

This release is focused on performance at scale: managing 1000s of users on a 24 hour global basis has been greatly simplified with the introduction of our web balancer and web manager components.

These tools provide out-of-the-box dynamic expansion of web nodes without relying on third party tools. Additionally, the storage and maintenance of user preferences and dashboards have been centralized, shifting away from the need to manually maintain these settings across servers and regions.

Significantly, the AMI web server can now handle twice the number of live streaming users per node, through a combination of adaptive caching, deduplication, and smarter data compression/transmission techniques.

As always, we continue to push the boundaries of what can be done on the web through further improvements to our IDE and end user experience features.

New Components:

- WebBalancer –
A fully integrated load balancer for automatically routing and distributing users when multiple AMI Web Servers are deployed.
- WebManager –
Used to store dashboard and user preference settings. When running multiple web servers, dashboards and other settings can now be centrally managed. Also, methods to amiscrypt were added to allow developers to seamlessly read/write files remotely.

Optimization:

- Memory Reduction –
The Web Servers Memory model has been optimized for streaming data with a 50% reduction when multiple users are logged in, and the savings increase as multiple users are added. This is achieved by creating a single copy of the data, with user-specific overlays, etc.
- Smarter Snapshots & Auto-caching –
If a client makes multiple, overlapping subscriptions then a state-of-the-world request to the ami center can be skipped.
- MongoDB Adapter –
Faster conversion from JSON to tables, schema detection has been improved. Better logging. Support of select * with auto type detection/conversion.

End User Features:

- Support for hot keys, so that users/developers can set customized key-strokes to any type of action(s).
- Added visual feedback for user-click events for Dialog Buttons, Tabs, Top menu icons
- Window icons, and developer mode Big green buttons
- Fresh new look for front-end error dialog
- Tables support checkboxes for viewing and editing, and a specialized editor for numeric cells

Developer Tools:

- Quickly jump to documentation within script editor
- Quickly jump to method implementation within script editor
- Added additional metrics in the Object Browser tool for Datamodels Tab: Auto Run, Auto Requery, Conflate Query, Limit, Datasource, Timeout
- Object Browser tool now includes custom methods
- Object Browser tool stores most recent searches

AMIScript:

- Ability to directly invoke callbacks. Added a new call “Callback” and companion methods:
 - » getCallbacks() –
Returns all callbacks for a given object,
 - » getCallback(...) –
Return the callback based on name and param type
 - » invokeCallback(...) –
Execute a callback
- Added ChartPanelAxis: methods:
 - » getZoom, setZoom –
retrieve and set current zoom of a chart
 - » getVisibleOffset, setVisibleOffset –
set position of a zoomed in chart
 - » getVisibleLength – total size of chart, when zoomed in
 - » getVisibleMaxValue, getVisibleMinValue, setZoomByMinMaxValue
- Other Methods / Classes:
 - » splitIgnoreCase(...) –
split a string ignoring case of the delimiter
 - » Object class –
Base class that all objects in AMI extend Object
 - » Iterable class –
A class that can be iterated over with for-each loop
 - » session::getWindows() –
returns a list of all windows within the dashboard
 - » session::getWindowsMap() –
returns a map of all windows within the dashboard
 - » Window::getInnerPanel() –
returns the top level panel within dashboard window
 - » FileSystem::readBinaryFile –
Read contents of a file (if webmanager is used, will RPC to webmanager’s file system)
 - » FileSystem::writeBinaryFile –
Write contents of a file (if webmanager is used, will RPC to webmanager’s file system)
 - » Panel::onKeyEvent call back –
Intercept user key events
 - » FormCheckboxField class –
Represents a checkbox in a form
 - » FormTextareaField class –
Represents a textarea field in a form
 - » FormDivField class –
Represents a div field in a form
- Global variables can now be referenced inside relationships

New Options:

- `ami.web.http.slow.response.warn.log.request.size` - used to control the size/detail of messages logged during a slow response.
- `ami.amilog.stats.period` option - How often should the statistics of the AMI process be appended to `AmiLog.amilog`
- `Ami.web.show.wait.icon.after.duration` - How much time should lapse in the server before the user is presented with a wait/loading icon

Other:

- Replay mode can now be throttled using the `MAX_PER_SECOND` option to controlling rate of messages
- Added Real-time downstream mode (selected or all) option for panels
- Accessing files now honors symbolic links

If you have any questions about these new features or would like a demo, please email info@3forge.com.