Change request log

1. Concept Location

Step #	Description	Rationale
1	After installation, we ran the code to identify and understand what changes were to be made.	
2	After logging in, we played around with the different selections present.	This was carried out to see what changes occur in the user interface to find any clue to identify the location or URL to localize the location
3	The next step we did was to look into the code to understand the code flow.	From step two we identified the URL of the page remains the same irrespective of which watch list is used.
4	We inspected the Chrome Console to identify which element we were supposed to make changes with along with the component responsible for rendering the table.	If we figured out how data is being loaded into this page, we would identify which class is responsible for fetching data.
5	Using the search all code files feature in VSCode Tool, we found the file responsible for creating the watch_list page.	*VSCode has global search as well as local file search which was used to paruse Code Base*
7	Inspected watchList.jsp	DisplayWatchList() responsible for loading watchList data from Java Code
8	WatchListDwr.java	@WatchListDao derives from @ BaseDao wich has a SetPrettyText Function which can be overridden SetPrettyText we can edit the PointValueTime here so that double floating point precision change does not persist in the database.
9	dataPointDetails.jsp	responsible for loading table history from Java Code
10	RenderedPointValueTime.java	@RenderedPointValueTime is responsible for storing the PointValueTime time, value and annotation data. Editing PointValueTime here so that double floating point precision change does not persist in the database.

Time spent (in minutes): 47

Classes and Methods inspected

1. Common
    a. /src/com/serotonin/mango/vo/DataPointVO.java
    b. /src/com/serotonin/mango/vo/User.java
    c. /src/com/serotonin/mango/DataTypes.java
    d. /src/com/serotonin/mango/vo/WatchList.java
    e. /src/com/serotonin/mango/vo/User.java
    f. /src/com/serotonin/mango/rt/dataImage/types/MangoValue.java
    g. /src/com/serotonin/mango/rt/dataImage/types/NumericValue.java
    h. /src/com/serotonin/mango/rt/dataImage/PointValueTime.java
    i. /src/com/serotonin/mango/rt/dataImage/types/BinaryValue.java
    j. /src/com/serotonin/mango/rt/dataImage/types/ImageValue.java

k. /src/com/serotonin/mango/rt/dataImage/types/AlphanumericValue.java

2. **WatchList**
    a. /war/WEB-INF/jsp/watchList.jsp
    b. /src/com/serotonin/mango/web/dwr/WatchListDwr.java
    c. /src/com/serotonin/mango/web/dwr/BaseDwr.java
    d. /src/com/serotonin/mango/web/taglib/Functions.java
3. **HistoryTable**
    a. /war/WEB-INF/jsp/dataPointDetails.jsp
    b. /src/com/serotonin/mango/web/dwr/beans/RenderedPointValueTime.java
    c. /src/com/serotonin/mango/web/dwr/DataPointDetailsDwr.java

2. Impact Analysis

Step #	Description	Rationale
1	*At the beginning, it makes sense to change the MangoValue class itself, but the data will persist if we make any change to that row as all the data is pushed back to Database*	*This change is for viewing purposes only and should not be reflected onto the database.*
2	*We want to change only rendering text without affecting values itself classes / function which are responsible such as PointValueTime*	*PointValueTime is being used to store, retrieve from database, changing directly will affect the data itself*
3	*Using the data PointValueTime and converting value to text at setPrettyText which is being rendered on to the jsp*	*SetPrettyText resides at BaseDwr which is being used to render data of various other Classes. Changing this function might affect other classes across code base.*
4	*RenderedPointValueTime which also does similar things and is called only when data is being rendered.*	*RenderedPointValueTime is responsible for storing converted string value of PointValueTime. Changing here might not affect other places as its only purpose is to render value*

Time spent (in minutes): 38

4. Common
    a. /src/com/serotonin/mango/vo/DataPointVO.java
    b. /src/com/serotonin/mango/rt/dataImage/types/MangoValue.java
    c. /src/com/serotonin/mango/rt/dataImage/types/NumericValue.java
    d. /src/com/serotonin/mango/rt/dataImage/PointValueTime.java
5. **WatchList**
    a. /war/WEB-INF/jsp/watchList.jsp
        i. WatchListDwr.DisplayWatchList()
    b. /src/com/serotonin/mango/web/dwr/WatchListDwr.java
        i. CreateWatchListState()
        ii. SetPrettyText()
    c. /src/com/serotonin/mango/web/dwr/BaseDwr.java
    d. /src/com/serotonin/mango/web/taglib/Functions.java
        i. GetHtmlText()
6. **HistoryTable**
    a. /war/WEB-INF/jsp/dataPointDetails.jsp
        i. DataPointDetailsDwr.GetHistoryTableData()
    b. /src/com/serotonin/mango/web/dwr/beans/RenderedPointValueTime.java
        i. RenderedPointValueTime()
    c. /src/com/serotonin/mango/web/dwr/DataPointDetailsDwr.java
        i. GetHistoryTableData()

3. Actualization

Step #	Description	Rationale
1	*WatchListDwr inherits from class BaseDwr, which in createWatchListState setPrettyText is called which is responsible for rendering PointValueTime double value to string*	*BaseDwr's setPrettyText is generice and hence does not attend to our special needed for WatchListDwr*
2	*Overriding setPrettyText in WatchListDwr and reducing the double precision*	*We override setPrettyText so that our changes only reflect to WatchListDwr and does not affect globally where BaseDwr is used as Base class also not altering only for* NUMERIC data.
3	*In case of GetHistoryTableData* RenderedPointValueTime is used to display the value	RenderedPointValueTime class is being use to render into which Is not being used anywhere, making change in RenderedPointValueTime does not affect anyother location.
4	*Changing* RenderedPointValueTime so that it stores only appropriate double precision data	*We change the value in* RenderedPointValueTime so that it stored correct value which is being displayed keeping in mind only the NUMERIC data is being acted upon

Time spent (in minutes): 29

7. **WatchList**
    a. /src/com/serotonin/mango/web/dwr/WatchListDwr.java
        i. SetPrettyText() Overridden
    b. /src/com/serotonin/mango/web/dwr/BaseDwr.java
8. **HistoryTable**
    a. /src/com/serotonin/mango/web/dwr/beans/RenderedPointValueTime.java
        i. RenderedPointValueTime()
    b. /src/com/serotonin/mango/web/dwr/DataPointDetailsDwr.java

4. Validation

Step #	Description	Rationale
1	Inspection Case: After the change inspected if the changes were in place for all users Logged in using a different user and verified if all values were being set to the required precision Expected Output: Values should be displayed with at most two decimal places	The idea behind this is to check if all the change is reflected all over the system. The test passed.
2	Test case: setting the precision to a different value. Inputs: setting the precision to a different value. Expected output: *v*alues should be displayed with at most two decimal places.	This case was to make sure that the change made works for any precision. The test passed.
3	Inspection case: verifying the value is being truncated and not being rounded off. Write down a couple of values before setting the precision value and compare the same with values after setting the precision. Expected output: value must have been truncated and rounded off.	This test case is more like checking if we meet the requirements for change1 i.e. values are truncated not rounded off. The test case passed

Time spent (in minutes): 33

5. Summary of the change request

Phase	Time (minutes)	No. of classes inspected	No. of classes changed	No. of methods inspected	No. of methods changes
Concept location	47	18	-	5	-
Impact Analysis	38	11	-	12	-
Actualization	29	4	2	8	2
Validation	33	-	-	-	-
Total	147	33	2	25	2

6. Conclusions

To make subtle changes in terms of precision of double value a comprehensive understanding of the source code is necessary as changes made carelessly might affect deeply even changing the sanity of the database. In order to make the change which is only few lines of code a comprehensive understanding of codebase, its impact, effect has to be made.