| **Name** | **Description** | **Type** | **User Group** | **Issue ID** |
| --- | --- | --- | --- | --- |
| swAdmin2: Network id use with one account | A message will now display if you try to save an existing user account (Edit user) or new user account (Create new user) with a Network ID that is already in use by another user account. | Improvement | Administrators | 15276 |
| AUTOSAR Instance and Interface Deployment: New extensions | We now offer two new AUTOSAR deployment extension views - Instance Deployment and Interface Deployment. The views support the Adaptive AUTOSAR workflow by aiding in generation of deployments.  The Interface Deployment extension supports creation of Interface Deployments including Method, Event,  and Field ID generation from a configured ID interval. The view also manages Event Groups.  The Instance Deployment view supports creation of SOME/IP service instances complete with service instance to port prototype mappings, SOME/IP service instance to machine mappings, and SOME/IP service instance configs. |  | Architect |  |
| AUTOSAR Topology Editor: New extension | The first version of an AUTOSAR Topology Editor extension view is now available.  The AUTOSAR Topology Editor is a dedicated user interface developed to support the creation and maintenace of the topology.  The view supports working with a topology from either a Node-Channel or a Channel-Node perspective:  **Node - Channel**  **Channel - Node** | Improvement | Architect | 15272, 15279, 15330, 15264, 15336, 15277 |
| Clone Item View: Support for inheritance | Inheritance is now supported for the optional <IgnoreDescription> element.  If needed, one can now include the SID of the parent item type which will result in ignoring the description for all sub-types. It is also possible to ignore the description of all cloned item types by adding:  <IgnoreDescription>I</IgnoreDescription> | Improvement | Architect | 15338 |
| Coverage mapping item: Support multiplicity single on the right side | There is now support for both single and list multiplicities on the right-hand (secondary) side of the view whereas previously there was support for only list. A new, optional attribute called *distinctMappingPerPair* on <Mappings> determines if each pair should have a distinct mapping. A "true" value enforces 1:1 relation and will always create a new mapping item for the selected pairs. A "false" value enforces replacement for the existing left mapping item. The default is "false". | Improvement | Architect | 15340 |
| Path Query: Additional support for float | The Path Query Language now supports the following for float:  **Operators**  + operator   * integer + float * float + float * float + integer   - operator   * integer - float * float - float * float - integer   \* operator   * integer \* float * float \* float * float \* integer   The result of these will be float.  **Evaluation**  Float attributes are now evaluated as floats all the way. Previously, @ATR1 = "4.3" would return "true". This is no longer the case. Instead, @ATR1 = 4.3 will return "true".  **Methods**  In addition, the following methods have been added:  ToFloat   * Transforms and type-casts a float (or integer) value string into an float value.   [String].ToFloat -> Float  Example:  /part::5ACV.Select(@5CVA.ToFloat).Sum  Results in the sum of the enum-values of part-attributes of (string) attribute-type Customer Value [5CVA].  Round   * Returns the rounded integer part of a float argument.   [Float].Round -> Integer  Example:  probability:=0.99999999; probability.Round  Returns '1'  RoundTo   * Returns the rounded value, to the specified number of decimals, of the argument floating point number.   [Float].RoundTo -> Float  Example:  pi:=3.1415926535897932384626433832795; pi.RoundTo(2)  Returns a two decimal approximation of pi, i.e. '3.14'. | Improvement | Architect | 12596, 15400 |
| Path Query: Split string by deliminator | Given a delimiter, it is now possible in the Path Query Language to split a string into a list of strings separated by ','.  String.Split(delimiter: String) -> [string] | Improvement | Architect | 14862 |
| XML Import/Export: Ability to Exclude Attributes | For XML Import and XML Export configurations, there is now an option to allow you to exclude attributes called <ExcludedAttributes>. This can be used instead of <IncludedAttributes> when, for example, you want to include all attributes except for one.  <ExcludedAttributes>ISES;ISSC</ExcludedAttributes> | Improvement | Architect | 15390 |
| Coverage matrix: Various improvements | The NotApplicableMappingItem has been removed from the HazardIdentification extension's Coverage Matrix view. The functionality has been replaced with the optional <ConstraintItems> tag in the <MappingItem> configuration. This gives the possibility to combine different mapping elements when necessary. In addition, the <MappingItem> tag has been extended with two optional settings <MappingItem itemType ="SID" notApplicable="true/false" quickAddMapping="true/false">. The notApplicable attribute will mark the corresponding cell in the coverage matrix gray. The quickAddMapping attribute allows for a single-click mapping to be created.  The <MappingItem> now has support for multiple instances meaning that several MappingItems can be a part of each CoverageMatrix instance.  The CoverageMatrix view now supports the following attribute types:   * Identity * User * Custom Identity * Computed * Custom Computed   The Custom Identity attribute type requires context input. The selected item is automatically used as input but it is possible to add information from a parameter by using the  <CoverageMatrixParameter select="**$ParameterName"**/>  The Mapping item name can now be set with the use of the SystemWeaver Path Language. Paths can be initiated by using standard path language syntax #{?} with starting points from the row item, column item or the currently selected item. Each item can be used as a starting point by using the $CurrentItem, $Row or $Column variable tags.  <DefaultMappingName>**$column.... +'of' + $row...+ $pSelected/..+ $mainItem/....**</DefaultMappingName>  **Example:**  The GUI has been updated with a new layout to better utilize the view. The faulty mappings are now found at the bottom of the view and the pane is hidden when there are no faulty mappings. It can be opened by clicking on the ..... splitter. | Improvement, Breaking change | Architect,  User, Breaking change | 15381, 15138, 14980, 15357, 14870, 15151, 14873, 14869, 15155, 14868 |
| Attributes: New Signature attribute type | There is a new attribute data type called Signature for digital signing. Data dimension Single is supported.  You sign by pressing and entering your Windows credentials.  Only items with a status of "CS released" can be signed.  Signature displayed in attribute preview:  The Username and Display name is saved as it exists at the time of signature. It is not affected by any future changes to these name properties.  Once a Signature attribute is added to an item and is used to sign, the attribute can never be removed from the item. A signed item can never be deleted. Since you cannot delete the item, and it has a status of "CS Released", this item and any referenced items, including previous versions can never be deleted.  You can also sign issues. The meaning of signing an issue is determined by the workflow definition. | Improvement | Architect, Extensions API, User | 15297, 15295, 15064, 15255, 15259, 15335, 15333 |
| Configurable graph: Various enhancements | The following enhancements have been made to the configurable Graph view.  **Support for using Path Query to select style for node and edge**.  It is now possible to use a path query to select a style.  Node  <Node style="#{? path query}"/>  Edge  <Edge from="." to="." style="#{? path query}"/>  Selection of style by providing the name of the edge or node style as a string is still possible.  **Support for using Find (Ctrl+F) in a graph to locate nodes and ports.**  Hits are highlighted with a green frame.  **Neighborhood for a selected port**  In addition to viewing the neighborhood for a selected node, it's now possible to show the **Neighborhood** for a selected port.  **Lock the Graph in Primary View**  The view can now be locked in the primary view position enabling you to keep the selected item in focus in the graph as you navigate through the structure tree. With the view locked, when you make selections in the tree, you will see them highlighted in the view.  And combined with the use of the **Secondary item view**, you can make edits to, e.g., attributes, description, etc., to an individual item selected in the graph while keeping the graph in view. Working this way, the structure tree view can basically be minimized altogether since most work tasks can be completed directly in the graph.  **Refresh button**  The existing Refresh option using the F5 key is now also available as a button on the view's toolbar  **Port Selections Highlighted in Tree**  Selected ports in the view are now highlighted in the structure tree.  **Legend Option**  It is now possible to configure a legend for a graph using a new <Legend> tag along with *legendLabel* attributes in <NodeStyle> and <EdgeStyle> tags. The legend can be viewed in the same way as Neighborhood.  More detailed information about Legend configuration can be found in the application Help, and an [example configuration](https://support.systemweaver.se/support/solutions/articles/31000160149-legends-in-configurable-graphs) is available on the Support Portal.  **Improved Message for Tree and Balloon Layout Problems**  "tree" or "balloon" layout requires that the graph data is topologically a true tree structure, i.e. no edge loops are allowed. When using layout="tree" or "ballon", and the data is not a tree, a "Exception: Graph is not a tree" message will display in the view. | Improvement | Architect, User | 14292, 15133, 9329, 15075, 15208, 15110, 14249, 15395, 15396, 15119, 15123, 15148 |
| Item Ribbon: New multi-button configuration options | When using a configured graph in an <ItemView> in a MultiButton, the optional *locked* attribute can be used to specify if the graph will be locked or not upon loading. Valid values are "true" or "false". The default value is "false".  There are now additional attributes for <SecondaryItemView> which can also be used when this view is included in a MultiButton. The new attributes are *configItem*, *locked* and *orientation*.   * The optional *configItem* attribute works similar to *configItem* in <ItemView>. * The optional *locked* attribute specifies if the secondary item view will be locked or not. The default value is "false". * The optional *orientation* attribute specifies the position of the secondary item view. Possible values are "right", "top", or "bottom". The default value is "right".   <ItemRibbonTabConfig>  ...  <MultiButton caption="Component Graph" visibility="Dynamic" hideIfDisabled="true">  <ItemView name="XXXGraph" configItem="BIDA:ComponentGraph" locked="true"/>  <SecondaryItemView name="Grid" configItem="BIDA:DeFunO" locked="true" orientation="top"/>  </MultiButton>  ...  </ItemRibbonTabConfig> | Improvement | Architect, User | 15408, 15409, 15355 |
| ItemVersionChanges: Option to include message when no difference | There is a new attribute for ItemVersionChanges for a change log called *includeNoChangeMessage* which provides the option of including a message "No changes, additions or deletions have been made to attributes, parts, and description content." in the ChangeLog when no changes have been made between versions of items.  If the attribute is set to "true", then the message will display in that case. The default value is "false" meaning that you must have this attribute included and set to "true" if you want to have the message included.  <Report>  <ForEach select="/REQS">  <AddToChangeLog/>  </ForEach>  <Section title="Change Log - updated items">  <ForEachItemInChangeLog type="updated">  <ItemVersionChanges includeNoChangeMessage="true"/>  </ForEachItemInChangeLog>  </Section>  </Report>  Result:   |  | | --- | | **(2) => (3):** | | No changes, additions or deletions have been made to attributes, parts, and description content. | | Improvement | Architect, User | 15372 |
| Parameters: Support for default values | There are now two options for setting a pre-selected value for a Parameter: the existing *hintContextPath* and a new attribute called *defaultValue*. They can be used in the same view configuration for different parameters, but cannot be combined/used together for the same parameter.  <Parameter caption="String" name="String" as="String" defaultValue="'String1'">  The supported data types for *defaultValue* are:   * Boolean * String * Date * Integer * Int64 * User * Object   The default value must be a valid parameter value, otherwise the default value will be empty. | Improvement | Architect, User | 15256 |
| Secondary item view: New layout options | There are now additional layout options for the Secondary item view. Aside from the default right position, and existing Undock options, you can now select to position the Secondary item view to the top or bottom of the view area. The options can be found when expanding the Close button:  In the below example, a grid in the secondary view is located at the top of the view area while the primary view contains a graph.  It is also possible to configure the layout via an *orientation* attribute for the position of the Secondary item view in a MultiButton on the ribbon (Item Ribbon: New multi-button configuration options). | Improvement | Architect, User | 15408, 15409, 15355 |
| PartType config: New AllowMultiInstance constraint | An <AllowMultiInstance> constraint tag is now available for part type configuration to constrain the addition of parts that already exist under the selected owner item.  The constraint is expressed by an XML element:  <AllowMultiInstance value="" allowDifferentVersions=""/>  Valid attribute valuesare 'true' (default) or 'false'.  With AllowMultiInstance set to "false", users cannot add multiple instances (occurrences) of the part under the same owner item.  The optional attribute *allowDifferentVersions* determines whether or not users can add other versions of an existing part. This option can only be used when AllowMultiInstance is set to 'false'.  With allowDifferentVersions="false": | Improvement | Breaking change, Architect | 15030 |
| Extensions API: Various enhancements | **Attribute editors**  In interface IswDialogs, there are new methods in the extensions API to launch the standard swExplorer dialogs for editing of attribute values. The unbound versions are for when you want to edit the value before the item/issue has been created. Not all attribute-types have an unbound version so there is a method to check if there is.          bool EditAttributeOk(IswAttributeType attributeType, IswObj obj);          bool EditAttributeUnboundOk(IswAttributeType attributeType, ref byte[] data);          bool HasUnboundEditor(IswAttributeType attributeType);          string GetAttributeDisplayString(IswAttributeType attributeType, byte[] data);          string AddNewUniqueAttribute(IswItem currentItem, IswItems contextItems);  **ExecutePathAsString** The Mapping item name can now be set with the use of the SystemWeaver Path Language. Paths can be initiated by using standard path language syntax #{?} with starting points from the row item, column item or the currently selected item. Each item can be used as a starting point by using the $CurrentItem, $Row or $Column variable tags.  <DefaultMappingName>**$column.... +'of' + $row...+ $pSelected/..+ $mainItem/....**</DefaultMappingName>  Executes path within variables and returns a string.  **Unique ID**  There is now support for the custom attribute Unique Id in the Extension API. It supports the creation of unique ID attribute values suitable for requirement IDs, etc., similar to Identity attributes. It builds on the use of two custom attributes - one that defines the context (AUIC) and one that defines the ID (AUID). | Improvement | Extensions API | 15129, 15258, 15329, 15406 |
| Extensions API: Version update | The Extension API has been updated. All extension views developed in-house will need to be recompiled with the new version. |  | Extensions API user |  |
| Client-Server: Latency indicator | The swExplorer now provides a latency indicator. It shows the approximate time it takes to go from the client application to the SystemWeaver server application and back. It can can be useful when diagnosing potential internet connectivity issues, and is not necessarily indicative of a problem.  **Typical Approximate Latency Times**  In milliseconds (ms)  Below are typical latency times under normal load conditions.   * Server and client on same machine: 2 * Server and client on different machines in same geographical location: 8-15 | Improvement | IT Admin | 15388 |
| Mirror Server: Log Progress while updating from main | The mirror server log file now provides more information about the cache update upon start-up. Previously, the log indicated:  2021-05-05 16:47:11,270 [5424] INFO bcall - Updating local cache from main...  It now provides more information about its progress in updating the cache file by including an entry for every 10000 writes that are read to the cache file.  2021-05-05 16:47:11,270 [5424] INFO bcall - Updating local cache from main...  2021-05-05 16:47:11,617 [14140] INFO root - Virtual Bytes: 4663.062528  2021-05-05 16:47:13,530 [5424] INFO journal - Updating cache, transaction: 60000  2021-05-05 16:47:15,849 [5424] INFO journal - Updating cache, transaction: 70000  2021-05-05 16:47:18,151 [5424] INFO bcall - Cache updated.  2021-05-05 16:47:18,258 [16584] INFO main - Sub> Starting tcp server.  2021-05-05 16:47:18,287 [16584] INFO main - Sub> TCP Server started on port: 2222  The main server log file now gives an indication of approximately how many more transactions are left to read to the cache file.  2021-05-05 16:45:33,448 [25536] INFO journal - Journal at transaction: 60000  2021-05-05 16:46:28,273 [25536] INFO journal - Journal at transaction: 70000  2021-05-05 16:47:11,066 [25536] INFO journal - Reading journal from 50563 to 79748  Most installations do not have over 10,000 writes in one day. In that case, you would only see one of these log entries. If you see many, it is an indication that many write operations occurred while the mirror was stopped, or are still occurring. Information about the volume of write operations can be very helpful to administrators for planning future downtimes. | Improvement | IT Admin | 15373 |
| Mirror server: Reconnect to main | If a mirror server loses connection to the main server, either because of network problems or if the main server is restarted, the mirror server will now attempt to reconnect automatically every 30 seconds. As soon as the main server can be reached again, it will reconnect. This means that there is no longer a need to restart a mirror server if you stop a main server. | Improvement | IT Admin | 15382 |
| Remove Part: Assertion failure | The fix resolves an assertion failure thrown when removing a part in certain scenarios that caused a server out of sync. | Bug fix | IT Admin | 15380 |
| Server: Option to limit .swjournal file size | When using journaling for the mirror server solution, it is now possible to set a minimum number of most recent entries to be saved in the swjournal file which will limit the size of the file. The configuration is done in the swServer.ini.  MinJournalEntries=x  If total > MinJournalEntries \* 1.1, then remove MinJournalEntries \* .1  Example: If you specify 10000, the number of entries will grow to 10000, then be somewhere between 10000 and 11000. In other words, without the limitation, the journal will store the total number of journal entries, e.g., 500000, whereas with the journal limitation, the server will remove old entries and maintain only the most recent 10000-11000 entries per the configuration.  The **Journal file info** tool in the swDatabaseManager will show the "First transaction id" and "Last transaction ID".  Connecting to journal file: C:\Work\Testing\R37\15089\_journalsize\15089.swjournal  Journal file identified  First transaction ID: 10655  Last transaction ID: 21001  Organizations that want to limit the journal size will need to decide how many transactions to keep. Monitoring the .swjournal and using the swDatabaseManager tool to see how many transactions it contains and, e.g., how many there are on average per day, can help in making this decision. | Improvement | IT Admin | 15089 |
| Server: Update db from journal at start | There is now an option in the main server configuration to allow the main server to run transactions found in the journal file upon start-up that are not found in the database. During normal operation on a local disk, there is no risk of this scenario occurring. However, this option does provide an automated way to handle an unforeseen synchronization problem should the need appear.  We do not recommend the use of SyncJournalEntries=True by default, since it can potentially hide network/disk problems that you otherwise would want to be aware of. Please consult Systemite if you are considering using this option.  **Warning:** If this option is set to "True" and you want to revert to a backup database copy as a means of "wiping out" unwanted database transactions, you will want to run **Remove extra entries** on the journal file before you start the server against the backup database. | Improvement | IT Admin | 15073 |
| RestAPI: Description reference text missing from export | This fixes an issue where in exports of plain description texts that contain Description References, the Description References as texts were not exported. | Bug fix | Rest API | 15032 |
| Rest API: Problems with SSL | This fixes an issue that was found where AspNetCore was trying to use HTTP2 on a machine that doesn't support it without manual configuration (Windows Server 2012 R2/Windows 8.1 and below). This is a known Bug fix that has been fixed in the master (cutting edge) branch of AspNetCore. There is now a new property called "allowHttp2" in appsettings.json to allow users to disable HTTP2. In addition, there is a new property called "certificateSubjectName" for specifying the Subject/Name of the SSL certificate. | Bug fix | Rest API, IT Admin | 15117, 15060 |
| AUTOSAR Exporter: Extension version now available | Our current SystemWeaver.AutosarExport standalone application for AUTOSAR exports is now available as an extension view that support exports of Classic and Adaptive AUTOSAR.  Also, the AUTOSAR Exporter now supports export of System Extract as well as export of a System Manifest  and Service Interfaces on the Adaptive AUTOSAR side. | Improvement | User | 15359, 15324, 15311, 15365, 15215 |
| Component graph: Issue with ComponentGraph output in report | This fixes an issue with the formatting of the component graph when included in a report or document. | Bug fix | User | 15362 |
| Component graph: Problem with recursive graphs | This fixes an issue where routing for sub-architectures was not handled effectively. This has now been addressed. | Bug fix | User | 14696 |
| Context visualizer: Right-click menu missing | The right-click menu with Export and Print options in the Context visualizer for Graphs has been reinstated. | Bug fix | User | 15425 |
| Document view: Assertion failure | This fixes an assertion failure error thrown when attempting to edit columns in the structure tree grid while the Documents view was loaded. | Bug fix | User | 15421 |
| Issue editor: New pop-up window | The **Ctrl+Alt+O** keyboard shortcut for opening an item in a separate window now works for opening an issue as well. This new **Edit Issue** window is supported both under Projects when selecting an issue in an issue view, and in a tree grid when focusing on an issue in the grid. | Improvement | User | 15325 |
| Issues: Optimize iteration handling | We have optimized the issues views when many iterations exist in the project. | Improvement | User | 15135 |
| Mirror Server: Optimize InitAncestorPartHandles | We have optimized part handling which results in improved mirror performance. The optimization will be most noticeable in larger, high-load installations. | Improvement | User | 15191 |
| Navigate with mouse side buttons | The programmable mouse side buttons can now be used to navigate back and forward between items in a structure tree as well as views, and issues, in the same manner as when using the Items ribbon Navigation options. | Improvement | User | 15315 |
| Parts View: Color indicates deprecated parts | Deprecated part types will now be highlighted in red in the view and a message indicating that the part type is deprecated with display in a new Error Info column. See Parts View: New info column. | Improvement | User | 15139 |
| Parts View: New info column | There is now an optional Error Info column that will provide information if a part type has been changed in such a way that it is no longer supported by the meta model. The column can be added by right-clicking on the head row and selecting it.  The part type multiplicity has been changed to "Single" and there are multiple occurrences of the part type, (which is no longer supported):  The part type is no longer supported by the meta model. i.e., part type has been changed/removed:  The part type is deprecated: | Improvement | User | 15156 |
| Reports: Various Path Query optimizations | Various optimizations have been made to Path Language queries, primarily to increase the performance of reports that contain numerous queries. The improvement is most noticeable with structures containing Simple parts as opposed to Structural parts. | Improvement | User | 15197 |
| StructureTree: Various optimizations | The loading of simple parts in the structure tree has been optimized. This enhancement entailed the minor upgrade to the API from version 1.52.0 to 1.52.1.  In doing this optimization, we also changed the initial view setting for item structures. Previously, a **Default** structure tree setting of all structural parts along with a fixed list of legacy part types was loaded for new item types or for item types where items of that type are yet to be created. This has now been removed. When opening an item of a new or unused item type for the first time, users will not see any part types by default. Architects will need to set up the needed **Default** view. | Improvement | User  Architect | 15310, 15096, 15371 |
| Configurable graph: Introduction of editability | The built-in configurable Graph view now offers an option to enable users to not only visualize the data in the configured graph, but also graphically design the data directly in the graph. And because it is a generic view, it can be used with all models and use cases in SystemWeaver.  The option is enabled as part of a graph configuration through the use of new tags for:   * Adding parts * Removing parts * Mapping connection types   We have also made available a first version of context menu configuration for adding and removing parts to provide users with a context menu style similar to the one found in the structure tree.  Editability can also be turned off in the configuration of the graph.  When enabled, users can perform the following operations directly in the graph in accordance with the meta model, in the same way as they can in the structure tree:   * Create new parts and their resulting graph nodes * Adding parts and their resulting graph nodes * Remove existing parts and their graph nodes (single or list) * Update nodes (name and attribute) * Connect nodes   For connecting nodes, the available connection types are:   * Explicit connections * Implicit connections * Connect through mapping item * Connect through anchor item   Right-click context menu options and shortcut options for editing in the Graph view are similar to those in the structure tree, e.g., New, Add, Copy (Ctrl+C), Paste as (Ctrl+V) with the addition of **Connect Ports**. Not only can the Copy and Paste options be used within the graph, you can also copy from a grid or the structure tree and paste in the graph. To create a connection, select the objects, right-click and select the appropriate Connect option. The label for this option is configurable.  It is also possible to configure the Remove part option for the menu using a string or path query.  For released items, New, Add, Paste, and Connect options will be grayed out.  Secondary item view: New layout options**:** Aside from the default right position, it is now possible to position the Secondary item view at the top or bottom of the view area. With editing enabled in the graph, you can, for example, copy and paste items between a grid and a graph.  More detailed information about the new Graph tags and how to work in edit mode can be found in the application Help.  Example configurations are available on the [Support portal](https://support.systemweaver.se/support/solutions/folders/31000116038). | Improvement | User,  Architect | 15006, 15368, 15214, 15066, 15355, 15002, 15389, 15409, 15020, 14848, 15190, 15408, 14849, 15130, 15274, 15036 |
| Grid: Option to optimize | The <Grid> tag now includes an optional *optimized* attribute which defines whether or not each cell value, e.g., in TextColumn and ItemAttributeColumn, will be recalculated as needed or not. We have seen speed improvements of a factor 10 using this option and, therefore, recommend its use if a grid is slow.  <Grid optimized="">  Valid values are "true" or "false". Default value is "false".  **True:** Each value will only be calculated once when you are working in the grid. This is suitable if the grid is for obtaining a "snapshot" that will be exported to Excel, etc. Grid generation and Excel export times are optimized. To refresh/recalculate the grid cell values at any time during your viewing, hit F5.  **False:** Each value is recalculated as needed when you are working/editing in the grid. This is suitable if you always want the values to be updated. Grid generation and export times will include obtaining all relevant up-to-date values from the database. | Improvement | User,  Architect | 15015, 15321, 15221 |
| API: Minor version update from 1.52.0 to 1.52.1 | The SystemWeaver API has been upgraded from version 1.52.0 to version 1.52.1. Although this minor upgrade does not technically require that all users receive and utilize the R37 version of the client applications (swExplorer, swAdmin2, swArchitect), we do recommend it and it would be required to access all of the client enhancements and Bug fix fixes listed in the release notes.  To obtain any newly introduced item types, part types, or attribute types, API applications developed in-house will need to be recompiled with the newer version of the SystemWeaverClientAPI.dll. The Rest API would need to be upgraded to obtain them as well.  Note that clients of version 1.52.1 cannot log in to 1.52.0 versions of the server. | Improvement, Breaking change | User, IT Admin | 15371, 15386 |
| Application icons and caption | **New application icons**  The following applications have received new icons:  **Test server status and port info**  In addition, the swTestServer application icon will now indicated if it is running or not:  Indicates the test server is stopped  Indicates the test server is running  Also, the server port number is now displayed in the swTestServer application caption, which also displays when hovering over the executable on the Windows taskbar.  Note: When swapping out older versions of these executables, you may need to clear your Window icon cache to view the new icon immediately. | Improvement | User, IT Admin | 15097, 15132 |