Exchange Online and versions of Exchange starting with Exchange Server 2013.

FIX – Microsoft Exchange Adapter – When processing an email retrieved from Exchange, if the Content-Type could not be found in the Internet Message Headers an “Object reference not set to an instance of an object” exception would be thrown.

NEW – Salesforce.com Adapter – Added the ability to download image and binary files using REST calls.

MOD – Salesforce.com Adapter - Added a dynamic resolver to hide the metadata generation property when OAuth is set to true.

NEW – Adapter Framework - A new Configuration property has been added to the ESBAAdapterBase class. This property will receive the active Neuron ESB configuration when the adapter is created. The Neuron ESB Adapter Framework that ships in the Samples Browser has been modified to use this new property.

FIX – Dynamics CRM 2011/2013 Workflow Adapter Plugin – When trying to install the workflow adapter a System.BadImageFormatException exception may be thrown on the file Neuron.Crm.Workflow.Setup.CustomAction.dll. This has been fixed.

Installation and Setup

MOD – The default settings for the optional Erlang and Rabbit MQ components has been changed so they are no longer selected by default.

MOD – 2 Neuron ESB assemblies were not properly versioned, defaulting instead to 1.0.0.0. This should have no impact. The 2 assemblies are RestControllers.dll and RestHostingService.dll. They are used by the Neuron ESB Discovery service to expose various management REST services.

Runtime

FIX – The Test Client and Workflow Host app.config files have been modified to probe the Neuron ESB Adapters and Pipelines directories for referenced assemblies. If the new Adapter Endpoint Process Step or Adapter Endpoint Workflow Activity were used at runtime, a file not found error could occur.

NEW – REST Documentation - In Neuron EBS 3.5 we released REST API documentation that shows the available REST endpoints and describes what they do. These REST APIs expose solution, runtime, deployment and endpoint management features. This patch provides improved REST API documentation including the ability to test the REST API directly from the help documentation page. To view the help documentation for an installed instance of Neuron ESB 3.5, navigate to the following URL in your browser:

<http://localhost:51002/help>

NEW - REST API - GET /neuronesb/api/v1/Activity/{instance}/{messageid} Now returns a new optimized ESB Message serialization format. This REST API can be used to return an ESB Message from the Failed Audit table within the Neuron ESB database.

FIX – WS-Discovery API – When using the Neuron ESB Test Client, displaying the Connection Settings dialog may log the following erroneous warning to the Neuron ESB Event log: Discovery Online Event received an unknown type from 'net.tcp://<machinename>:9021/Announcement'

Neuron ESB Explorer

FIX – x86 version of the Neuron ESB Explorer did not properly expose Process Step properties for editing. Also, Workflows could not be created using the Create New Workflow dialog within the Neuron ESB Explorer. This issue was isolated to the x86 version of the Neuron ESB Explorer.

FIX – Various UI fixes were made

NEW – A “Refresh” button has been added to the “Getting Started” screen that displays the RSS feed for the Neuron ESB product team blog. This allows users to refresh the RSS feed on demand.

FIX – When trying to load the assembly references set on a C# code process step, if the path of assembly is not in legal format or dependencies of the referenced assemblies are not found, the Neuron Explorer may crash

FIX - Selecting the delete button to delete the assembly references on c# code process step could cause the Neuron explorer to crash.

Workflow

FIX – Attempting to cancel a workflow instance within the Workflow Tracking UI could generate an either an “An error occurred while performing operation Stop on workflow Instance X” or a “Value cannot be null” error message.

FIX – SOAP Collection within the Correlation Set dialog was not editable.

FIX – Neuron ESB Messages returned or sent out by a Workflow Instance at runtime were missing the workflow endpoint name, workflow instance id and workflow name context properties.

FIX – When using the Code Workflow Activity Editors within a Loop Activity, the item property was not accessible via intellisense in the tabbed view of the Code Editor.

FIX – The TransformXml Workflow Activity parameters collection has been changed to a Dictionary<string,object> collection. The values for parameters can now be set at runtime by using any of the Code Editor Workflow Activities just prior to executing the TransformXml Workflow Activity.

NEW - A new Workflow Activity named TransformMessage has been added to the Workflow Toolbox. This activity accepts a Neuron ESBMessage as a parameter and outputs a new ESBMessage containing the transformed output of the source message's body. The Workflow Activity allows customers to dynamically pass in XSLT parameters by using either XPATH, Neuron ESB custom message properties or Neuron ESB environmental variables.

To use a custom property, enter the following into the value field for the parameter:

`{property:<prefix>.<name>}`

For example, if I have defined the custom property “Person.Name” with value “Michael Jones”, then using “{property:Person.Name}” will insert the value “Michael Jones” into the parameter value when the XSLT is evaluated.

For an environment variable:

`{env:<name>}`

If I have the environment variable “MachineName” set to “MJONES01”, then using “{env:MachineName}” will result in the parameter having the value “MJONES01”.

Finally, for an XPath expression:

`{xpath:<xpath-expression>}`

If the source XML looks like this:

```
<person>
  <firstname>Michael</firstname>
  <lastname>Jones</lastname>
</person>
```

Using “{xpath:/person/firstname}” will use the value “Michael” for the parameter value.

Samples

MOD – The correlated Send and Receive sample has been modified to receive all correlated messages on sub topics.

MOD – The correlated Send and Receive and Correlated Workflow samples did not have the default localhost selected in the Deployment Settings of the Availability Group. Also, the Availability Group would show a Stopped state instead of a Disabled state in Endpoint health. If a user ran attempted to run these samples without localhost enabled, they would receive a “The given key was not present in the dictionary” error. This has been fixed.

Neuron 3.5.0.5 Change History

For information on Neuron ESB 3.5 release

<http://www.neuronesb.com/product/latest-release/>

Neuron 3.1.0.448 Change History

Runtime

NEW – WMI FailedMessageEvent – a LargeMessage variable has been added to this event. If the failed message body is greater than 1MB in size, the LargeMessage variable will return True, otherwise False.

Adapters

NEW – ODBC Adapter – Support has been added for using Dynamic Constants in the Value property within the Parameters collection UI associated with both the Polling and Post Polling statements within the Adapter property grid. *CurrentDateTime*, *CurrentDateTimeUTC* and *GUID* are accepted value parameters in the Polling and Post polling parameters dialog box. The first 2 can be used with an optional format specifier. All can support optionally persisting their values in the initial Poll statement and using those values in Post Poll statements.

For parameter values in the polling statement, examples of valid entries would be (2nd and 3rd arguments are optional):

- {\$CurrentDateTime}
- {\$CurrentDateTime,ddmmyyyyhhmmss}
- {\$CurrentDateTime,ddmmyyyyhhmmss,true}
- {\$CurrentDateTimeUTC,ddmmyyyyhhmmss}
- {\$CurrentDateTimeUTC,ddmmyyyyhhmmss,true}
- {\$GUID,,true}
- {\$GUID}

For parameter values in POST polling statement, examples of valid entries would be (2nd argument is optional):

- {\$CurrentDateTime}
- {\$CurrentDateTime,ddmmyyyyhhmmss}
- {\$CurrentDateTime,ddmmyyyyhhmmss}
- {\$CurrentDateTimeUTC,ddmmyyyyhhmmss}
- {\$CurrentDateTimeUTC,ddmmyyyyhhmmss}
- {\$GUID}

If the *True* flag is passed in on the polling statement parameter and then the same variable (minus the True flag) is used on the post polling statement, then the post polling value will be retrieved from a stored collection and used.

NEW – File Adapter – A new File Age Filter has been added and can be applied when the adapter is set to polling mode. This filter is applied/added to the Last Write datetime stamp of the file being processed. If the Last Write time (after the applied File Age Filter value) is greater than the current datetime, the file will not be processed. This can be used to prevent scenarios such as processing files that aren't yet completely written to disk.

Neuron 3.1.0.438 Change History

Business Processes

FIX – Execute Process Process Step- A race condition could occur when a process calls another sub-process and an error is thrown from a different step within a sub-process. An "An item with the same key has already been added." Exception could result.

Neuron ESB Explorer

FIX – Reporting was instance specific when multiple instances of the Neuron ESB are installed. If users opened a solution with a database in an instance other than the one it was created in, reporting results in Message History, Failed Messages and Workflow Tracking would not be returned.

FIX – Conditions Code Editor Dialog – Save and Apply button would be unavailable under low display resolutions

Neuron ESB Runtime

FIX – If the database was not available at the time when the Neuron ESB Runtime is starting, startup would be aborted. The Neuron ESB runtime is now allowed to start but will log an error in the event log alerting user to the database access failure.

Adapters

NEW – ActiveMQ Adapter – Has been updated to use the 1.6.4 Active MQ libraries

FIX - ActiveMQ Adapter - fixed issue where publish mode adapter did not enter a stopped state when the username/password were incorrect.

NEW – Dynamics CRM Subscription Adapter - Added support to report the total number of available records on a fetch xml query when the count attribute is used to limit the result set. For example, if count is set to 2, and there are a total of 12 records, the result set will only include the first two results but the totalRecordCount will be 12.

Neuron Topics

FIX – Rabbit MQ – When running solutions created in older versions of the Neuron ESB Explorer, the underlying queues may not be created as durable queues, even though the Topic's Recoverable property is set to True. For example, once the Party disconnects, the underlying queue would be deleted, preventing the store and forward of the message at a later time. This was due to a legacy setting being applied. This has been corrected.

Neuron 3.1.0.423 Change History

Neuron ESB Messaging

FIX – Single instance parties can fail to connect when configured in endpoints during reboot – If the Neuron ESB runtime was stopped abruptly or the server was rebooted, endpoints using parties that are configured for single instance mode may fail to start up with an erroneous error suggesting that the party is already connected.

Business Processes

NEW – Sign XML and Verify Signed XML Process Steps – Changed Certificate property to a drop down box that displays all certificates registered in the Security ->Authentication->Credentials section of the Neuron ESB Explorer.

Adapters

FIX - ActiveMQ Adapter - Improved support for ActiveMQ topics. Users can now choose between durable and non-durable consumers. Durable consumers are not deleted from ActiveMQ when Neuron disconnects, and when Neuron reconnects it will connect as the same consumer. Also fixed an issue that when ActiveMQ restarts, Neuron endpoints did not automatically reconnect to the ActiveMQ Server.

NEW - ActiveMQ Adapter - Added Environmental Variable binding support for ActiveMQ Username and Password.

FIX - CRM 2013 subscription adapter - Fixed bug where security tokens expire after one day and the adapter endpoint needs to be restarted.

NEW - SharePoint Subscription Adapter - Added support for custom Lookup columns when adding or updating document library items. When setting a lookup column, users must pass in the ID of the lookup value (not the title).

Neuron 3.1.0.412 Change History

Neuron Explorer

FIX – MSMQ Management – On Windows 2012 servers users may receive an erroneous error message i.e. “retry messages queues are not supported with this version of msmq”

Adapters

FIX – Active MQ Adapter – Support has been added for monitoring Topics as well as Advisory Topics

FIX – FTP/FTPS/SFTP – Certain operations can generate the following error, “This final line's reply code does not match the first line's reply code”.

FIX – FTP/FTPS/SFTP – Xceed libraries have been updated to 5.6

FIX – SFTP Adapter – Would throw “Attempted to read past the end of the stream.” Exception on disconnect

FIX - CRM 2013 Adapter - Updating an entity via the SaveCrmRecord command didn't work with lookups.

Neuron 3.1.0.406 Change History

Neuron Database

FIX – Neuron Database Creation broken in build 3.1.0.295

Neuron 3.1.0.404 Change History

Adapters

FIX – CRM 2013 Adapter – When performing a FetchXml query that contains a join, and the query returns an optionsetvalue, the result field value would be Microsoft.Xrm.Sdk.OptionSetValue.

FIX – CRM 2013 Adapter – When creating a Campaign Response in CRM, an exception was thrown stating that the regardingobjectid type must be set.

FIX – ActiveMQ Adapter – When a failure occurred while publishing message to the bus, the adapter would successfully put the message in the failed messages table but did not report an error.

Neuron Explorer

FIX – When a database is configured with SQL Authentication, the Delete All button in the Failed Messages report would throw an exception.

Neuron 3.1.0.400 Change History

Business Processes

FIX – Decision Step – If a branch was disabled, the entire Decision step would not execute.

Service Endpoints

FIX – A NullReferenceException would occur in a client connector if the WS-Addressing information was not present in the SOAP message.

FIX – When the TransportWithMessage security mode was selected, Service Connectors could only be configured with certificate type Credentials. These can now be configured with username/password type Credentials.

Logging

FIX – The Maximum Log File Size configured for the Neuron ESB Service runtime was not being honored and defaulted to 100MB.

Neuron Explorer

FIX – Message History and Failed Message Reports – When selecting the view related messages context menu, an error would occur. This has been fixed. To implement, run the \Neuron ESB v3\Sql\0007_UpdateTo3_1_0.sql sql script against any existing database.

FIX – Selecting “View Related Messages” in either Message History or Failed Message Reports would throw an exception.

Neuron 3.1.0.390 Change History

Neuron Explorer

FIX – Rabbit MQ management - When users navigate to Deployment->Manage->RabbitMQ within the Neuron ESB Explorer, an Index out of Range exception may occur if there is not a Rabbit MQ server defined within the deployment group. This has been corrected

Neuron 3.1.0.386 Change History

Business Processes

FIX – Encrypt XML Process Step – When decryption is used, the following error is thrown: “**Exception: Padding is invalid and cannot be removed.**” This has been fixed.

FIX – Dependency Icon on Process Library toolbar could throw unhandled exception if Process is not selected.

Neuron 3.1.0.384 Change History

Security

FIX – Credentials – When saving a Domain User credential, its internal access type was set to WindowsGroup rather than WindowsUser. This has been corrected

Neuron 3.1.0.383 Change History

Adapters

FIX – Azure Service Bus Adapter – work was done to clean up resources when error occurs or adapter stops polling

Service Endpoints

FIX – REST Endpoints – if NTLM or other security options other than transport:windows and none were selected, an erroneous warning would be written to the log. The REST endpoint would still work however. This fix removes the erroneous warning message.

Neuron 3.1.0.382 Change History

Service Endpoints

FIX – Raw Body on REST based Service Connectors – This property was not being enforced. Previously it had to be manually set in a Business Process. This has been corrected. The current behavior now is if the property is manually set, it will be enforced, otherwise the property defined on the service endpoint will be enforced.

Neuron 3.1.0.381 Change History

Neuron Runtime

FIX - Updated the management service so that it would no longer re-throw SQL based errors, causing errors to be logged twice.

FIX – Use Remote TCP Clients – This was previously disabled. This has been re-enabled

FIX – Neuron Log File Cleanup – Various fixes. Also set to look for old logs to clear pursuant to configure server settings every 10 minutes.

Neuron Explorer

FIX – Certificate testing – When a certificate has been selected by subject name, pressing the Test button would always result in no certificates found, even though it existed. This has been fixed.

Business Processes

FIX – KeyNotFoundException in Code Process step could crash Neuron ESB Runtime. When using incorrect locking at runtime, under the right conditions the Neuron ESB Runtime may stop. This has been corrected.

FIX – Print Process would generate unhandled exception – This was due to a recent upgrade of a helper assembly. This has been corrected.

Neuron Topics

FIX – Removed recording of server and client keep alive pings when in verbose mode

Neuron Auditing

FIX – Failed Auditing when using MSMQ and No subscribers - Fixed issue where after a couple of audits due to no subscribers present for the publish message, object disposed errors would be generated.

FIX – Failed Auditing when using TCP or Named Pipes and No Subscribers – This was inactive. Now the messages will be audited if the Neuron Audit database is installed and configured for the active deployment group.

Neuron 3.1.0.369 Change History

Neuron Explorer

FIX – MSMQ Management – When resubmitting a batch of messages from an existing MSMQ Queue using the Republish form, some messages would fail to publish, throwing an object exception. This would happen if the message was not previously selected, which would cause its underlying body to be retrieved. This has been fixed.

FIX – Resubmitting messages – Users could not manually overwrite the topic in the topic dropdown box when there was a wildcard subscription. This has been fixed.

Adapters

FIX – ActiveMQ Adapter - Previously, if the ActiveMQ adapter that is configured in publish mode encountered an error while publishing a message to the bus, the adapter would place the failed message into the failed message table and continue to pull messages from ActiveMQ, but not publish them to the bus or write them to the failed messages table, resulting in a message loss. Now when the adapter encounters an error, it will still write the failed message to the failed messages table and continue to process messages from ActiveMQ, but the messages pulled from ActiveMQ will be processed correctly – either published to the bus or written to the failed messages table.

NEW – Dynamics CRM Subscription Adapter - Added support for retrieving OptionSetValues and EntityReference names when using the GetCrmRecord command. When the FetchXml returns an OptionSetValue, the return element will contain both the numerical value and the text description of the option set value. When the FetchXml returns an EntityReference, the return element will contain both the Id and the name of the referenced entity.

Neuron 3.1.0.367 Change History

Business Processes

FIX – ODBC Process Step – if used sequentially, in a Request/Response pattern within a Business Process, the return message on all subsequent calls will inherit the Namespace, Connection String, Root node and Row node name of the first returned message due to the ability to dynamically set these properties at runtime. Now, dynamic properties will not be used unless their respective design time properties are blank.

NEW – Detect Duplicates Process Step – Modified to auto detect the database during design time testing. No longer a need to feed it a connection string using a custom property

Adapters

FIX – ODBC Adapter – if used sequentially, in a Request/Response pattern within a Business Process, the return message on all subsequent calls will inherit the Namespace, Connection String, Root node and Row node name of the first returned message due to the ability to dynamically set these properties at runtime. Now, dynamic properties will not be used unless their respective design time properties are blank.

NEW – ODBC Adapter – The new “Connect on Startup” property has been added to allow users to control whether or not the Neuron ESB runtime will attempt to connect to the data source when it starts up the adapter. Default is set to false.

Neuron 3.1.0.363 Change History

Business Processes

FIX – Error may occur when attempting to drag an existing Process folder under another folder

FIX – Error may occur when user places cursor on existing process within a folder and selects “Create Folder” from the context menu.

FIX – Error may occur when attempting to Paste an existing Code Step within the Catch block of the Exception Process Step

Neuron Client API

FIX - Modified the Party static constructor to set the log4net logging level based on the configured value of the esbTraceSwitch trace switch in the program's configuration file.

Neuron 3.1.0.359 Change History

Neuron Transports

FIX – Named Pipes – The newly introduced Max Concurrent Sessions and Max Concurrent Instances properties were not being saved if modified from their defaults.

Neuron 3.1.0.358 Change History

Service Endpoints

MOD – Service Connectors – Rollback partial feature :http headers with non REST Bindings:

- In 3.1.0.312, we introduced the capture and restore of HTTP headers for all http based protocols. Previous to 312, we only did this for REST. We have decided to revert back to only capturing and restoring HTTP headers for the REST binding as it was interfering with our native capabilities to mediate between protocol boundaries.

Neuron Explorer

FIX – Rabbit MQ Management – When viewing the Rabbit MQ management screen (i.e. ‘Deployment -> Manage -> Rabbit MQ’ from the Navigation menu) when there is no Rabbit MQ servers configured for the selected deployment group, a “System.ArgumentOutOfRangeException: Index was out of range” exception will occur. This has been fixed.

Neuron 3.1.0.350 Change History

Neuron Logging

FIX - Modified the context.Instance.TraceInformation and context.Instance.TraceError methods. Both methods were outputting their log content regardless of the value of the esbTraceSwitch value in the ESBSERVICE.exe.config file. Additionally, we added the context.Instance.TraceWarning method to allow users to output Warning messages to the Neuron ESB Log files.

Business Processes

FIX – Copy and Paste Process Steps – When copying and pasting process steps from one process to another, the final copied steps would not be saved correctly. This has been fixed.

FIX – Publish step within Parallel Step – Using a Publish step within a Parallel step would throw an “Object reference not set to an instance of an object.” Exception. This has been fixed.

Neuron 3.1.0.343 Change History

Service Endpoints

FIX – Keep Alive and HTTP Headers – Extended this functionality so that the capture and restore of HTTP Headers for HTTP based bindings can be controlled by the existing “Capture” and “Restore” custom header checkboxes that exist on the Client Connector and Service Connector tabs. Also, extended the “Keep Alives Enabled” checkbox on the Binding tab of Service Endpoints to apply to all HTTP based bindings. Previously these only applied to REST.

FIX – Action property not in sync with HTTP header property – if the Context.Data.Header.Action property is changed, it will now override the existing HTTP header property.

Neuron 3.1.0.340 Change History

Neuron Auditing

FIX – Using Auditing with MQSeries adapter may cause exception preventing auditing of messages. Using Neuron Auditing for messages published by an MQSeries adapter endpoint which has “Include Metadata Properties” enabled (located on the General tab of the adapter endpoint) will sometimes cause a “hexadecimal value 0x02, is an invalid character” exception within the auditing service, preventing the message from being persisted to the Audit database. This did not affect message delivery though. The exception occurred due to the MQSeries source queue generating a non XML conformant message id. This has been corrected within the auditing service.

Service Endpoints

FIX – The default values for the Max Concurrent Sessions and Max Concurrent Instances settings on Service endpoints have been changed to match Microsoft guidance. Specifically, Max Concurrent Sessions now defaults to 100 times the processor count, whereas Max Concurrent Instances defaults to the sum of Max Concurrent Sessions and Max Concurrent Calls. All three parameters can be modified within the Neuron ESB Explorer.

FIX – REST Binding – When doing a POST, the REST binding has been modified to exclude sending back the transfer encoding, connection and keep alive http headers to the calling client.

Business Processes

FIX – Service Process Step – Has been modified to check for an HTTP context before attempting to read custom binding information within an application configuration file to support hosting in IIS.

FIX – Service Process Step - Action validation has been removed so as to work with SOAP services that do not require an action to be specifically supplied.

FIX – Service Process Step – Modified Meta Data retrieval to always use http get mode on url extensions of “?wsdl”, falling back to meta data exchange mode on failure. The reverse for non “?wsdl” extensions. Previously, some extensions for “?wsdl” would be resolved with later but would generate an exception when creating the channel to the service.

Neuron Topics

FIX – Peer Network Channel – Various fixes were put in place.

FIX – Rabbit MQ Channel – If using the Client API and Pause() was executed off of the Subscriber object, a thread abort exception would occur in the ESB Channel Manager. This has been resolved.

Neuron Configuration

NEW – A Max Concurrent Sessions and Max Concurrent Instances property has been exposed for all internal Neuron services and all TCP/Named Pipes and MSMQ based Topics. The default values has been set to assume an 8 processor/core server configuration, which can be modified. Existing configurations should be opened within the Neuron ESB Explorer and then resaved to accept the new configuration modifications.

Neuron 3.1.0.321 Change History

Neuron Configuration

FIX – Import/Export Configuration – Fixed issues when importing and overwriting values in an existing configuration. Areas affected are importing of: Topics, Publishers, Subscribers, XML Schemas, XSL Transformations, XML Documents, WSDL Documents, Adapter Endpoints, Service Policies, Administrators, Deployment Groups, and Bridges. Status bar now shows that configuration is modified after import. Command line tool now supports importing/exporting of Administrators.

FIX – Permissions Error when modifying Configuration – When attempting to modify a Neuron ESB configuration a “System.IO.IOException: Cannot create a file when that file already exists” may result if the permissions of the underlying entity XML files were changed. This fix will instead throw an exception that specifically says the issue is a Permissions issue and the file affected, directing users to correct the permissions.

Neuron Runtime

FIX – Start up fails - If more than one Topic, Service or Adapter endpoint had the same ID within their respective XML file, Neuron would throw an ambiguous dictionary error and fail to start up. This has been corrected. Now Neuron will start up but throw a very specific error such as:

Exception: Unable to add the endpoint 'Test_Mail' of type 'MsmqPublishingService' with the ID of '570fadb7-f141-47a8-b394-75d9418b958b' to the Endpoint Tracking system. Another endpoint 'Mail_Test' of type 'MsmqPublishingService' already exists with that ID. Please ensure that the IDs of all Neuron ESB Entities are unique.

This error usually results when users manually copy an entity XML file to create a new one, but fail to change the ID property to something unique. Changing the property to a unique value will solve the issue.

Service Endpoints

FIX – Keep Alive support for REST – In some cases, the first REST call to a REST endpoint may fail due to Connection and Keep Alive headers being forwarded back to the calling client. All subsequent calls would succeed. In those

cases users can now disable Keep Alive on the Binding tab of a Service Endpoint within the Neuron ESB Explorer to ensure all calls are successful.

NEW – Max Concurrent Sessions and Instances – These are now exposed on the Client Connector tab of Service Endpoints. Previously, the default values were set to 2147483647. The new default value for Max Concurrent Sessions, Max Concurrent Instances as well as Max Current Calls is set at the number of processors/cores * 16 plus the number of processors/cores * 10. These values can now be modified in the UI. Modifying these service throttles should be done to control performance and throughput of the Client Connectors (Neuron ESB hosted Services).

Adapters

NEW – Dynamics CRM Adapter - Support for “Upsert” with the Dynamics CRM 2013 Subscription adapter has been added.

The Dynamics CRM 2013 adapter now supports an upsert with the SaveCrmRecord command. To enable this functionality, a new attribute called “ignoreFailedLookup” has been added to the Lookup element. The default value of this attribute is false, indicating to the adapter to throw an exception when the lookup doesn’t return any results. When set to true, the adapter will continue processing the request. When the SaveCrmRecord command is executed, any fields that are configured for a lookup will still attempt to reference the lookups that are part of the request. If the lookup succeeded, then the value that was returned from the lookup will be used. If the lookup had failed, then the entity field will be ignored while performing the create or update.

To perform an upsert, you have to include a lookup to retrieve the ID of the entity you want to either create or update. If the lookup of the entity ID succeeds, then the SaveCrmRecord will result in an update. If the lookup fails, the command will result in a create. Here is an example of a contact upsert:

```
<NeuronCrmAdapterMessage>
  <CrmCommands Transactional="true">
    <Lookups>
      <Lookup id="contactid" entity="contact" lookupfield="contactid" ignoreFailedLookup="true">
        <searchfield name="lastname" value="Smith" typename="string" operator="equal" />
      </Lookup>
    </Lookups>
    <SaveCrmRecord allowCreate="true" id="contact" entity="contact" identityfield="contactid">
      <properties>
        <contactid typename="Uniqueidentifier" lookupid="contactid" />
        <firstname typename="String">John</firstname>
        <lastname typename="String">Smith</lastname>
      </properties>
    </SaveCrmRecord>
  </CrmCommands>
</NeuronCrmAdapterMessage>
```

Note – the ignoreFailedLookup attribute has no effect on the SetCrmRecord or DeleteCrmRecord commands. Both of these commands require an entity ID to perform their respective actions.

NEW – File Adapter – The ability to configure either Polling or Eventing for file detection has been added. The previous default behavior was Eventing which was backed up by polling. Now users can specifically configure to use only one or the other. When Polling is specified, users can configure the polling interval in seconds.

Additionally, when polling is configured, users can specify the file attribute to sort by (i.e. name, create date, last accessed date, last write date) as well as the sort order.

FIX – Sales Force Adapter – Rest support – Now setting the ESB Message Http.Headers.ContentType equal to the content type being returned by Salesforce REST call. Previously all content types were enclosed in double quotes.

Neuron 3.1.0.312 Change History

Service Endpoints

NEW – Custom Behaviors – In previous versions an error would be generated by a service endpoint (client connector) when starting up if a custom endpoint or service behavior was applied but a behavior of that “Type” already existed on the endpoint. Now if the endpoint or service behavior “Type” exists, it will be removed and the custom behavior of the same “Type” will be added to replace it.

NEW – Service Connectors – Modifications to work with JSON and detect content type http headers:

- If the outgoing call is to a REST endpoint, the SOAPAction http header will be cleared. Additionally, if the ContentType of the http header property of the ESB Message object has been set, that value will be used to control the outgoing content type of the request.
- Previously, on replies returned by a service connector or messages received by a Client Connector, only the HTTP header properties would be retrieved if the binding was REST. Now they are retrieved for an HTTP binding protocol. Also, previously the content type was not inspected. In many cases this would result in a flag being placed on the ESB Message indicating the payload was binary. This would happen in the case of JSON payloads. This has been changed so that the content type is now inspected and the binary payload flag is set appropriately.
- Limiting the capture of custom SOAP headers to NON REST bindings

Process Steps

FIX – JSON Process step:

- A check for the http content type header has been added to auto detect JSON.
- The assembly picker dialog was limited to displaying only serializable classes. This has been corrected.
- When converting to binary objects/classes, serialization options are automatically expanded in cases where object is not serializable.
- When converting to XML, http content type header will be set to “application/xml”. When converting to JSON, http content type header will be set to “application/json”

FIX – Audit Process step – When testing in design mode with a new ESB Configuration, an Audit Proxy Null error would occur, even if the solution was running on the local machine and the correct Party ID and Topic were entered in the Edit Message dialog. This has been corrected.

FIX – Process Execution Process step – When testing this in the Process Designer, if a user changes the underlying process being called, that change would not be reflected the next time the user runs the Execute Process step without first closing all processes. This has been fixed.

Topics

FIX – Rabbit MQ – When recovering from a connection failure, the following warnings may appear in the event log suggesting that messages already exist in a collection:

- Existing message already exists in collection with same key
- An ACK was received for multiple messages on topic 'X', but the channel does not have the original messages.

This collection is for internal tracking. The error was due to a race condition which has been resolved. The warnings have been changed to verbose logging.

FIX – Rabbit MQ – When In reconnect mode, a race condition would occur throwing an exception causing a loop of reconnects. This has been corrected.

FIX – Peer – Occasionally this would report an object null exception on a dictionary during configuration change events.

Neuron Logging

FIX – “NeuronEventLog:” messages would appear in the Neuron ESB trace based logs.

Neuron 3.1.0.301 Change History

Neuron Monitoring

FIX – RabbitMQ Endpoint Health Monitoring – The RabbitMQ user accounts configured for deployment groups were not being used to access the RabbitMQ health monitoring site, providing the information to report in Endpoint Health. This has been fixed.

FIX – Rabbit MQ topics in Endpoint Health would not reflect updated user credentials in connection until restart of Topic.

Adapters

FIX – Salesforce.com – When sending a message through to the adapter, if the ESB Message Soap Action header was not set, a fault would be thrown. This has been corrected.

Configuration

FIX – Rabbit MQ Configuration – Username and password were not stored as encrypted within Neuron ESB Configuration

FIX – Deprecated Rabbit MQ elements removed from deployment groups. When creating a new deployment group, older Rabbit MQ elements used in versions prior to Neuron ESB 3.1 were still added with default values. These have been largely removed and are no longer used

Queue Management

FIX – Rabbit MQ Queues – were still using old default values to connect and purge rather than new collection of Rabbit MQ Connections introduced in Neuron ESB 3.1

FIX – Rabbit MQ Queues – if an error occurred during connection, the error information would not be reflected in the UI.

Neuron Explorer

FIX – Null Reference exception could be thrown if creating a new deployment group and trying to add a Rabbit MQ Server to the collection

Neuron 3.1.0.297 Change History

Neuron Topics

FIX – RabbitMQ topic memory leak – Under load, the publisher side of the topic could result in a memory leak due to acks/nacks not finding active messages to release. This has been corrected

FIX – RabbitMQ throws thread abort exception when disconnecting party. This has been corrected

Neuron 3.1.0.287 Change History

Adapters

FIX – Dynamics CRM 2011/2013 Plugin – During installation, the plugin would install incorrect versions of the Neuron ESB client assemblies. This has been corrected.

Messaging

FIX – Submitting messages directly through a Publisher object was not setting the machine, username and SID properties. This has been corrected

Neuron Monitoring

NEW – WMI (Windows Management Instrumentation) endpoint monitoring events have been added to Neuron ESB. Users can now subscribe to status events for Adapter or Service endpoints using simple WMI queries. For example, within a .NET application a query watcher may look like this:

```
static void Main()
{
    var managementScope = new ManagementScope("\\\\.\\root\\Neudesic_ESB_v0");
    managementScope.Connect();

    var eventEndpointQuery = new WqlEventQuery("EndpointStateChangeEvent");
    eventEndpointQuery.Condition = "Type = 'AdapterEndpoint'";

    var watcher = new ManagementEventWatcher(managementScope, eventEndpointQuery);
    watcher.EventArrived += EndpointChangedEventHandler;
    watcher.Start();

    Console.WriteLine("Listening for events. Press Enter to exit.");
    Console.ReadLine();

    watcher.Stop();
}
```

The associated **EndpointChangedEventHandler()** could look like this:

```

static void EndpointChanged(object sender, EventArgs e)
{
    string zone = e.NewEvent["Zone"] as string;
    DateTime eventDate =
        System.Management.ManagementDateTimeConverter.ToDateTime(e.NewEvent["Datetime"] as
        string);
    string type = e.NewEvent["Type"] as string;
    string name = e.NewEvent["Name"] as string;
    string state = e.NewEvent["State"] as string;
    string hostName = e.NewEvent["Hostname"] as string;
    string instanceName = e.NewEvent["EsbInstanceName"] as string;
    string application = e.NewEvent["Application"] as string;
    string deploymentGroup = e.NewEvent["DeploymentGroup"] as string;
    string endpointId = e.NewEvent["Id"] as string;
    string info = e.NewEvent["Message"] as string;

    var message = string.Format(
        CultureInfo.InvariantCulture,
        "Name={0}, Endpoint Type={1}, State={2}, Instance={3}, Zone={4}, Machine={5},
        Application={6} , DeploymentGroup={7}, Info={8}, DateTime={9}",
        name,
        type,
        state,
        instanceName,
        zone,
        hostName,
        application,
        deploymentGroup,
        info,
        eventDate.ToString());

    Console.WriteLine(message);
}

```

The “Type” property returned can be either “ClientConnector”, “ServiceConnector” or “AdapterEndpoint”. The “State” property returned can be the text representation of any member of the following enumeration:

```

public enum ServiceState
{
    StateUninitialized,
    StateStarting,
    StateStarted,
    StateStopping,
    StateStopped,
    StateFailed,
    StatePaused,
    StateDisabled,
    StateOutOfService
}

```

If the Service State returned is “StateFailed” then the associated “Message” property will contain the error message.

Neuron 3.1.0.284 Change History

Adapters

NEW – Salesforce.com Adapter - REST & OAuth support has been for the Salesforce adapter. The REST functionality supports both Xml and Json message encoding. This is for the runtime only - metadata retrieval to come later.

FIX – ActiveMQ Adapter - Fixed mapping of ActiveMQ adapter subscribe mode metadata properties to ActiveMQ message properties. If Metadata was enabled, sending a message would cause an exception: Exception while sending message String was not recognized as a valid Boolean. This was due to a recent upgrade of the underlying ActiveMQ libraries.

FIX – Dynamics CRM 2013 Adapter - Fixed bug that gave incorrect exception when an attribute was missing. Now it will return an appropriate exception.

Service Endpoints

FIX – Service Connectors - fixed issue when trying to use a credential on a service connector:
Exception: An item with the same key has already been added.

FIX – Client Connectors – Validation against Access Control Lists when Windows Security is selected would cause a 401 unauthorized error to be thrown in all cases.

Neuron 3.1.0.275 Change History

Neuron Explorer

FIX – When saving the Neuron Configuration in either online or offline mode and either the user (in offline mode) or the Neuron ESB service account (in online mode) did not have sufficient permissions to the underlying Neuron Configuration folder, a generic “Failed to save ESB Configuration” error would be presented. This has been corrected to present the user with a message to verify the permissions on the specific configuration folder.

Neuron Configuration

NEW – Support has been added for manually modifying the Enabled flag within an XML document representing either an Adapter or Service Endpoint within the Neuron Configuration. If the flag is modified while the configuration is running within the Neuron ESB runtime, the changes will be picked up and loaded as long as the change is made within the defined ConfigurationReloadDelay window specified in the esbservice.exe.config file. By default, this is 1 minute.

Service Endpoints

FIX – When a JSON request was sent to a Service Connector configured with a REST binding and the target REST service returned a JSON response, that response would be returned as serialized XML. This has been corrected.

Neuron Database

FIX – Microsoft Sql Azure – When running the 0007 update sql script would fail against SQL Azure. Microsoft recently modified their support for string literals. This has been corrected. Anyone using SQL Azure for the Neuron Database should run the updated “0007_UpdateTo3_1_0.sql” script located in the default sql script directory (i.e. C:\Program Files\Neudesic\Neuron ESB v3\Sql).

Adapters

FIX – Dynamics CRM 2013 Adapter – When attempting to use this adapter, it would throw the following error: “*System.IO.FileNotFoundException: Could not load file or assembly 'Microsoft.IdentityModel..’*”. This has been corrected.

NEW – Active MQ adapter – User credential support has been added to the adapter. Anonymous or Username can be selected for security mode.

Neuron Process Designer

FIX – Audit Step causes Object Reference Exception using the new Debugger

Neuron 3.1.0.268 Change History

Business Processes

FIX – Javascript Process Step – When opening editor a System.IO.FileLoadException may be thrown.

Neuron 3.1.0.264 Change History

Neuron Explorer

FIX – Erroneous datetime format error could be displayed when attempting to test a process in the process designer.

Neuron 3.1.0.260 Change History

Neuron Explorer

NEW – Edit Test Message Dialog – Process testing – Modified dialog to support viewing HTTP header properties after importing a saved ESB Message (*.esbmsg) from reporting.

FIX – Selecting Delete by right clicking on Process or Folder in Process Library did not delete the selected item

Neuron 3.1.0.257 Change History

Neuron Explorer

NEW – Message Viewer from Message History and Failed Message reports now displays all HTTP properties if captured by a Neuron Client Connector. These can be found on the ‘Neuron Properties’ tab. All HTTP properties are preceded by “HTTP”.

Neuron Test Client

NEW – New HTTP properties – HTTP.RemoteAddress and HTTP.RemotePort are now included in the Http collection section within the Message Header list located on the Message History tab of the test client.

Neuron Message

NEW - New HTTP properties – HTTP.RemoteAddress and HTTP.RemotePort have been added. When using a Neuron Client Connector (service hosted by Neuron), the calling user’s IP address and Port will be captured (if

available) and preserved in these properties. These properties can also be accessed within a Code process step by accessing the context.data.http header.

Neuron 3.1.0.256 Change History

Neuron Explorer

FIX – Subscription Conditions – Saved Conditions that were selected as part of a subscription would not be displayed in the Conditions column of the subscription list located on the General tab of Publisher/Subscriber configuration. This has been fixed. Absent an Ad Hoc condition, the previously saved Condition name will be displayed in the Conditions column. If there are multiple Conditions selected then the name of the first condition followed by "..." will be displayed. If there is a previously saved Condition and an Ad Hoc condition defined, then only the Ad Hoc Condition's expression will be displayed.

FIX – Opening a non-Neuron ESB configuration folder would cause an unhandled exception. This has been fixed.

Neuron Process Designer

FIX – Publish Step causes Object Reference Exception using the new Debugger

FIX – Dependency button for Process List could cause an unhandled exception if a Process was not selected.

FIX – Unhandled exception– When creating a new process (before its saved) and if adding multiple steps, then deleting objects, if the designer surface was selected and delete was pressed again, an unhandled exception could occur.

FIX – Mouse would remain as Hourglass if Debugger was stopped by user during Debugging session.

Adapters

FIX -- Dynamics CRM 2013 – More friendly exception handler has been added to handle connection errors.

FIX -- Dynamics CRM 2013 – A null exception could sometimes be reported as a warning in the Neuron ESB Event log during shutdown of the Neuron ESB runtime.

Neuron 3.1.0.242 Change History

Adapters

NEW – Dynamics CRM 2013 – Support has been added to support Dynamics CRM 2013 hosted online by Region.

Neuron 3.1.0.238 Change History

Neuron Topics

FIX – MSMQ based Topics – When using the Neuron Client API on a remote machine attempting to connect to an MSMQ based topic, the following exception may be generated: "Cannot determine whether a queue with the specified format name exists". This has been fixed.

Neuron 3.1.0.236 Change History

Neuron Topics

FIX – Rabbit MQ Topic Memory Leak – If Rabbit MQ based topics are defined, a memory leak could occur over time due to internal monitoring. This could result in exhausting ports on the machine. This has been corrected.

Neuron Client API

NEW – A new Reply() overload has been created to allow users to submit reply messages based on an exception object.

FIX – When attempting to connect to a Neuron ESB Server which is down, the timeout for the connection would sometimes take up to 60 seconds to occur due to internal retry semantics. This has been corrected so that the timeout occurs immediately.

Neuron 3.1.0.233 Change History

Neuron ESB REST Based Monitoring

FIX – Endpoint Health Monitoring – In Neuron ESB 3.1 we released REST based interfaces for exposing all the deployment configuration information, server management capabilities and Endpoint Health capabilities. The Neuron ESB 3.1 REST interfaces for runtime management can be found here: <http://localhost:51002/help> . Endpoint Health interface was not using cached connections to the Neuron ESB Runtime control service. This service proxy is based on TCP sockets. Some operating configurations may not release the socket resources immediately. Overtime this could cause socket exhaustion on those machines. This has been fixed.

Neuron Client API

NEW – An asynchronous send message function, SendTaskAsync() have been added to replace existing deprecated methods. This uses the .NET TPL functions. For example:

```
using (Publisher publisher = new Publisher())
{
    publisher.Connect();
    Task<ESBMessage> task = null;
    task = publisher.SendTaskAsync("Contacts.Lookup", "Test", SendOptions.Request);
    task.ContinueWith(t =>
    {
        if (t.Result.Header.BodyType.Equals("Contact"))
        {
            Contact contact = t.Result.GetBody<Contact>();
        }
    });
    task.Wait();
}
```

Neuron 3.1.0.231 Change History

Adapters

FIX – Microsoft Exchange Adapter – Updated support to the Microsoft Exchange Web Services Managed API 2.1. This adapter will require that .NET 4.5 be installed. Also, root polling of Inbox would return folder not found error. This has been corrected

FIX – Active MQ Adapter – Changed registration name from “Name Adapter” to “Active MQ Adapter”.

Neuron 3.1.0.222 Change History

Business Processes

FIX – Publish process step – When using the publish process step in multicast mode, duplicate messages could be published if there was not a Cancel step following the publish step. This was because we were not resetting the Topic property back to its original value after the publish step. This has been corrected.

FIX - Publish process step – When testing the process step in request mode within the process designer, sometimes the Edit Reply Message dialog would appear prompting for a reply message. After the reply message was copied in, it could throw a ThreadStateException. This function is obsolete and has been removed. With Neuron ESB 3.1 the publish step can be tested live at design time. Users can use a test client to receive the request message and send back a response to the Process being tested in the designer.

FIX – All Code Steps – The auto completion of parentheses and brackets would sometimes display extra parentheses and brackets. This has been corrected.

FIX – Clear All Breakpoints on the process designer did not clear breakpoints on non-Code process steps. This has been corrected.

FIX – C# Code Editor – Editor could sometimes crash if a curly brace was entered following a return from an IF statement. This would only happen in some environments. This has been corrected.

FIX – Importing a Process with a Code step. If a process with a code step was imported using the mouse right click context menu, and the user opened the code step editor and error would occur. The context menu has been removed. Users should use the Import toolbar button.

Adapters

FIX – CRM Plugin Adapters – These have all been updated to deploy all dependent Neuron assemblies into the CRM Async Service's BIN directory (default location: `C:\Program Files\Microsoft Dynamics CRM\Server\bin`), rather than the Global Assembly Cache.

Transports

NEW – MSMQ based topics now support request/response type of messaging using multiple instances of publishers and subscribers. In previous releases, this would usually result in errors/timeouts and incorrect responses being returned to the publishers. This new capability requires the creation of specific response and request queues. These can be created either through the MSMQ Management console of the Neuron ESB Explorer, or through the scripts generated through the Neuron ESB Explorer.

FIX – Rabbit MQ based Topics – If the Rabbit MQ service was down and the parties attempted to connect, they would return an offline event as expected. However, if the Rabbit MQ service was later started, the parties would not automatically detect the start and connect to the Topic. In some cases a “Value cannot be null” error would be reported in the event log. This has been corrected.

Database

FIX – When upgrading a Neuron ESB Audit Database to version 3.1, some database would cause the following errors could occur:

1. `SqlError in ActivityUpdateSession - 8144 Severity 16 - Procedure or function ActivityUpdateSession has too many arguments specified.`
2. `Column names in each table must be unique. Column name 'Instance' in table 'dbo.EndpointStatus' is specified more than once`

These have been corrected.

Neuron Explorer

FIX – Drag and dropping of processes into folders was disabled. This has been corrected

FIX – Unhandled exception would result when trying to delete a process folder. This has been corrected

FIX – Endpoint Health Monitoring – By default, every 2 seconds endpoint health queries the Neuron ESB Runtime for health statistics to report. It does this by creating a service proxy to the runtime, getting the report and then closing the proxy. This service proxy is based on TCP sockets. Some operating configurations may not release the socket resources immediately. Overtime this could cause socket exhaustion on those machines. This has been fixed.

FIX – Endpoint Health Monitoring – Specifically for Rabbit MQ based topics. If the Rabbit MQ service was down, and the Topic's transport was changed to another Transport, the transport name in the Endpoint Health viewer would "flip" back to "RabbitMQ" every 15 seconds with new stats. This was because the heartbeat service was not shut down when the Topic's transport property was changed. Also, the Rabbit MQ's health monitoring service would continue to run in the background. This has been corrected.

FIX – Endpoint Health Monitoring – Specifically for Rabbit MQ based Topics. Every 15 seconds the message processed and message rate would flip to different numbers inconsistent with the current number of messages processed. This has been corrected

Service Endpoints

FIX – REST Endpoints – When using policy for a REST endpoint and specific HTTP Status Codes are selected to define an error condition some may not actually evaluate properly in an underlying protocol exception is thrown in relation to the status code, as in the case of a 409 status code. This has been corrected.

Neuron 3.1.0.183 Change History

Topics

FIX – MSMQ Based Topics – If the MSMQ service was shut down or connectivity was lost to the MSMQ service and automatic dead letter handling was enabled on the Topic, the Neuron ESB Runtime service may stop and restart. Also, if the MSMQ service was restarted, the Topic and its respective publishing/subscribing configured endpoints would not reestablish a connection. Both of these circumstances have been corrected.

Service Endpoints

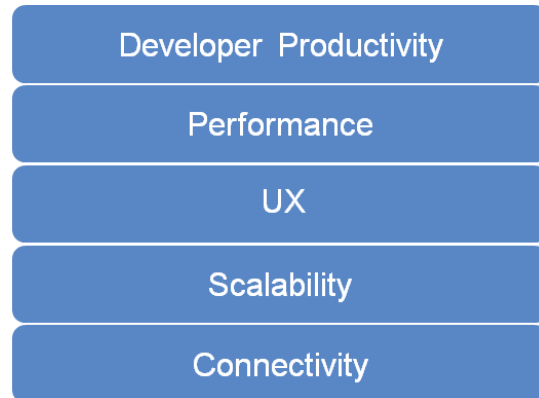
FIX – Client Connectors – If an error occurred during publication of the message through a Client connector, an Object error would be returned rather than the original error. This has been corrected.

Neuron 3.1 (3.1.0.174)

Neuron ESB 3.1 Release Themes

The Neuron ESB 3.1 release contains a number of enhancements and features that continue to make it the easiest, most cost effective and intuitive Application and Service Integration platform to use for the Microsoft .NET environment.

Although there are significant UX improvements as well as new performance, connectivity and scalability options, the predominate focus of this release is developer productivity.



Neuron ESB 3.1 provides significant UX, debugging tools and features that make building, testing, debugging and managing complex business processes far easier than it was with previous versions. More details of these features are included below but some of the highlights in this release are:

- Step by step debugging of Business Processes
- Support for Categories, Folders and Filters
- Design time support for all Business Process Steps
- Set Breakpoints, F5/F11 support and Disable Process Steps
- Tabbed view for multiple Processes
- Tabbed view for multiple Code Step Editors
- Language support for VB.NET, JavaScript and full C# classes

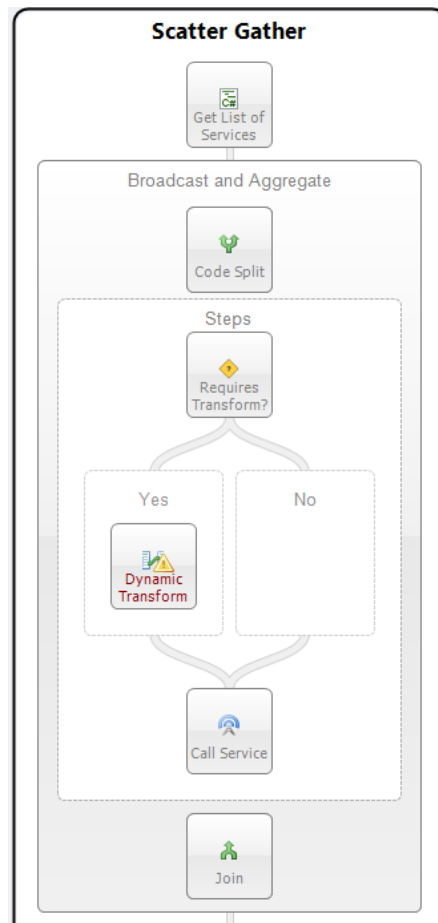
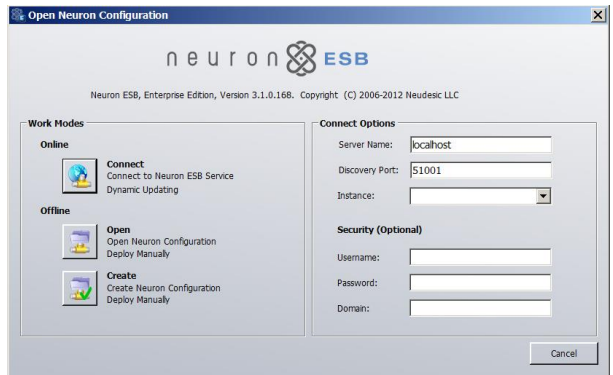
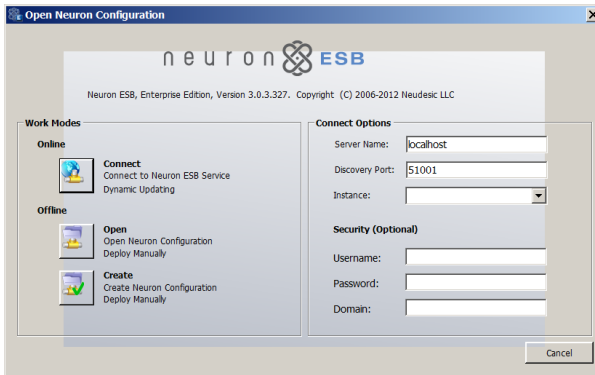
Neuron ESB 3.1 also provides new connectivity options with Salesforce.com and Microsoft Dynamics CRM 2013 adapters that support full metadata harvesting and sample message generation.

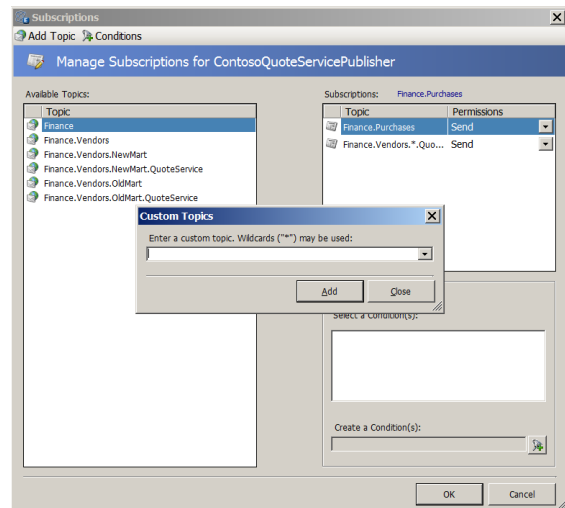
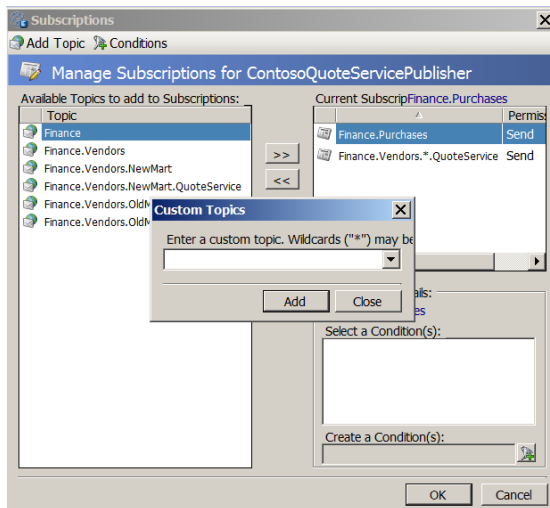
In addition, Neuron ESB has extended support for RabbitMQ by supporting full clustered/mirrored environments with zero message loss and interruption of the Neuron ESB messaging pub/sub system.

Many more feature enhancements and additions have been included in this release the details of which are included below.

User Experience

There was significant work done in the Neuron ESB 3.1 release to support users running the desktop display mode at 125%. Previously, Neuron ESB only supported 100% display mode. This would cause many of the UI elements, text and dialogs within the Neuron ESB Explorer to appear either somewhat distorted and, in some cases, nonfunctional. Below are some before and after images:

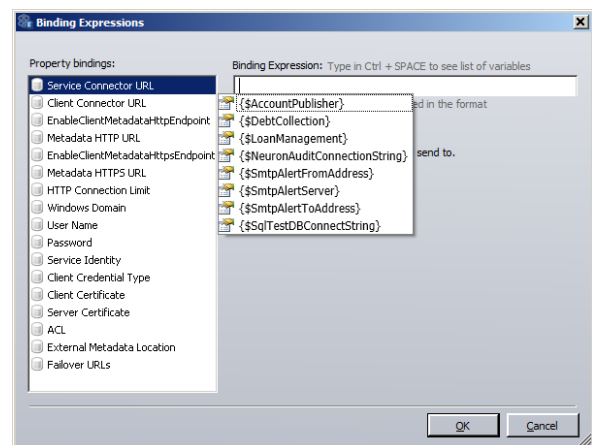




Almost every feature of Neuron ESB 3.1 has a significant UX component, or was influenced by UX. This section is intended to highlight more of the general UX work delivered with Neuron ESB 3.1. It does not in any way express the full extent of that work. More examples of the UX work can be found in many, if not all of the sections of this document.

Environmental Variables

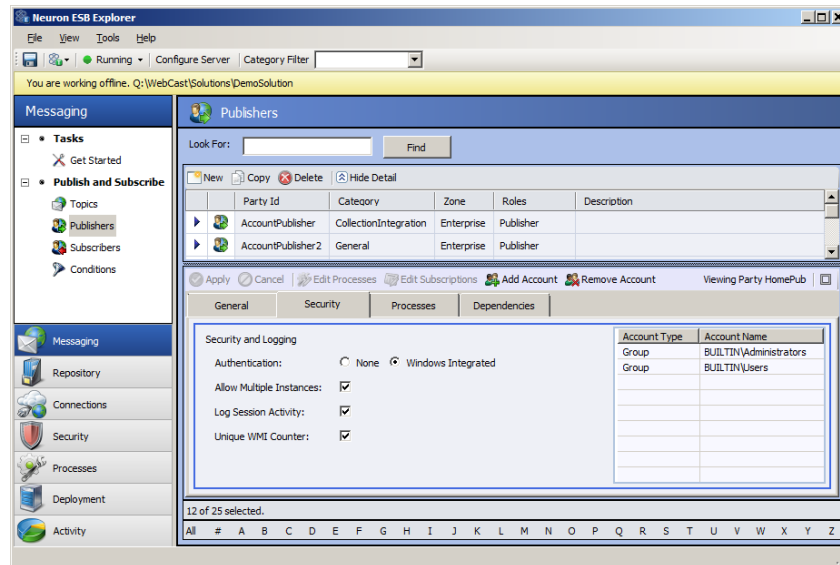
Neuron ESB has always shipped with the ability to use Environmental Variables for configuration of any Adapter or Service Endpoint, Database connection string or Process Step. By using the Bindings Expression dialog box, users can bind environmental variables directly to the properties exposed by those entities. Many developers access the values of Environmental Variables directly in Code Steps within Processes. However, in previous versions of Neuron users were required to manually type in the names of those variables with the proper formatting.



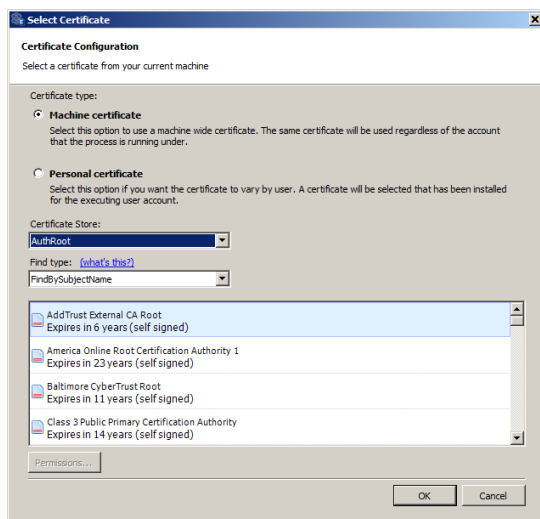
In Neuron ESB 3.1, intellisense has been added to the dialog. By pressing ctrl+space, a drop down list of all available Environmental Variables (filtered by the Category Filter) will be displayed for selection.

Security Configuration

One of the most obvious changes previous users of Neuron ESB will find is that Neuron ESB Parties (i.e. Publishers and Subscribers) are locked down by default. When Publishers or Subscribers are created in Neuron ESB 3.1, the Microsoft Windows accounts that can host and use them are restricted to those accounts that are members of the local ("BUILTIN") Administrators and Users group that exists on the Neuron ESB server. If hosting the Neuron ESB Client API on remote machines, the user account hosting the API must either be a member of those groups on the Neuron ESB server, or the user account must be added to the Party's security membership through the Security Tab of either the Publisher or Subscriber role within the Neuron ESB Explorer. Security can also be removed entirely, the default behavior in previous versions of Neuron ESB.



In addition to locking down Publishers and Subscribers, most of the Security section within the Neuron ESB Explorer has been refactored to simplify the user experience related to creating and managing the accounts and certificates that can be used to configure endpoints and other elements with Neuron ESB.



In previous versions of Neuron ESB, users were required to manually type in search criteria for certificates that existed in a specific store, a troublesome and error prone process. This was required when creating certificate type of Credential in the Security store as well as when a certificate would be required for service or client connector endpoint. In some cases, the Neuron ESB Security Credential store was essentially disconnected from the Service Endpoints for Issuer Credentials, requiring duplicate information to be entered in many different places.

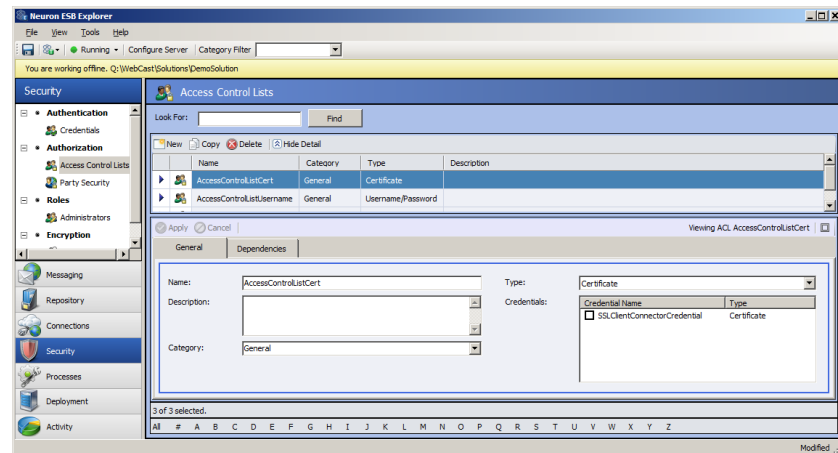
In Neuron ESB 3.1, all certificate selections within Neuron ESB have been linked back into the Neuron ESB Security store located in the Security->Authentication->Credential section of the Neuron ESB Explorer. Additionally, all certificate selection

is controlled through a new UX experience, eliminating the need for users to manually search for and type in search criteria.

In previous versions of Neuron ESB, when user credentials were required for service connector endpoints, users would have to manually enter in account information within the Service Connector tab of the endpoint, rather than use the user Credential created and stored in the Security->Authentication->Credential section of the Neuron ESB Explorer. This happened in other areas as well.

In Neuron ESB 3.1 we've standardized on the Active Directory selector dialog to allow users to select Windows accounts when creating Windows Account type credentials in the Security store. When used, Neuron ESB will also resolve and store the associated SID of the account. All account selections within Neuron ESB have been linked back into the Neuron ESB Security store.

One last area where Neuron ESB 3.1 normalizes the Security experience is during the creation of Access Control Lists. In previous versions of Neuron ESB the credentials that comprised this list were entirely disconnected from the Security store. In other words, when a list was created, users were not prompted to select an existing Credential from the Security store. Instead they were forced to create entirely new ones, with a different UX experience, that were often duplicates of what already existing in the Credential store. In Neuron ESB 3.1 we've unified the experience so that Access Control Lists are now comprised of a selection of Credentials that already exist in the Security Store:



Consistent Entity Management

Neuron ESB 3.1 delivers a new consistent Entity Management experience throughout the Neuron ESB Explorer. Regardless of the entities, such as Topics, Publishers/Subscribers, WSDL, XML, Databases, etc., users should find a consistent interface that varies little from one entity to another. For example:

- Text boxes representing entity names as well as other properties have been extended to support long entity names.
- Dropdown controls that display category, entity or other information have been modified to automatically widen to support the longest entry in the list.
- Descriptions have been modified to support multiple lines and scroll bars.
- Some entity interfaces (such as Service Endpoints) have been reorganized to make data entry and the selection of options more intuitive.
- In many entity interfaces, user selection is enabled/disabled based on context and user entry to reduce opportunities of configuration errors.

MSMQ Management

Neuron ESB 3.1 provides users a way to manage, view and edit pending messages awaiting delivery when using MSMQ based topics. Although this capability existing in the Neuron ESB 3.0 release, the user experience was overwhelming in that it provided unnecessary information to the general user.

[illegible]

The dialog that allowed users to associate WSDL (metadata) with Neuron ESB hosted services has been redesigned to make it more intuitive and easier to use. When a service endpoint (specifically a client connector) is hosted using either http or http (non REST), WSDL can be configured for the endpoint by clicking on the Metadata button located on the Client Connector tab. This is only enabled if the Url prefix is either http or https.

Configure Client Connector Metadata

☐ Do not expose metadata for the client connector.

☐ Access metadata at external location.

External metadata location:

☒ Host a static metadata document.

WSDL Document:

Relative WSDL Path (Optional):

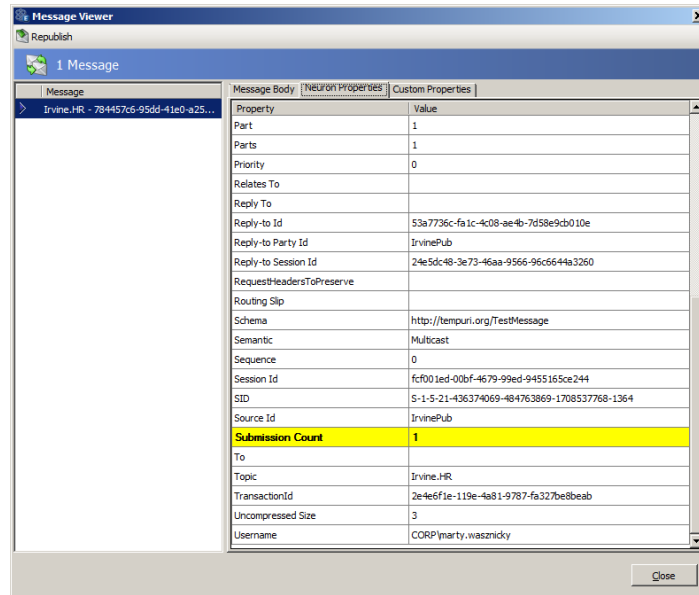
☒ Replace service port URLs with the Client Connector URL

☐ Use HTTPS for the Client Connector URL

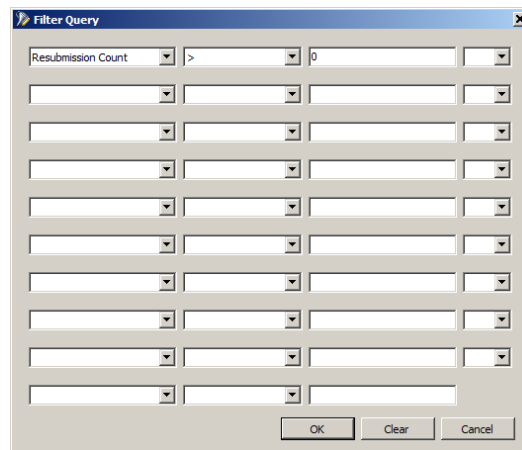
Save Cancel

Previously, Client and Service connectors which were not configured would be displayed in Neuron ESB Explorer's Endpoint Health. In Neuron ESB 3.1 only configured Client and Service connectors are listed.

This has been added to both Message History and Failed Message reports. If a message is resubmitted using the message viewer, its Submission Count Property will automatically be incremented on the message that is resubmitted. Once the message is audited again, the Submission Count property can be viewed within the Neuron Properties tab of the Message Viewer, highlighted in yellow.



Users can use the filter feature of both the Failed Message and Message History reports to query for any messages that have been resubmitted more than once:

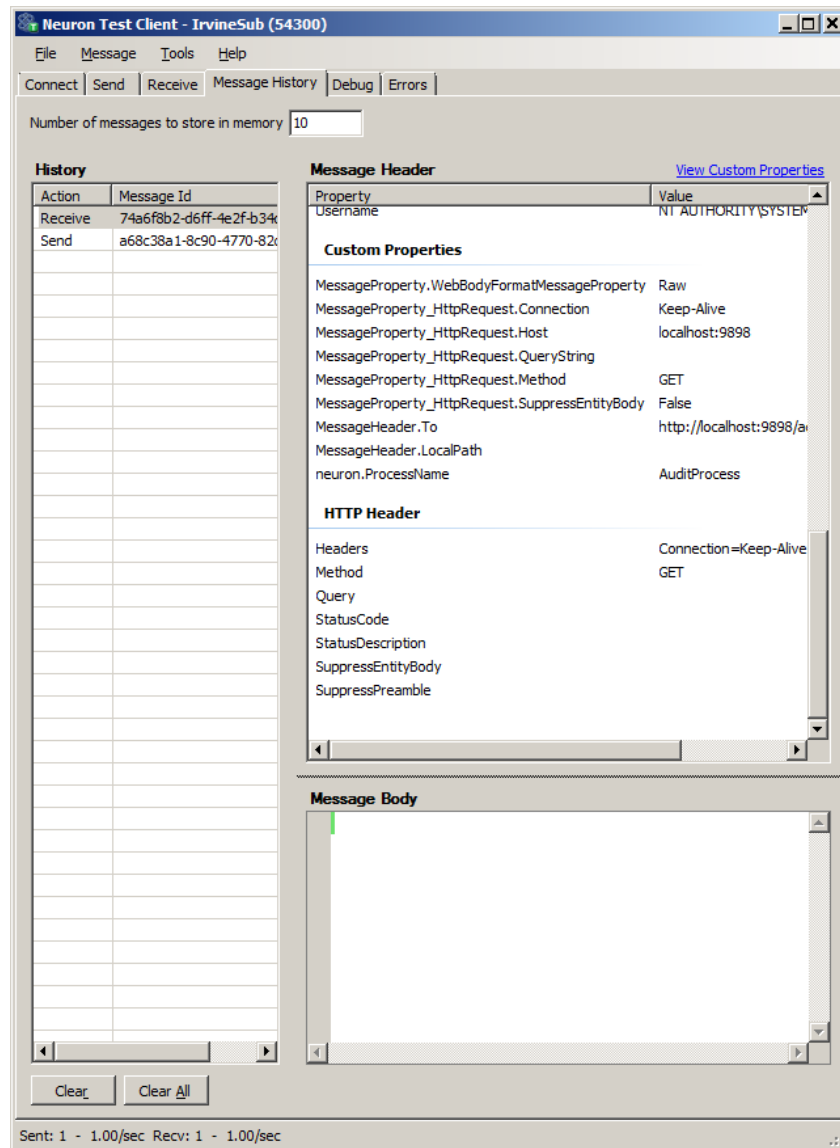


Message Viewer navigation

Message Viewer navigation within the Message History and Failed Message reports has been added. Now, when viewing a message using the Message Viewer, users can click on message records in the underlying report and that record will be displayed in the Message Viewer. There is no longer a need to close and reopen the Message Viewer to view a different record.

Extending Neuron ESB Test Client

The Neuron ESB Test Client has been enhanced to show all Neuron ESB Header, Custom Properties and HTTP Header properties in the Message History window:



FIX – When viewing binary messages within the Message Viewer, the hex editor would not resize correctly if the Message Viewer dialog was resized. This has been corrected.

FIX – Previously, if the solution’s bootstrap address had been changed to anything other than 5000, republishing messages from either the Message History or Failed Message reports would fail.

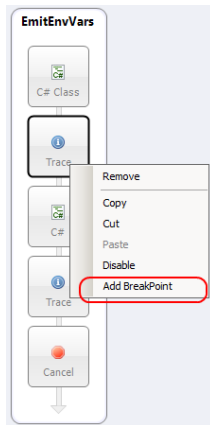
Developer Productivity and Business Processes

Neuron ESB has always provided a way for developers to test the Business Processes that are created within the Neuron ESB Explorer’s Process Designer. Testing entailed submitting a message using the “Edit Test Message” dialog and watching, as each Process Step in the Business Process executed. Users would see a green marker move through the Business Process, highlighting each Process Step as it was executed. However, the testing process lacked true debugging support. For example, there was no way to stop at a specific point in the Business Process and actually inspect the state and context of the Process and/or a specific Process Step. There was no way to debug custom code written within a C# Code Process Step. This required developers to insert custom trace and logging statements prior to testing. There was also no ability to disable branches or Process Steps within a Process.

If extra Process Steps were included during testing, these would have to be removed from the Process before deployment.

Neuron ESB 3.1 introduces true Business Process debugging support at design time so that developers can quickly test, diagnose and fix the Business Processes AS they are developing them. The Business Process debugging experience within Neuron ESB 3.1 is synonymous to the experience provided by Microsoft Visual Studio for debugging .NET applications. For example, developers can set Breakpoints, disable Process Steps, use F5/F10/F11 keys, view exceptions and view the context of the Process and Process Steps.

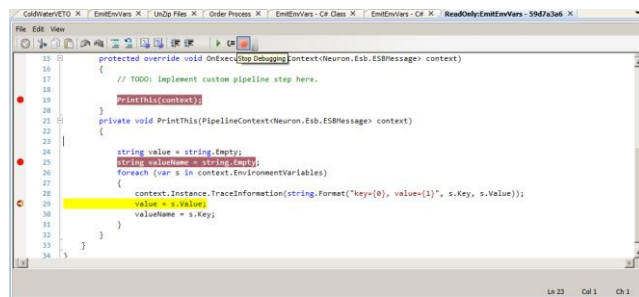
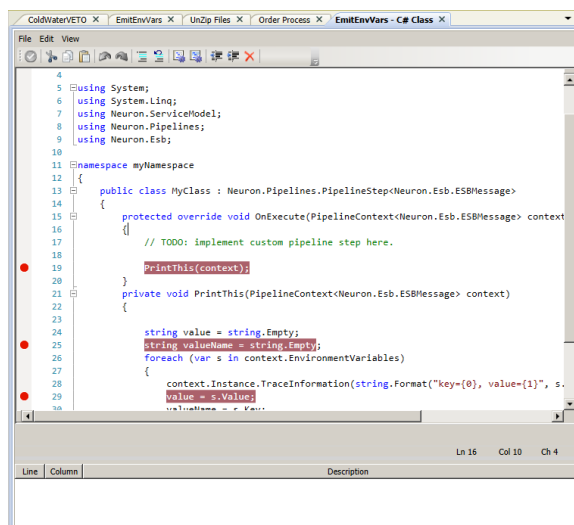
Breakpoints



Setting Breakpoints is both easy and intuitive and can be set on both on Process Steps as well as within any code editor belonging to a C#, C# Class and VB.NET Process Step. Within a Business Process, Breakpoints can be set on any Process Step and selecting “Add BreakPoint” from the context menu. Once a breakpoint is added, the Process Step will change to a reddish brown color. To retain all Breakpoints between opening and closing of the Business Process, the Process must be saved.

Within a VB.NET, C# or C# Class Code Editor, setting Breakpoints is as easy as clicking to the left of the line of code that the debugger should stop on. A red dot will be displayed to the left of the line of code (Breakpoint Indicator Margin) and the line of code will be highlighted in red.

Once a Breakpoint is set and the debugging process has started (by selecting the new “Start/Resume Debugging” toolbar button) , the green marker will move to the first Process Step that has a Breakpoint. Once the green marker moves to a Code Process Step (whether or not it has a Process Step level Breakpoint set on it) that has Breakpoints set within its Code Editor, the debugger will automatically open the Code Editor and stop on the line of code that has the Breakpoint. The line stopped on will be highlighted in yellow and the debug toolbar of the Code Editor will appear.



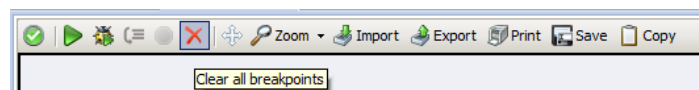
From there, F5 (continue), F10 (step over) or F11 (step into) keys can be used to walk through each line of code in the editor. The debugger can also be stopped by clicking the “Stop Debugging” toolbar button (F1 key). Once on the last line and F5 (or the continue toolbar button is pressed) the

debugger will move to the next Process Step that has a Breakpoint set on it.

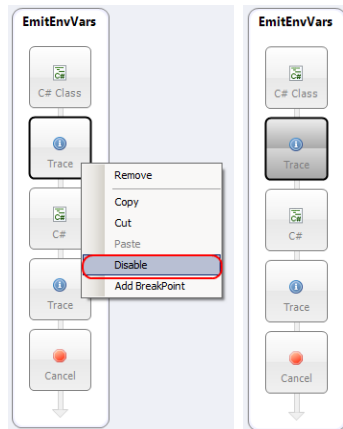
If an unhandled exception occurs, the line will immediately be highlighted in orange. By moving the mouse cursor over the Breakpoint Indicator Margin to the left of the line of code, a tooltip will appear with the error message. If an unhandled exception occurs on a Process Step, the debugger will stop on that Step and highlight it in red. The exception information can be viewed in the Trace Window.



To remove all Breakpoints in a Code Step, there's a red "X" icon on the toolbar. It clears all Breakpoints. The Process Designer has a similar toolbar button that does the same, but for Process Steps.



Disabling Process Steps



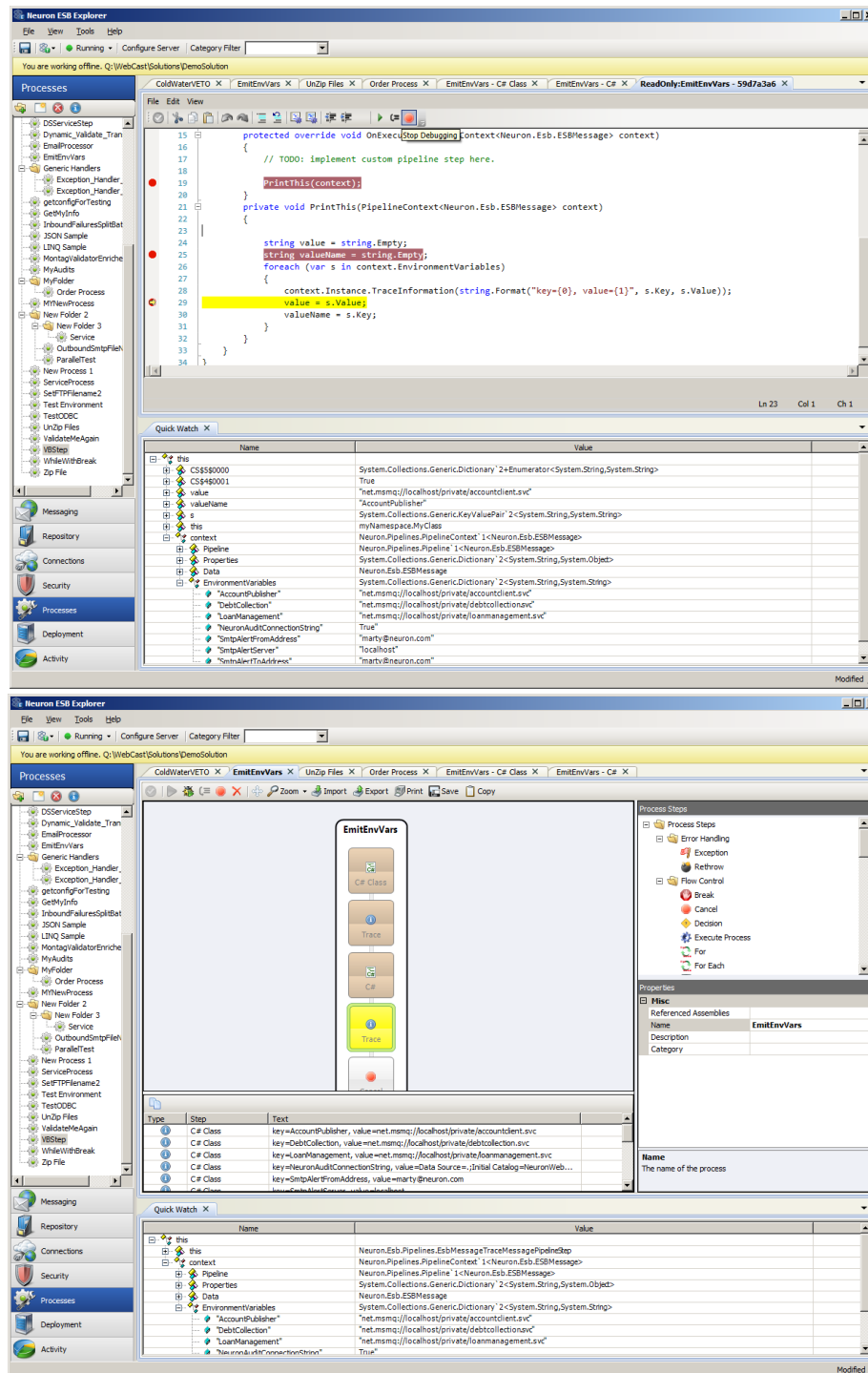
Disabling Process Steps has been enabled for both design time and runtime environments. If a user disables a Process Step, that Process Step will be stepped over and not executed, either when testing in the Process Designer or at runtime.

Within a Business Process, disabling a Process Step is as easy as selecting “Disable” from the context menu. Once a Process Step is disabled, the Process Step will change to a dark grey color. To retain the disabled condition at runtime, the Business Process must be saved.

Quick Watch Window

Neuron ESB 3.1 introduces support for a Quick Watch window that provides the ability for developers to view local variable information (much like Microsoft Visual Studio Auto window). If a Breakpoint is hit during the debugging process, the state of the ESB Message can be evaluated in the Quick Watch.

The state of custom variables, Environment Variables and various context properties like context.Data.Properties, context.Properties, context.States, etc. can also be examined as the user steps through each Process Step within a Process.

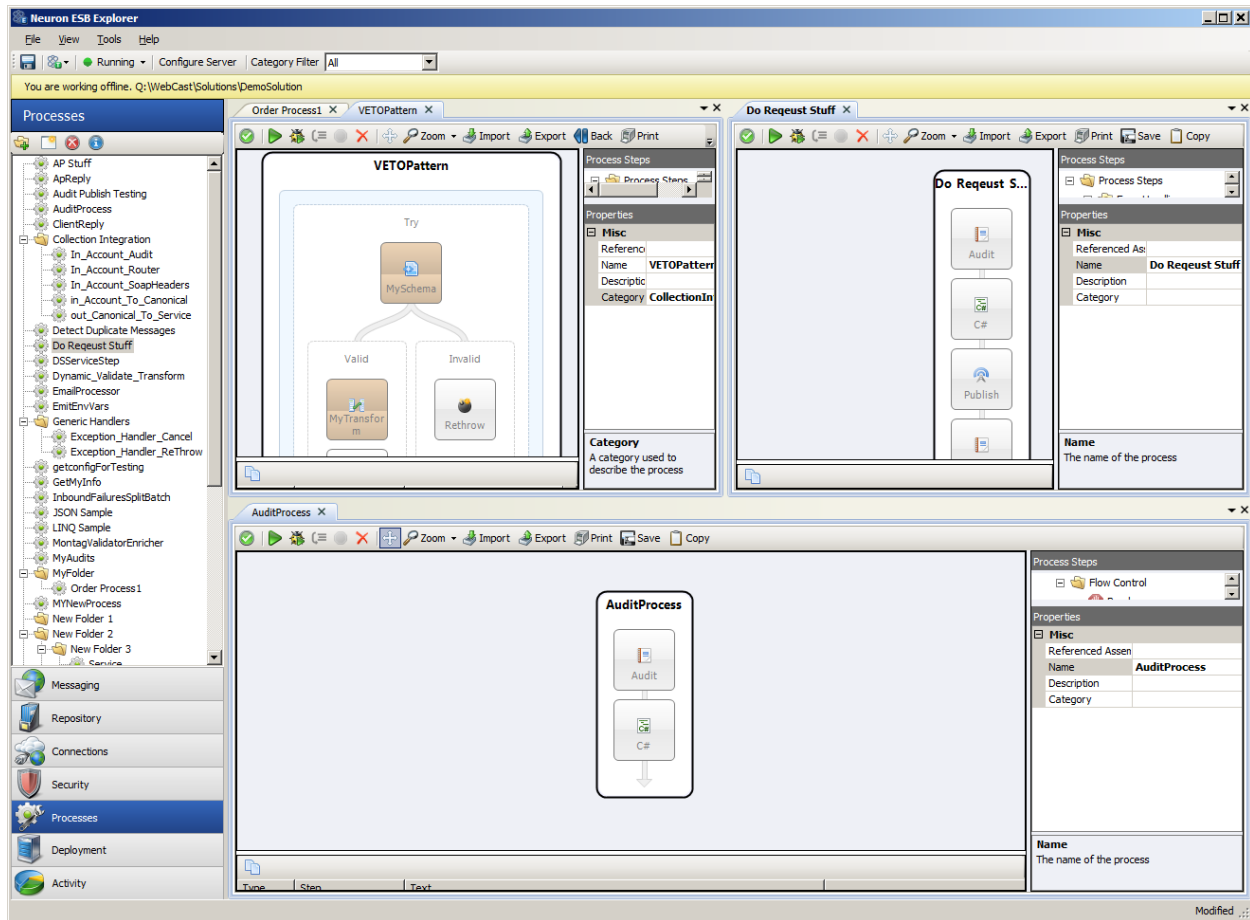


Tabbed View of Business Processes

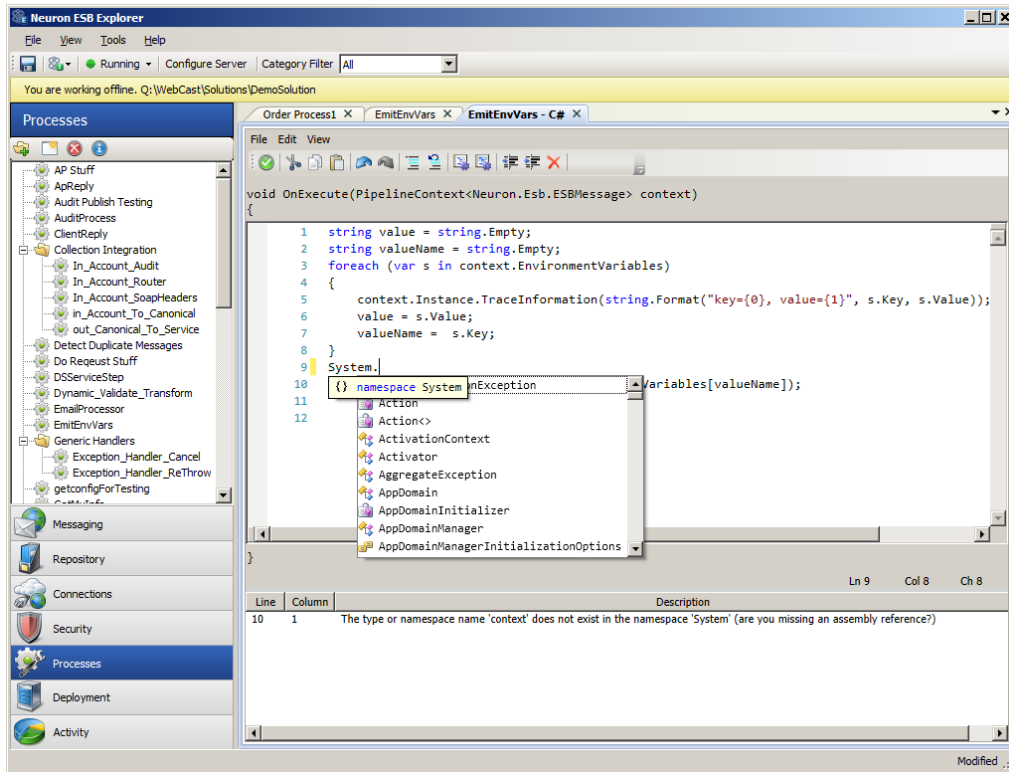
In previous versions of Neuron ESB only one Business Process could be opened within the Process Designer at time in order to be worked on. If a user needed to copy elements of one process over to another, the first process would have to be closed before the second process could be opened and viewed.

Neuron ESB 3.1 introduces a new Tabbed Dialog view of Processes within the Process Designer. This allows multiple Processes to be opened at the same time, allowing the user to navigate between them all by simply

selecting their respective tabs. Additionally, the tabs are dockable and detachable, allowing several processes to be arranged within the same window.



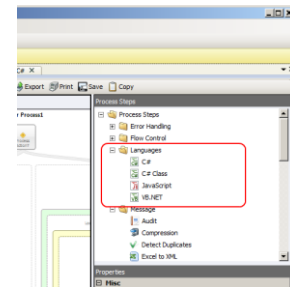
Or processes can be detached and worked on in dedicated windows:



Extended Language Support

Neuron ESB 3.1 introduces support for new languages as well as expanded support for C# and the overall code writing experience. Users can now write VB.NET, JavaScript or full C# classes directly within a Business Process without any need to generate and reference .NET assemblies. All languages are represented by individual process steps under the Languages folder of the Process Step library

A new developer toolbar has been included with all the Code Steps

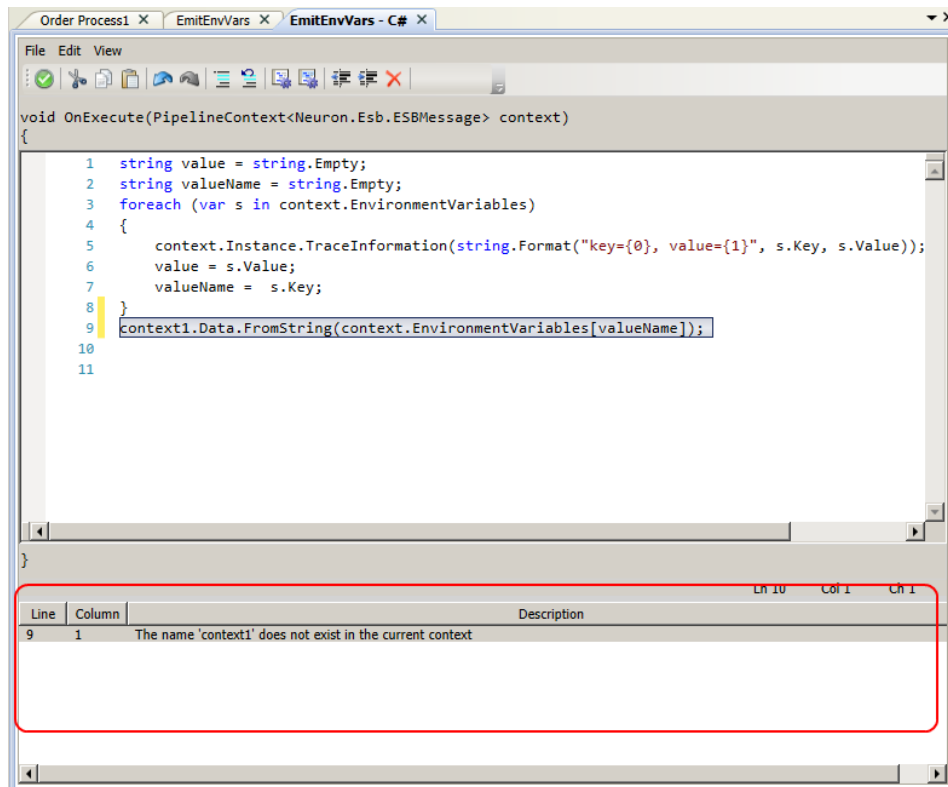


This provides developers access to common tasks such as undo, comment, format, indent, delete all breakpoints, add assembly references, display line numbers, etc.

We have also added support for adding breakpoints on a specific line of C# (within the C# or C# Class Code Steps) and VB.NET Code Step. Developers can add breakpoints on code lines which will cause the Neuron ESB debugger to stop on the specific line when it hits, similar to what developers experience within Microsoft Visual Studio. Once broken into the debugger, a modified Toolbar will appear allowing the user to continue, step over or stop the debugging process. While in Debug mode, developers can view the state of context and process within the Debug Window.

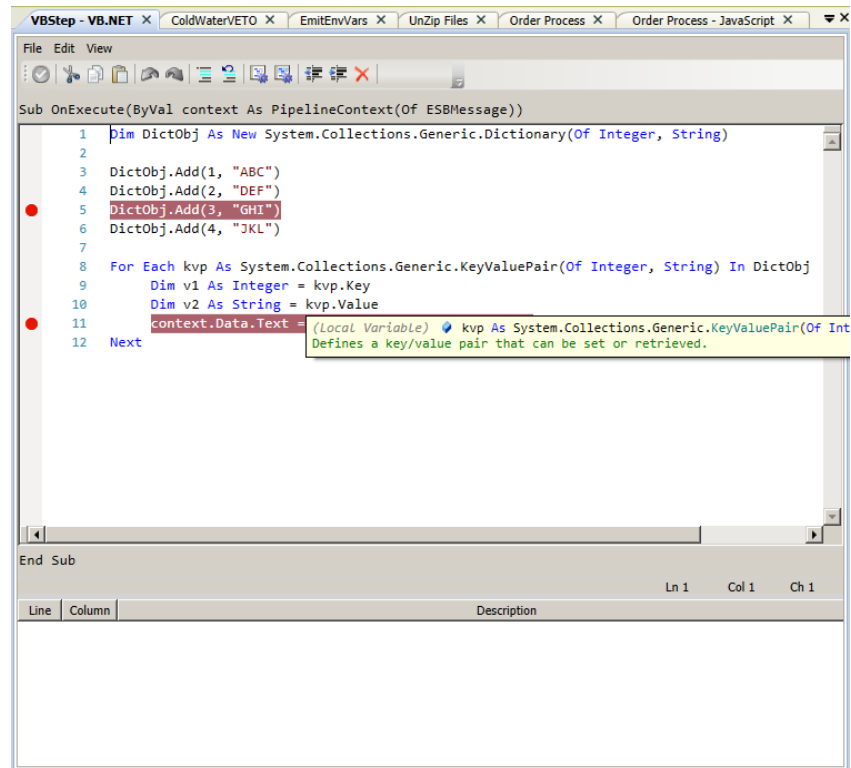


Also, a new compile window has been added to all the Code Steps that allow users to see the design time compile error, line number and column where the error is occurring as shown in the C# Code Editor Image below:



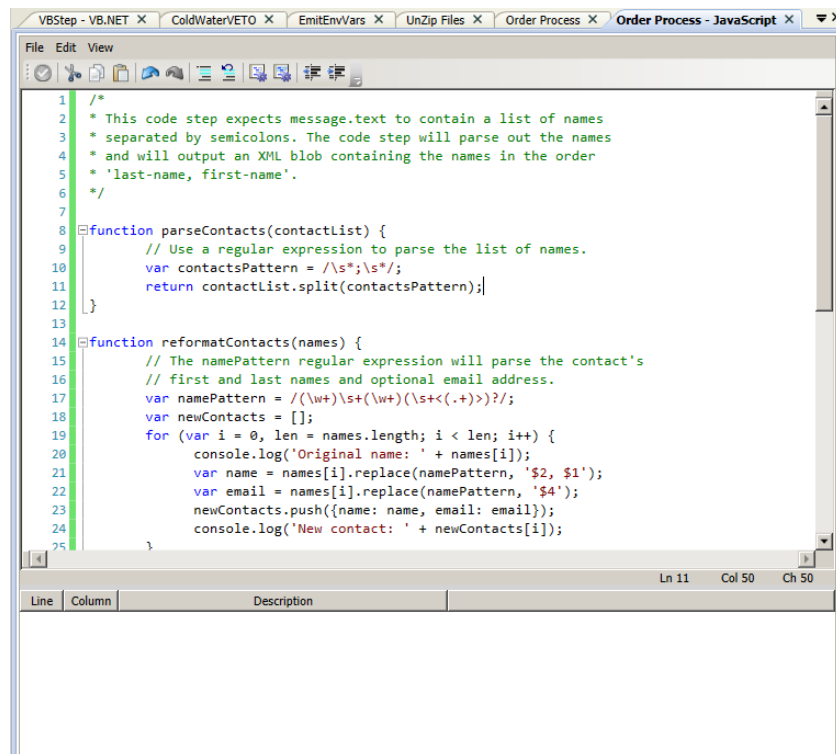
VB.NET

Writing VB.NET code within a Process is as simple as dragging the VB.NET Code Step onto the Process Designer, right click and selecting edit. Users will receive the same experience as if using the C# Code Step:



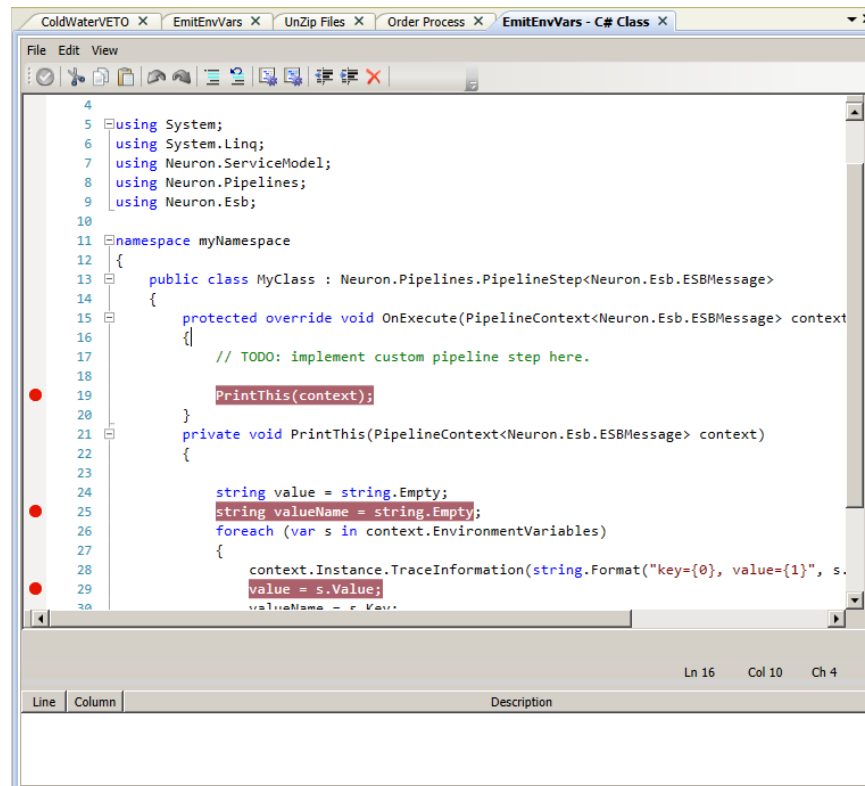
JavaScript

Writing JavaScript directly in a Process is just as easy. Drag a JavaScript Code Step directly onto the surface of the Process Designer, right click and select edit to open the Code Editor:



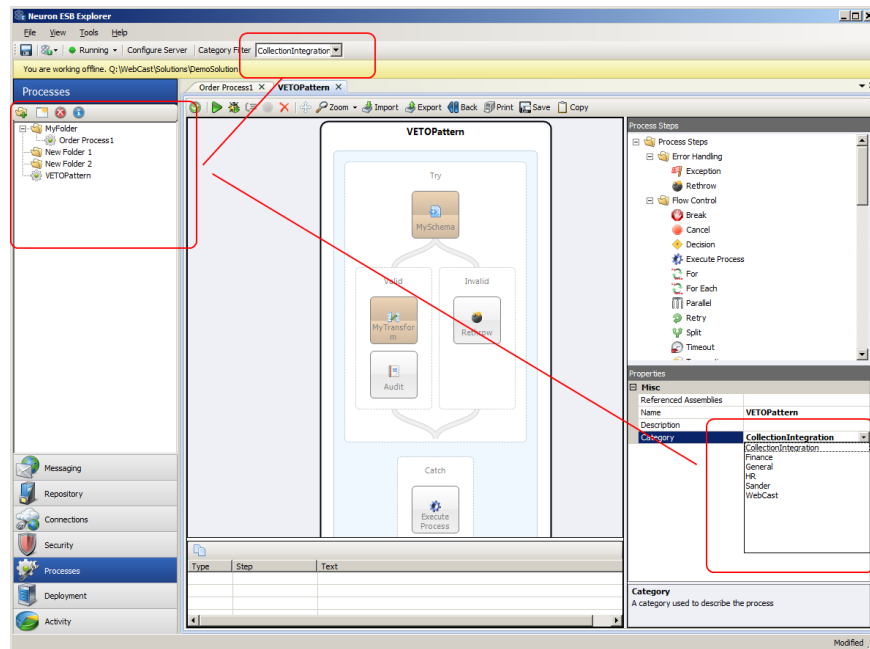
C# Classes

The new C# Class Code Step offers some advantages over the existing C# Code Step that previously shipped with Neuron ESB. The C# Class offers developers a way to more easily organize their code, reducing its complexity. It also allows developers to add using statements and define their own name spaces and new classes. The C# Class Code Step provides an existing template that developers can modify by changing the namespace and class names. From there developers can add additional methods or new classes:



Category and Descriptions

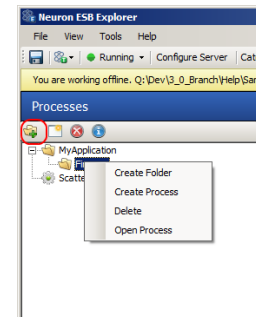
All processes now have both a category and description property. If assigned a category, the category selected in the Category Filter toolbar dropdown of the Neuron ESB Explorer will automatically filter the view of processes in the Process Library.



Business Process Folders

Users can now create folders to organize and store their processes in. An “Add Folder” button has been added to the Process library toolbar. Sub folders can be created by right clicking on an existing folder and selecting “Create Folder” from the context menu. Users can also rename the folder by simply clicking on the name and in the process library for it to become editable.

Once folders are created, existing processes can be dragged and dropped into the folders. Once the folder organization is complete, the solution must be saved.



Business Process Step Folders

In Neuron ESB 3.1 all process steps have been reorganized into a folder hierarchy with the following categories (folders). All new process steps are highlighted in blue italics:

Process Step Folder	Process Step
Error Handling	Exception
	Rethrow
Flow Control	Break
	Cancel
	Decision
	Execute Process
	For
	For Each

	Parallel
	Retry
	Split
	Timeout
	Transaction
	While
Languages	C#
	<i>C# Class</i>
	<i>JavaScript</i>
	<i>VB.NET</i>
Message	Audit
	Compression
	Detect Duplicates
	Excel to XML
	JSON
	Pop
	Push
	Set Property
	Trace
Security	<i>Decrypt XML</i>
	<i>Encrypt XML</i>
	<i>Sign XML</i>
	<i>Verify Signed XML</i>
Services	Publish
	Rules – WF
	Service
	Workflow
Storage	MSMQ
	ODBC
	Store

Table Query

Xml Query

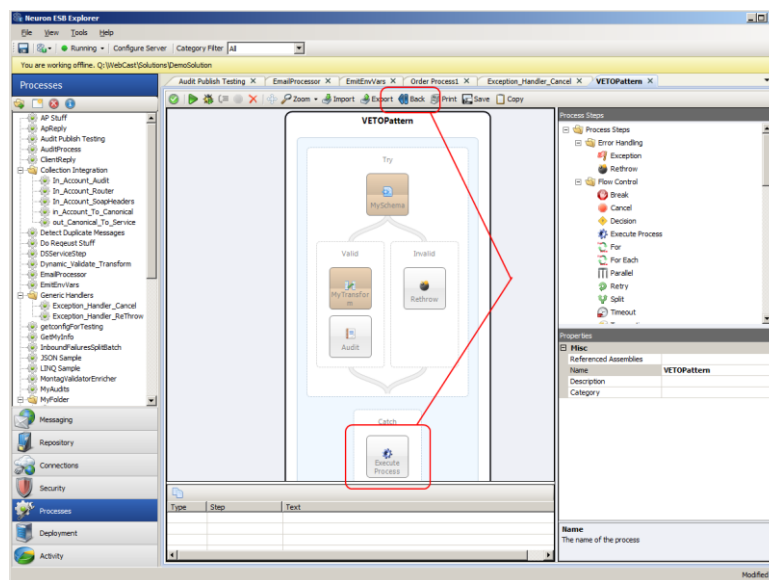
XML

Transform – XSLT

Validate - Schema

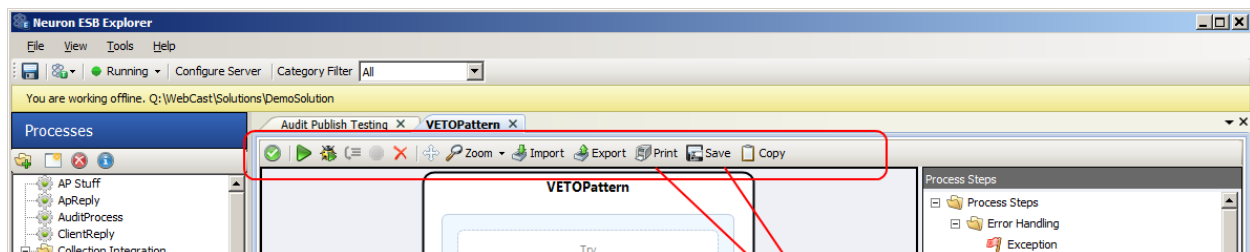
Business Process Navigation

Reusable processes are an important development pattern. Neuron ESB Processes can be reused/called by other existing processes by using the Process Execution Process Step. If a user double clicks on this Process Step, the Process that it calls will be displayed along with a Back button on the toolbar. This provides the developer a way to easily navigate back and forth between parent and child processes.



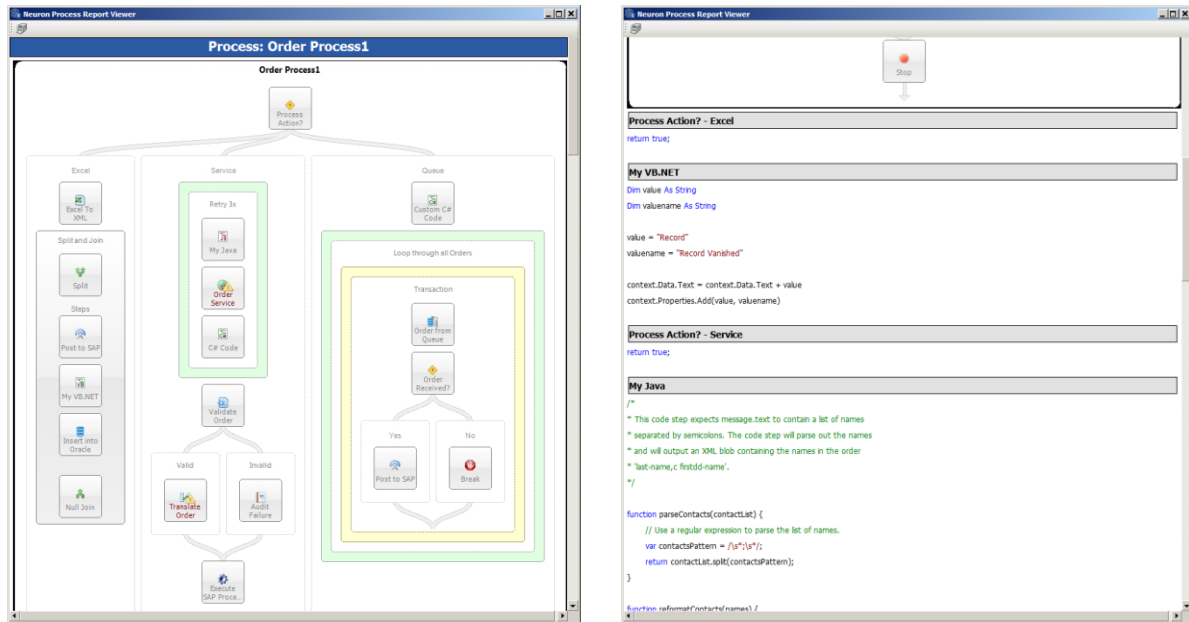
Business Process Printing and Save Image

All processes within the Neuron ESB Process Designer can now be saved as a JPEG or Printed as a report through the new Process Designer's toolbar:



Clicking the "Save" button will prompt the user for a location to save the JPEG image of the process to.

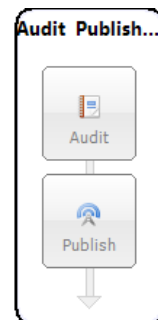
Clicking the "Print" button will display a Print Preview form with an image of the process along with any custom code that any Process Step may contain. All custom code is separated in the report by the name of the Process step that contains it and is formatted with Rich Text.

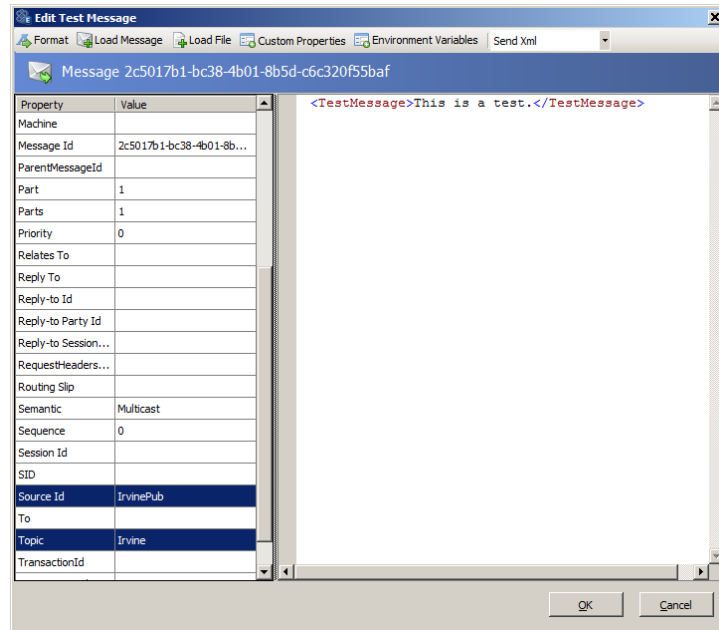


Business Process Steps

Extended Design Time Testing Support

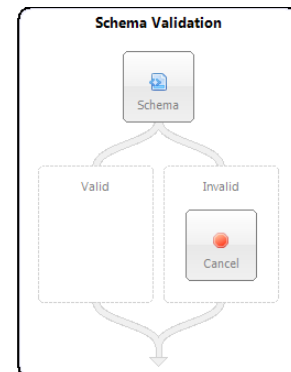
In Previous versions of Neuron ESB, almost all Process Steps could be tested directly in the Process Designer with the exception of the Audit and Publish Steps. In Neuron ESB 3.1, the Audit and Publish process steps have been modified to support design time debugging within the Neuron ESB Business Process Designer. The only requirement is that the solution which is to be tested must be hosted and running in a Neuron ESB runtime on the local machine. The Neuron ESB Explorer can be opened in either online or offline mode for this to work. Additionally, when submitting a message to test, a valid SourceId and Topic must be provided in the "Edit Test Message" dialog. These are used to create a connection to the live Neuron ESB runtime service hosting the solution:





Extended Xml Schema Validation Process Step

In previous versions of Neuron ESB the “Validate-Schema” process step only supported file based locations for the “schemaLocation” attribute for schemas that contained import or include references. For example, when a WSDL is downloaded, the schemas included in that WSDL will often import from a different schema, using a “schemaLocation” attribute to direct the schema parser on where to find it. Neuron ESB previously only supported a file-based location, such as this:



```
<xsd:schema xmlns:tns="urn:partner.soap.sforce.com"
xmlns:ens="urn:subject.partner.soap.sforce.com" xmlns="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" targetNamespace="urn:partner.soap.sforce.com"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:import schemaLocation="C:\SchemaS\Account_sObject"
namespace="urn:subject.partner.soap.sforce.com" />
```

At runtime, if there was a Validate-Schema process step, the import (or include) would be resolved to the local file system.

In Neuron ESB 3.1 support has been added to resolve schema locations to the ESB Repository, like this:

```
<xsd:schema xmlns:tns="urn:partner.soap.sforce.com"
xmlns:ens="urn:subject.partner.soap.sforce.com" xmlns="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" targetNamespace="urn:partner.soap.sforce.com"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:import schemaLocation="esb:Account_sObject"
namespace="urn:subject.partner.soap.sforce.com" />
```

At runtime, if the schemaLocation attribute starts with esb: Neuron ESB will look for the schema in the configuration's repository. The name following the esb: is the name of the schema in the repository.

Exception Pattern built into Audit Process Step

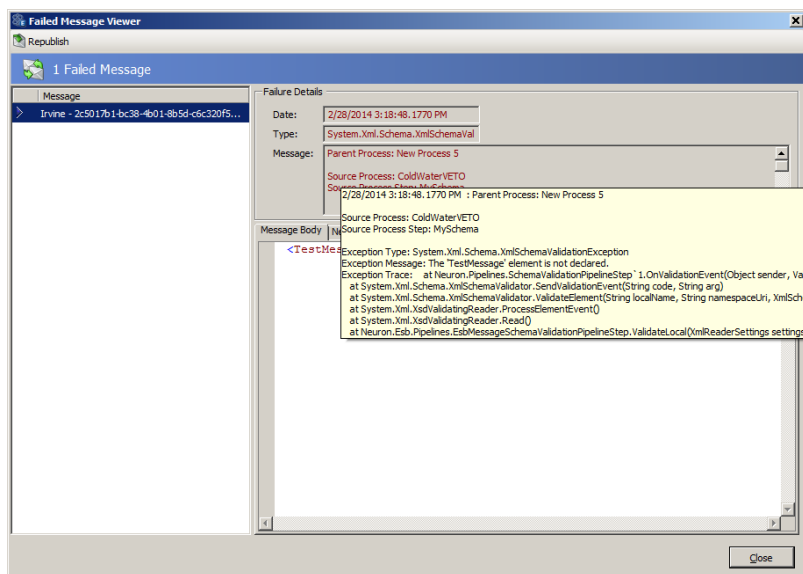
In previous versions of Neuron ESB, users were encouraged to adopt a reusable exception management pattern to employ in all of their processes. Generally this would entail creating a generic process that would handle and audit the exception and that process could be called from the Catch block of any other process using the Process Execution process step. Part of that pattern included the enrichment of the original exception using a C# Code Process step similar to this:

```
PipelineException parentEx = (PipelineException)context.Properties["PipelineException"];

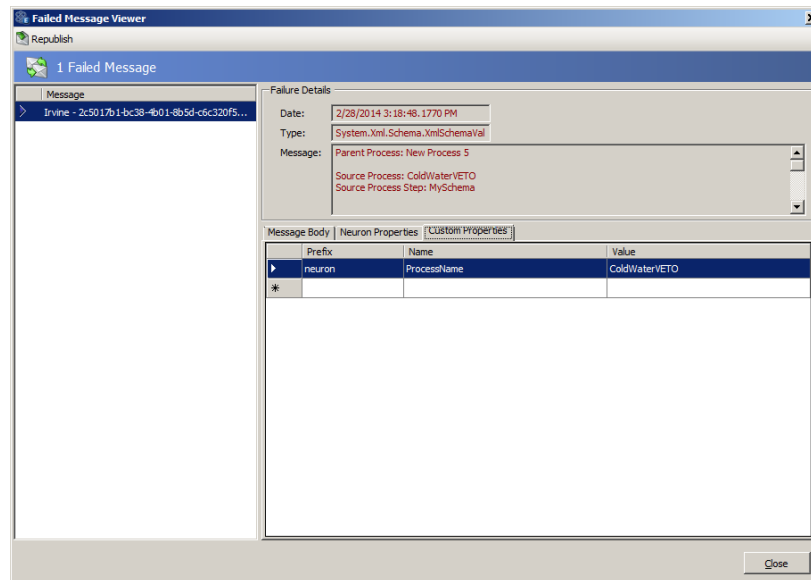
string newExceptionMessage = String.Format(System.Globalization.CultureInfo.InvariantCulture,
    "\r\n\r\nProcess Error Message: {0} \r\n\r\nSource Process: {1}\r\nSource Process Step: {2}\r\nSource Error Message: {3}\r\n\r\n",
    parentEx.Message,
    parentEx.Pipeline,
    parentEx.PipelineStep,
    parentEx.InnerException.Message);

context.Properties["CurrentException"] = new System.Exception(newExceptionMessage);
```

With Neuron ESB 3.1 this is no longer necessary. When a failure occurs and the message is audited using the Audit process step, the parent process name, source process name (where the error occurred) and the process step name that generated the exception are now captured. Also, the appropriate inner exception is retrieved if a child process generated the exception. All of this information is now recorded in the Failed Message Report and can be viewed in the Message Viewer.



The source process name (where the error occurred is also captured within the database as a custom property when using the Audit Process Step in Failure mode (action property)).



When using the Audit Process step in either Receive or Send mode (action property), the property will contain the name of the current process the Audit Process step is executing in.

Simplifying Pop/Push Process Steps

The Pop and Push Process Steps allow users to simplify the saving and restoration of a message at runtime within a Business Process. In many occasions, it may be desirable to save the original incoming message (Push) before it is transformed, enriched or otherwise changed as part of the process and when finished, restore (Pop) the message to its original state.

In previous versions of Neuron ESB, developers would be required to use a C# Code Step directly following the Push Process Step. This Code Step would have one line of code i.e.:

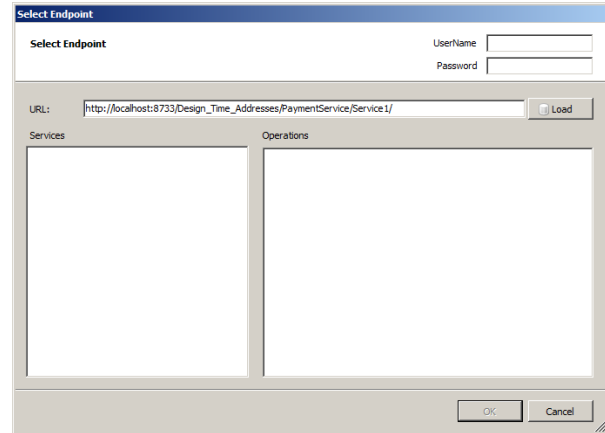
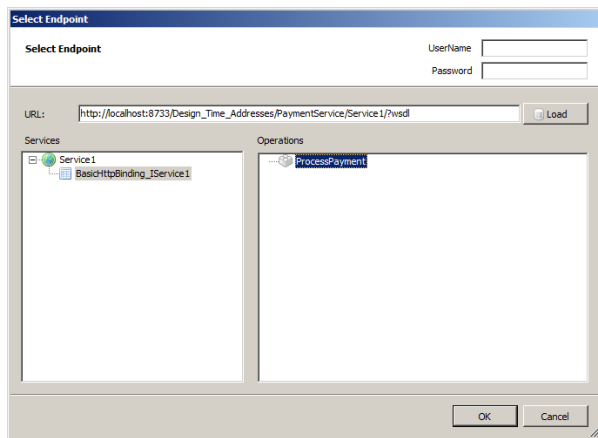
```
context.Data = context.Data.Clone(true);
```

In Neuron ESB 3.1 this is no longer required.

Extending the Service Process Step

In Neuron ESB 3.1 the Service Process Step has been extended to support WSDL extensions as well as maintaining its current support for WCF "Metadata Exchange Endpoints". In previous versions of Neuron ESB, if the either the Metadata Url property was used or, a user selected to import the metadata by selecting "Select Endpoint..." from the context menu, only "Metadata Exchange Endpoints" could be used.

In Neuron ESB 3.1, WSDL support has been added, allowing non WCF services to be easily accessed through use of WSDL at runtime. If using the Select Endpoint metadata wizard (displayed below), if an endpoint is entered without either a WSDL or Metadata Exchange Endpoint, WSDL will be assumed and that extension will be automatically entered for the user:



Also, if a WSDL or Metadata Exchange Endpoint is not found, an error will be displayed to the user. In previous versions of Neuron ESB, the wizard would silently fail, displaying no error at all.

Lastly, the Service Process Step can still be manually configured to call any WCF or NON WCF service endpoint using its associated property grid. This allows service calls to be made without have to retrieve and parse the Metadata for each call.

New Timeout Process Step

The Timeout Process Step executes the contained Process Steps on a background thread. To succeed, the contained Process Steps needs to complete successfully before the timeout period expires, otherwise a timeout exception will be raised on the Business Process.

For example, if the Timeout Process Step is configured for 5 seconds, the contained steps must complete their execution before the 5 seconds expires, otherwise a timeout exception will be raised and the Business Process may terminate. Even though the contained Process Steps are executed on a background thread, the Timeout process step executes synchronously. The Business Process will not continue until either the contained Process Steps complete or a timeout occurs.

The Timeout Process Step is compatible with the use of transactions, such as transactional MSMQ topics. If a transaction is active when the Business Process runs, the contained Process Steps will execute within the context of the transaction that the Business Process is participating in. Failure of the contained Process Steps will cause the transaction to roll back and the message will be returned to the party's message queue.

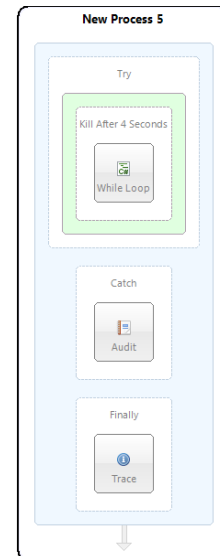
Although this cannot generally be used to abort a poorly constructed looping scenario, a cancellation token has been built into the context object which can trigger the termination of the processing occurring in the background thread. This will only happen between Process Steps.

For example, the Timeout Process Step would not terminate the following constructed loop if used in a C# Code Process Step:

```
while (true)
{
    // Do something here
    System.Threading.Thread.Sleep(1000);
    context.Instance.TraceInformation("hi");
}
```

However, if the loop were reconstructed to use the Cancellation Token, the loop (and the background processing thread) will be terminated directly by the Timeout Process Step by using the `IsCancellationRequested` property of the `CancellationToken` object of the context object.

```
while (!context.CancellationToken.IsCancellationRequested)
{
    // Do something here
    System.Threading.Thread.Sleep(1000);
    context.Instance.TraceInformation("hi");
}
```



New Decrypt/Encrypt Message Process Steps

The Encrypt Message and Decrypt Message steps employ the XML Encryption standard to protect XML messages being sent between parties. The Encrypt Message and Decrypt Message process steps employ the AES encryption standard based on the Rijndael encryption algorithm to protect sensitive data from being exposed to unauthorized parties.

The Encrypt Message and Decrypt Message process steps will look up the encryption key from the Neuron ESB configuration. Developers may also specify a custom initialization vector that will be used to enhance the security of the encrypted data. Using the Encrypt Message step, it is possible to either encrypt the entire XML message, or to encrypt a fragment of the message specifying using an XPath expression.

New Sign/Verify Message Process Steps

The Sign Message process step uses the RSA algorithm to attach a cryptographic signature to an XML message. The Verify Message process step will use the same RSA algorithm to verify that a signed message is valid. Both the Sign Message and Verify Message process steps use X.509v3 certificates stored in the Windows certificate store and configured in the ESB configuration to sign or verify the XML message.

The Sign Message process step requires the presence of a private key attached to the certificate in the certificate store. Using the private key, the Sign Message process step will use the XML Signature standard to calculate the cryptographically-secure signature for the message and will add the signature to the XML body of the message.

The Verify Message process step will use the public key associated with a configured X.509v3 certificate to verify that the signature attached to an XML message is valid and was signed by the private key associated with the public key that is used to verify the message's signature.

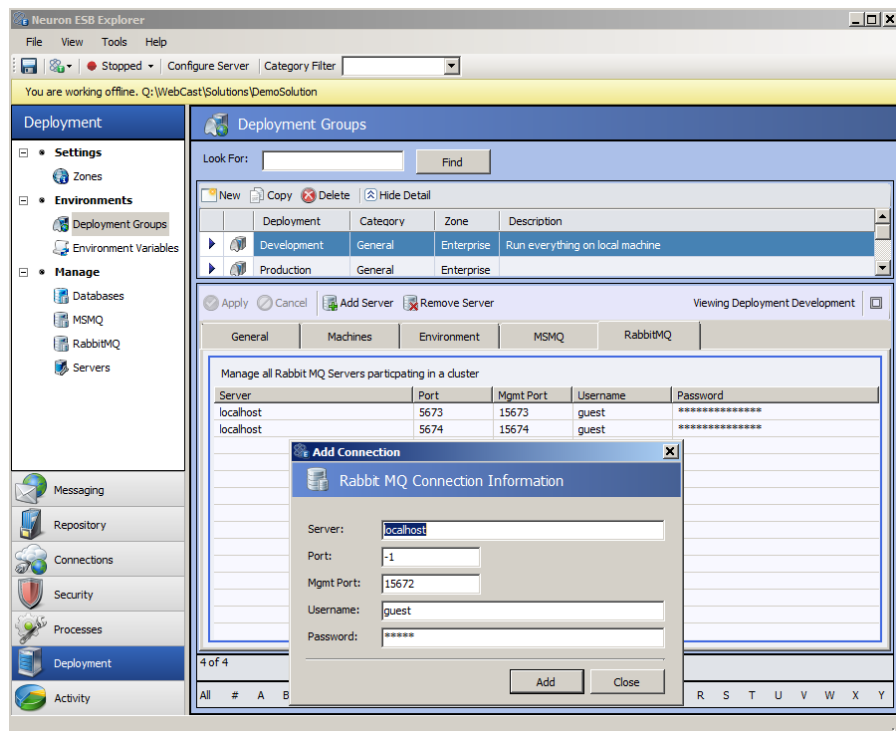
Connectivity

Topic Transports

RabbitMQ Clustering for Topics

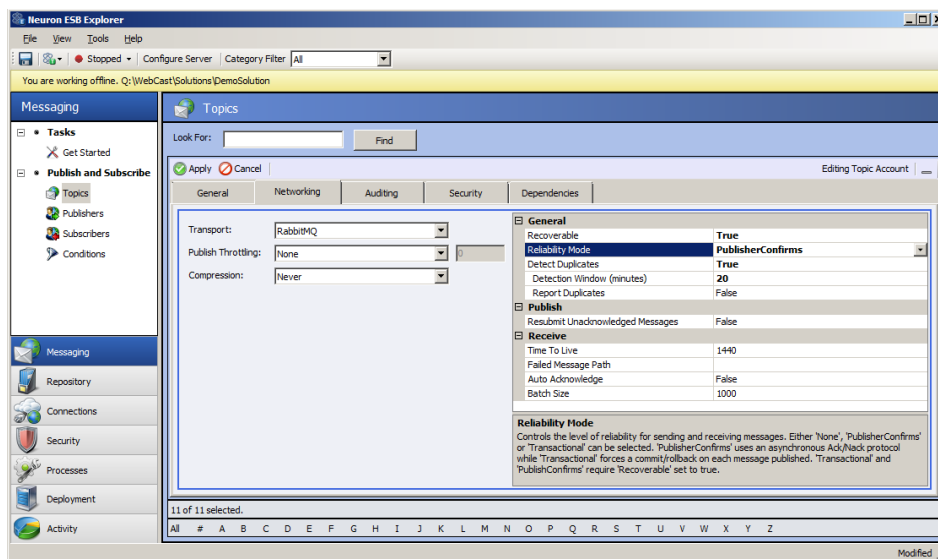
In Neuron ESB, Topics can be configured to use any number of transports including TCP, Peer, Named Pipes, MSMQ and RabbitMQ. MSMQ or RabbitMQ would typically be used if durable/persisted or transacted messaging was required end to end.

In Neuron ESB 3.1, RabbitMQ support has been substantially upgraded to support Clustered/Mirrored RabbitMQ environments.



In clustered RabbitMQ environments, mirroring of all Neuron ESB infrastructure queues should be configured through the RabbitMQ administration portal. Once this is done, each member of the cluster should be entered in the RabbitMQ tab of the Neuron ESB Explorer's Deployment Groups section as displayed in the image above.

New property options on the Topic's transport network property page determine the behavior as well as level of reliability that will be provided if either a member of the cluster or all members of a cluster fail. These property options can also guarantee once only delivery of the message.



The RabbitMQ transport for a Topic can be configured to run either in PublishConfirm (asynchronous type of transaction support based on acks/nacks), Transactional or no Transaction mode. It can also be configured to detect and discard any duplicate messages as well as resubmit all unacknowledged messages once connectivity is restored to the cluster if one or all machines in the Rabbit MQ cluster fail.

For example, consider the case where there are 2 RabbitMQ nodes configured in a mirrored cluster. If messages were being published and received through Neuron ESB while one of the nodes went down, the underlying Neuron ESB Runtime would automatically catch the failure condition, report it, and seamlessly roll over and send/receive messages to the remaining node in the RabbitMQ cluster. The original failed message would be resent to the surviving node and (if using PublishConfirms), any unacknowledged messages will be republished to the surviving node as well to ensure no message loss. Any messages that fail to be redelivered would be written to the Neuron ESB Audit database where they can later be resubmitted.

If both nodes go down, once the primary node is restored, all unacknowledged messages (if using PublishConfirms), will be resent to the primary node. Any failures will be written to the Neuron ESB Audit database where they can later be resubmitted.

Miscellaneous

FIX – RabbitMQ Dead Letter Monitoring – Neuron ESB automatically monitors a dead letter queue for all of its infrastructure queues. If the Neuron ESB dead letter processor fails on startup, any endpoint that successfully starts up after the fact will restart the dead letter processor.

FIX - RabbitMQ Endpoint Health statistics - The endpoint health statistics (errors and warnings) were never in sync with the statistics in the base child class, hence even if restarted and the statistics were cleared, they would be repopulated by the old number and always show red errors. This would be intermittent. This has been fixed.

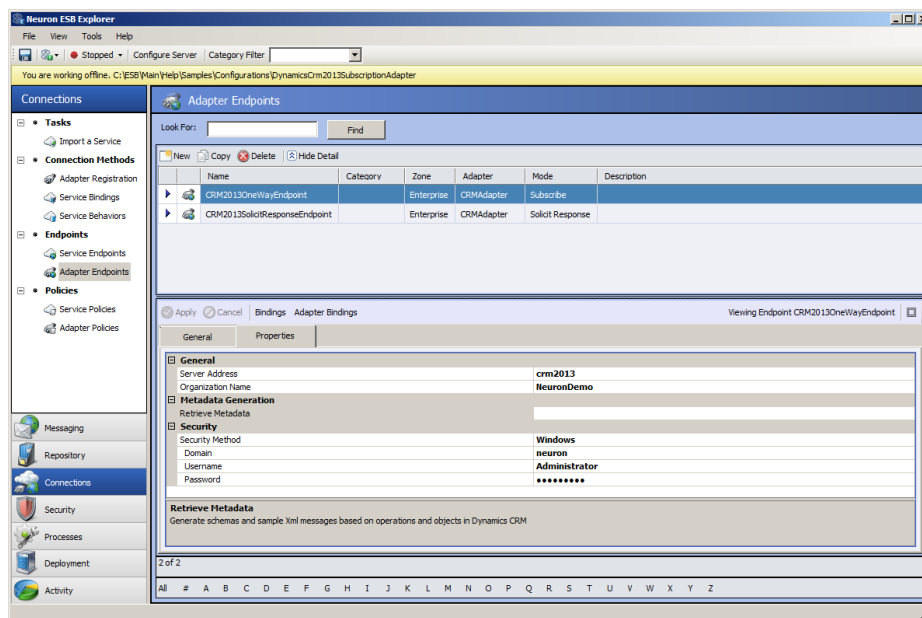
Additionally, if a RabbitMQ based topic failed to start up when the Neuron ESB runtime was started, the timer would never be initialized so there was nothing watching to try to refresh the statistics reported in Endpoint Health within the Neuron ESB Explorer in case where the RabbitMQ topic would later be restarted.

Adapters

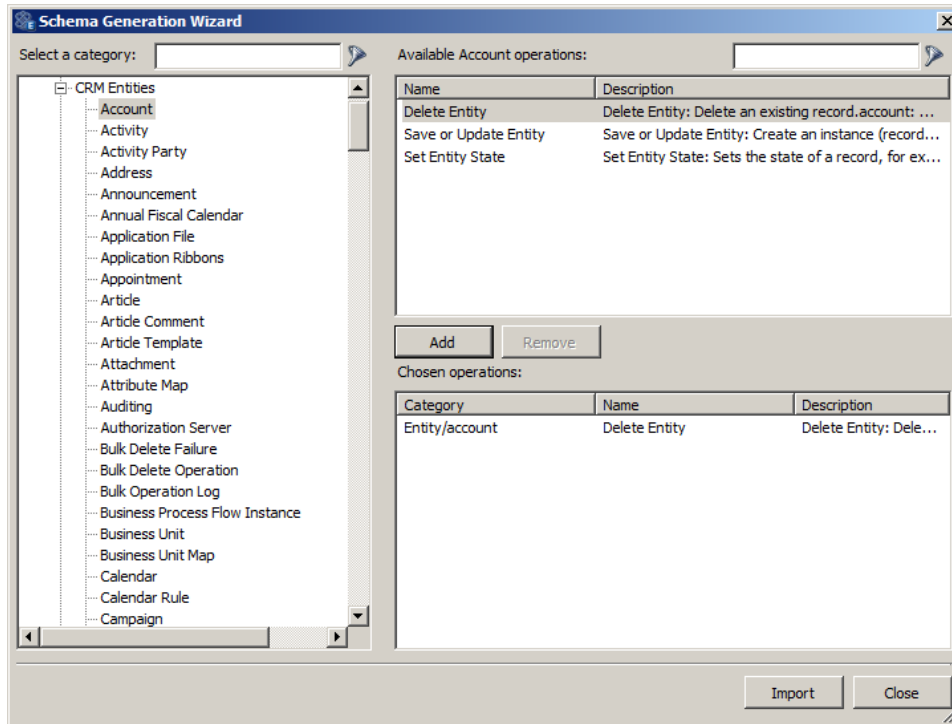
Microsoft Dynamics CRM 2013 Adapter

This is a new adapter included in the 3.1 release. This adapter supports both one way subscribe as two-way, solicit response mode. Users can either send updates or inserts into Dynamics CRM 2013, or make FetchXml Query requests against Dynamics CRM 2013. This adapter also supports meta-data harvesting. Users can browse the operations exposed by Dynamics CRM2013 and elect to generate Xml Schemas and sample Xml Messages for the various operations.

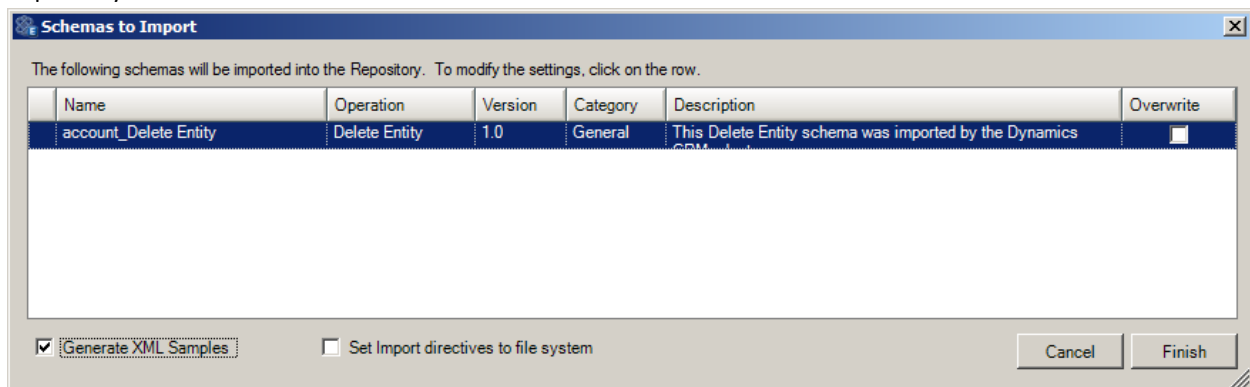
This adapter supports multiple security options, standard Windows credentials, Live Id credentials and Federated.



The Meta data generation wizard can be accessed through the “Retrieve Metadata” property of the adapter endpoint.



Operations can be selected by clicking the Add button. After operations are selected, the Import button will display the selected operations, allowing users to edit their properties and to optionally choose to generate sample Xml messages. The Finish button will store all the generated Xml Schemas and messages into the Neuron Explorer's Repository.

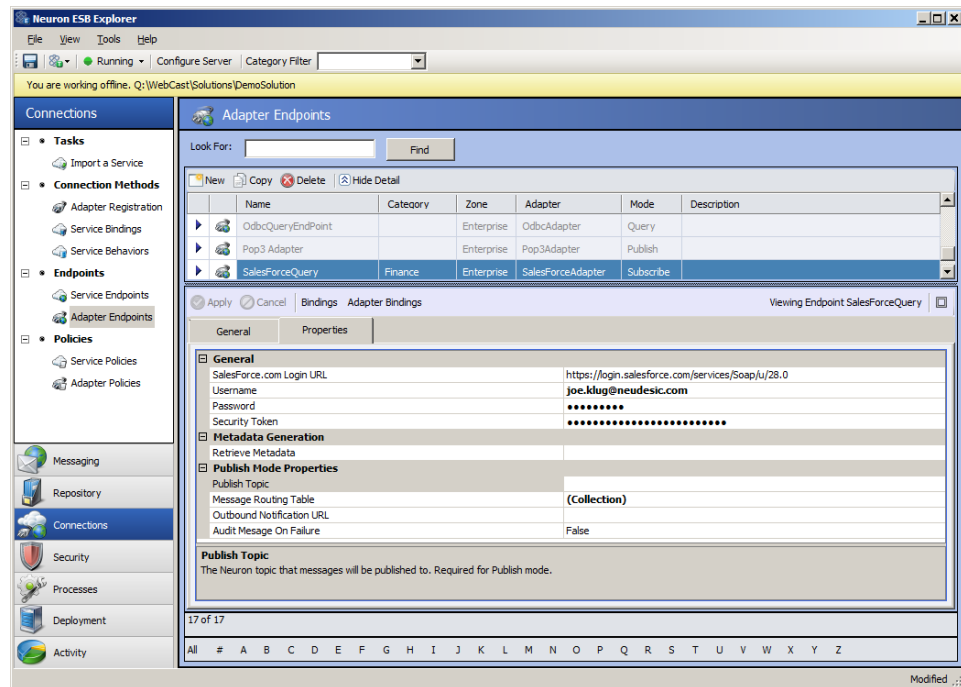


Neuron ESB 3.1 ships a sample demonstrating how to use the new adapter. This can be found in the Neuron Explorer's Sample Browser.

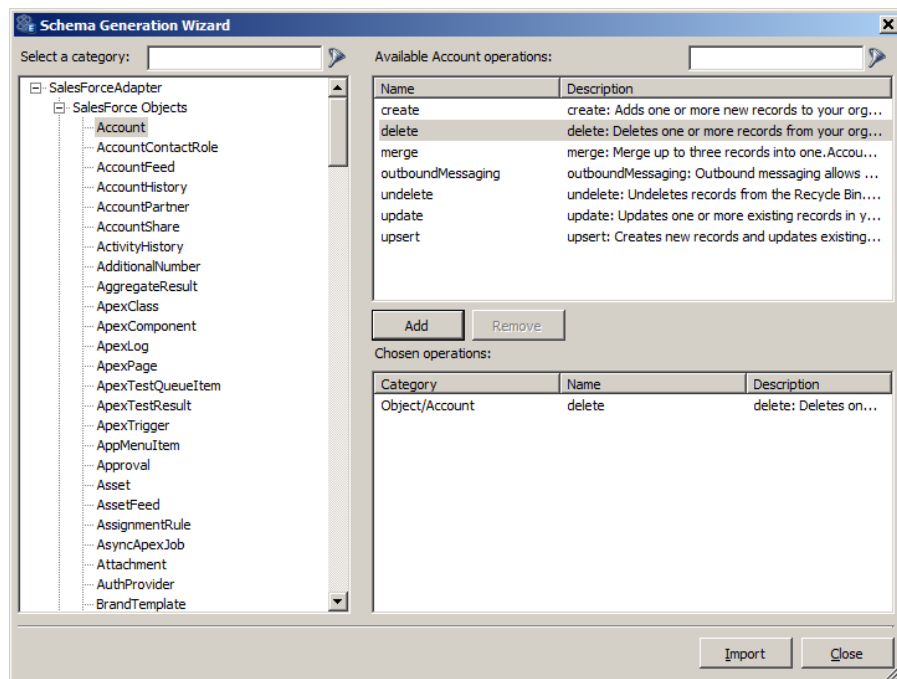
SalesForce.com Adapter

This is a new adapter included in the 3.1 release. This adapter supports the publishing of outbound notifications directly from SalesForce.com to Neuron ESB as well as two-way, solicit response mode. Users can either send updates or inserts into SalesForce.com, or make Query requests against SalesForce.com. This adapter also supports meta-data harvesting. Users can browse the operations exposed by SalesForce.com and elect to generate Xml Schemas and sample Xml Messages for the various operations.

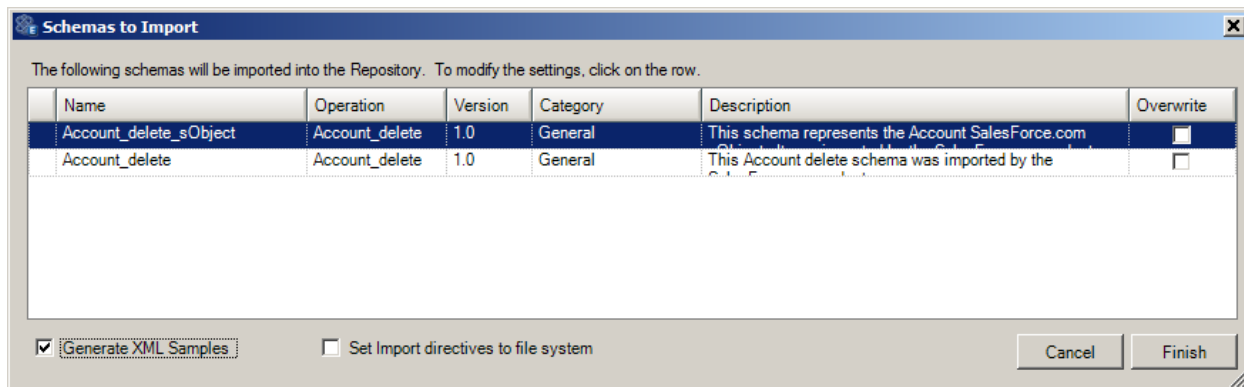
When using the Salesforce.com adapter in Publish mode, the user must supply a URL that Neuron will host. All outbound notifications that are received can then be mapped to specific topics using the “Message Routing Table” property of the adapter endpoint.



The Meta data generation wizard can be accessed through the “Retrieve Metadata” property of the adapter endpoint.



Operations can be selected by clicking the Add button. After operations are selected, the Import button will display the selected operations, allowing users to edit their properties and to optionally choose to generate sample Xml messages. The Finish button will store all the generated Xml Schemas and messages into the Neuron Explorer's Repository.



Extended Apache Active MQ Adapter

Several modifications were made to this adapter to achieve higher throughput in environments with limited IO capabilities. We added support for a configurable number of ActiveMQ Consumers to publish messages to Neuron ESB and a configurable number of producers to send messages to ActiveMQ from Neuron ESB. Support was also added for different acknowledgement types when reading messages from ActiveMQ – Individual Acknowledge, client acknowledge, transactional, auto acknowledge. For sending messages to ActiveMQ, we added support for both synchronous and asynchronous sends. Additionally, the client libraries were upgraded. We now use:

- Apache.NMS.ActiveMQ – 1.6.2
- Apache.NMS – 1.6.0

Updated Neuron ESB Adapter Framework

This has been updated and is now included in the Neuron Explorer's Sample Browser. This is a sample project template that demonstrates how to build Neuron ESB Adapters using any message pattern such as request/response, solicit/response, one way publish, one way subscribe. Many helper functions have been pushed into main Neuron ESB assemblies making adapters much easier to develop.

Miscellaneous

FIX – POP3, Microsoft Exchange, Azure Service Bus, and FTP/SFTP/FTPS Adapters – At runtime, these adapters may use the Neuron ESB Audit Database for specific functions, if enabled. We found that under certain conditions the adapters would erroneously detect that there was no Neuron ESB Audit database configured for the environment. Hence those specific features, if enabled, would remain effectively disabled. This has been corrected.

FIX – Microsoft Exchange Adapter - System.ArgumentException i.e. "The value must be greater than 0" would be thrown if a slash wasn't used with the folder name.

Service Endpoints

Miscellaneous

NEW – Client Credentials, Service Credentials and Access Controls Lists are all populated by the Credentials created and maintained in the Security section of the Neuron ESB Explorer. If these were set in previous versions of Neuron ESB, they will have to be reconfigured to use the Credential store.

NEW - Windows authentication is now supported when using the REST binding with the Transport:Windows security setting.

FIX - Fixed service endpoints so when policy retries are defined, they show up as warnings similar to adapter endpoint retries when policies are used. Additionally, warnings and errors are incremented for the WMI performance counters for adapter endpoints.

Performance

Neuron Runtime

Asynchronous Startup and Shutdown

In previous releases, on start up the Neuron ESB runtime would start up each Service and Adapter endpoint, one at a time. This could result in both significant startup and shutdown times if there were many endpoints defined in a solution. This has been changed in Neuron ESB 3.1 so that all endpoints are started up and shutdown asynchronously.

Neuron Installer

Neuron ESB WMI Performance Counters

The creation of these has now been moved to the Setup.exe installer. Previously, these were created each time the Neuron ESB Runtime was started. This feature is represented on the Feature Install page of the installation wizard by the “ESB Service Management Objects” and is disabled (unchecked) by default.



Neuron 3.0.3.331

MSMQ Topics

FIX – Msmq based topics can be configured for automatic dead letter and poison queue monitoring. When these options are configured, if any messages are moved to either queue, Neuron ESB will automatically read the messages from the queues in a transaction and either write them to the Neuron ESB Audit Database or to the file system, depending on the user provided configuration settings. However, if a failure occurred, a final attempt would be made to store the message on disk. If that failed, for whatever reason, the error information would be written to the Neuron ESB Event log, but the message itself would be discarded. With this fix, now an error will be

thrown, causing the transaction to roll back and the message restored to the underlying queue. If Neuron ESB enters this state it's critical that the user corrects the error condition immediately. The only known conditions that could cause the rollback to occur is if 1.) There was no disk space left on the Neuron Server, or the path provided by the user or 2.) The Neuron ESB runtime service lacks permissions to write to the disk location.

Neuron 3.0.3.327

Business Processes

FIX - Modified how we report Code Process Step Compiled Errors. Previously, if a compile error occurred in a Code Step at runtime, the exception in recorded in the logs would be:

Exception Type: Neuron.Scripting.CodeCompilationException

Exception Message: 1 compilation error(s) encountered in pipeline source code.

The actual compile error that occurred is now appended to the message.

Adapters

NEW- Added credential support to the Dynamics CRM4/2011 subscription adapter. Users can now enter the Domain, Username and Password directly as properties of the adapter. Previously the adapter endpoint would need to be configured using the "Run As" feature of the adapter runtime which required users to configure Kerberos appropriately on the Neuron ESB Server.

TCP and Named Pipes Topics

FIX - In previous releases, Parties communicating with TCP/Named Pipes Topic exchanged heartbeat messages. If heartbeat messages were not received by the Topic's Publishing Service within a specific interval, the Publishing Service would disconnect the Party from receiving published messages. Under heavy load conditions, background threads may not be available in the .NET thread pool to service the heartbeat protocol for the Party. This could result in Parties being forcefully disconnected by the Publishing Service. If disconnected, the Party would attempt to reconnect to the Publishing Service. Once in this state the Party may either not receive published messages, or may not be able to send pending reply messages back to the Publishing Service. These cases could also manifest themselves as Socket Exceptions. The heartbeat protocol has been changed so that as long as the Party is actively receiving messages the Publishing Service will not disconnect it.

Service Endpoints

FIX – In previous releases the receive timeout property for REST, wshttp, basichttp, nettcp, msmq, ws-fed and named pipes bindings when using Reliable Sessions was set to 60 minutes. This is now set to timespan.maxvalue.

Neuron 3.0.3.316

MSMQ Transports

FIX – MSMQ Queue Creation – When creating the underlying MSMQ Queues for Neuron Parties through either the Neuron Explorer or through the use of the generated Windows Script file queues that had very long names (over 92 characters) would return as not being created. However, these queues would be created, but with improper permissions. Users attempting to use the Neuron Client API to connect to the associated MSMQ based Topic would receive errors messages indicating the format name was wrong; the queue didn't exist or the user lacked the

necessary permissions. This has been corrected to ensure that the queue does not in fact get created and that an error is displayed within the MSMQ Management screen of the Neuron ESB Explorer indicating that the path name exceeds the MSMQ path size limit.

Neuron 3.0.3.312

Neuron Explorer

FIX – Unhandled exception could occur if connectivity to internet was interrupted while trying to retrieve RSS feed update from Neuron Product Team Blog

FIX – All Samples have been updated.

Neuron Runtime

FIX – MSMQ Publishing Services - Anytime a configuration was saved regardless if it was a change in an actual MSMQ based Topic, all MSMQ publishing services representing MSMQ based topics would be automatically restarted. This could be perceived as a distinct pause in processing. Also, if transactions were not enabled, message lost was possible.

Neuron 3.0.3.308

Business Processes

FIX – Table Query Process Step – When specifying a TEXT select statement that does not include Field Names and instead includes TSQL functions i.e. “select RTRIM(FirstName) from PhoneBook where ID=5”, the Table Query Process Step will return an error similar to “Name cannot begin with the '>' character, hexadecimal value 0x3E.”. This can has been fixed. However, this can be made to work without this by providing an alias after the function. For example: “select RTRIM(FirstName) **As FirstName** from PhoneBook where ID=5”.

Neuron 3.0.3.307

Service Endpoints

FIX – When a failure occurred on a Service Endpoint (service connector) and Auditing was enabled on Policy, the publisher rather than the subscriber id would be reported as the source.

FIX – When a service endpoint encounters an endpoint not found exception, under some circumstances the underlying exception stream returned from the network stack would be empty. This would result in the error being reported in the Neuron ESB Event log, but only a generic error would be reported in Auditing. This has been corrected.

Neuron 3.0.3.306

Service Endpoints

FIX - When receiving a SOAP request with multiple root elements (unwrapped), the client connector would throw the following exception: Exception: There are multiple root elements. This has been corrected. The Client Connector now supports multiple root elements.

Neuron 3.0.3.305

Service Endpoints

FIX – Service Faults and Request/Response message patterns. In previous releases, if a service fault was returned from the endpoint, the fault would be forwarded back to the client caller. However, the exception that would be reported in the Neuron ESB Event log and Auditing would indicate a generate Application Exception with the message “Message delivery failed”. This has been correct to show a FaultException with the identical fault details and reason code returned to the client.

Neuron 3.0.3.304

Neuron ESB Service

FIX – Login error on startup when using nonstandard Port for Neuron Audit Sql database. When specifying a Port with the server name for the Neuron ESB audit database, the Neuron ESB Runtime would report a login failure on startup. This has been resolved.

Neuron 3.0.3.301

Adapters

FIX – SFTP Adapter – A bug introduced with build 3.0.3.298. When in publish mode, when attempting to delete file which was downloaded, a disconnect error would be reported.

Business Processes

FIX – Parallel Process Step – Accessing context.Properties bag within branches would produce a key already found error. This has been corrected.

FIX – Code Process Step – Existing Code process steps would report a compile error due to a missing namespace/assembly reference introduced in build 3.0.3.282. This could be manually fixed by adding a reference to the System.Data.DataSetExtensions.dll assembly. However, Neuron will automatically update all existing Code Process steps now.

Neuron 3.0.3.298

Adapters

FIX – SFTP Adapter – a Maximum Packet Size property (default 8KB) has been added as a design time property. Some SFTP servers do not return their accepted maximum packet size. Previously, this would result in a forced connection from the server when transferring larger files, resulting in a zero byte file on the SFTP server. Exceed libraries for dependent FTP/FTPS/SFTP Adapters and Compression Process Step have been updated to 5.4.

Neuron 3.0.3.292

Neuron Explorer

FIX – Opening the Event Log from the Server Management form could sometimes cause the appearance of a lockup. This was due to activity being generated on a thread other than the main UI thread.

Neuron 3.0.3.291

Neuron Repository

FIX – XSD Schema contents would be saved incorrectly for newly added XSD Schemas within the repository. Meta data would be saved, but not the schema contents

Neuron 3.0.3.290

Neuron Runtime

NEW – Modified the Neuron ESB runtime startup so that Performance Counters for Neuron ESB Topics and Endpoints would only be created IF they are enabled on the Server tab of the Zone settings screen within the Neuron ESB Explorer. Previously, these were always created regardless of the settings. Under some circumstances the creation of Microsoft Performance Counters can take anywhere from 10 seconds to minutes depending on the current Windows Operating System environment.

Neuron 3.0.3.288

Adapter Endpoints

NEW – Adapter Endpoint Policies – In previous releases if a policy was defined for an adapter endpoint and a failure to send occurred, the error event would not be written to the Neuron ESB Event log into the final retry failed. This has been changed so that every attempt to send will generate a Warning event in the Neuron ESB Event log. The final retry attempt will still log an Error event in the Neuron ESB Event log. If No policy is defined, that failure event will be logged as an Error event in the Neuron ESB Event Log

Service Endpoints

NEW – Service Endpoint Policies – In previous releases, if no policy was defined, the failure event would be logged as a Warning in the Neuron ESB Event log. That failure event is now logged as an Error event in the Neuron ESB Event Log.

Neuron 3.0.3.287

Service Endpoints

NEW – Custom Behaviors – In previous versions an error would be generated by a service endpoint when starting up if a custom endpoint behavior was applied but a behavior of that “Type” already existed on the endpoint. Now if the endpoint behavior “Type” exists, it will be removed and the custom endpoint behavior of the same “Type” will be added to replace it

FIX – The ServicePoint Connection Limit is now being set with the same value applied to the ServicePointManager DefaultConnectionLimit. This value is determined by the DefaultHttpConnectionLimit appSettings configuration within the esbservice.exe.config file. The default value is 100. Previously the ServicePoint Connection Limit was being set to 2.

FIX – Setting the “Certificate Revocation” property of the Service Endpoint to “NoCheck” will now set the ServicePointManager’s CheckCertificateRevocationList property to false. Previously, setting this property only set the behavior of the certificate assigned to the service connector, and did not affect server side behavior

FIX – Setting the “Certificate Validation” property of the Service Endpoint to “None” will now return “true” to all server side certificate validation calls made to the service endpoint. Previously, setting this property only set the behavior of the certificate assigned to the service connector, and did not affect server side behavior

Neuron 3.0.3.284

Service Endpoints

FIX – Query String parameters have been changed to ensure that ‘&’ and other special characters are encoded correctly.

Neuron 3.0.3.282

Neuron Configuration

FIX – Databases – An error would occur if a Neuron Database server was configured with a sql server name with an appended port number (i.e. “server:port”). The database would fail to save in the Neuron Configuration.

Business Processes

FIX – Code Process Step – System.Data.DataSetExtensions and System.Xml.Linq have been added as references/using statements for all Code Process Steps

Neuron 3.0.3.276

Service Endpoints

NEW – The HTTP Connection limit has been set to a default of 100 for all outbound HTTP Server calls. This can be reconfigured within the esbservice.exe.config, located in the appSettings under the DefaultHttpConnectionLimit key.

Neuron 3.0.3.273

Neuron Auditing

FIX – Failed Auditing Properties set incorrectly. When using either the MSMQ or RabbitMQ Dead Letter or Poison Message Queue automatic processors which move these types of messages directly into the Neuron Failed Message database, the Party ID was always being set to the Publisher’s ID, the Source ID was empty and the direction was set to Send. This has been corrected. The Party ID is now set to the Subscriber’s ID, the Source ID is set to the original Publisher’s ID and the direction is now set to Receive. Additionally, when using Service Policy’s the Source ID for resulting message in the Failed Message table would be blank. This now shows the appropriate

Publisher's ID. Adapter Policies would show the Direction as Send. This has also been corrected to indicate Receive as well.

Configuration

FIX – Save As from the Neuron Explorer's file menu would just save the ESB Solution project file. It did not save the entire directory of entities. This has been corrected.

Adapters

FIX – POP3 Adapter – Various fixes and tracing were added to ensure messages received were deleted and to ensure connectivity.

Neuron 3.0.3.267

Adapters

FIX – File Adapter – When subscribing to messages using a file adapter endpoint, if a filename property is not provided either at runtime or design time, the session ID would be used to create the outbound file name. This has been changed to use the Message ID to ensure uniqueness

Topics

FIX – TCP and Named Pipes Topic Transports – When using either of these transports under load, a memory leak could occur due to a deadlock situation with the heartbeats. This has been resolved.

Neuron 3.0.3.265

Topics

FIX – MSMQ Network Transport – Automatic Poison and Dead Letter Queue handlers could fail with large messages

NEW – RabbitMQ Network Transport – Added automatic Dead Letter Queue handler for RabbitMQ topics. Messages in Dead Letter are now moved and stored into the Neuron Failed Message table.

Adapters

FIX – CRM 2011 Workflow Adapter – Setup would fail if discovery service for Dynamics CRM was setup to use https.

Neuron 3.0.3.255

Adapters

FIX – File Adapter – Now supports using a full filename as a filter to poll against. Polling now requires that the FileSpec property have a wildcard character (i.e. “*”) preceding the actual file extension to search for to make this consistent with .NET.

FIX – CRM 2011 Workflow Adapter – Now supports using FetchXml to return master/child relationships using linked entities

Service Endpoints

FIX - The client connector was losing the XSD namespace from the SOAP envelope when reading out the SOAP body.

Processes

FIX – Execute Process Step – “An item with the same key has already been added” exception would occur when using the context.State property bag is initialized and used in a Parent Process and then the message is passed to a child process via the Execute Process step.

Neuron 3.0.3.248

Adapters

FIX – File Adapter – Rename option on subscribing file adapter endpoint would throw an exception reporting an invalid path indicating extra extension separators and names.

Runtime

FIX – Small memory leak was fixed

Topics

FIX – MSMQ based Topics - The Failed Message Path property was not getting recognized if used. Also, its state was not being persisted

Neuron 3.0.3.243

Business Processes

FIX – When saving XSLT documents in Online Mode, under certain conditions the contents of the Xslt document may be removed. This has been fixed.

Neuron 3.0.3.240

Neuron ESB Runtime

FIX – In some cases, unless auditing was enabled at the topic level, some exception conditions within the pub/sub channels would not be accurately reported.

FIX – Neuron ESB Startup time limit is now a configurable property in the ESBSERVICE.exe.config file. Defaults to 10 minutes.

Topics

NEW – MSMQ Transport – If the Transaction Timeout was configured at the Topic level, the default value of 10 minutes was still being used. This has been fixed.

NEW – MSMQ Transport – if Receive Error Handling is in effect (when transactions = true) warnings will be reported if runtime detects a Reject or Move, while exceptions will be reported if Drop or Fault are specified. This deals specifically with Poison Message scenarios.

Neuron Audit Database

FIX – When creating the Neuron Audit database using the Azure SQL Create script, errors would occur indicating missing objects. This has been fixed.

Neuron ESB Explorer

FIX – Conditions dialog did not preserve custom property setting when launching to edit existing condition

FIX – Dependency tab columns were truncated for Databases

FIX – MSMQ Management Viewer – When viewing ESB Messages in an underlying queue for either a Party or Topic, an exception could occur if the message was large.

Business Processes

FIX – Audit Process Step property grid properties are now dynamic and will only show failure type and failure detail properties if action is set to Failure

FIX – Using either the context State or PipelineProperties for cache when called through the Process Execution step within a process would fail.

FIX – When using the Code Process Step in Edit mode, a “System.InvalidOperationException: Collection was modified; enumeration operation may not execute.” Exception could occur.

Neuron 3.0.3.212

Neuron Auditing

FIX – Failed Auditing would fail if the internal response object of a WebException was null.

Business Processes

FIX – Split Process Step – The default Maximum number of threads has been set to 10 from 0.

NEW – Publish Process Step – Request/Reply Timeout setting is now a design time property that defaults to 60 seconds

Neuron Explorer

FIX – Encryption Keys – When storing encryption keys, embedded new line and carriage returns would be stripped out by the serializer.

FIX - Unhandled Exception may occur in Endpoint health when restarting Neuron Runtime service.

MSMQ Topics

NEW - Transaction Timeout for msmq network properties are now dynamic so they only show if Transactions = true

Rabbit MQ Topics

FIX - When using Rabbit MQ Subtopics and the Publish process step with the Semantic set to Request, a no subscriber warning would be generated

Neuron 3.0.3.206

MSMQ Topics

NEW – MSMQ Transactions have been extended to 10 minutes by default

Neuron Runtime

FIX - Blocked out debugging code that reported an error if a reply was received by a party, but the corresponding request could not be found.

Neuron 3.0.3.203

Business Processes

NEW – Split Process Step – When used in Asynchronous mode, the maximum number of threads to use is now exposed as a configurable property of the Process Step

MSMQ Topics

FIX – Dead letter and Poison Queue automatic processing – If the Neuron Audit database was not installed and the automatic poison and dead letter queue processing were enabled and the “Failed Message Path” property of the MSMQ Topic was not set, an error would result in the following error:

An error occurred while attempting to process a dead letter: The process cannot access the file 'C:\Program Files\Neudesic\Neuron ESB v3\<Topic_Party>_DeadLetter_20130920080954.msg' because it is being used by another process.

Neuron 3.0.3.199

Runtime

FIX – Reply Messages received during Save – During Save operations any yet to be delivered reply messages waiting to be received from a Publish Process Step within a Business Process would time out pursuant to the RequestReplyTimeout property of the Client Context and log a Timeout Exception in the Neuron ESB Event log. If there were more than one reply message in flight, the timeouts would escalate resulting in the perception that processing has paused.

Neuron 3.0.3.198

Topics

FIX – Several Null Exceptions were fixed within TCP, Named Pipes and MSMQ Publishing Services that could, under specific circumstances cause message processing to stop until endpoints were restarted

Neuron 3.0.3.193

Policies

FIX – Adapter Policies – The failure action would fire immediately, rather than log and execute the individual retries. This was a bug introduced into the 3.0.3 line. Adapter Policies worked as expected in the 3.0.2 line. This has been fixed.

Topics

NEW – MSMQ Transport – A Transaction Timeout property with a default of 60 seconds has been added to the MSMQ Network properties for Topics. This allows users to configure long running Business Processes that require transactional rollback to the underlying subscriber queue. If the Timeout needs to be set to anything greater than 10 minutes, then the machine's "machine.config" files for the 32-bit and 64-bit .NET Frameworks must also be modified to overwrite the default machine wide transaction timeout of 10 minutes. This can be done by adding the following xml fragment:

```
<system.transactions>
  <machineSettings maxTimeout="01:00:00"/>
</system.transactions>
```

Neuron Explorer

FIX – MSMQ Management – Message box would appear anytime Check Queues or Message Count Functions were executed which alerted the user erroneously that the operations were not supported against remote queue servers. This has been removed as these operations are supported against remote queue servers.

Neuron 3.0.3.190

Neuron Audit Database

NEW – Modified the GetFailedMessagesForProcessing Stored Procedure to accept a Parameter (i.e. @Rows) to limit the number of failed audit messages to return in each batch poll when called from an ODBC Adapter Endpoint.

Adapters

FIX – SMTP adapter could generate an Access Violation exception under heavy load. This has been fixed

Neuron 3.0.3.184

Topics

FIX – If Dead Letter Processing was enabled on more than 1 MSMQ based Topic, the Party information could be incorrectly recorded in the Neuron Failed Audit Database

Neuron Runtime

FIX – Added a gate to prevent any message processing to occur on startup of Neuron Runtime Instance until all endpoints and topics are started. The previous behavior could result in No Subscriber warning messages be logged, when in fact there were subscribing endpoints, but they simply were not fully started yet.

Neuron 3.0.3.176

Business Processes

FIX – Detect Duplicate Process Step – Error would occur retrieving audit database connection string when used within a Step Block of a Split shape.

Neuron Runtime

FIX – Changed order of endpoint startup to ensure Endpoints start up before Publishing Services

Topics

FIX – MSMQ Channel would fail if Transactions were not enabled. Bug introduced with Build 3.0.3.171

Neuron 3.0.3.171

Topics

NEW – Added the Send Timeout property to MSMQ based Topics and extended the max transaction timeout of the publishing service

Neuron Explorer

FIX – An error would occur when trying to import pre 3.0 configurations into 3x Explorer. Specifically a null exception would be caused by encrypted user accounts. This has been fixed.

NEW – A command timeout has been added to the Database configuration form. This controls the command timeout for the Auditing service. On high demand, this can be increased. Default is 5 minutes.

Adapters

NEW – Microsoft Dynamics CRM Workflow Adapter Plugin – Using the Entity send, Neuron only returned the name value if the Entity reference and not the ID in the returned XML. Neuron now adds an ID attribute on any Entity Reference returned.

Neuron 3.0.3.165

Neuron Explorer

FIX – RSS feed could cause Getting Started Page unhandled exception if networking was disabled

FIX – RSS Feed – fixed html encoding

Samples

FIX – Retry Service Sample – Service step properties were not set

Neuron 3.0.3.162

Adapters

FIX – File Adapter – If Create or Append options were selected on the Subscribe mode File Adapter endpoints, the error “Empty path name is not legal.” would result. This bug was introduced in build 3.0.3.147 and is now fixed

Neuron 3.0.3.161

Neuron Explorer

FIX – New RSS feed could cause Getting Started Page unhandled exception if networking was disabled

FIX – Samples link on Getting Starting page would fail to launch Samples Browser

FIX – Endpoint Health – When stopping or restarting an endpoint (i.e. topic, adapter or service) the endpoint would be stopped or restarted on all servers listed in the active deployment group by default. New right click context menus have been added so users can choose to do this action for the current server or all servers.

FIX – Server Management – If alternative logging location was enabled in the registry, it would result in a phantom “Neuron Instance” with instance name of “Unknown” appearing in the server management console.

Business Processes

FIX – Call Service Step – Would through a null exception if dynamic urls were not employed and used within process designer.

FIX – Call Service Step – Call would fail for net.msmq bindings if client credentials were configured

Neuron 3.0.3.153

Neuron Explorer

NEW – MSMQ Management Viewer – When viewing existing messages within a Party Queue, users can now edit the message and save it back to the underlying Queue via a “Save to Queue” context menu as well as a “Save Message” tool bar button.

Neuron 3.0.3.151

Neuron Explorer

FIX – Adapter Endpoints – Changed detailed data, specifically adapter type, mode and description, would not be updated in the master list when Apply was clicked

FIX – Encryption Keys – Copy Key to Clipboard button was added so that the actual text of the encryption key could be copied and viewed.

Neuron 3.0.3.150

Transport

FIX – MSMQ Transport – When using Policy to set Retry Options, TTL (Time to Live) was getting set to value one minute greater than the expiration of the policy. This could cause backed up messages to expire before being retried if there was a failure. TTL is not set to 100x (times) the value of expiration or 24 days, whichever is less. This can be modified in the UI to support a smaller period of time.

Neuron Explorer

FIX – Binding Expressions Dialog – When running under a display mode greater than 100% the bottom command buttons could appear to be cut off.

NEW – Added Neuron ESB Product Team Blog RSS feed to Getting Started screen.

Business Processes

NEW – Removed need to manually enter path and file name of assembly to add as a reference in the Code Process Step. This now uses common dialog function

Neuron 3.0.3.147

Adapters

FIX – Dynamics CRM Subscription Adapter – When receiving messages from MSMQ transactional topics, the adapter could execute several retries, even though each attempt was successful.

NEW – File Adapter – Added Rename file write option. If file exists, the filename will be incremented with zeros to construct a new file name to write to.

Database

FIX – Upgrade script from 3.0.2 to 3.0.3 – Transactional semantics have been rolled in to prevent failed upgrades

Neuron 3.0.3.146

Neuron Explorer

FIX – Endpoint Health Monitoring

- If an endpoint or topic was deleted, it would always show up in endpoint health.
- If the transport of a topic was changed, the old transport would always be displayed in endpoint health until restart of Neuron ESB Service.
- When restarting adapter or service endpoints, the endpoint would stop/start recycle twice.
- When restarting any endpoint or topic errors, warnings, message count and message rate were not being reset.
- If the publishing service's name was changed, it would not be reflected in endpoint health until restart of Neuron ESB Service
- Changed AMQP references to RabbitMq

NEW - Added a "Clear Panel" button on Endpoint Health. This will clear all warning and error counts from the panel and displayed endpoints and publishing services within Endpoint Health.

NEW – Added a "Clear Errors and Warnings" context menu to allow users to reset the error and warning count for the selected endpoint or publishing service

Transport

FIX – MSMQ Publishing Service – Message loss could sometimes occur if Endpoint was stopped or reset while in an Adapter or Service Policy retry mode.

Neuron 3.0.3.145

Business Processes

FIX – If there was an error loading a Business Process or a Business Process Step, the Neuron ESB Explorer would not load the ESB Solution until the problem Business Process or Business Process Step was manually removed from the configuration. This has been fixed so that the ESB Solution will load.

Neuron 3.0.3.144

Adapter Endpoints

FIX – CRM 2011 Workflow Plugin - Updated crm2011 plugin so that when using fetchxml, empty nodes are returned. Previously empty nodes were excluded from the xml returned

Neuron 3.0.3.139

Configuration

NEW – Configuration Encryption

- The Rabbit MQ username/password located in deployment groups are now stored encrypted.
- Environmental Variables are now stored encrypted

Neuron 3.0.3.137

Transports

FIX – MSMQ Topic Transports – Certificate support would fail when selecting Security mode = transport. This has been fixed.

Neuron 3.0.3.136

Configuration

NEW – Configuration Encryption

- Encryption Key data are now stored encrypted

Neuron 3.0.3.134

Transports

NEW – MSMQ Topic Transport – Modified existing properties so that they dynamically displayed based on user selection. An “Ordered Delivery” property has been added. Setting this property to True will automatically enable ordered delivery of messages from the publisher, over the Topic to the subscriber. A new “Use Policy for Retries” property has been added and is only visible if Transaction is set to True. If set to True, another property, “Endpoint Policy” populated with all the existing Adapter and Service Policies will become visible. If this is set with a Policy, the policy’s information for Retries (transactional MSMQ) will be used to override the Topic’s retry settings at

runtime. Additionally, there is no need to assign the policy to the publishing adapter or client connector to set the message's expiration. The published message's expiration property will be automatically set by the transport.

ESB Runtime Pub/Sub

NEW – Support has been added for publishing .NET classes that can only be serialized with the BinaryFormatter i.e. Dictionary objects, etc.

Neuron ESB Explorer

FIX – When working in Online Mode, the new warning dialogs introduced in build 3.0.3.128 could sometimes cause the UI to lock up. This has been fixed.

Business Processes

FIX – Decryption for process step property names ending in “Key” or “Passphrase” were not getting decrypted properly.

Adapter Endpoints

FIX - Decryption for Adapter Endpoint property names ending in “Key” or “Passphrase” were not getting decrypted properly.

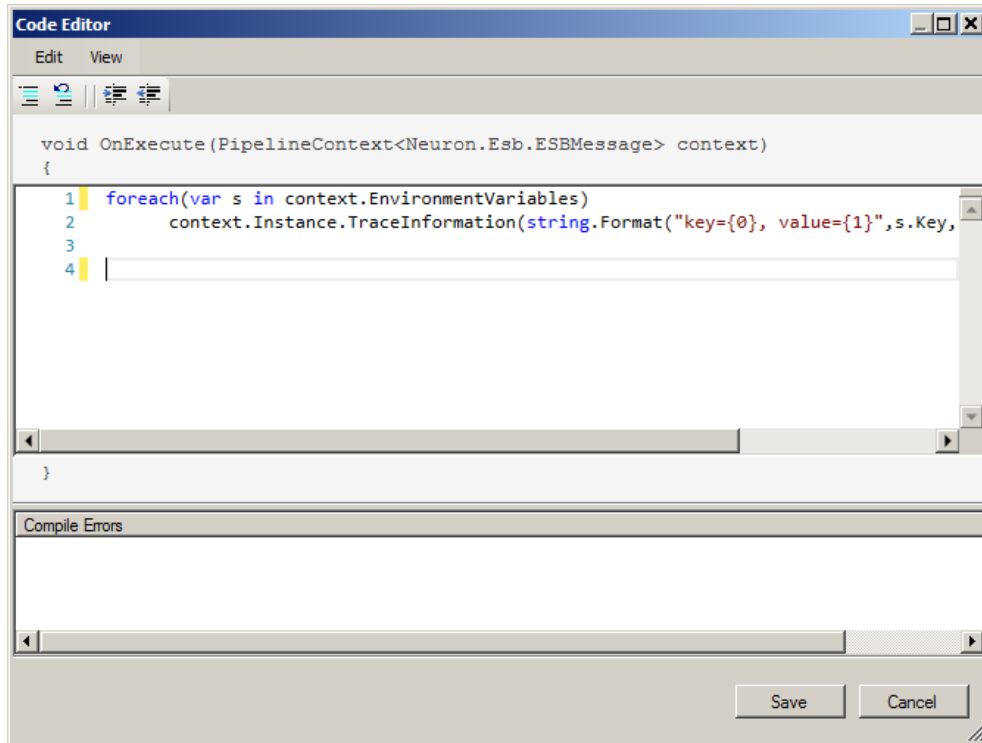
FIX – File Adapter Endpoints – Modified “Filename” property introduced in Build 3.0.3.126. Now if the “Type” property is also independently set for out bound messages, the “Filename” property will automatically use the “Type” extension instead of the original one for serializing out files.

Neuron 3.0.3.126

Business Processes

NEW – A command and connection timeout property have been added to the Store, Table Query and Xml Query Process Steps. Default timeout for connection is 15 seconds while the default timeout for commands is 30 seconds.

NEW – Code process step – Comment/Uncomment, Indent/Unindent commands have been added to the Code Editor accessible through a new toolbar and menu items.



NEW – A new JSON Converter Process Step has been added. This step removes the need to use custom code to handle JSON serialization activities within process. The JSON process step will convert XML to JSON, JSON to XML, .NET classes to JSON, JSON to .NET Classes, XML to .NET Classes, .NET Classes to XML. If serializing out to JSON from a .NET Class or from XML, the date time format can be controlled through the “DateTime Format” property.

Neuron ESB Explorer

FIX – When users on Windows XP would try to change their configuration using the Configure Server folder selection dialog, an access violation would sometimes occur. This has been fixed

Adapters

FIX – FTP Adapter – the ‘ftp.Size’ property was not getting populated when retrieving FTP files.

NEW – File Adapter – the ‘file_in.Filename’ and ‘file_out.Filename’ properties have been added to the File adapter. The properties contain the full file name and extension of the file retrieved, or the file to be sent.

Configuration

FIX – minor fixes have been made to silent install parameters

FIX – When exporting a process that uses a custom process step the ExportConfig.exe will throw this exception: System.NullReferenceException: Object reference not set to an instance of an object. This has been fixed.

Neuron 3.0.3.102

Service Endpoints

NEW – SSL support has been added for REST based endpoints

FIX - When a service endpoint returns a 4xx Status Code, Neuron would return a 500 status code and wrap the actual status code and description in headers, for example:

```
HTTP/1.1 500 The remote server reported an error.  
Content-Length: 0  
Server: Microsoft-HTTPAPI/2.0  
StatusCode: 401  
StatusDescription: Authentication failed. Invalid token.  
Date: Fri, 26 Jul 2013 13:01:58 GMT
```

This has been corrected.

Topics

REMOVED – The BizTalk channel has been depreciated and removed.

Configuration

NEW – All passwords and connection strings entered for process steps, adapters and configured databases are now encrypted and are represented by the “*****” in the configured property grids.

Business Processes

FIX - Modified the validations for the Transform - Xslt and Validate - Schema process steps. This was causing a problem when the name of the schema or transform was being provided programmatically.

NEW – The Zip/UnZipped Process Step has been renamed to Compression. Its previous context property prefix of “zip” has been changed to “compression”. Besides its current support for all the Zip compression methods, additional support has been added for Raw Deflate (compatible with Microsoft’s DeflateStream and PHP zlib’s compression) and GZip. When upgrading, if the older “zip/unzip” process step was used, it should be removed from the business process before upgrading or the business process will not be accessible.

REMOVED – The “Transform – Service”, “Validate – Service” and “Rules – BizTalk” process steps have been depreciated and removed.

NEW – Service Process Step – Users can now dynamically configure the service URL at runtime by setting the context property i.e. `context.Data.SetProperty(“Addressing”, “To”, http://localhost/someService)`. **NOTE: when using dynamic property, dummy value should be entered into the service url property grid.**

Neuron ESB Installer

FIX – Installation would fail on non-english operating systems because the Administrators group was not being localized correctly in the multi-instance setup. This has been fixed

Adapters

FIX – Microsoft Azure Service Bus Adapter – JSON message bodies were being set as binary message bodies during publication to a Topic from Azure. This has been corrected to now attempt to determine and convert to native encoding for all received messages.

FIX – Microsoft MSMQ Adapter – Did not throw error and fail message send when properties on MSMQ message headers were set with invalid values

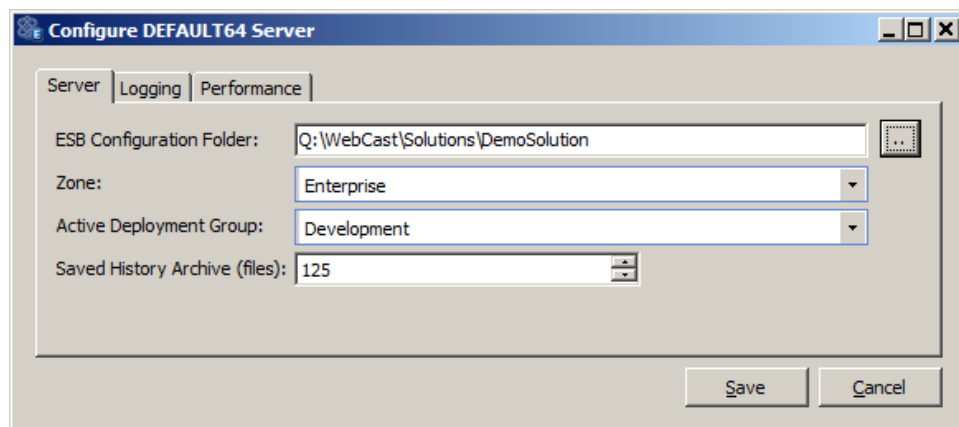
NEW – SMTP Adapter – Added support for SSL as well as username/password, anonymous and default credentials log in to SMTP server.

FIX – SFTP Adapter – Some SFTP servers maintain extended locks on files, preventing deletion. This has been fixed.

FIX – Microsoft Dynamics CRM Subscription Adapter – During x64 installations, the incorrect image of the dependent Microsoft.Crm.Sdk* dlls were being installed.

Neuron ESB Explorer

NEW – Added support for configuring automatic deletion of older Neuron ESB Saved Configuration archive files within the History folder of Neuron ESB Solutions. Saved History Archive (files), defaulting to 0 (unlimited number of archived files), has been added to the Configure Server dialog which can be accessed either from the Neuron ESB Explorer toolbar or from the Neuron ESB Explorer Server management screen (i.e. Deployment -> Manage -> Servers, right click on service and select “Configure Service...” from the context menu).



FIX – When using Windows XP, users could sometimes experience an exception when opening the Folder browse dialog box and navigating between different folders. This has been fixed.

FIX – Database management – Editing the description or security related properties of a database did not trigger the initialization of the “Apply” button to allow persistence of the changes. This has been corrected.

FIX – The Server Topology drop down box for Deployment Groups has been changed to a check box and relabeled to “Use Remote TCP Clients” to more accurately represent the functionality. This only needs to be enabled if using the Neuron ESB Client hosted in remote .NET applications which communicate via TCP based Topics. The database must also be selected.

Neuron 3.0.3.85

Service Endpoints

FIX – REST and OData – When using a REST service connector, encoded values were being re-encoded, resulting in incorrect URLs being sent to OData web services.

Adapters

NEW – Microsoft Dynamics CRM Subscription Adapter – Support has been added for Customer datatype lookups.

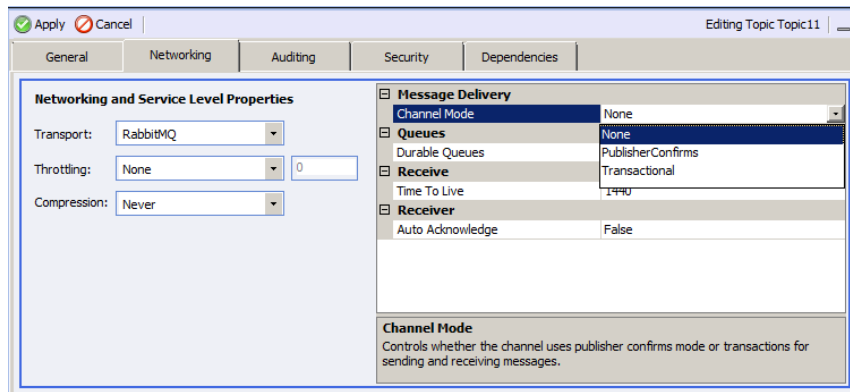
NEW – Microsoft Azure Service Bus Adapter – Support has been added for JSON, Binary and all other message payload formats. New polling options have been added, specifically polling interval and batch size per polling interval. Support for Azure custom brokered message properties has been added. Azure custom properties can be retrieved on incoming messages and can be added or modified to outgoing messages. The following Azure message properties are supported and can be modified or applied to outgoing messages: *CorrelationId*, *MessageId*, *ContentType* and *Label*. The following Azure message properties are supported and can be retrieved from incoming messages: *EnqueuedTimeUtc*, *EnqueuedSequenceNumber*, *Size*, *CorrelationId*, *MessageId*, *SessionId*, *ContentType* and *Label*. The ESB Message context property prefix for Azure message properties is “*azure*”, while the prefix for Azure custom properties is “*azure.custom*”.

FIX – FTP/SFTP Adapter – The Timestamp compare feature has been entirely re written to support FTP folder directories that have over 100,000 files that cannot be deleted. Performance optimizations were also made.

Bindings Adapter Bindings	
Viewing Endpoint Endpoint1	
General Properties	
(Connection)	
Server Name	mwasznicky04.corp.neudesic.net
FTP Connections	2
Port	21
Transfer Mode	Stream
Transfer Type	Binary
Passive FTP	True
FTP Operation Timeout	240
(Security)	
Secure Connection	None
Secure Data Channel	False
Certificate	
Anonymous Connection	True
HTTP Proxy	False
Publish Mode Properties	
Publish Topic	Topic11
Polling Interval	60
Error Reporting	Error
Error On Polling	ReportAllErrors
Delete After Download	False
Enable Timestamp Comparison	True
Check Window	00:04:00
Archive File After Download	False
FTP Folder	
File Mask	*.txt,*.xml,*.txt,*.complete,*.tmp
File Ready Scheme	True
Method	ExcludeFileExtension
File Extension	*.tmp
Publish Empty Message	False
Audit Message On Failure	True
Subscribe Mode Properties	
FTP Folder	
Generate Unique Filename	False
PreAllocateStorage	False
File Ready Scheme	False
Overwrite	True
Audit Message On Failure	False
Enable Timestamp Comparison If True, compares the last write timestamp of the FTP file to retrieve, with the timestamp of the previously retrieved file of same name. If timestamps are the same, the file is not retrieved unless the Check Window property has been exceeded.	

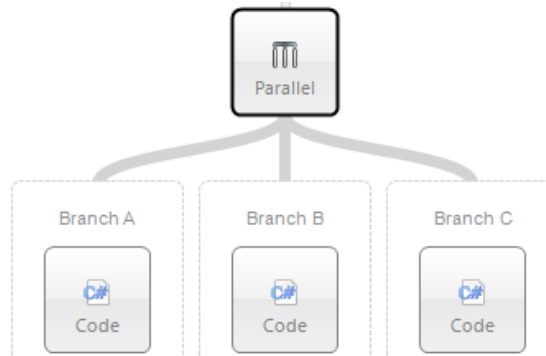
Topics

NEW – The AMQP transport has been renamed to RabbitMQ. Support has been added for RabbitMQ 3.1.3 from 2.8.5. Support has been added to allow configuration of Message level Time to Live. Neuron does automatic monitoring of the dead letter queues, similar to what is done for MSMQ. If a message expires, it will be moved to the Neuron Failed message database. Support has been added to allow users to choose between RabbitMQ's transaction and publisher confirms reliability options at the Topic level (Channel Mode property). Routing is now done at the Exchange level for Topics and Subtopics and Queues are automatically created at startup.



Business Processes

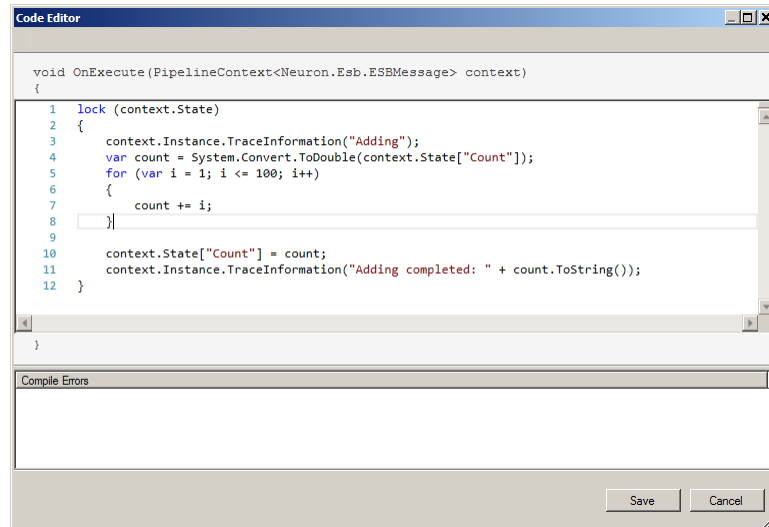
NEW - Parallel process step – This new step allows actions to be executed in Parallel within the Business Process. Multiple parallel branches can be added; each branch by default will run in its own thread. The total number of concurrent threads (i.e. branches) is configurable within the property grid.



NEW – A new State object, off of the context object has been added to manage concurrent access to state across all running instances of the same business process. This State object is thread safe and can be accessed and updated concurrently between threads. Setting the property is as easy as:

```
context.State["Name"] = "Joe";
```

Alternatively a lock can be used to control longer running access against a property in the State object as shown below:



NEW – Transform XSLT Process Step – This has been enhanced to so that custom message properties, environment variables, or XPath expressions in in the parameters collection for XSLT stylesheets.

To use a custom property, enter the following into the value field for the parameter:

```
{property:<prefix>.<name>}
```

For example, if I have defined the custom property “Person.Name” with value “Michael Jones”, then using “{property:Person.Name}” will insert the value “Michael” into the parameter value when the XSLT is evaluated.

For an environment variable:

```
{env:<name>}
```

If I have the environment variable “MachineName” set to “MJONES01”, then using “{env:MachineName}” will result in the parameter having the value “MJONES01”.

Finally, for an XPath expression:

```
{xpath:<xpath-expression>}
```

If my source XML looks like this:

```

<person>
  <firstname>Michael</firstname>
  <lastname>Jones</lastname>
</person>

```

Using “{xpath:/person/firstname}” will use the value “Michael” for the parameter value.

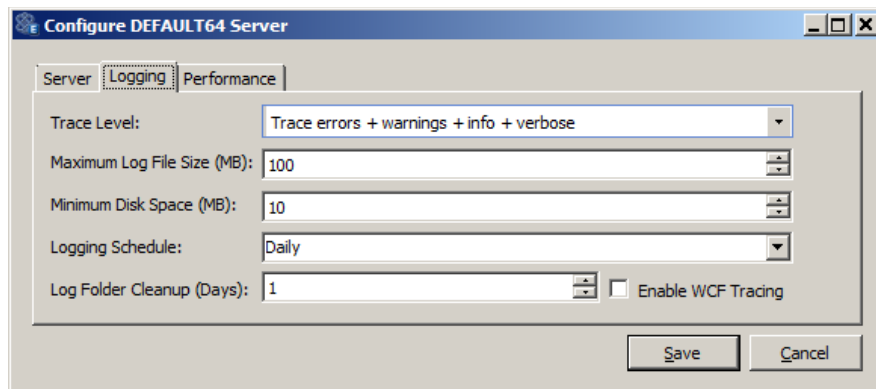
FIX – Service Process Step – If a service url was entered without its complementary bindingName property, an error would result at runtime stating binding cannot be null. This has been changed to provide the user a more

informative error message with information necessary to correct the error. Both design time and runtime validation has been changed

Neuron ESB Explorer

NEW - Neuron ESB Explorer – IF in online mode, users will be notified if the ESB Service runtime they are connected to goes offline. They will also be notified when the service comes back on line. If users try to shut down the ESB Service runtime while in online mode while there are pending changes to be saved, users will be prompted to continue or cancel so that changes can be saved first.

NEW – Added support for configuring automatic deletion of older Neuron Log directories within Neuron ESB Explorer. Log Folder Cleanup (Days), defaulting to 10 days, has been added to the Configure Server dialog which can be accessed either from the Neuron ESB Explorer toolbar or from the Neuron ESB Explorer Server management screen (i.e. Deployment -> Manage -> Servers, right click on service and select “Configure Service...” from the context menu).



Neuron 3.0.2.127 Change History

Business Processes

FIX – A bug was introduced in build 3.0.2.56 that prevented users from setting the ProcessOnReply property to False within a Process connected to the OnReceive event of a Party. Setting this property to false was required before making a request/reply type of call using a Publish step within a process connected to the OnReceive event. Setting this property was also previously required in a process attached to the OnPublish event of a Party, if any Processes attached to the OnReceive event of that Party were required to run. This fix eliminates the need to set the ProcessOnReply header property in any scenario.

Adapters

FIX – The ODBC Adapter’s “For XML” functionality will now work with any version of the SQL Server Native Client ODBC driver.

Neuron 3.0.2.121 Change History

Runtime Performance

FIX – In memory transports saw decreased throughput of up to 4x due to a bug introduced in build 3.0.2.80. This has been resolved.

Neuron 3.0.2.120 Change History

Neuron ESB Explorer

FIX – When a user selected “SAVE AS” from file menu, if the default “Documents” system folder did not exist, an exception would result. This has been resolved.

Neuron 3.0.2.119 Change History

Installer

FIX – Logging bug prevented multiple instances of Neuron from being successfully installed. Setup.exe would generate an error.

Neuron 3.0.2.118 Change History

Service Endpoints

FIX – If a protocol or communication exception occurred when using a REST service connector in a Request/Response message pattern, the user would receive an error similar to:

*Exception Type: System.ServiceModel.ProtocolException
Exception Message: The remote server returned an unexpected response: (400) Bad Request.
Exception Trace:
Server stack trace:*

This would NOT include the web response from the REST service. This has been fixed.

Neuron 3.0.2.113 Change History

Adapters

NEW - SFTP and FTP Adapters – A complimentary property has been added to the Timestamp feature, CheckWindow. The CheckWindow property determines the length of time a file will be resolved as a duplicate when using the Timestamp feature. The default value for this is 1 day.

FIX - FTP Adapter – Disconnect event would not be recorded in the Neuron logs correctly

FIX – FTP/FTPS Adapters – When using multiple connections, errors that occur were not be properly registered. This has been fixed. Also, optimizations have been made.

FIX – Microsoft Exchange, FTP/SFTP, POP3, Microsoft Azure Service Bus Adapters – Error On Polling and Error Reporting properties were not being enforced.

FIX – Microsoft Azure Service Bus Adapter – Wait Duration time was previously defaulting to zero seconds. This has been changed to default to 60 seconds.

Business Processes

FIX - When testing processes within the Process Designer, if objects were stored in the context.Properties dictionary, these objects would persist after the test runs. If a non-serializable object was stored in the dictionary, the process could not be saved.

Setup

FIX - Running setup, the instance name was not getting validated against existing installed instances. If another instance was installed using the same instance name, it would overwrite existing installation.

Neuron Explorer

FIX – When using Windows XP, users could sometimes experience an exception when opening the Folder browse dialog box and navigating between different folders. This has been fixed.

Neuron 3.0.2.94 Change History

Adapters

NEW - File Adapter - Has been enhanced to support multiple File Mask properties within a single adapter endpoint. File masks can be separated by either a comma or semi colon i.e. '*.log;*.txt;*.xml'

NEW - SFTP and FTP Adapters - Has been enhanced to support multiple File Mask properties within a single adapter endpoint. File masks can be separated by either a comma or semi colon i.e. '*.log;*.txt;*.xml'

FIX - SFTP and FTP Adapters - In publish mode, duplicate transport message properties representing the FTP/SFTP adapters may be present. This has been corrected

FIX - MQSeries Adapter - Has been changed to return all Queue types in property grid at design time

Neuron ESB Explorer

FIX - AMQP Management - An unhandled exception would occur when right clicking on row and selecting "purge messages" menu item when rabbit mq is not started. Also returning more useful errors messages when queue has no messages and when queue has yet to be created. Previously, amqp just returned 400 and 404 errors

FIX - AMQP Management - Added delete prompt so user can cancel delete action when selecting purge messages context menu. Also, context menu is now context aware.

FIX – AMQP Management - Changed durable flag display so that if row is not active, durable flag is null rather than false

TEST Client

FIX - The messaging menu items were enabled even though the test client was disconnected.

Topics

FIX – AMQP Network properties - modified amqp channel property descriptions.

FIX – AMQP – Custom messages properties were not being persisted through from publisher to subscriber

Auditing

FIX - SID error when creating an ESB Message within a Code Process step and that message is later audited.

Exception: Procedure or function 'MessageAuditStoreMessage' expects parameter '@SID', which was not supplied

Service Endpoints

FIX - logic error when opening the client metadata form from the client connector tab. The logic error was falsely reporting that a WSDL document did not exist.

Neuron 3.0.2.85 Change History

Adapters

FIX – MQSeries Adapter – In publish mode, monitoring remote queue definitions located in a local Queue Manager, users may receive a “MQRC_OPTION_NOT_VALID_FOR_TYPE” error.

Service Endpoints

FIX – When assigning WSDL metadata to a client connector, if the WSDL was later removed the client connector would fail to start with a “The given key was not present in the dictionary.” error. This error has been corrected and validation has been enhanced.

FIX – When assigning WSDL metadata to a client connector and choosing HTTPS, the TCP/IP port in the WSDL was getting set to 80 instead of 443. This has been corrected

Neuron Explorer

FIX – Opening existing ESB Solutions from the Neuron Explorer in Offline mode would sometimes cause an invalid error (“System.ApplicationException: Cannot load the file”) due to the selected folder name being appended to the full folder path on Windows XP Operating Systems. This has been fixed.

Neuron 3.0.2.80 Change History

Policy

FIX – Retry Expiration Interval was being ignored when set to a value of zero on both Adapter and Service Policies. This has been fixed.

Adapters

FIX- ODBC adapter endpoint would not startup if a connection could not be made to the datasource configured in the supplied connection string. This has been changed to record an error in the event log but still allowing the adapter endpoint to start.

NEW – ODBC adapter – added design time Sql Injection Check property that allows users to configure the ODBC adapter endpoint to validate or not against sql injection attacks.

Business Processes

FIX – Process steps pasted into the root process element (i.e. decision branch, etc) were not being added to the underlying Process block.

Neuron 3.0.2.74 Change History

Neuron ESB Explorer

FIX – When opening an ESB configuration on certain builds of the Windows XP operating system, users would experience the following unhandled exception: *System.Runtime.InteropServices.COMException (0x80040154): Class not registered (Exception from HRESULT: 0x80040154 (REGDB_E_CLASSNOTREG))*. This was due to an incorrect version of shell32.dll. This has been corrected.

Neuron 3.0.2.72 Change History

Transports

FIX – When using the client API to connect to MSMQ based topics remotely, an error would be thrown indicating an *ArgumentNullException* error. This has been fixed.

Service Endpoints

FIX – When passing query string parameters to a client connector, a duplicate key error would occur if there were duplicate query string parameters. This has been fixed.

Business Processes

FIX - A cancel exception was incorrectly resulting in the catch block of the exception process step getting executed if a Cancel process step was used within the Try Block.

FIX – When using the Process Execution step and setting the PipelineName property of that step dynamically, the property was not being removed after being read. This could cause a recursive loop. This has been fixed.

Neuron 3.0.2.63 Change History

Business Processes

FIX – The value of an environmental variable would not be used if configured in a Process Step located on a Process which in turn was called by the Process Execution step. This has been fixed.

FIX – The path directory dialog box for the Process Execution would not populate the “ESB Solution Source” property when a directory was selected.

FIX – If a business process was associated with the OnReceive event of a Publisher, it would not be executed when the reply message was received by the publisher. This has been fixed.

Adapters

NEW – SFTP and FTP adapter – Folder, Filename, Server and Port are all properties that can be dynamically set at runtime for outbound (subscribe mode) adapter endpoints.

Service Endpoints

NEW - Added an optional configuration item that can be added to the ESBSERVICE.exe.config file that will override the connection limit globally for all servers that Neuron sends HTTP requests to. The following line can be added to the ESBSERVICE.exe.config file:

<configuration>

```

...
<appSettings>
    ...
    <add key="DefaultHttpConnectionLimit" value="10"/>
</appSettings>
...
</configuration>

```

The DefaultHttpConnectionLimit setting will override the default limit of 2 connections to each outbound HTTP server that is being enforced by the .NET Framework. Given the example above, the default number of concurrent connections to any remote server will be 10.

FIX – When using the Client<T> of interface to call a service endpoint, if the service endpoint was configured to use an environmental variable, the call would use the configured URL instead if a change was made to the configuration and saved. This has been fixed.

FIX – On startup, Neuron would fail to report if an error occurred when opening the proxy to a service endpoint, resulting in a faulted exception when the endpoint would be called. This has been fixed

Neuron 3.0.2.33 Change History

Business Processes

FIX – When adding Business Processes to Neuron Parties users would experience an error such as “Environment Variable X does not exist” if the Business Process contained a reference to an Environmental variable.

FIX – Test Message Viewer in the Business Process Designer would appear in the incorrect size if desktop preference was set for anything other than 100%

FIX – When executing the Zip/UnZip process step, an object reference error could result.

FIX – When using the Detect Duplicates process step, if an environmental variable was used to configure the connection string of the Neuron Audit Database, its value would not be used.

FIX – ODBC Process Step – When receiving messages over MSMQ, Transactional topics, the ambient transaction would sometimes be enlisted, ignoring the serialization and timeout options of the ODBC Adapter configuration. This could have also resulted in extended locking of the resources. The Transactional property has been changed from “True/False” to “None/Required/RequiresNew” to fix this.

Configuration Management

FIX – both ImportConfig.exe and ExportConfig.exe would throw exceptions when executed. Also new flags, -verbose and --o, has been added to provide logging to the console and optionally output the log to a text file specified by the user.

FIX – When importing archive packages (*.esb) via the Import User Interface in the Neuron ESB Explorer, WSDL documents would not get imported.

FIX – The create script function of MSMQ management within the Neuron ESB Explorer would fail to include the names of the queues to create within the generated scripts.

Adapters

NEW – SFTP Adapter has been added. This also supports the following optional features:

- Supports MD5 SSH host key or fingerprint
- Supports private Key (*.PPK) generated by puttygen.exe
- Supports Delete after Download as an option for publish mode
- Archive on Publish feature. Allows users to configure the publish side adapter to archive retrieved files on the existing SFTP server after successful download as an alternative to deleting the source file.
- Include Sub Folders on publish mode. Publish mode adapter can be configured to scan subdirectories of the specified root for files to publish to the bus, relieving the need to specify a unique adapter endpoint per SFTP folder to monitor.
- Create matching sub folder option when Archive on Publish feature. If the publish mode adapter is configured to scan recursively through sub folders, it can optional recreate the sub folder hierarchy under the archive folder specified.
- Trace writer feature. When Neuron ESB service is running in verbose mode a new trace file will be written to the Adapters directory that will contain diagnostic information regarding the SFTP commands being sent to server.
- Overwrite file option and File Ready features added to Subscribe side SFTP Adapter. Users can now configure adapter to use a temp file extension during upload. Once upload is done, the temp file extension will be renamed to original file extension, signaling file is ready to be accessed by other processes. Users can also specify whether to over write the target file name, if it already exists.
- Timestamp feature to avoid duplicate files during publish mode. If user enables compare by timestamp, the collection will survive between shut downs of the SFTP Adapter endpoint or Neuron ESB Runtime service

FIX – MSMQ adapter subscribe mode. A “Stream Message Body” property has been added. By default this is set to true, which is the old behavior of the adapter. When the Use ActiveXMessageFormatter property is set to true, the new “Stream Message Body” property should be set to false to ensure the ActiveXMessageFormatter is used to serialize the message to the underlying MSMQ Queue in subscribe mode.

FIX – FTP adapter Timestamp feature would sometimes fail to persist to Neuron Audit database.

FIX – FTP adapter Timestamp feature. If an environmental variable was used to configure the connection string of the Neuron Audit Database, its value would not be used.

FIX – FTP Adapter may throw null exception if Publish on Audit Failure was enabled and FTP Connections was configured greater than 1.

FIX – Azure Service Bus Adapter may throw null exception if Publish on Audit Failure was enabled.

FIX – Microsoft Exchange Adapter may throw null exception if Publish on Audit Failure was enabled.

FIX – POP3 Adapter may throw null exception if Publish on Audit Failure was enabled.

FIX – FTP Adapter would throw a 550 reply error when in Publish mode if a file mask was specified and no files were found matching the file mask.

FIX – ODBC Adapter - In 64 bit installs of Neuron, ODBC errors would be thrown in the ODBC Connection Builder UI if either the Load, Create or if users queried the ODBC Drivers by enabling the radio button.

FIX – ODBC Adapter – When receiving messages over MSMQ, Transactional topics, the ambient transaction would sometimes be enlisted, ignoring the serialization and timeout options of the ODBC Adapter configuration. This could have also resulted in extended locking of the resources. In cases of Batch mode, partial batch committals would result rather than entire rollback of the batch on error. The Transactional property has been changed from “True/False” to “None/Required/RequiresNew” to fix this.

NEW – Single Instance mode for Adapter Endpoints have been enhanced to work across Neuron ESB Runtimes instances on a single server rather than only across physical server boundaries. When enabled, Single Instance Mode prevents more than one instance of the specific adapter endpoint from being started at any one time between ESB runtime instances whose solutions are configured to use the same Neuron Audit database. If the running adapter endpoint stops or becomes disabled for any reason, another instance of the same adapter endpoint will be automatically started in one of the other running solutions. The only requirement is that the participating solutions are configured to use the same Neuron Audit database, the adapter endpoint name is identical and the adapter endpoint “id” is identical. The “id” is specified in the XML definition of the endpoint. An endpoint can be exported from one solution and imported to another to preserve the “id”.

Policies

FIX – Changed the default for MSMQ Retry Interval in Service and Adapter policies from 1 second to 60 seconds

Auditing and Reporting

FIX – Primary key has been added to Audit Tables

FIX – Resubmitting binary messages via the Message Viewer would sometimes with an empty stream error.

Service Endpoints

FIX – When assigning WSDL to a client connector, an exception may occur on startup of the endpoint indicating bad XML due to extra characters that may sometimes be read.

NEW - DomainCredential, UserNameCredential, and PasswordCredential have all been made bindable properties for service endpoints. This means that Environmental Variables can be used to configure their values

NEW – Added an override check box to the “Configure Client Connector Metadata” form that allows clients to consume the client connector URL over HTTPS even if it is manually configured for HTTP.

Neuron 3.0.2.8 Change History

Service Endpoints

- FIX – When a service endpoint goes offline temporarily, the Neuron Service Connector would fail to reconnect, even when the service endpoint comes back on line. The user may receive either a CommunicationException or WebException. This would require the user to restart the Service Connector Endpoint within Endpoint Health.
- FIX – REST services: When an exception was returned to the Neuron Client Connector for a request, it was being returned with a 200 HTTP status code and not getting processed correctly. The user would receive an error such as:

“There was an error checking start element of object of type [x]. The data at the root level is invalid. Line 1, position 1. ---> System.Xml.XmlException: The data at the root level is invalid. Line 1, position 1.....”

If the HTTP status code is not being set appropriately by the endpoint, then it will now be set automatically to 500 and be processed correctly as an exception.

- FIX - When setting The HTTP header property Transfer-Encoding = chunked on service request messages the client would receive an HTTP 504 (GatewayTimeout) error
- FIX – REST services: Neuron incorrectly builds the Query string in Neuron 3.0.x. The logic for building the query string for a REST request was incorrect. The ‘&’ character was being pre-pended to the string instead of being used as a separator between parameters.

Server Management

- FIX – Users were unable to provide a set of credentials to both connect and manage remote Neuron servers that were not part of the same domain. This has been fixed so that users can enter alternative set of user credentials on the connect form for the Neuron ESB Explorer. These credentials will be used to connect as well as perform all server management functions within the server management screen within the Neuron ESB Explorer.
- FIX – the server name in the Connect form was not being correctly preserved with user preferences

Adapters

- NEW – FTP Adapter – Added the following new features:
 - Archive on Publish feature. Allows users to configure the publish side adapter to archive retrieved files on the existing FTP server after successful download as an alternative to deleting the source file.
 - Trace writer feature. When Neuron ESB service is running in verbose mode a new trace file will be written to the Adapters directory that will contain diagnostic information regarding the FTP commands being sent to server.
 - Overwrite file option and File Ready features added to Subscribe side FTP Adapter. Users can now configure adapter to use a temp file extension during upload. Once upload is done, the temp file extension will be renamed to original file extension, signaling file is ready to be accessed by other processes. Users can also specify whether to over write the target file name, if it already exists.

Neuron 3.0.1.37 Change History

Installer

- FIX – SETUP.EXE would fail if the registry subkeys for changing the Neuron ESB log file location below existed:
 - HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Neudesic\Neuron ESB v3\ESB\SystemLogFolder
 - HKEY_LOCAL_MACHINE\SOFTWARE\Neudesic\Neuron ESB v3\ESB\SystemLogFolder
- This has been fixed.

User Interface

- NEW – Replaced current open folder dialog used for opening ESB Configurations or saving ESB configurations with a new UI. The new UI uses a customized dialog for Windows XP and the standard Shell folder selection dialog for Windows Vista and above. More intuitive.

Adapters

- NEW – Microsoft Dynamics CRM Subscription Adapter : Added the Microsoft.CRM.Sdk.dll, Microsoft.CRM.SdkTypeProxy.dll and, Microsoft.CRM.SdkTypeProxy.XmlSerializers.dll to the installer to support the adapter
- FIX – Microsoft Dynamics CRM 2011 Workflow adapter plugin: Dependent Neuron Client API dlls are now automatically added to the GAC. In previous versions this had to be manually done.
- FIX – Microsoft Dynamics CRM 2011 Workflow adapter plugin: Default registry entries are now created during installation. In previous versions users had to first run the Neuron ESB CRM Workflow Adapter Configuration Tool after install.
- NEW – ODBC Adapter: Added Command and Connection timeouts as configurable properties
- NEW – SQL Adapter: Added Command timeout as a configurable property
- NEW - FTP Adapter: Added persisted timestamp feature. If user enables compare by timestamp, the collection will survive between shut downs of the FTP Adapter endpoint or Neuron ESB Runtime service
- FIX - ODBC Adapter: Fixed rename bug when using text odbc driver to parse flat files. Also added extended exception handling to gracefully handle issue when file doesn't exist
- NEW – ODBC Adapter: New Batch mode capabilities have been added. Previously users could specify batch table inserts. However, all column inputs were treated as strings and enclosed in quotes during execution. Now data type information can be added as an attribute to the column to prevent this for data types such as an int or bool. Additionally, a new format has been introduced that also allows Batch mode capabilities to call stored procedures rather than just insert directly into a table. The formats are as follows:

```
<Batch>
  <Statement type="StoredProcedure" name="uspUpdatePhoneNumber">
    <Fields>
      <Field type="int" name="Id">2</Field >
      <Field type="varchar" name="PhoneNumber">22222222</Field >
    </Fields>
    <Fields>
      <Field type="int" name="Id">21</Field >
      <Field type="varchar" name="PhoneNumber">21212121</Field >
    </Fields>
    <Fields>
      <Field type="int" name="Id">24</Field >
      <Field type="varchar" name="PhoneNumber">24242424</Field >
    </Fields>
  </Statement>
</Batch>
<Batch>
  <Statement type="Table" name="PhoneBook">
    <Fields>
      <Field name="Total" type="int">12345</Field >
      <Field name="FirstName">Mary</Field >
      <Field name="LastName">Jane</Field >
    </Fields >
    <Fields >
      <Field name="Total" type="int">123444445</Field >
      <Field name="FirstName">Joe</Field >
      <Field name="LastName">Box</Field >
    </Fields >
    <Fields >
      <Field name="Total" type="int">87888</Field >
      <Field name="FirstName">John</Field >
      <Field name="LastName">Smith</Field >
    </Fields >
  </Statement>
</Batch>
```

```
</Statement >
</Batch>
```

Business Processes

- NEW – ODBC Process Step: Added Command and Connection timeouts as configurable properties
- NEW - Detect Duplicate Process Step: Modified so that the audit database is resolved at runtime, eliminating need for user to supply a connection string at design time. Also extended the size of the unique id to 200 from 50.
- NEW – Zip/UnZip Process Step: This has been added. New Zip Object model has also been added that can be used directly from a Code Process Step when unzipping a collection of files
- NEW – Added the ability for users to access or add Environmental Properties for specific Deployment Groups when testing processes within the Neuron ESB Explorer Process Designer. Environmental Variables can now be tested in the design time environment.
- FIX – Audit Process Step: When importing existing 2.6 ESB Configurations, the default for auditing properties and bodies were automatically set to FALSE. This has been correct so that these properties will automatically be set to TRUE
- NEW – Schema and Transform Process Steps: Added a property that allows users to simply select the name of the schema or transform to be used at runtime, rather than import them in (which still remains an option). The names will be used to look up the XSLT or XSD documents from the configuration at runtime instead of typing in the XSLT or XSD or maintaining two copies in both the configuration and process step definition.

Reporting/Auditing

- FIX – Republish message from Audit and Failed Audit would fail if Bootstrap Service port was configured with anything but 50000.

Repository

- FIX - When saving XML documents in the repository where the existing documents were being "opened" and not "created". This caused a bug if the revised XML was shorter than the original file. The new file was being saved, but the remaining text at the end of the file was not being removed.

Neuron 3.0.0.839 Change History

Transports

- NEW - MSMQ Transport Property for Topics – Users can now configure the “Custom Dead Letter Queue” property with an environmental variable entering the syntax “{\$myvariable}” as the value for the property

Neuron 3.0 Change History

User Interface

- NEW – The entire Connect/Create/Open dialog of the Neuron ESB Explorer has been replaced with a new user experience and asynchronous functionality. On connect, if runtime service is not started, user will be prompted to allow the Explorer to start the service, eliminating need for manual restarts. User selection is saved as a preference

- NEW – Neuron ESB Explorer – Significant modifications have been made to the Neuron ESB Explorer, both from a functional and usability perspective. All long-running operations and processes within the Neuron ESB explorer are now asynchronous. This means operations can be cancelled and navigation to other functional areas are no longer blocked, resulting in a more consistent, usable and reliable experience for users.
- NEW – Samples Launcher - Neuron ESB now ships with an improved “Samples” launcher along with 50 samples, 18 of which are new to Neuron ESB 3.0 and demonstrate various capabilities.
- NEW – Topic Subscription builder and Business Process Selection for Publishers and Subscribers has been completely refactored

Configuration

- NEW - The Neuron ESB 3.0 configuration storage format has evolved from the single file format (i.e. *.ESB file available in Neuron ESB 2.x) to a directory structure consisting of folders representing each entity type. Within each folder, an XML formatted file represents a specific entity such as a Topic, Endpoint, Business Process, etc. The Neuron ESB Explorer is designed to work and store all modifications to the directory structure. To migrate older configurations, users can create a new configuration with the Neuron ESB Explorer and then choose File->Import and select the older *.ESB file. Once the import completes the user can save the new configuration to a folder.
- NEW – Export Utility - new Export utility can be accessed from the File menu of the Neuron ESB Explorer. When “Export” is selected, the Export dialog box appears and allows users to select the individual entities to be exported (or all of them). If an entity is selected that has dependencies, the user is prompted with those dependencies and given the option of including them in the export. Once the selection is complete, all the selected entities are exported out to a new *.esb package file.
- NEW – Import Utility - new Import utility can be accessed from the File menu of the Neuron ESB Explorer. When “Import” is selected, the user is prompted to select an existing *.esb configuration file. Either a new Neuron 3.0 *.esb package file or an older Neuron ESB 2.x *.ESB configuration file (if migrating a Neuron ESB 2.6 solution to Neuron ESB 3.0) can be selected. The contents of the *.esb file will be displayed in the Import dialog box.
- NEW –Neuron ESB 3.0 provides two new command line tools, to automate the import and export of entities from one ESB configuration to another. The command line tools are located in the default instance directory of the Neuron ESB installation folder, and are named ExportConfig.exe and ImportConfig.exe.

To make it easier to re-export the same objects or groups of objects, the ExportConfig.exe program recognizes the presence of response files on the command line. A response file is a simple text file that contains one option per line and can be generated from the Import/Export dialogs. For example, a response file that exports a Topic, Publisher, and Subscriber would look like this:

```
--topic=Topic1
--party=Publisher1
--party=Subscriber1
```

To use the response file on the command line, prefix the path to the response file with an @ symbol. For example:

```
ExportConfig @TopicsAndParties.rsp C:\MyConfiguration
C:\TopicsAndParties.esb
```

Services

- NEW – REST based Object Model added – Neuron ESB 3.0 provides a new object model. In previous versions of Neuron ESB users had to modify custom property names and text values to manage and manipulate REST communication within Neuron. Neuron ESB 3.0 introduces a new HTTP class that can be directly referenced within a Business Process Code Step
- NEW – JSON support for REST - Neuron ESB 3.0 ships the Newtonsoft JSON library which can also be referenced directly within the Business Process Code Step editor.
- REST support for Service Policies - Neuron ESB 3.0 introduces REST HTTP Status code support for service policies. Users can now configure a service policy to retry and fail depending on the HTTP Status code returned on a REST service endpoint.
- WSDL support for Neuron ESB hosted Services (Client Connectors) - Neuron ESB 3.0 has added the ability to attach and expose WSDL from a Client Connector. WSDL can be imported using the Import Service wizard and stored in the new WSDL repository and then later selected from any configured Client Connector.
- NEW – When existing services are imported using the Import Service wizard, the WSDL and Schemas are automatically saved within the Schema and WSDL repository.
- NEW – base addressing for Client Connectors has been changed to allow port sharing. Now 2 Client connectors with the same base port address can be created and started i.e.:
 - <http://localhost:9000/myservice>
 - <http://localhost:9000/myserviceOops>

Adapters

- NEW - Neuron ESB 3.0 introduces several new enhancements and capabilities to increase the overall effectiveness and user experience of the ODBC adapter. In Neuron ESB 3.0 the ODBC Adapter's connection builder UI has been re-designed to provide asynchronous support to long running operations such as querying a driver's capabilities and testing connections. Also, once a driver is selected its keywords are queried and displayed for the user as a tooltip, providing guidance for the connection string construction.
- NEW – Support for catching Access Violations from ODBC Drivers within the ODBC Adapter has been added
- NEW - Added support for forcing failure status on CRM 2011 processes using the Neuron CRM 2011 workflow adapter
- NEW – Microsoft Azure Service Bus Adapter - Neuron ESB 3.0 includes an Azure Service Bus Adapter to provide reliable communication on either endpoint of the Azure Service Bus. The Azure Service Bus Adapter provides full support for the latest capabilities provided by the Windows Azure SDK version 1.7.

The Azure Service Bus adapter supports the following Azure Service Bus Brokered Messaging features:

- Send to Azure Service Bus Queue
- Send to Azure Service Bus Topic
- Receive from Azure Service Bus Queue
- Receive from Azure Service Bus Subscription

In addition, the Neuron Azure Service Bus adapter simplifies the development experience by providing additional capabilities required to ensure reliability typical in production scenarios without the need to write custom code including:

- Smart Polling
- Eventual Consistency

- Transient Error Detection and Retry

Auditing

- NEW – Support has been added so users can configure the Neuron ESB Audit database using Microsoft SQL Azure. Scripts to create the database can be found in the “\Neuron ESB v3\Sql\Azure” install directory.
- NEW – Users can now configure Auditing to optionally persist both Body and Custom Properties. This can be configured either at the Topic level or through the use of the Audit Process Step.
- NEW – Bus monitoring of failed Messages has been added. Users can now configure either an ODBC or SQL adapter endpoint to poll the Neuron Audit database and call the ***GetFailedMessagesForProcessing*** stored procedure. This stored procedure will return a batch of failed messages, including some of the meta-data. A Neuron ESB Business Process can then asynchronously split the batch into individual messages, transform it (using XSL), and email it out to subscribing entities with the original message as an attachment.

Management

- NEW - Neuron ESB 3.0 now provides ability to centrally manage and configure all Neuron ESB runtime service instances on all servers within the environment, using the new Server Management Console within the Neuron ESB Explorer. The server management functions have been refactored to be asynchronous in nature, allowing operations against multiple instances simultaneously. Each Neuron ESB runtime service instance, regardless of server location, can be configured, as well as started or stopped, through the Neuron ESB Discovery service. Users can select multiple instances and select to start or stop them all at once.
- NEW – Automatic detection and reload of changed ESB Configuration. When using the Neuron ESB Explorer in offline mode, if a change is made to the underlying configuration by another process or user, the user will be notified of the change and prompted to optionally load the new changes.
- NEW – Endpoint Health Monitoring has been refactored to be asynchronous. Users can now jump directly to the event log or to the appropriate queue management screen based on context
- NEW – Dependency Viewers - Neuron ESB 3.0 introduces Dependency Viewers and alerts on all Neuron ESB entities managed within the Neuron ESB Explorer. The Dependency Viewers are visible within a dedicated Tab on an entity’s configuration screen. This Dependencies tab can be found on the following Neuron ESB entity configuration screens:
 - Topics
 - Publishers
 - Subscribers
 - Conditions
 - Databases
 - Environmental Variables
 - WSDL Documents
 - Adapter Registrations
 - Service Bindings
 - Service Behaviors
 - Credentials
 - Access Control Lists
 - Encryption Keys

- When a user attempts to delete an entity that has dependencies, they are prompted with the list of dependencies so that they can make informed decisions.
- NEW – Global Categories - Within Neuron ESB, every entity (i.e. Topic, Publisher, Subscriber, Adapter, Adapter Endpoint, etc.) can be configured with a Category. Categories are a way to visibly group related entities within a solution. Adding a Category is as easy as typing in a new name in the category field and clicking “Apply”, followed by “Save”. Categories are global in nature with the Category filter drop down now located on the main Neuron ESB Explorer Toolbar. If users enter a new Category within any entity, it will immediately be available to select as an available Category in any other entity type within the Neuron ESB Explorer. If a user selects a Category on the toolbar filter, then only those entities within the Neuron ESB Explorer that are associated with that Category will be visible. This provides users a way to group entities in application related clusters.
- NEW - The Event Log viewers within the server management console have also been refactored to provide real-time updates as events occur, the ability to export and email the event log by the source of the log (Neuron ESB runtime service name) and the ability to clear the log. Using the Event Log viewers, users can see detailed information, as well as the categories and source of that information.
- NEW – Event logging to the Neuron ESB Event log has been modified. A new Neuron ESB v3 Windows Event log is installed to record all Neuron ESB version 3 events. All Neuron ESB instances are registered as Sources. For example, if an instance named “Default64” is installed, events from that instance will be recorded with a source of “ESBv3_DEFAULT64”
- NEW – A new SCOM management pack ships with Neuron ESB and is located in the “\SCOM” instance’s sub directory
- NEW – A new Neuron ESB Discovery service is installed, running on port 51001. This discovery service is set to start automatically at startup and registers and monitors all installed instances of the Neuron ESB runtime on a Machine. It automatically starts instances if prompted by users within the Neuron ESB Explorer. This removes need to manually startup Neuron ESB runtime instances. The Neuron Discovery service is also used to remotely manage and configure all Neuron ESB Runtime instances on a machine.

AMQP/Rabbit MQ

- NEW – AMQP/RabbitMQ has been added as a new Transport that users can configure Topics with.
- NEW – AMQP Management screen has been added within the Neuron ESB Explorer allowing users to see real time statistics as well as the ability to purge underlying queues

MSMQ

- NEW – Automatic Poison Queue and Dead Letter Queue handling has been added to Neuron Topics. When activated (true by default) all messages that get written to the Poison Queue or Dead Letter Queue will automatically be moved to the Neuron Audit Failed database.
- NEW - FailedMessagePath property has been added to the MSMQ Transport so that users can configure where dead letters or poison messages are written to if auditing is disabled
- NEW – Security attributes have been added to MSMQ based Topics. Users can now configure authentication and authorization on MSMQ topics and use Certificates
- NEW – Messages that are published, but have NO subscribers defined will be stored in the Neuron ESB Audit Failed database with the error of No Subscriber
- NEW – An entirely new MSMQ management experience has been added to the Neuron ESB Explorer. Users can execute the following functions, accessible through the toolbar or through context menus:
 - Create Queues
 - Check Queue Existence

- Purge Queues
- Delete Queues
- Delete Messages
- View Messages
- Edit/Resubmit Messages
- Message Counts

These new features (except for Create and Delete Queues) can be used to manage Queues for either local or remote Neuron ESB servers. Editing pending queued messages and resubmitting them for redelivery is also available from within the Neuron ESB Explorer's Queue Management console.

MISC

- NEW – The Topic level Encryption option of "RIJ" has been changed to "AES."
- NEW – Microsoft Dynamics CRM 2011 Workflow adapter plugin is now shipped with the product and can be found in the "\Plugins\Crm 2011 Neuron Publication" instance's sub directory
- NEW - Microsoft Dynamics CRM 4 Workflow adapter plugin has been relocated to the "\Plugins\Crm 4 Neuron Publication" instance's sub directory
- NEW – The Microsoft BizTalk Server plugin has been updated to support Microsoft BizTalk Server 2010 and has been relocated to the "\Plugins\BizTalk Server 2010 Channel" instance's sub directory
- CHANGE – The Microsoft SharePoint Server 2007 plugin has been relocated to the "\Plugins\SharePoint 2007 Neuron Publication" instance's sub directory
- CHANGE – The Microsoft SharePoint Server 2010 plugin has been relocated to the "\Plugins\SharePoint 2010 Neuron Publication" instance's sub directory
- NEW – The default "Sample" solution has been updated to the New Neuron ESB Configuration store and has moved to the "\Sample" instance's sub directory
- NEW – Neuron ESB Setup has been modified to include a bootstrap program (Setup.exe) that allows users to install optional components (like Erlang and RabbitMQ). This will detect if other requirements are installed (like .NET 4) and install them if necessary
- NEW – Neuron ESB version 3 has been updated to be compatible with the Microsoft .NET 4 framework
- NEW – The Neuron ESB Explorer plugin has been updated to support Microsoft Visual Studio 2010 and 2012
- NEW - A configuration statement was added to the installer and application configuration files to disable FIPS policy for cryptographic algorithms
- NEW - Added binding redirects for 2.6 versions of custom adapters and custom process steps. This should allow existing custom adapters and process steps to work with Neuron 3.0 without the need to recompile them against our new assemblies

Business Processes

- NEW - Business Process Code Step editor has been updated to support the latest .NET language enhancements, including LINQ automated IntelliPrompt and lambdas.
- NEW – Users can optionally expose Line Numbers within the Business Process Code editor
- NEW – Copy, Cut, Paste and dynamic Zoom have been added to the Business Process designer
- NEW – External assembly references can now be added and managed at the Business Process level, rather than in each individual Code step
- NEW – Import and Export toolbar buttons have been added to the Business Process Designer

- NEW – Test form for submitting messages to test in the Business Process Designer has been modified with toolbar buttons and the ability to load an existing ESB message previously saved from Neuron Auditing or the Neuron ESB Test client for Process Testing
- NEW – Users can now define a custom folder hierarchy within the Process Step library UI to display custom Process Steps. The folder hierarchy can be created during registration of custom Process Steps by specifying an optional “path” attribute within the “neuronpipelines.config” file
- NEW – The new Detect Duplicates Process Step can be configured to use the current Neuron Audit database. Users can configure the criteria (either at design time or runtime) that define what makes a message unique. Users can also configure the time window in which a duplicate message is to be prevented from being received.
- NEW - A new Process Step that allows a For Each type looping construct to be configured within a Process has been provided. The existing Break Process Step can be used within the For Loop. This has advantages over the For Loop Process Step, as any .NET collection that supports enumeration can be used with fewer configurations.
- NEW – The new ODBC Process Step provides powerful capabilities to users building Business Processes that need to integrate directly with ODBC based data sources (i.e. Excel, SQL Server, Oracle, DB2, MySQL, and Lotus Notes, among others). The ODBC Process Step supports one way inserts and updates, as well as query (i.e. Request/Response) types of message patterns. Optionally, the ODBC Process Step can also generate XML Schemas that represent the result sets returned. This Step also leverages the new ODBC connection string builder that the Neuron ESB 3.0 ODBC Adapter users.
- NEW – The process Code Editor’s window can now be maximized
- NEW - Added dynamic connection string support for the Table Query, Xml Query and Store Business Process Steps. The connection string properties for the steps are:
 - XmlQuery.ConnectionString
 - TableQuery.ConnectionString
 - Store.ConnectionString

These can be configured by calling the SetProperty() method of the ESBMessage object. For example:

```
Context.Data.SetProperty("Store","ConnectionString", "my connection string");
```
- FIX - Table Query and Store steps would fail when XPATH was provided but not used as Command Parameters
- FIX – An unhandled exception would occur if user opened a Code step that accesses an assembly that could not be found