GenBroker – Caching OPC Items

APPLICATIONS NOTE

October 2013

Description: Guide to use OPC Data caching
OS Requirement: Win 2000, XP Pro, Server 2003, Vista, Server 2008, Windows 7
General Requirement: Basic understanding of GenBroker communications

Introduction

GenBroker supports OPC item caching. You can specify the time interval that the cache will maintain an item's value. When a client releases an OPC item (i.e. a tag), the item is not actually removed from the server; it is moved into the item cache. Then, when the client requests the same item again, the item is retrieved from the cache instead of being requested on the OPC server again. You can define how long OPC items should be stored in the cache. Alternatively, GenBroker may deactivate the items put in the cache, thereby decreasing the number of updates coming in from the OPC servers (otherwise, the items in the cache are also updated).

The cache mechanism is optional. When caching is on, it improves the speed of screen switching. Caching takes place for both OPC Direct operation and operations via GenBroker channels.

The OPC Item Cache check box enables/disables the cache. Cache Time defines how long (in seconds) the items would be kept in the cache. When "Deactivate Cached Items" is checked, the items are deactivated when put into the cache.

When an item is "deactivated", its connection to the OPC Server remains, but updates are no longer sent.

How It Works

Assume that you have two GraphWorX32 displays (Display1 and Display2) on a PC, both getting data via GenBroker. If you switch back and forth between these displays, the following will occur for each scenario involving caching.

Caching OFF

- GraphWorX32 opens Display1
- GenClient calls AddItems on the server to put the items of Display1 on scan
- Server sends updates for Display1 items
- GraphWorX32 switches to Display2
- GenClient calls RemoveItems to remove items in Display1 from the server
- GenClient calls AddItems to add Display2 items
- Server sends updates for Display2 items
- GraphWorX32 switches back to Display1
- GenClient calls RemoveItems to remove items in Display2 from the server
- GenClient calls AddItems to add Display1 items
- Server sends updates for Display1 items

Caching ON, Deactivate OFF

Assuming Display2 is opened for a shorter time than the Cache timeout. (Differences from "Caching Off" are in RED)

- GraphWorX32 opens Display1
- GenClient calls AddItems on the server to put the items of Display1 on scan
- Server sends updates for Display1 items
- GraphWorX32 switches to Display2
- GenClient puts Display1 items into cache
- GenClient calls AddItems to add Display2 items
- Server sends updates for Display1+Display2 items
- GraphWorX32 switches back to Display1
- GenClient puts Display2 items into cache
- GenClient gets Display1 items from cache and reuses them
- Server sends updates for Display1+Display2 items
- After Cache Timeout the Display2 items are removed from cache (so the cache gets empty)
- GenClient call RemoveItems for Display2 item
- Server then sends updates only for Display1 items

Caching ON, Deactivate ON

Assuming Display2 is opened for a shorter time than the Cache timeout. (Differences from "Caching ON, Deactivate OFF" are in RED)

- GraphWorX32 opens Display1
- GenClient calls AddItems on the server to put the items of Display1 on scan
- Server sends updates for Display1 items
- GraphWorX32 switches to Display2
- GenClient puts Display1 items into cache
- GenClient deactivates Display1 items
- GenClient calls AddItems to add Display2 items
- Server sends updates only for Display2 items (Display1 items are still on the server, but they have been deactivated)
- GraphWorX32 switches back to Display1
- GenClient puts Display2 items into cache
- GenClient deactivates Display2 items
- GenClient gets Display1 items from cache and reuses them
- GenClient activates Display1 items
- Server sends updates only for Display1items

- After Cache Timeout the Display2 items are removed from cache (so the cache gets empty)
- GenClient call RemoveItems for Display2 item (**Note:** they were on the server, but they were deactivated)

Where to Enable Caching

1. Start GenBroker by going to Start → Programs → ICONICS Tools → GenBroker Configurator
2. Click on the "Edit" button to bring up the configuration window.
3. On the tree on left side, expand Advanced → Common → Optimization to turn on OPC Item Caching as shown in Figure 1

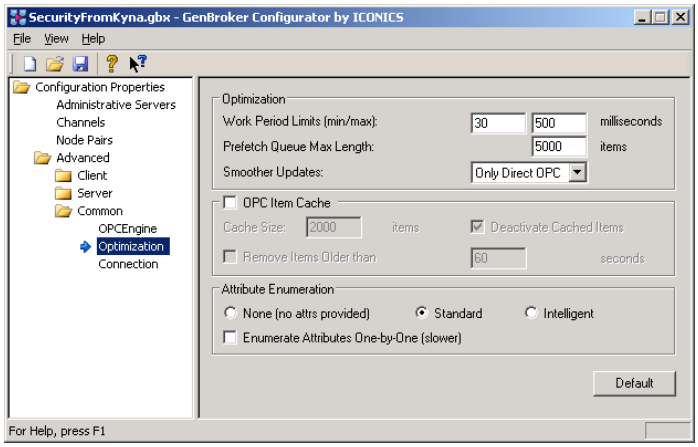

Figure 1 - GenBroker Optimization

Consider The Following

- Adding and removing items is typically a lengthy operation, but there may be exceptions.
- Activating and deactivating items may be a lengthy operation depending on the OPC server.
- Having items on the server and in GenClient cache uses some memory.
- Having lots of items updating quickly consumes CPU, but values typically do not change very quickly

Conclusion

Caching OFF Advantages and Disadvantages

- Minimizes number of items on the server (items are always removed when a display is closed)
- Minimum items gets updated
- Maximum of calls to remove and add items
- Good for cases when many values changes very quickly and when deactivating items takes a long time (same as removing them)

You should turn the cache off when:

1. Doing few screen switching with few tags
2. Switching many screen with many tag updates along with slow activation/deactivation

Caching ON, Deactivate OFF Advantages and Disadvantages

- Minimizes number of all calls to the server
- Has all items active and therefore taking up CPU
- Good for servers with slowly changing values
- Has quick response when switching displays - the items are taken from cache then

You should cache with the deactivate option off when:

1. Switching many screens with a few tag Updates, and the Activation/Deactivation is slow.

Caching ON, Deactivate ON Advantages and Disadvantages

- Numerous items on the server and in the client - consumes memory
- Inactive items do not send updates - so they do not load CPU
- If activation and deactivation is quick, then the response is quick as well

You should turn on caching and deactivation when:

1. Screen switching is heavy with many Tag Updates, and Activation/Deactivation is fast
2. Screen Switching is heavy with many Tag Updates and Activation/Deactivation is slow