

Product Overview

- developerWorks®, SupportPac, MQSeries.net™

**InQuest™ & SynQuest™ family of
Solutions for WebSphere® MQ**

InQuest Family Design Goal

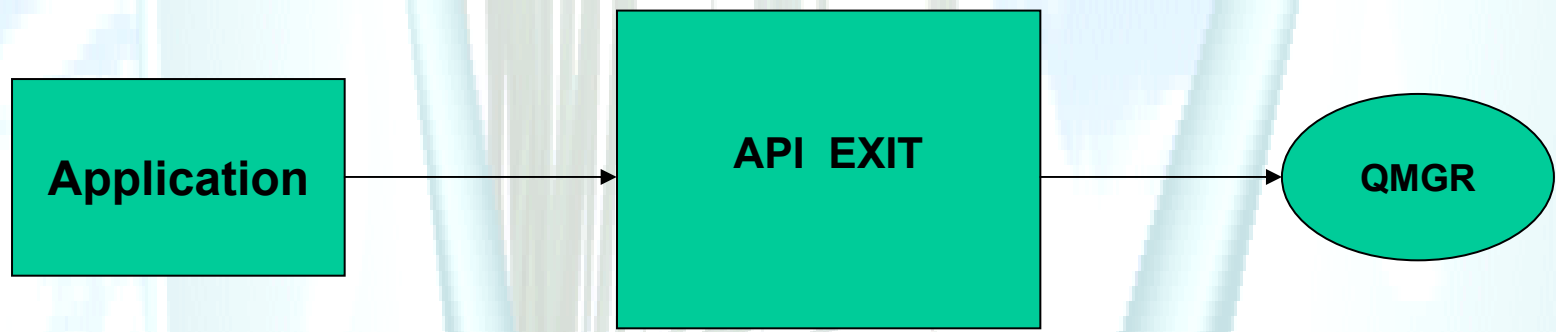
- To build a completely configurable very fast and high-performance tool that could help users professionally utilize the API-exit technology without worrying about coding, gotchas, performance, support,
- Requirements:
 - Work on all call types.
 - Both Before and After MQI
 - Filter on anything that's available within the exit
 - Take actions based on evaluation
 - Actions both within, and outside of MQ
 - Industrial Strength, High Performing, Fully Vendor Supported
Filtering, Replicator, Logger, Synchronizer, MirrorQ,

InQuest™ for WebSphere® MQ

- A user-configurable tool that utilises the power and functionality of the WMQ API Exit to intercept WMQ API calls that meet certain criteria (as specified in **Filters**), and to take user specified actions (as defined in **Actions**). A product that will provide fully supported easy to use, filtered, no coding needed, high performance access to MQ calls and automatically take actions to do many things authorized users want done.

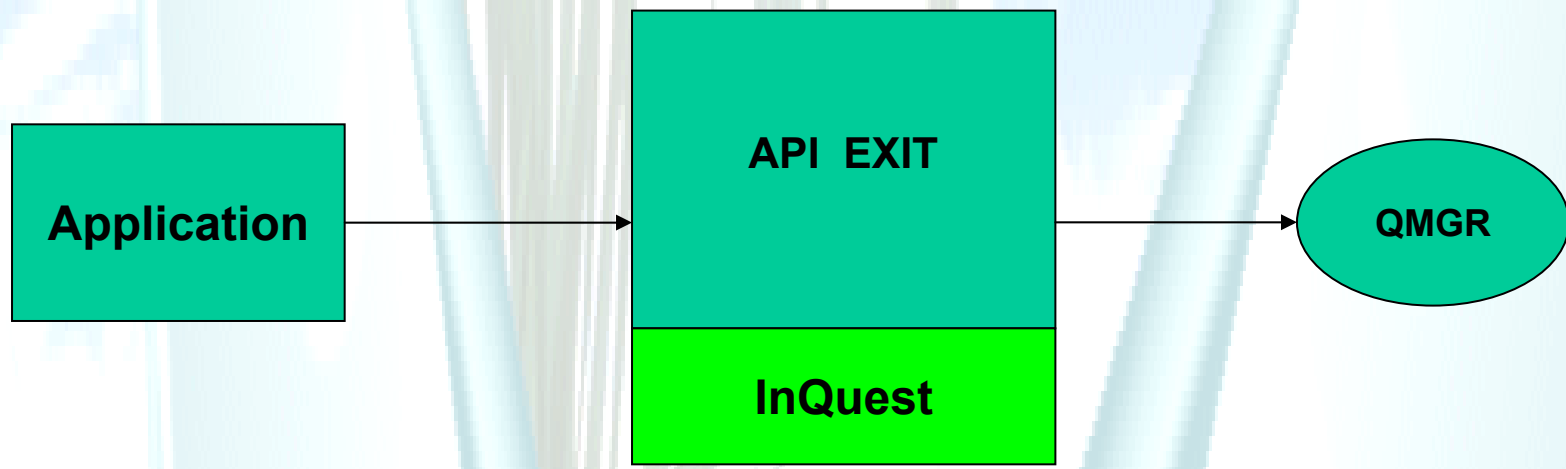
API Exit Basics

- An *API exit* is a program module that monitors or modifies the function of MQI calls.



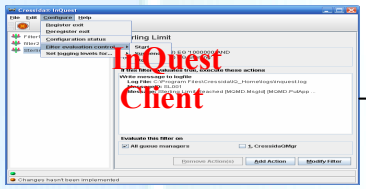
API exits let you write code that changes the behaviour of WebSphere MQ API calls, such as MQPUT and MQGET, and then insert that code immediately before or immediately after those calls.

The product family....



As part of the API Exit InQuest can intercept potentially every MQ call and evaluate the options, values and data in that call.

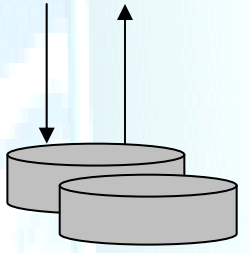
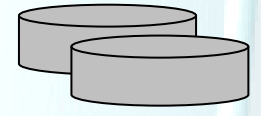
Architecture



Implement

InQuest Agent

Configuration Files



Repository

Command Processor

Shared Memory Server

- SMA
- SMA
- SMA

WebSphere MQ Application

InQuest API

- Intercept
- Filter/Evaluate
- Action
- InQuest API EXIT

MBA

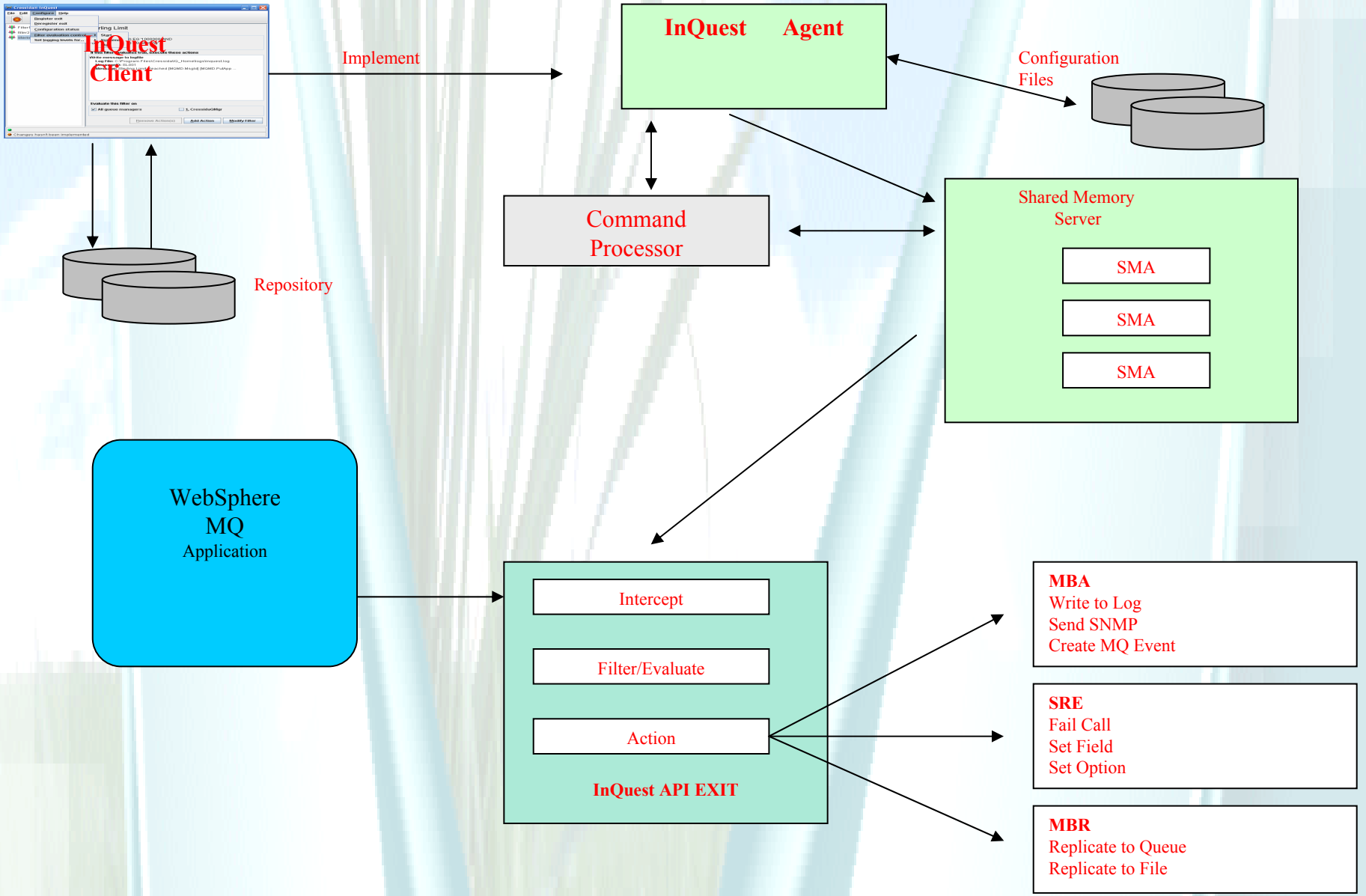
- Write to Log
- Send SNMP
- Create MQ Event

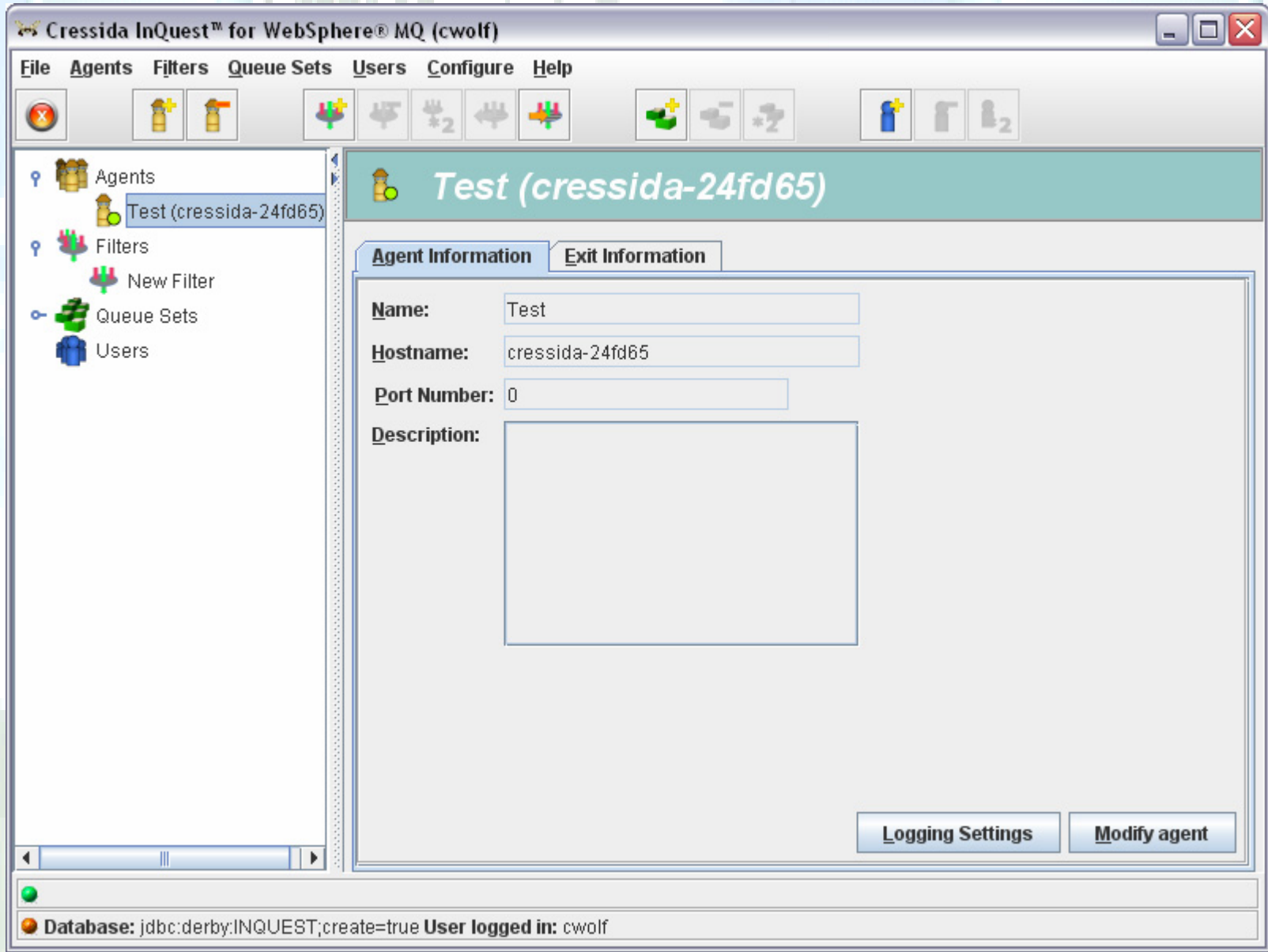
SRE

- Fail Call
- Set Field
- Set Option

MBR

- Replicate to Queue
- Replicate to File





The screenshot shows the Cressida InQuest™ for WebSphere® MQ (cwolf) application window. The title bar includes the application name and standard window controls. The menu bar contains File, Agents, Filters, Queue Sets, Users, Configure, and Help. The toolbar below the menu bar contains various icons for agent management, such as adding, deleting, and refreshing agents. On the left, a tree view shows the hierarchy: Agents (selected), Test (cressida-24fd65), Filters, New Filter, Queue Sets, and Users. The main area displays the configuration for the selected agent, 'Test (cressida-24fd65)'. It has two tabs: 'Agent Information' (active) and 'Exit Information'. The 'Agent Information' tab contains the following fields:

- Name:** Test
- Hostname:** cressida-24fd65
- Port Number:** 0
- Description:** (empty text area)

At the bottom right of the main area are two buttons: 'Logging Settings' and 'Modify agent'. A status bar at the bottom left shows a green indicator and the text: 'Database: jdbc:derby:INQUEST;create=true User logged in: cwolf'.

Filtering

Filters defined via GUI Interface for

PUT
PUT1
OPEN
GET

Fields available depend on call type
but can include...

MQMD
MQOD
MQPMO
MQAXC
MQDLH
MQXQH
Message Data
LocalTime

Add New Filter

Filter Name: New Filter(1)

Field: LocalTime

Modifiers:
 DayOfWeek TimeOfDay

Starting at: 0 Ending at: 0

Operator:
 NOT BETWEEN

Value:
User Defined 1;5

Resulting Filter Expression:
DAYOFWEEK(LocalTime) BETWEEN (1;5)

Buttons: Add, Update, Delete

Resulting Filter Expression:
DAYOFWEEK(LocalTime) BETWEEN (1;5)

Buttons: Group, Ungroup, Up, Down, Logic

Bottom Buttons: < Back, Next >, Finish, Cancel

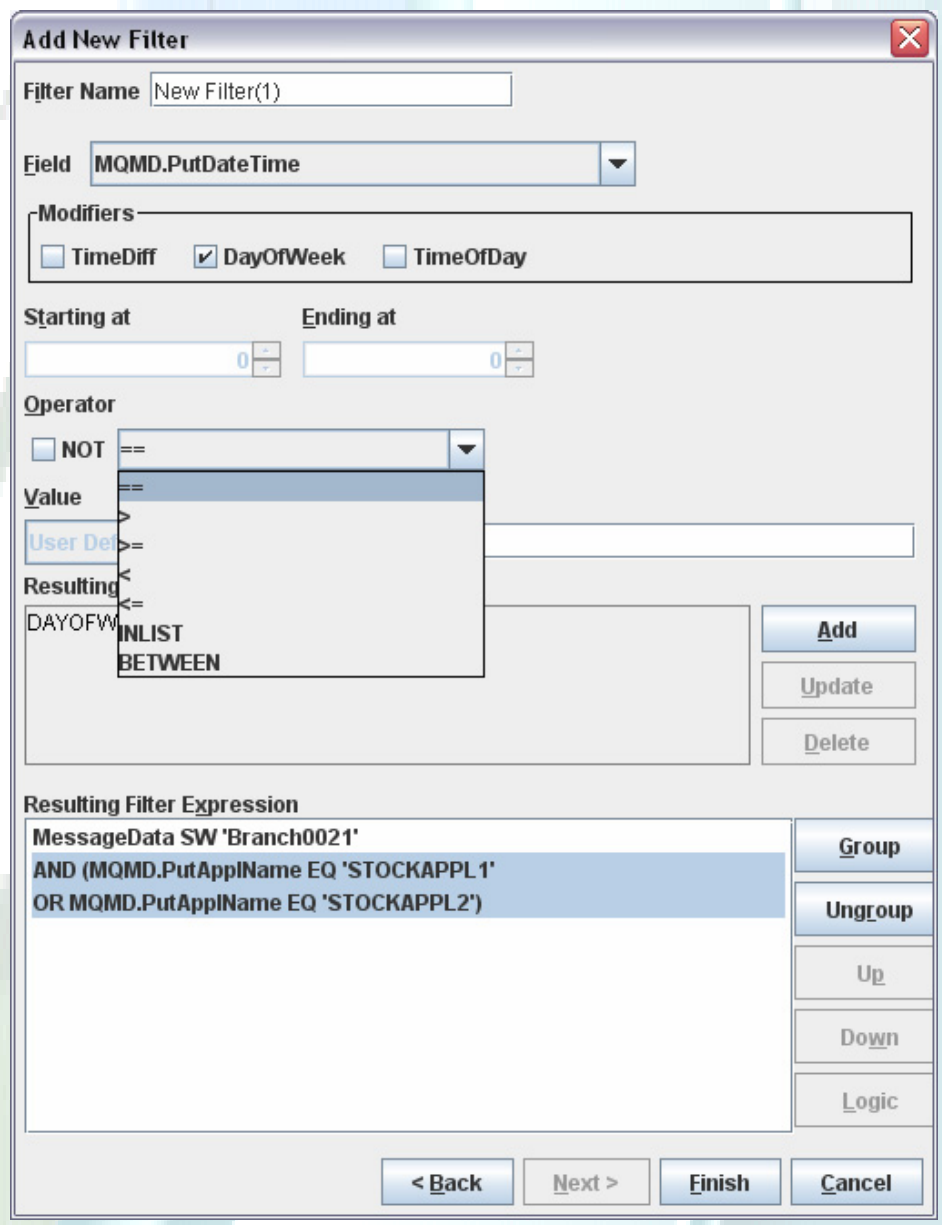
Filtering (cont'd)

Multiple Operators available

- ==
- >
- >=
- <
- <=
- EQ
- SW
- EW
- CT
- &
- INLIST
- BETWEEN

Expressions combined using

- AND
- OR
- AND NOT
- OR NOT
- GROUPING



Add New Filter

Filter Name: New Filter(1)

Field: MQMD.PutDateTime

Modifiers: TimeDiff DayOfWeek TimeOfDay

Starting at: 0 Ending at: 0

Operator: NOT ==

Value: User De

Resulting Filter Expression:

MessageData SW 'Branch0021'
AND (MQMD.PutAppName EQ 'STOCKAPPL1'
OR MQMD.PutAppName EQ 'STOCKAPPL2')

Buttons: Add, Update, Delete, Group, Ungroup, Up, Down, Logic, < Back, Next >, Finish, Cancel

Actions

InQuest is modular.

- Modules currently available :

MBA

Message Based Alerting

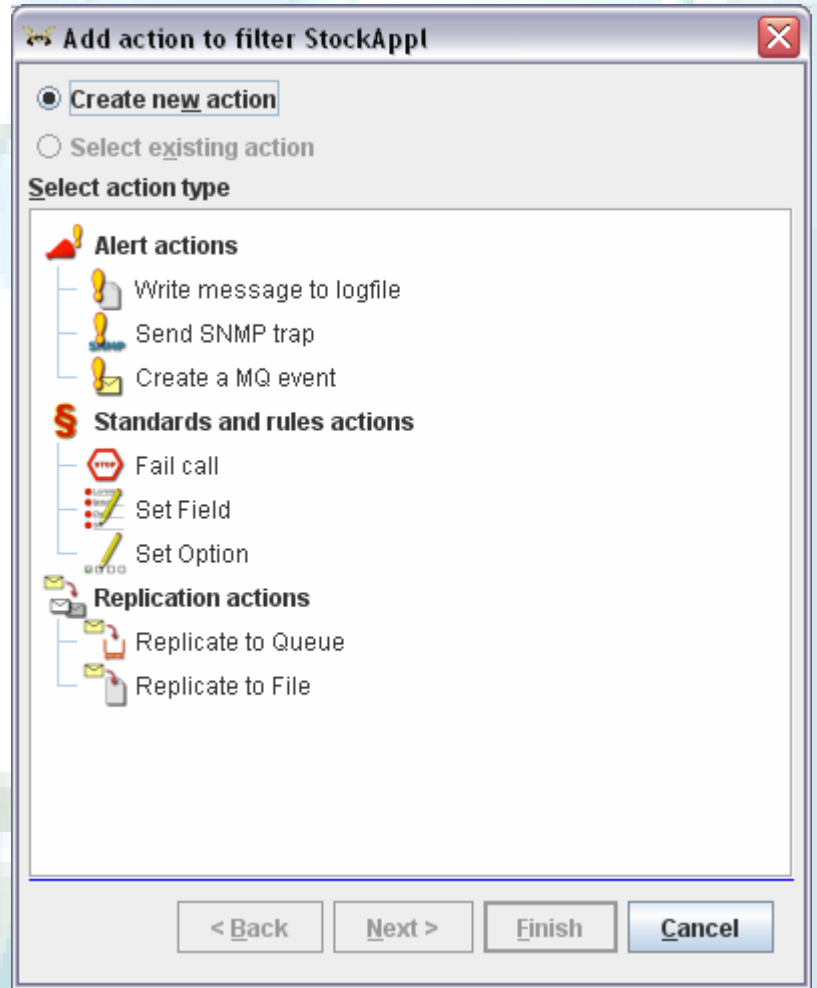
SRE

Standards and Rules Enforcer

MBR

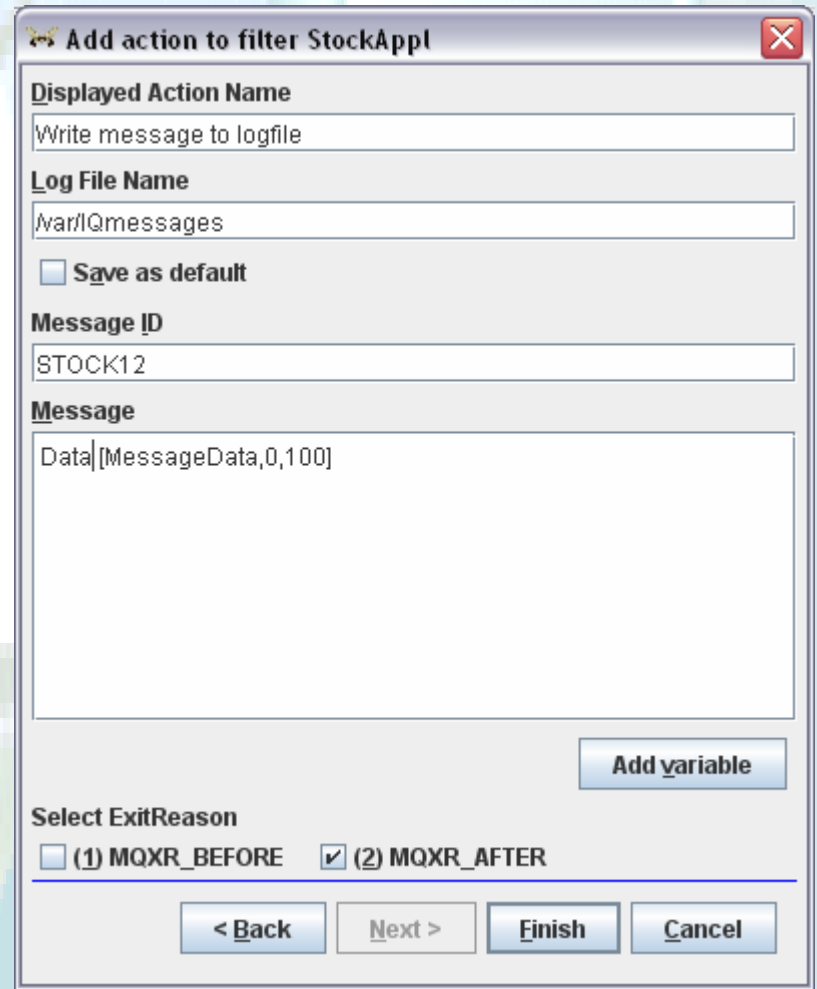
Message Based Replicator

Logger and Synchronizer



Message Based Alerting

- Three actions available
 - Write message to logfile
 - Put an MQ Event message to a specified queue
 - Both may include original msg data and any filterable field as substituted variable
 - Send SNMP Alert
 - to a specified host and community with specified severity



The screenshot shows a dialog box titled "Add action to filter StockAppl". It contains several input fields and checkboxes:

- Displayed Action Name:** Write message to logfile
- Log File Name:** /var/1Qmessages
- Save as default
- Message ID:** STOCK12
- Message:** Data|[MessageData,0,100]
- Add variable** button
- Select ExitReason:** (1) MQXR_BEFORE, (2) MQXR_AFTER
- Navigation buttons: < Back, Next >, Finish, Cancel

Examples where MBA may be used

- An unexpected userid generated a message
- An unexpected program (executable name)...
- A transaction has a value of over one million £
- An application is causing an MQ resource shortage
- An unexpected format type is used.
- A message is larger than 500K

MBA is ideal if you want alarms based on business information within seconds

Standards and Rules Enforcer

- Three actions available
 - Fail Call
 - Fail the call and set RC and CC
 - Set Field
 - Set values for specified fields to comply with standards
 - Set Option
 - Change option settings to comply with standards

The screenshot shows a dialog box titled "Add action to filter StockAppl". It contains the following fields and controls:

- Displayed Action Name:** A text box containing "Set Field".
- Select field:** A dropdown menu with "MQMD.UserIdentifier" selected.
- New value:** A dropdown menu with "User Defined" selected and a text box containing "ApplControl".
- Start offset:** A spin box with the value "0".
- Buttons:** "Add to list" (top right), "Remove" (bottom right), "< Back" (bottom left), "Next >" (bottom left), "Finish" (bottom right), and "Cancel" (bottom right).
- List of changes:** A text area containing:
MQMD.Persistence -> MQPER_PERSISTENT
MQMD.UserIdentifier -> 'ApplControl'

SRE Uses

- Control access based on message content
- Enforce good practices or programming standards

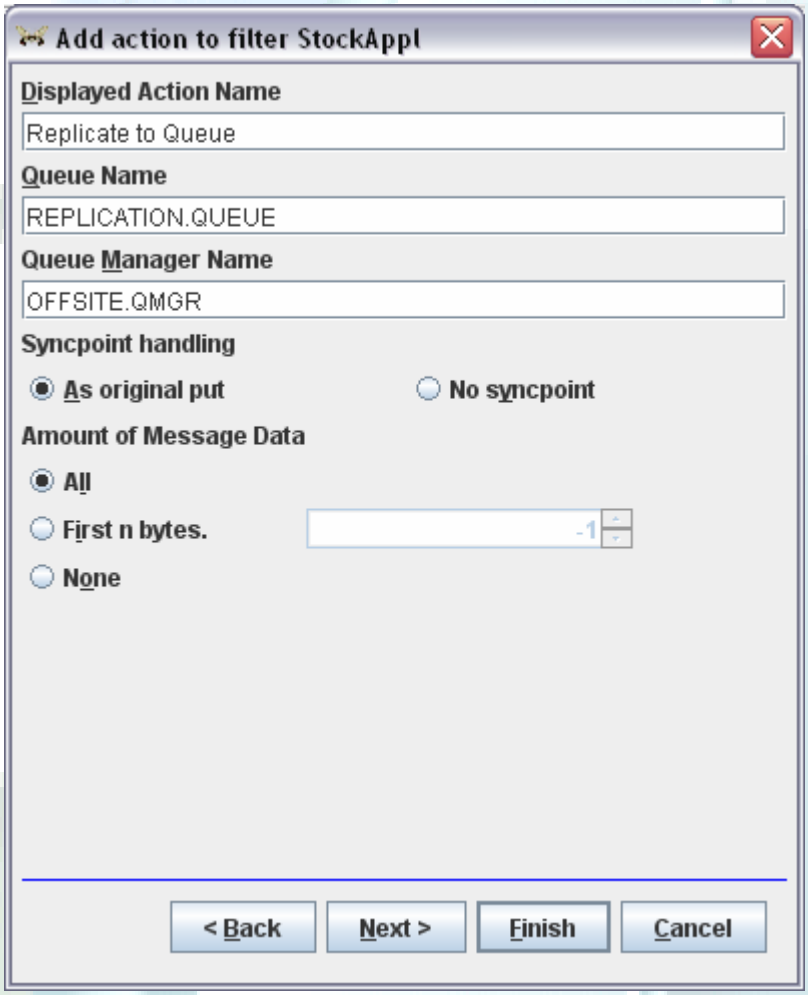
E.g.:

- MQPMO_SET_IDENTITY_CONTEXT is forbidden
- A dynamic queue name must start with ABC
- If an MQGET uses MQGMO_WAIT, the WaitInterval can not be greater than 5000, nor can it have the value MQWI_UNLIMITED, unless the option MQGMO_FAIL_IF_QUIESCING is used.
- Expiring messages are not allowed
- All messages on queue XYZ must be persistent

MBR

- Two actions available
 - Replicate to Queue
 - Copy and Log the message to another queue within the QMGR
 - Replicate to File
 - Copy and Log the message to a file, specifying format
 - Character delimited
 - XML
 - Raw

The replicated data can include all or only part of the original message data



The screenshot shows a dialog box titled "Add action to filter StockAppl". It contains several fields and options:

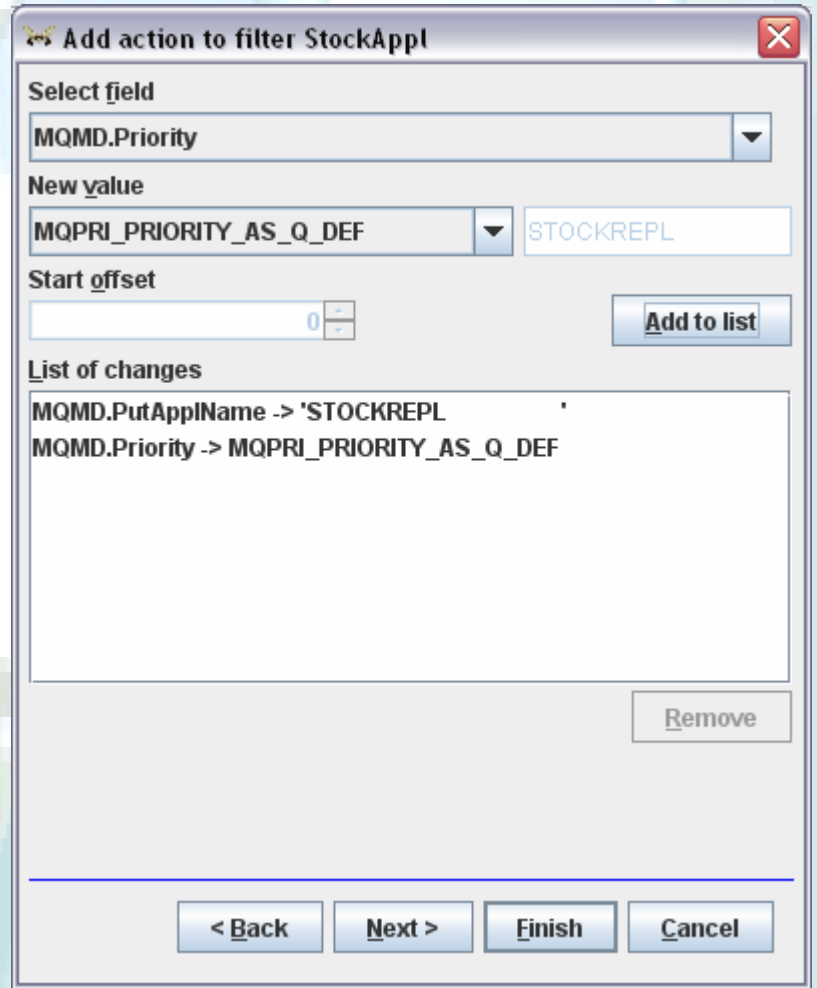
- Displayed Action Name:** Replicate to Queue
- Queue Name:** REPLICATION.QUEUE
- Queue Manager Name:** OFFSITE.QMGR
- Syncpoint handling:** As original put, No syncpoint
- Amount of Message Data:** All, First n bytes. (with a text box containing "-1"), None

At the bottom, there are four buttons: "< Back", "Next >", "Finish", and "Cancel".

MBR Uses

Optionally imbedded **setField** and **SetOption** actions may be used to change the value of one or more attributes of the replicated data

Note that the content of the original message data cannot be altered

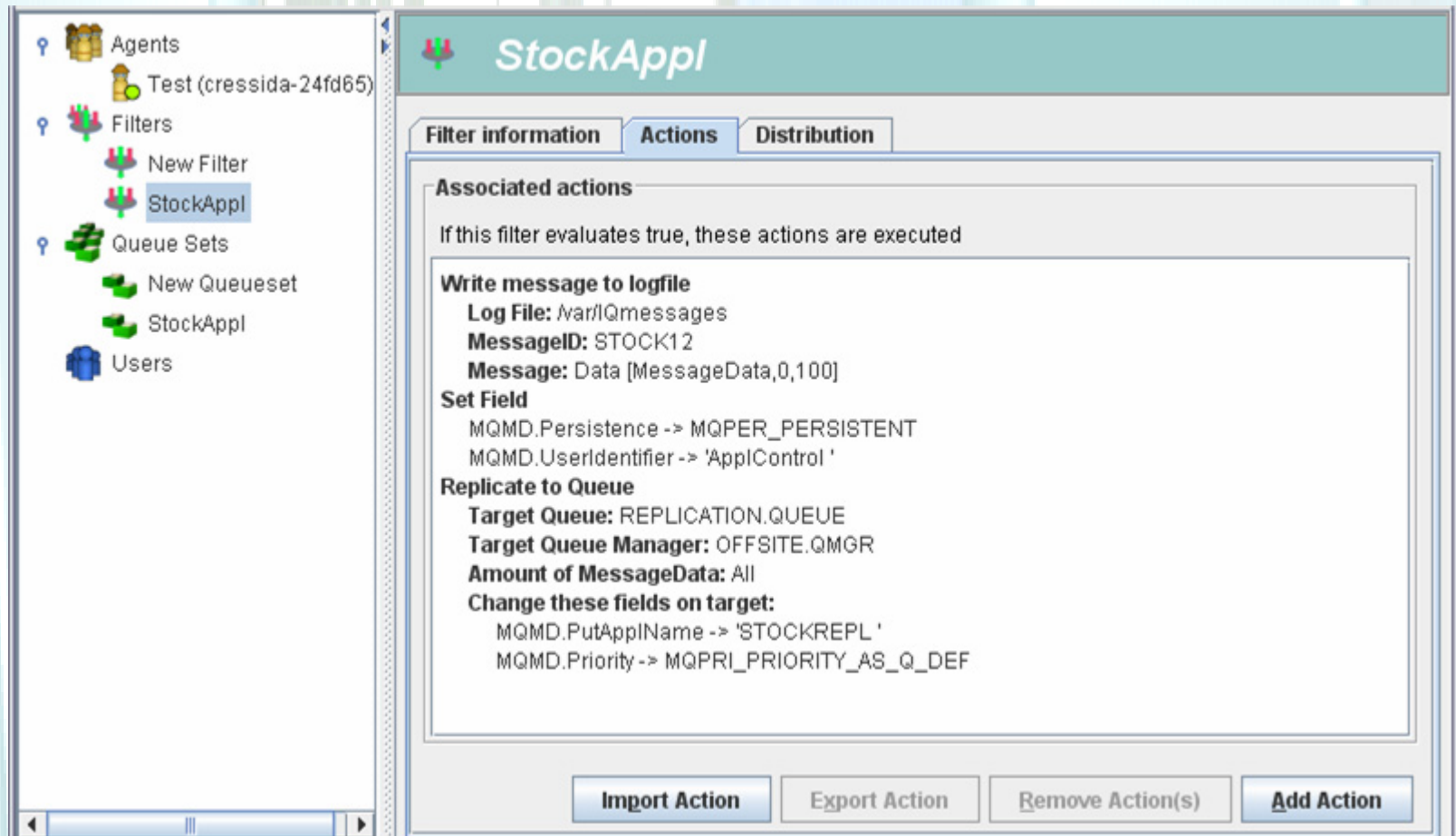


The screenshot shows a dialog box titled "Add action to filter StockAppl". It contains the following fields and controls:

- Select field:** A dropdown menu with "MQMD.Priority" selected.
- New value:** A dropdown menu with "MQPRI_PRIORITY_AS_Q_DEF" selected and a text input field containing "STOCKREPL".
- Start offset:** A numeric input field with "0" and increment/decrement buttons.
- Buttons:** "Add to list" (blue), "Remove" (grey), "< Back" (blue), "Next >" (blue), "Finish" (blue), and "Cancel" (blue).
- List of changes:** A text area containing:
MQMD.PutAppName -> 'STOCKREPL'
MQMD.Priority -> MQPRI_PRIORITY_AS_Q_DEF

Multiple Actions

- Any number of **Actions** can be associated with a filter and can be of different types
- All actions associated with all filters that evaluates to true for a specific call are executed



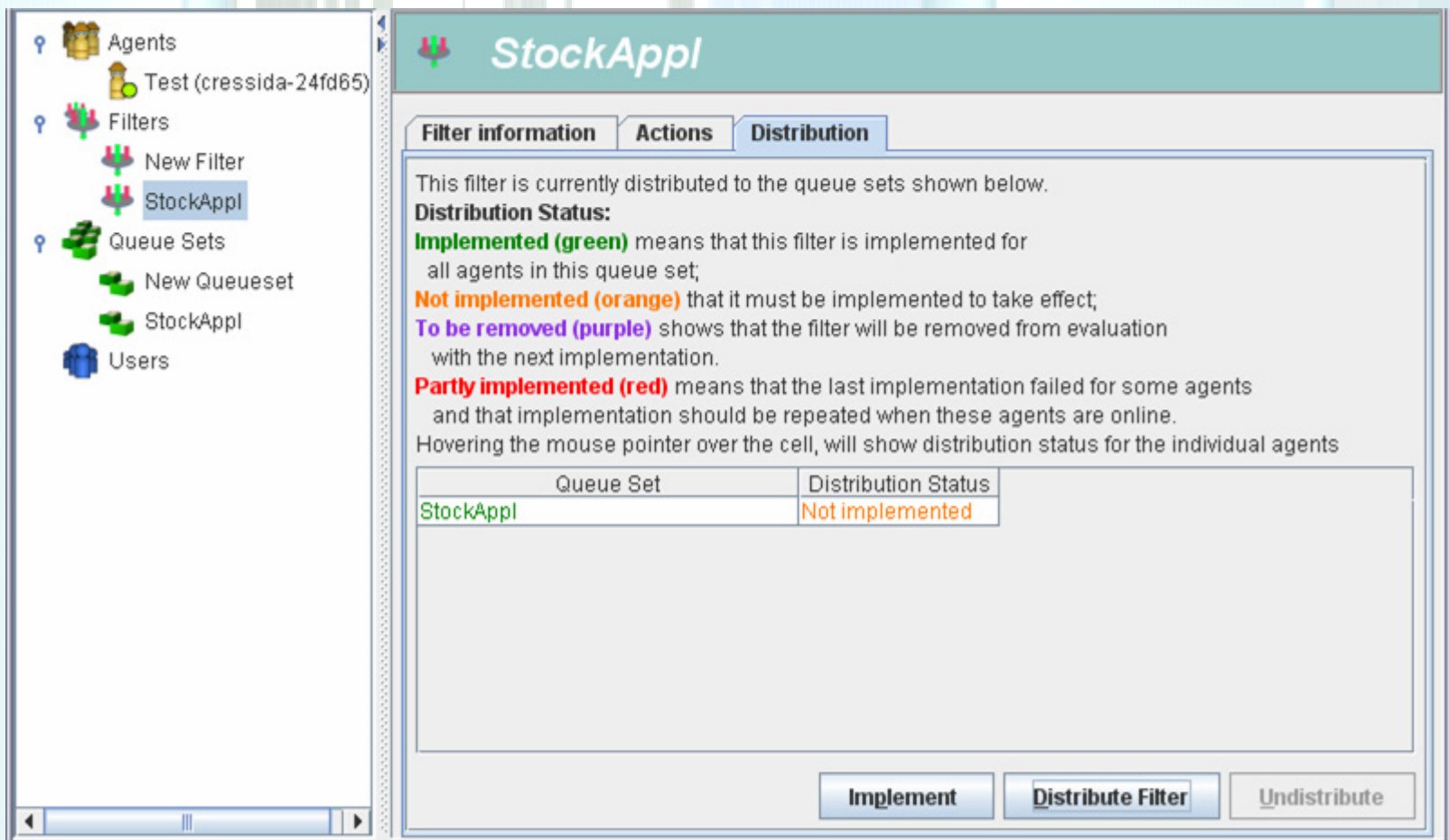
The screenshot shows the configuration interface for 'StockAppl'. The left sidebar contains a tree view with categories: Agents (Test (cressida-24fd65)), Filters (New Filter, StockAppl), Queue Sets (New Queueset, StockAppl), and Users. The main window has tabs for 'Filter information', 'Actions', and 'Distribution'. The 'Actions' tab is active, displaying a list of associated actions under the heading 'Associated actions'. Below this heading is the text 'If this filter evaluates true, these actions are executed'. The actions listed are:

- Write message to logfile**
 - Log File: /var/lqmessages
 - MessageID: STOCK12
 - Message: Data [MessageData,0,100]
- Set Field**
 - MQMD.Persistence -> MQPER_PERSISTENT
 - MQMD.UserIdentifier -> 'ApplControl '
- Replicate to Queue**
 - Target Queue: REPLICATION.QUEUE
 - Target Queue Manager: OFFSITE.QMGR
 - Amount of MessageData: All
- Change these fields on target:**
 - MQMD.PutAppName -> 'STOCKREPL '
 - MQMD.Priority -> MQPRI_PRIORITY_AS_Q_DEF

At the bottom of the window are four buttons: 'Import Action', 'Export Action', 'Remove Action(s)', and 'Add Action'.

Distribution

- Filters are distributed to queue sets
- Implements transparently adds queue names to filter definition



The screenshot shows the 'StockAppl' application window. On the left is a tree view with categories: Agents (Test (cressida-24fd65)), Filters (New Filter, StockAppl), Queue Sets (New Queueset, StockAppl), and Users. The main area has three tabs: Filter information, Actions, and Distribution. The Distribution tab is active and contains the following text:

This filter is currently distributed to the queue sets shown below.

Distribution Status:

- Implemented (green)** means that this filter is implemented for all agents in this queue set;
- Not implemented (orange)** that it must be implemented to take effect;
- To be removed (purple)** shows that the filter will be removed from evaluation with the next implementation.
- Partly implemented (red)** means that the last implementation failed for some agents and that implementation should be repeated when these agents are online.

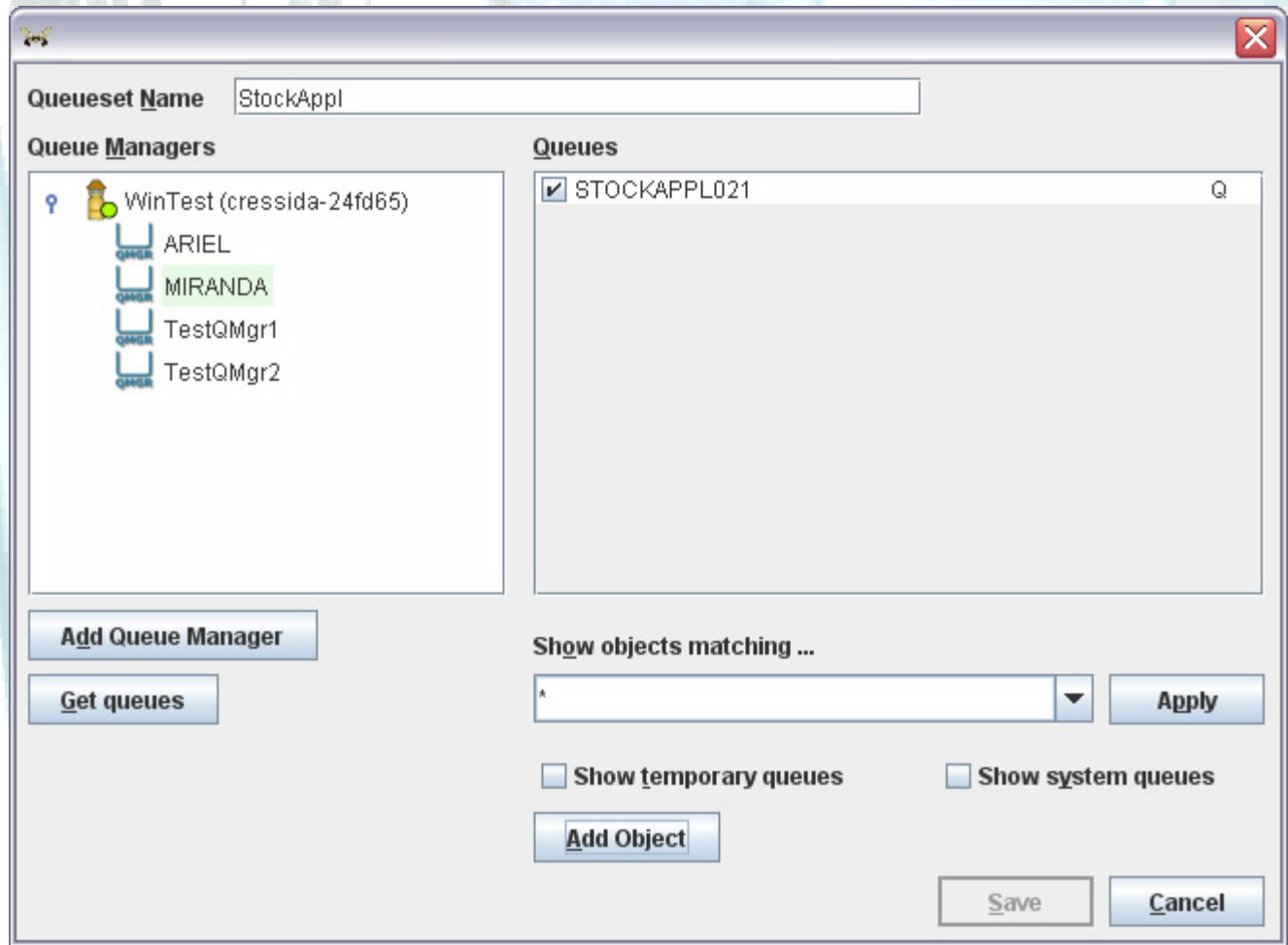
Hovering the mouse pointer over the cell, will show distribution status for the individual agents

Queue Set	Distribution Status
StockAppl	Not implemented

At the bottom of the window are three buttons: Implement, Distribute Filter, and Undistribute.

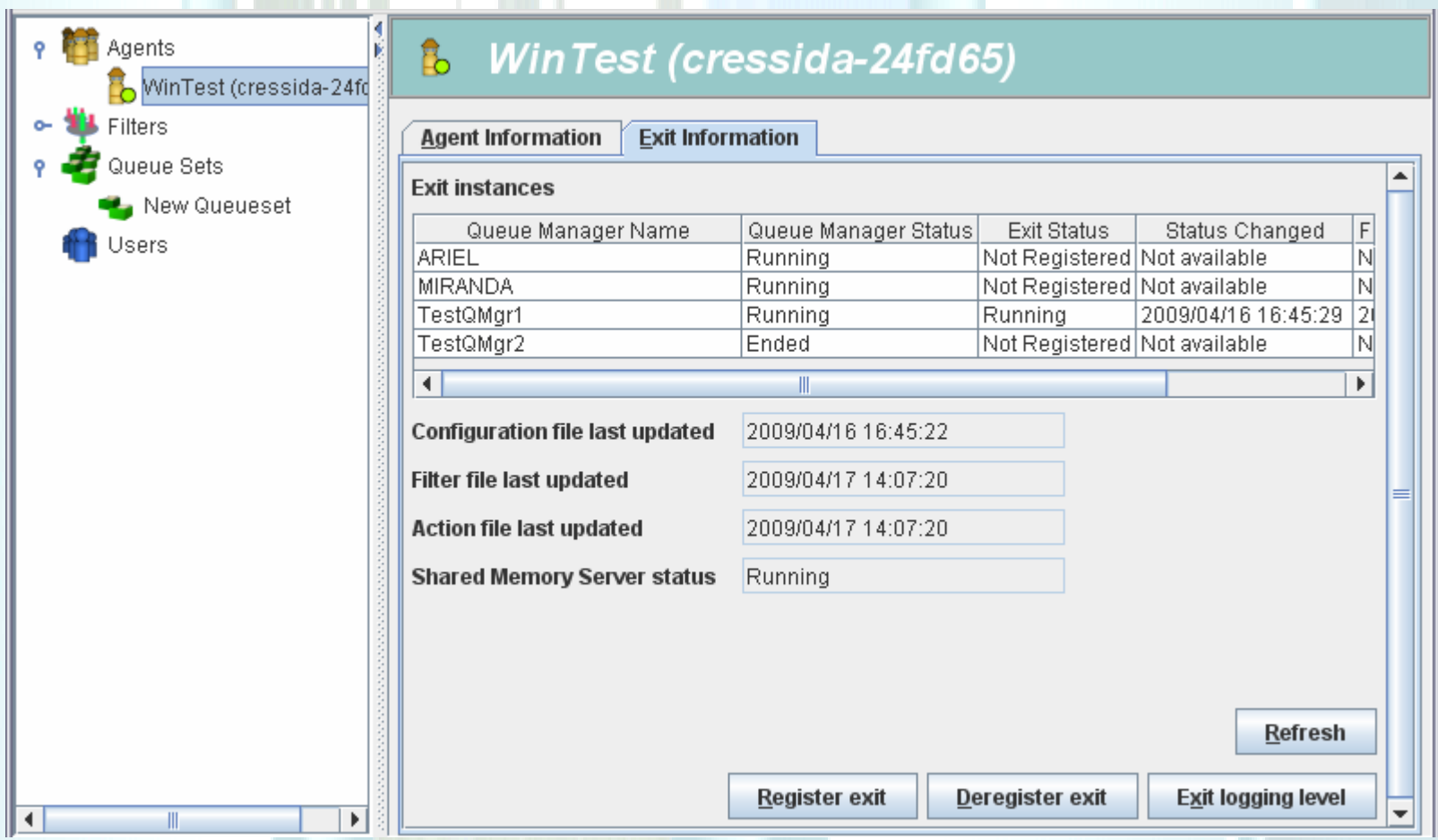
Queue Set

- A list of 1-many queues on 1-many queue managers



Agents

- Remote control of the API Exit
- Target for implementation



WinTest (cressida-24fd65)

Agents
WinTest (cressida-24fd65)
Filters
Queue Sets
New Queueset
Users

Agent Information | **Exit Information**

Exit instances

Queue Manager Name	Queue Manager Status	Exit Status	Status Changed	F
ARIEL	Running	Not Registered	Not available	N
MIRANDA	Running	Not Registered	Not available	N
TestQMgr1	Running	Running	2009/04/16 16:45:29	21
TestQMgr2	Ended	Not Registered	Not available	N

Configuration file last updated: 2009/04/16 16:45:22

Filter file last updated: 2009/04/17 14:07:20

Action file last updated: 2009/04/17 14:07:20

Shared Memory Server status: Running

Refresh

Register exit | Deregister exit | Exit logging level

InQuest Summary

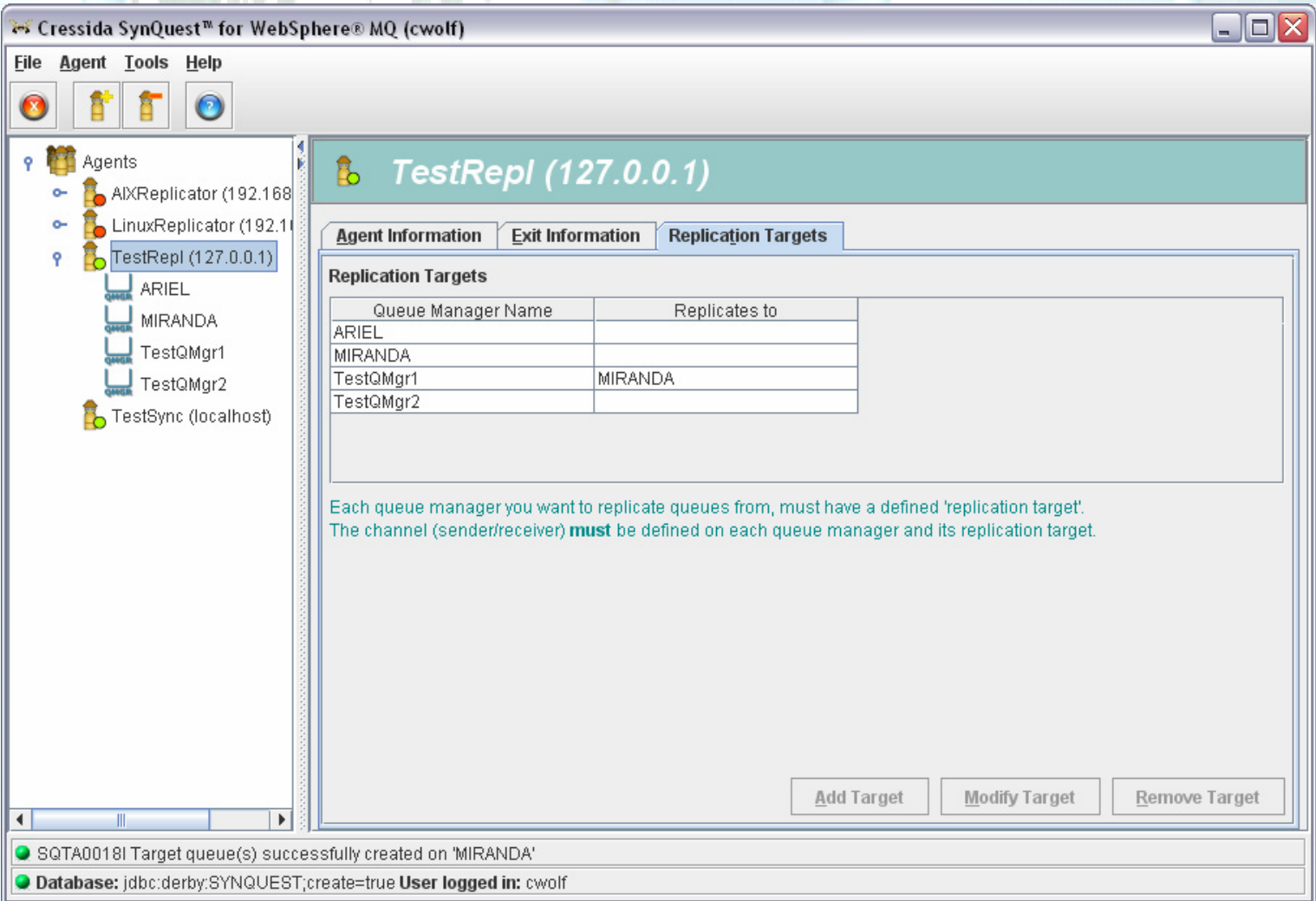
- Action oriented, flexible, no coding needed user-configurable MQ API handling family of tools
- Uses reliable existing IBM supplied technology
- Modular, expandable functionality
 - Message Replication, Logging, MirrorQ, Synchronization, Enforce Standards, Content Based Alerts, ...
- No API coding skills required
- Easy to use graphical front end
- Precise filtering
- High-performance
- Cressida supplied and supported

SynQuest™ for WMQ

- Production Strength Multi-Platform MirrorQ Solution
- Disaster Recovery & Hot Site Application
- Synchronize Local and Offsite Queue Managers
- User Selective Source Queue Replication
- High-Performing, No User Coding needed Synchronizing API Exit Agent
- Utilizes Well Trusted IBM WMQ Transportation Engine
- Based on Cressida InQuest Highly Optimized API Technology

Replication Agent

- Handles the source queue managers



Cressida SynQuest™ for WebSphere® MQ (cwolf)

File Agent Tools Help

Agents

- AIXReplicator (192.168)
- LinuxReplicator (192.1
- TestRepl (127.0.0.1)**
 - ARIEL
 - MIRANDA
 - TestQMgr1
 - TestQMgr2
 - TestSync (localhost)

TestRepl (127.0.0.1)

Agent Information Exit Information **Replication Targets**

Replication Targets

Queue Manager Name	Replicates to
ARIEL	
MIRANDA	
TestQMgr1	MIRANDA
TestQMgr2	

Each queue manager you want to replicate queues from, must have a defined 'replication target'.
The channel (sender/receiver) **must** be defined on each queue manager and its replication target.

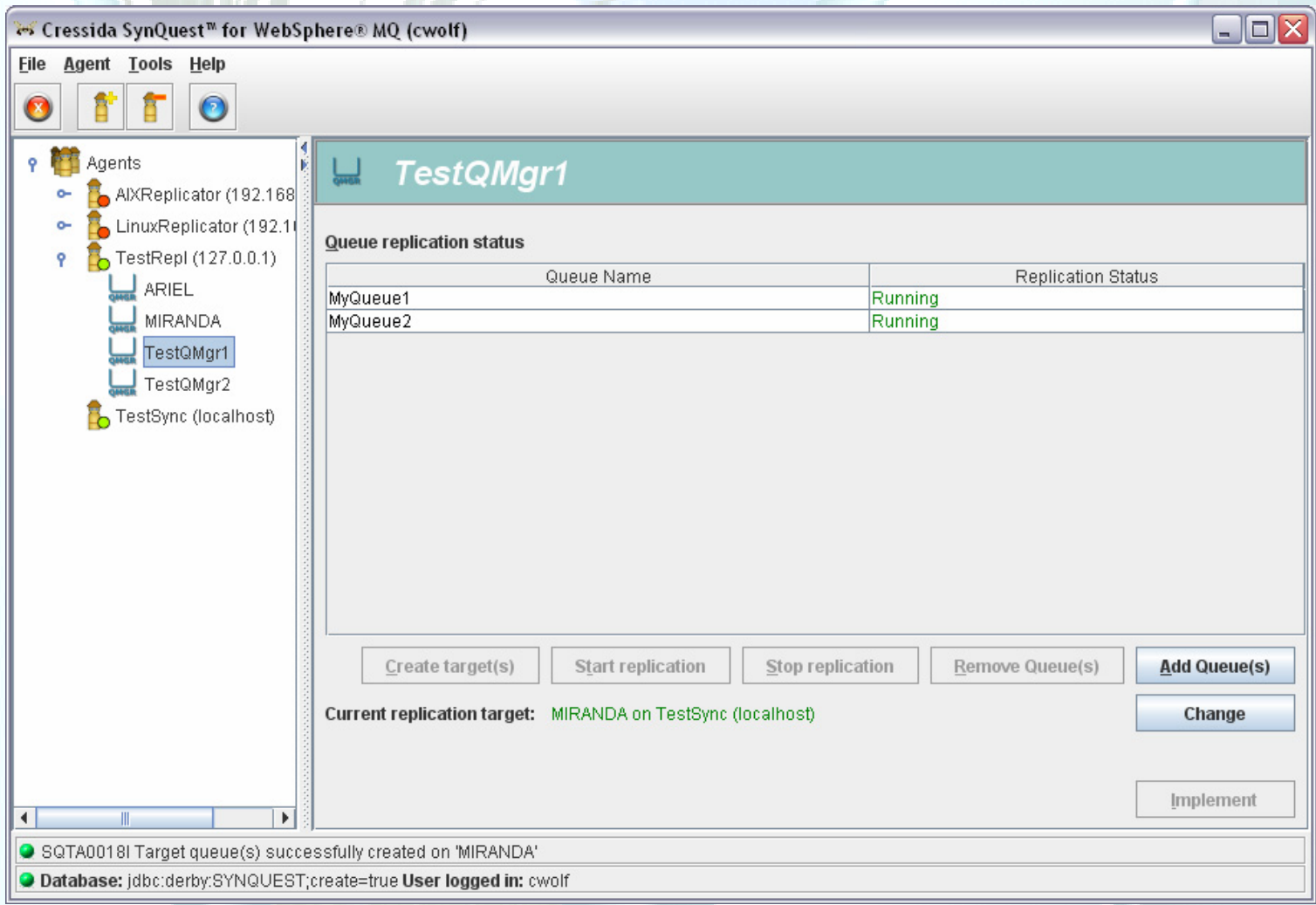
Add Target Modify Target Remove Target

● SQTA00181 Target queue(s) successfully created on 'MIRANDA'

● Database: jdbc:derby:SYNQUEST,create=true User logged in: cwolf

Source Queue Managers

- Defines which queues are replicated



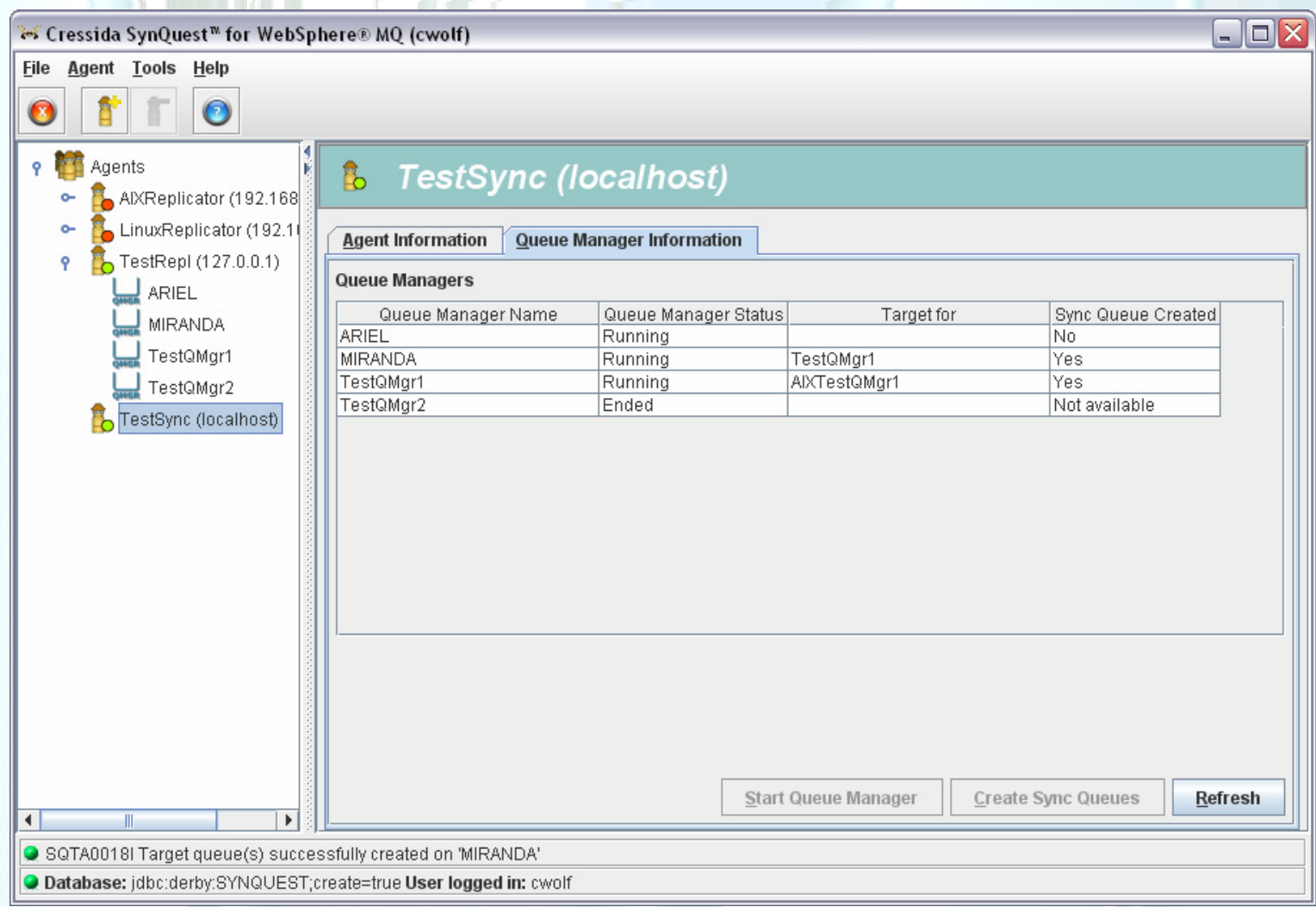
The screenshot shows the Cressida SynQuest™ for WebSphere® MQ (cwolf) application window. The interface includes a menu bar (File, Agent, Tools, Help), a toolbar with icons for stop, add, delete, and refresh, and a tree view on the left under 'Agents'. The tree view lists: AIXReplicator (192.1.68), LinuxReplicator (192.1...), TestRepl (127.0.0.1), ARIEL, MIRANDA, TestQMgr1 (selected), TestQMgr2, and TestSync (localhost). The main pane displays 'TestQMgr1' with a 'Queue replication status' table:

Queue Name	Replication Status
MyQueue1	Running
MyQueue2	Running

Below the table are buttons: Create target(s), Start replication, Stop replication, Remove Queue(s), Add Queue(s), Change, and Implement. The 'Current replication target' is 'MIRANDA on TestSync (localhost)'. A status bar at the bottom shows: SQTA0018I Target queue(s) successfully created on 'MIRANDA' and Database: jdbc:derby:SYNQUEST;create=true User logged in: cwolf.

Synchronizing Agent

- Handles the target queue manager



Cressida SynQuest™ for WebSphere® MQ (cwolf)

File Agent Tools Help

Agents

- AIXReplicator (192.168)
- LinuxReplicator (192.1
- TestRepl (127.0.0.1)
 - ARIEL
 - MIRANDA
 - TestQMgr1
 - TestQMgr2
- TestSync (localhost)**

TestSync (localhost)

Agent Information Queue Manager Information

Queue Managers

Queue Manager Name	Queue Manager Status	Target for	Sync Queue Created
ARIEL	Running		No
MIRANDA	Running	TestQMgr1	Yes
TestQMgr1	Running	AIXTestQMgr1	Yes
TestQMgr2	Ended		Not available

Start Queue Manager Create Sync Queues Refresh

● SQTA0018I Target queue(s) successfully created on 'MIRANDA'

● Database: jdbc:derby:SYNQUEST,create=true User logged in: cwolf

SynQuest Summary

- Action oriented, no coding needed MirrorQ & Disaster Recovery Functionality
- Uses reliable existing IBM MQ technology
- No MQ API coding skills required
- Easy to use graphical front end
- High-performance & Industrial Strength
- Multi-Platform enabled
 - Windows XP, 2003 (32 bits)
 - AIX 5.3 and 6.x, Solaris 8 & 9, HP-UX 11i V2 & V3
 - 2.6 based RedHat and Suse Linuxes ...