



Cressida Technology Ltd.
 1 Lammas Gate, 84A
 Meadrow
 Godalming
 Surrey GU7 3HT
 UNITED KINGDOM
 🇬🇧 UK (+44) 1483-239300
 🇺🇸 USA (+1) 914 238-7900
 📠 (+44) 1483-239383
 🌐 <http://www.cressida.info>

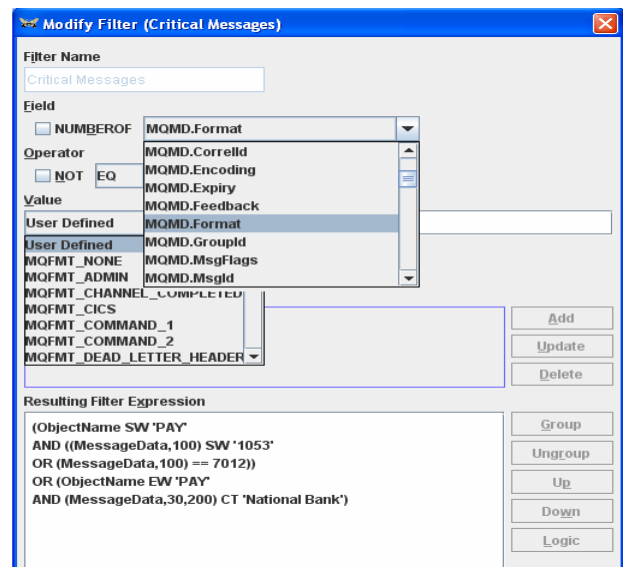
Cressida InQuest™ for WebSphere® MQ

Cressida InQuest™ for WebSphere® MQ (**InQuest**) is a user-configurable tool that utilises the power and functionality of the IBM® WebSphere® MQ (WMQ) API Exit to “intercept” MQI calls that meet certain criteria (as specified in **Filters**), and to perform specific functions (as defined in **Actions**).

The powerful and flexible **Filter Editor** allows the definition of **Filters** that combine any number of criteria to test if a certain API call requires further processing. Each criterion defines a comparison to be performed using the value of an attribute of the call against a comparison value. Depending on the data type of the attribute being tested, various comparison operators are available, including string scan and bit (option) test operations. The graphical front-end allows the creation of complex filter combinations with great ease, while the run-time component ensures that evaluation of these filters adds minimal overhead to the running applications.

Depending on the call type being intercepted, any combination of the following fields can be used when defining filter criteria:

- **ObjectName**
- **CompletionCode** and **ReasonCode**
- fields **AppName**, **AppIType**, **ConnectionName**, **Environment**, **SecurityId** and **UserId** from the **MQAXC** structure
- any of the **MQOD** fields
- any of the **MQMD** fields
- any of the applicable **Options** fields (**MQGMO**, **OpenOptions**, **MQPMO**)
- **the message Data** (MQGET AFTER, MQPUT and MQPUT1)
- any of the **MQDLH** fields (Dead Letter Queue only)
- fields **RemoteQName** and **RemoteQMgrName** from the **MQXQH** structure (Transmission queue only)



No limits are imposed by the Filter module on how many criteria are defined neither in a single filter, nor on how these are combined, nor on the total length that a filter string can become.

Any number of **Actions** can be associated with a filter, specifying what extra processing should occur if the filter evaluates to **TRUE**. Actions are grouped in individual **InQuest modules** that can be installed incrementally depending on the needs of your application. Some typical examples of actions include:

- Raise an SNMP alert or MQ Event if an amount in a message exceeds a certain limit
- Fail a GET with a long wait, if the option FAIL_IF_QUIESCING is not also specified, or Set the option if it's missing
- Replicate messages for one particular client to a file or to another queue

The modular design of InQuest allows the addition of features and functions as and when needed, without requiring changes to applications.

Key Benefits of the InQuest Infrastructure:

- Eliminate “application stubs” and/or “API wrappers”
- Simplify applications by externalising part of the application/business logic
- Dynamically activate logic changes without recompiling programs or recycling queue managers
- User friendly graphical front end interface
- Low run-time overhead

The next sections introduce the **InQuest modules** that are currently available. Based on customer demand and market feedback Cressida plans to introduce additional actions for these modules as well as introducing new product modules for Auditing, Accounting, Encryption, Workload Simulation.

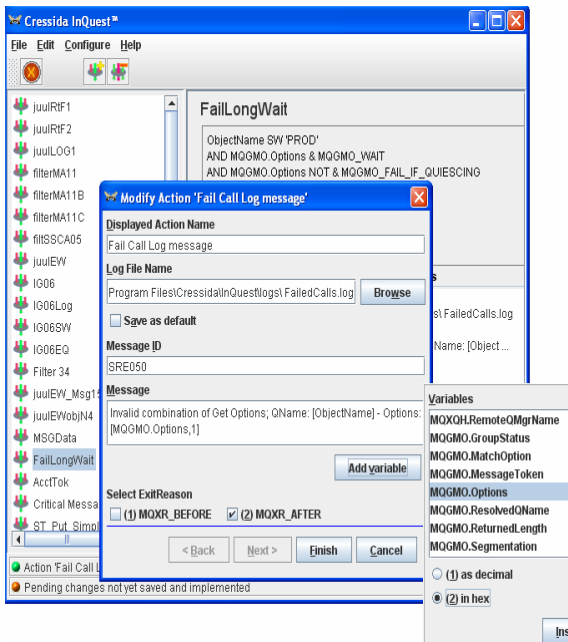
MBA: Message Based Alerting module

The Message Based Alerting module allows the user to add entries in a log file or to generate alerts based on any MQ API call that meets criteria defined by the user.

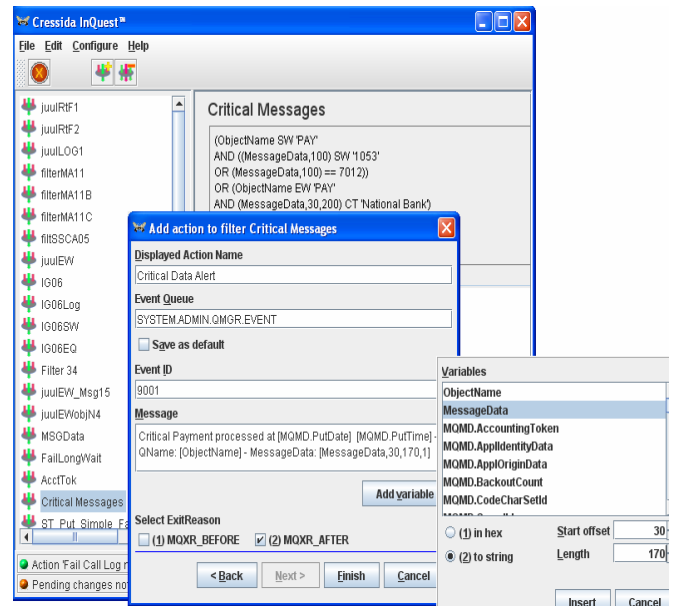
The alerts, i.e. actions, can be any of the following, either alone or in combination

1. Write an entry to a specified log file
2. Send an SNMP trap to a specified host and community with specified severity
3. Put an MQ Event message to a specified queue

The message included in the above alerts can be made up of user defined text and any of the WMQ fields or parts of the fields that are available for defining the filter criteria.



Write a message to a log file to record the fact that InQuest failed a GET with invalid options.



Create an MQ Event to signal the processing of a critical payment message.

Key Benefits of Message Content Based Alerting:

- Create selective audit or trace logs using any of the aforementioned fields as filter criteria
- Raise alerts based on, and including, real business data

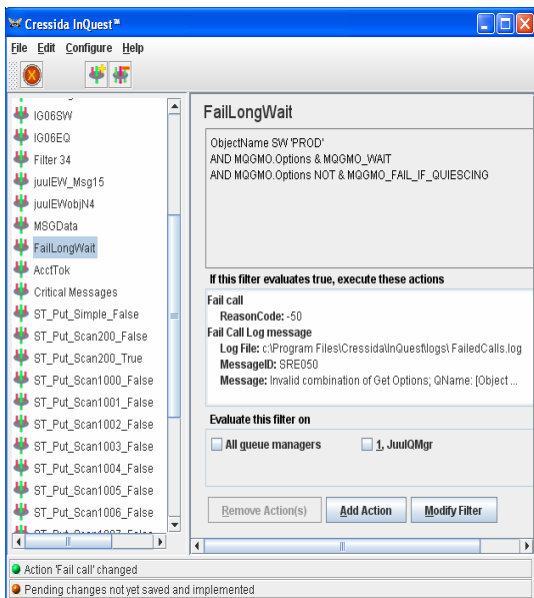
- Detect critical and/or “abnormal” application conditions in real-time
- Extend the current WMQ metric based monitoring tools to report “message content based” alerts and events
- Create true business application views in your monitoring dashboard
- Limit the volume of output data by including only relevant attributes and/or actual message data in the log files or alerts

SRE: Standards and Rules Enforcer Module

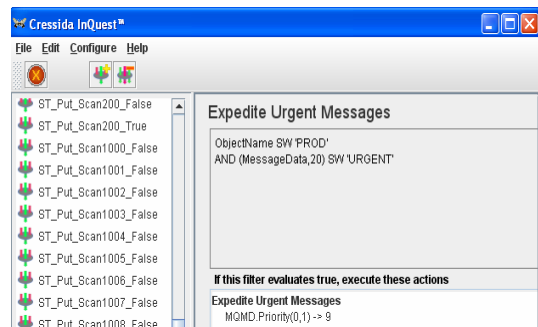
The Standards and Rules Enforcer module allows users to modify MQPUT, MQPUT1, MQGET and MQOPEN calls so that they conform to defined policies and required standards.

The actions can be any of the following either alone or in combination

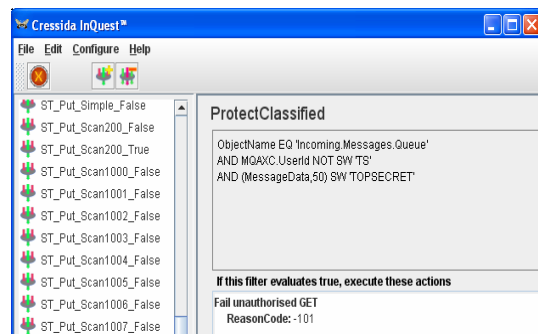
1. Fail Call: the call will not be processed by WMQ, and the application will receive MQCC_FAILED and a user defined value specified for MQRC
2. Set Field: change the value of one or more fields , e.g. set MQMD Priority to 5
3. Set Option: add or remove a flag or an option, without affecting other options that may be set



Fail a GET with WAIT if “Fail if Quiescing” is not also set



Make sure that “urgent” messages are sent with priority 9



Only allow access to classified users

Key Benefits of Standards and Rules Enforcer:

- Enforce “best practices” and site standards; non-conforming calls can either be dynamically corrected or rejected
- Change call attributes based on “message context” , e.g. raise or lower the message priority depending on the customer Id value in the message
- Implement security at the individual message level

MBR: Message Based Replicator Module

The Message Based Replicator Module allows the user to selectively replicate messages based on criteria defined within a filter.

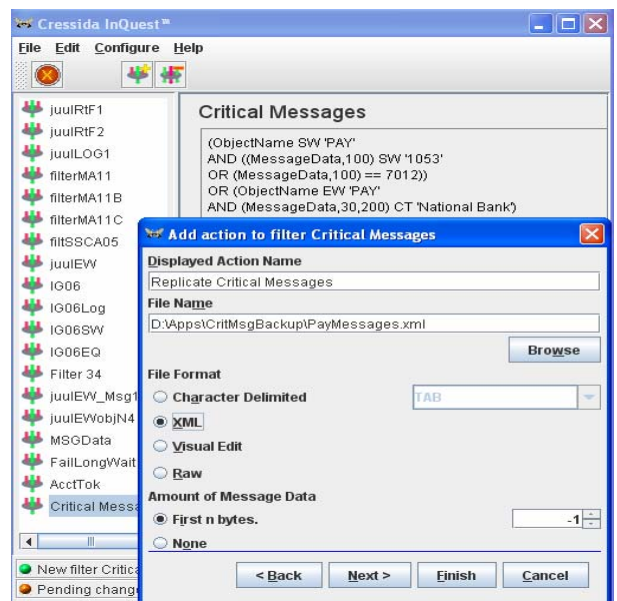
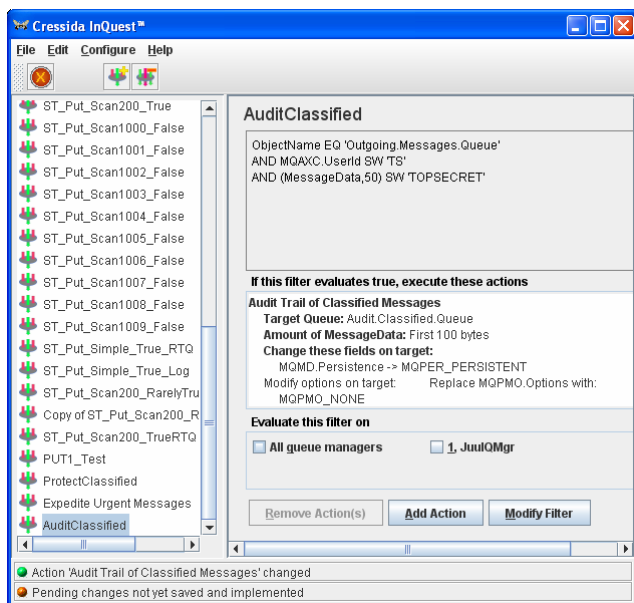
The actions can be any of the following either alone or in combination

1. Replicate to Queue
2. Replicate to File

The replicated data can include all or only part of the original message data so that the replication output can be limited to contain only the essential information.

Optionally, you can also use imbedded **setField** and **SetOption** actions to change the value of one or more attributes before replication takes place. This is extremely useful when replicating to Queue, when some attributes – if left unaltered – could potentially interfere with the original application, e.g. some of the Report options, or render the replicated messages unusable, e.g. Expiry. Note that the content of the actual message data cannot be altered.

When replicating to file, several output formats are available depending on your intended use of the replicated data.



Key Benefits of Message Content Based Replication:

- Selectively replicate messages based on content
- Limit the replicated data to the essential information
- Replicate data to the medium or format that best suits your needs

Cressida InQuest™ for WebSphere® MQ.

A flexible, configurable, powerful MQ API Exit tool from Cressida Technology Ltd.