| **Name** | **Description** | **Type** | **User Group** | **Issue ID** |
| --- | --- | --- | --- | --- |
| Admin: New theme applied to swAdmin2 client | The theme of the swAdmin2 client user interface (GUI) has been updated to fit our new SystemWeaver color scheme. Additional updates may follow in future releases.  It is also now possible to use network authentication to log in if you installation supports it. | Improvement | Administrator | 16929 |
| API: Error message thrown | This removes an error message "DoEvents, Exception: 85332, Event not handled: 46" thrown when an issue is created since there is no support for listening to this event in the C# Client API. No error occurred, the cache is still ok. | Improvement | API | 16793 |
| API: Fix view icon offset | This fixes an issue with the icon index being off by 1 when using the method: SWImages**.**TypeImageIndexToGuiImageIndex  The offset also caused some extensions with a hard-coded icon # to display the incorrect icon.  This is a breaking change **if** code changes in inhouse developed products were made to fix the offset. We recommend only using hard-coded numbers for GUI images and to use TypeImageIndexToGuiImageIndex for Type Images. | Bug fix | API | 16975, 15491, 16900 |
| API: Minor version update from 1.53.2 to 1.53.3 | The SystemWeaver API has been upgraded from version 1.53.2 to version 1.53.3. Although a minor version upgrade does not technically require that all users receive and utilize the new version of the client applications with the new server, we do recommend it, and it would be needed to obtain the client enhancements and bug fixes listed in the release notes. If you reference them with the specific version flag set to true for inhouse development, you must recompile. | Bug fix | API | 15018 |
| Configurable graph: Connect edges to ports with connections | It is now possible to connect edges to ports with connections. Earlier the edge would not be drawn if connected to a port with connections, but would be connected to ports without connections. This limitation has been removed now. | Improvement | Architect | 16194 |
| Configurable graph: Edge grouping | The EdgeStyle tag has a new, optional attribute called *edgeGrouping*. It defines whether or not to merge edges that have the same start and end. Supported values are: "group" or "none". Default, as before, is "none".  If *edgeGrouping*="group", all edges with this Edgestyle and with the same start and end, will be grouped to one (1) edge. The edge labels of the merged edges will be merged and separated by a new line.  **Examples:**  *edge*G*rouping*="none" or *edgeGrouping* not used  *edge*G*rouping*="group" | Improvement | Architect | 14767 |
| Configurable graph: Enable manual ID and objects for ports | It is now possible to manually configure the ID for ports in a configurable graph in the same way as one can for edges and nodes using attributes *id* and *objects*. This provides a way to create a dummy port, i.e., a port that does not represent an object in the database. More details can be found in the *SystemWeaver Script Language Reference Manual* in the application Help.  **Examples:**  <OutPort id="id-1"/>  <OutPort id="#{? 'dummy\_' + Handle}"/>  **Note:** It is currently not possible to configure dummy ports in sequence diagrams. | Improvement | Architect | 16194, 16549 |
| Configurable graph: New graph options | The following new XML tags have been added as Options for graph configuration.  <Options>  <HideUnconnectedPorts value="true"/>  <HidePortLabels value="true"/>  <HideEdgeLabels value="true"/>  <DelegateImplicitPorts value="true"/>  <MergeNodesWithIdenticalID value="true"/>  </Options>  **<DelegateImplicitPorts value=""/>** defines if the implicit ports are always delegated. By default this option is "false", i.e., that ports with the same connectionIDs are all connected together, no matter if they are under the same parent or not. If the option is set to "true", ports are only connected to other ports under the same parent. If a child port wants to send or receive a port from cousin ports, this communication should go through, and be delegated through, the parents.  **<HideUnconnectedPorts value=""/>** can be used to hide unconnected ports in the graph when set to "true". The default, or "false", is to display them.  **<HidePortLabels value=""/>** can be used to hide port labels in the graph when set to "true". The default, or "false", is to display them.  **<HideEdgeLabels value=""/>** can be used to hide edge labels in the graph when set to "true". The default, or "false", is to display them.  **<MergeNodesWithIdenticalID value=""/>** merges nodes with identical IDs in the graph when set to "true". The default, or "false", is to not merge them. Note that if a node appears under different parent nodes, it will not be merged. | Improvement | Architect | 16891 |
| Configurable grid: Expanded editability (Actions) | The following operations are now supported directly in Grids when configured using Actions.  **Set part name**   * When the Part Name option is enabled for a part type, it is now possible to add or update the part name instead of the default item name when using Actions. <Part> has a new *name* attribute. When part name is enabled, the *name* attribute uses the standard way of parsing text so that it supports path queries. Example:   <Part id="P" name="#{? $port.DefObj/ARDE.Select(Name).Join('')}" owner="ItemSW" match="$port"/>  **New note**   * With the implementation of note creation using the Script Language, users can now work with item and issue Notes in a grid when configured using Actions. To enable this, a new data type "Description" has been introduced. Example:   <Action id="MyAction">  <Parameter name="currentItem" as="Item"/>  <Parameter name="noteDesc" as="Description"/>  <Item id="A" match="$currentItem">  <Note sid="NND" description="$noteDesc"/>  <Note sid="NND" value="TextText...."/>  <Note sid="NND" value="#{? $currentItem.Name + 'TextText..'}"/>  </Item>  </Action>  **Select library**   * Previously, when creating items in the background as a result of an operation, the Library was hard-coded. Users can now choose which Library to add the items to using the **Select Library** dialog. A pre-selected value can be configured. Example:   <DialogSelectLibrary defaultValue="$Lib"/>  **Select item version**   * With the implementation of Select Item Version Dialog using the Script Language, users can now change an item version to any version using action. Example:   <ActionBinding actionId="DoIt" caption="Change the version">  <WithParam name="Req">  <DialogSelectItemVersion/>  </WithParam>  </ActionBinding>  **Presets for issue creation**   * Some properties, e.g., Type, Name, Status, AssignedTo, Iteration,.. etc., in the New Issue dialog can now be preset for users when they create issues. Example:   <Variable name="currentItem" as="Item" select="."/>  <ActionBinding actionId="AddIssue" caption="Add Issue from Dialog">  <WithParam name="issue">  <DialogCreateIssue project="AC1" issueType="Issue" name="#{? 'Issue for' + $currentItem.Name}" status="Closed" assignedTo="#{? CurrentUser}" Iteration="Backlog"/>  </WithParam>  </ActionBinding>  **Part matching for part with multiplicity = List**   * The *owner* attribute is no longer required when using *match* for updating parts.   **Part matching for part with multiplicity = Single**   * The *owner* attribute is still required, but the match will be done automatically based on *partType* attribute. So no need to use *match* attribute.   **Part removing for part with multiplicity = Single**   * The *remove* attribute can be used to remove a part with multiplicity= Single. Note that together with the *remove* attribute, it is still required to use *partType*, *owner*, and *defObj* attributes in order to be able to remove a part with multiplicity= Single.   **Set description for an item and issue**   * A new, optional *description* attribute has been added to <Item> and <Issue> tags to support adding a description (formatted text) to an issue/item from another item, issue, or note. Example:   <Action id="DoIt">  <Parameter name="desc" as="Description"/>  <Issue description="$desc"/>  </Action>  **Clone an Issue**   * Using the optional *clone* attribute,it is now possible to clone an issue. When cloning an issue, the cloned issue will get its type, properties (e.g., status, iteration, assigned to, etc.), description, attributes, name, attachments, and project from the original issue. Issue's ObjRef (relationships to items and issues) will not be cloned automatically. If needed, they shall be created manually. To clone into another project, i.e., not the same project as original issue, use the *project* attribute. Example:   <Action id="cloneIssue">  <Parameter name="issue" as="Issue"/>  <Issue clone="$issue"/>  </Action>  **Clone an Item**   * Using the optional *clone* attribute*,* it is now possible to clone an item. When cloning an item, the cloned item will get its type, description, attributes, name, and library from the original item. Item's parts will not be automatically cloned. If needed, parts should be created manually. To clone into another library, i.e., not the same library as original item, use the *lib* attribute. Example of cloning an item to the same library as the original Item:   <Action id="cloneItem">  <Parameter name="item" as="Item"/>  <Item clone="$item"/>  </Action>  **Nested menus**   * It's now possible to configure nested menus in grids using Action. Menus are generated based on the caption specified in <Action> or/and <ActionBinding>, adding (->) in the *caption* attribute will automatically create a nested menus. Example Structural menu:   <ActionBinding caption="Update (->) status (->) Done">  <Action caption="Update (->) status (->) ToDo">    Result:  **Deactivate menu options**   * Menu option activation can now be conditioned, using two new attributes, *enabled* and *disabledTooltip*, added to ActionBinding.   <ActionBinding actionId="DoIt" caption="Add Req" enabled="$req.Select(@TTT) = ['Yellow']" disabledTooltip="This option is only available for requirements which are not Yellow">  Result:  **DialogSelectItem recommendations**   * There is now support for configuring recommendations for adding parts. Example:     <DialogSelectItems itemType="ARSI" partType="DSIN" owner="$requirement" pathToOwner="container;requirement"/>  **Check if RowObject is empty (null)**   * It is now possible to check if the value of a RowObject is null. This can be used in both <Action> and <ActionBinding>.   Example: ActionBinding  <ActionBinding actionId="DoIt1" caption="Add Signal 1" enabled="$requirement != []">  Example: Action  <Action id="DoIt">  <Parameter name="requirement" as="[Item]"/>  ...  <Choose>  <When test="$requirement = []">  .....  </When>  </Choose>  </Action> | Improvement | Architect | 16223 |
| Configurable grid: Issue workflow restrictions/pre-conditions applied | Any applicable Issue Workflow restrictions or pre-conditions that are part of a specific Workflow are now applied when working with Issues directly in a configured grid. For example, if you attempt to set a particular Status for an issue using the grid, and the logic in the Workflow does not allow it, the configured message will inform the user. Likewise, if you configure an <Action> to create an issue, there is no need to add the default status in your <Action> as it will be set automatically by checking the workflow configuration.  Example of a status precondition thrown in grid:  This applies to the following editable (readOnly="false") columns:   * <IssueAttributeColumn> * <AssignedToColumn> * <IssueStatusColumn> * <NameColumn> * <DescriptionColumn> | Improvement | Architect | 16868, 16894, 16887 |
| Configurable grid: Selection of abstract item type sub-types | When using DialogSelectItem in ActionBindings for an abstract item type, the **Select type** dialog will now appear directly, and present non-abstract sub-types from which users can choose.  Previously, the **Open item of type Abstract ...** dialog appeared and users needed to click on **Select item type...** to obtain the Select type dialog. | Improvement | Architect | 16861 |
| DescriptionColumn: Editing of issue description | It is now possible to edit Issue descriptions included in a configured grid using DescriptionColumn when *readOnly*="false".  <Grid>  <Options>  <CellAutoHeight/>  </Options>  <Columns>  <NameColumn readOnly="false" width="150" caption="Issue name"/>  <DescriptionColumn readOnly="false" caption="Issue description" width="400"/>  </Columns>  <ForEach select="/issueref::IR">  <Row/>  </ForEach>  </Grid> | Bug fix | Architect | 16893 |
| Excel Import: Option to match on name or foreign id | As the Excel Import application is not a core product, we provide it to customers when it fits a customer's needs. This standalone application now offers an option to either match on item name, or match on foreign id. If there is a match, no new import is done. No match remains the default setting, i.e., all items will be imported. | Improvement | Architect | 16935 |
| Icons: New type icons | The following new Type icons have been added. They can be used as image icons for item, part, and issue types, as well as for multi-instance configurable views, and plugins: | Improvement | Architect | 16841 |
| Metamodel Diff: Error thrown exporting attributes to Excel | This fixes an error that was thrown when exporting the Attributes list in the Metamodel Diff tool to Excel. The error occurred when there was a default value set for an attribute. | Bug fix | Architect | 14247 |
| Metamodel Diff: Include deprecated status in attributes comparison | The Deprecated status (IsDeprecated) for Attributes is now displayed in the Metamodel Diff tool and included when exporting this list to Excel. | Bug fix | Architect | 15864 |
| Metamodel XML export/import: Deprecated status for attributes included | The Deprecated status for Attributes is now included in the XML export and import of the meta model. This applies to all meta model XML Export/Import options:  swArchitect   * XML Export/Import * Metamodel Cutter   swExplorer   * XML Export/Import when including metadata with data export   Exported XML:  <AttributeType sid="2APH">  <Name>Comment</Name>  <Info>Calibration parameters</Info>  <DataType>adtString</DataType>  <DataDimension>admSingle</DataDimension>  <Range/>  <EnumTypes/>  <Format>250</Format>  <Description>  <Binary><![CDATA[AQAAAAACNgEAAHicrZLfa8IwEMfjsL/RPe8te9xDwdpt6qMK+8lk4IS9xvbUQJpIGnX9E/dfrWnp=]]></Binary>  </Description>  <IsRestricted>False</IsRestricted>  <IsDeprecated>True</IsDeprecated>  <DefaultValue/>  <EmptyValueIsValid>True</EmptyValueIsValid>  </AttributeType> | Bug fix | Architect | 15864 |
| Script language: Access to other parameter values | When using <Parameter> with *defaultValue*, it is now possible to access other parameters' values, as shown in the below example. | Improvement | Architect | 16869 |
| Script language: Connecting parameters of type String, Boolean, Integer | It is now possible to use <Depends> to connect parameters of type **String**, **Boolean** and **Integer**. Until now, the use was supported for parameters of type Item.  The <Depends> tag defines a dependency of a Parameter to another Parameter, with a specified *name*. For example, a selection of a specific value in one parameter list can trigger the presentation of specific values in a second parameter. Example:  <Report>  <Parameters>  <Parameter caption="1st Selection" as="String" name="p1" defaultValue=" 'Yes'">  <Values>  <ForEach select="['Yes','No']">  <AddValue/>  </ForEach>  </Values>  </Parameter>  <Parameter caption="2nd Selection" name="p2" defaultValue=" 'Y1'">  <Log>---- #{? $p1}</Log>  <Depends name="p1"/>  <Values>  <Choose>  <When test="$p1 = 'Yes'">  <ForEach select="['Y1', 'Y2', 'Y3']">  <AddValue/>  </ForEach>  </When>  <When test="$p1 = 'No'">  <ForEach select="['N1', 'N2', 'N3']">  <AddValue/>  </ForEach>  </When>  </Choose>  </Values>  </Parameter>  </Parameters>  </Report> | Improvement | Architect | 15908 |
| Script language: New parameter types | With this release, we introduce two new parameter types: checkbox and text box.  A new, optional *type* attribute is used to specify checkbox or text box. As before, the default type is combobox. See the application Help for more information. | Improvement | Architect | 16945 |
| Script language: New prepare functionality | More work has been done on the Prepare functionality. Items that generally will be needed when working with the prepared items are also fetched, to reduce the number of server calls required for actually working with prepared data. For all items found by specified path expression, the DefObjs for all of their parts are also fetched, if they are items. | Improvement | Architect | 17005 |
| Script language: Split string to character | It is now possible to input a String and have it split into separate characters. This can be used, e.g., when you have filtered a string and then want to remove any type of character.  It could also be used when making attribute type definitions for ReqIF where the attribute type name there shall only contain alphanumerical characters.  **Syntax**  String.ToChar -> [Character]  **Example**   |  | | --- | | 'Hello World'.ToChar |   [H, e, l, l, o, , W, o, r, l, d] | Bug fix | Architect | 16657 |
| Script language: Support for DescriptionColumn with notes | It is now possible to use <DescriptionColumn> for Issue Notes and item Note.  Example Issue notes  <Grid>  <Options>  <CellAutoHeight/>  </Options>  <Columns>  <TextColumn caption="Changes" rowObject="issueNote">  #{?.}  </TextColumn>  <DescriptionColumn width="auto" caption="Issue Note" rowObject="issueNote"/>  </Columns>  <ForEachIssueInObj type="IR">  <ForEachNote type="NISS">  <RowObject name="issueNote">  <Row/>  </RowObject>  </ForEachNote>  </ForEachIssueInObj>  </Grid>  Example Item notes  <Grid>  <Options>  <CellAutoHeight/>  </Options>  <Columns>  <DescriptionColumn id="itemNote" caption="Note" objectName="itemNote" readOnly="false"/>  </Columns>  <ForEachNote type="NND">  <RowObject name="itemNote">  <Row>  </Row>  </RowObject>  </ForEachNote>  </Grid>  In Grids, Item Notes can be edited using the *readOnly*="false" attribute setting in DescriptionColumn. Issue Notes can never be edited. | Bug fix | Architect | 16893 |
| swArchitect: Loading Attributes is slow | The initial loading of the **Attributes** list in the swArchitect client has been optimized. | Bug fix | Architect | 15018 |
| swArchitect: Ordering of part default attributes | The feature for reordering the sequence of a part's **Default Attributes** is now more visible. Upon selection, the number is shown as editable. | Bug fix | Architect | 15523 |
| Workflow: Introduction of new feature | This release includes the first version of a Workflow process meta model and feature which can guide users through a work process consisting of activities and procedures. The workflow will inform the users which activities need to be done to fulfill a process, and also lets them know how the activities can be done in SystemWeaver (procedures). The user can then select a context in which they want to perform the activity, thereafter the user will be taken to the correct context and the appropriate views to perform the procedure will automatically load. Example Process data:  It's ideal to use for a particular domain, e.g., Cybersecurity, Adaptive Autosar, or for any other use case you have as our Workflow meta model is generic. The process can also include descriptions of procedures done outside of SystemWeaver. Procedures can be configured to guide users to the right context and the proper view(s) to use. This is a first version of the feature.  **Configuration**  Workflow is a Feature configured via *Configure the explorer*.  When configured and activated, the Workflow guide loads just below the structure tree via the **Overview** drop-down menu:  More details about the Workflow Process meta model and how to configure and use the feature can be found in the application Help. | New | Architect | 16780 |
| XML export/import: Misleading warning | This fixes a misleading "Invalid def obj type: XXXX in part of type YYYY" warning displayed when exporting, when, in fact, the def obj is correct. Now, the correct warning is displayed during XML import. | Bug fix | Architect | 13836 |
| ExtensionsAPI: Access to workflow permissions | There is now support in the SystemWeaverExtensionsAPI.dll for accessing issue workflow permissions.  IswWorkflowPermissions  Permissions here refers to workflow configurations with their definitions of Preconditions, NextStatuses, etc. See https://support.systemweaver.se/en/support/solutions/articles/31000149874-creating-a-workflow#Configuring-the-Workflow. | Improvement | Extensions API | 16748 |
| ExtensionsAPI: Open views in tabs in Multi-instance views | The Extensions API now supports the opening of all views including config item views (e.g., grid, graph, etc.) and extension views in both (single) ItemView and MultiInstanceItemView. To enable this, we have added OpenItemInView**()**and Host**.**GetCurrentNodePath**()** to both view types.  For OpenItemInView**()** , we have also added parameters for ConfigItem and ConfigId so all types of views and extensions can be opened.  void OpenItemInView**(**IswItem item**,** THowToOpenType howToOpenType**,** string viewName**,** string configItem **=** ""**,** string configId **=** ""**);**  **Simple view example**  You can retrieve the updated SystemWeaver templates from the SystemWeaver Support Portal where they are uploaded as attachments. | Improvement | Extensions API | 16766 |
| ExtensionsAPI: Updated VS templates | The SystemWeaver Visual Studio templates have been updated to support Visual Studio 2022. | Improvement | Extensions API | 16964 |
| ExtensionsAPI: Version update | The SystemWeaver Extensions API has been updated. This affects extensions placed in the swExplorerExtensions directory. Extension views must be recompiled against the updated SystemWeaverExtensionsAPI.dll. This is a breaking change. | Improvement | Extensions API | 16961 |
| Digitally signed installation files | The SystemWeaver installation files are now digitally signed. This will make it less likely you'll encounter anti-virus issues with SystemWeaver client and server applications. | Improvement | IT Administrator | 17018, 16612 |
| Server: New setting for mirror cache | There is a new option in the swServer.ini for the main server called AllowLocalCache. Supported values are True or False. True is the system default. If AllowLocalCache is set to False, mirror servers connecting to the main server can only run using in-memory cache (CacheInMemory=True).  As always, we do recommend the use of a cache file for mirror servers. CacheInMemory is a suitable option for smaller databases and/or QA environments only.  When upgrading to R42, both main and mirror servers must be updated. | Improvement | IT Administrator | 16670 |
| swConnectionTest: Status button not working | This fixes an issue where the Status operation resulted in an "Unable to connect to server" message. | Bug fix | IT Administrator | 12905 |
| Rest API: File version numbering | This fix ensures that the File version of the SystemWeaver.RestApi executable is updated. | Improvement | Rest API | 16474 |
| RestApi: Part with defObj as part gets empty name | This fixes an issue when you get an item that has an part with a part as defObj which returned an empty name for both parts due to the part's part getting its name from its own defObj. Names are now returned properly. | Improvement | Rest API | 16567 |
| ArchitecturalSystemSignals extension | The Architectural System Signal extension view has been updated to support multiple senders of the same signal allowing for platform architecture design: | Improvement | User |  |
| Clone item view: Identity attribute transformation not working | This fixes an issue where a configured transformation of an Identity type attribute to a String type attribute was not working. | Bug fix | User | 15986 |
| Clone item view: New options for referenced objects | When cloning items in a structure that includes parts with RefObj(s) (Referenced Object) to a part or item, there are now options to clone, ignore, or include RefObj(s) in the resulting clone. If no RefObj Option selection is made, ref objs are ignored by default, as before.  When you make your selections for RefObj, they are displayed in a new **RefObj option** column. The RefObj's name is also displayed.  A new attribute called *actionRefObj* has been added to the Clone item view configuration.  *actionRefObj* specifies the action for RefObj according to the part type. Supported values are include, clone and ignore.   * 'include' for RefObj means that the original object (Item or Part) included (as-is) in the cloned structure, without being cloned. * 'clone' for RefObj means that the original item or part will in fact be cloned. * 'ignore' for RefObj means that the original object (Item or Part) will be ignored.   If *actionRefObj* is not specified, the RefObj will be ignored by default, as before.  **Example**  <CloneParts>  <Part sid="6PRR" action="clone" actionRefObj="clone" traceparttype="QQQ">  <Include>@ABAN="include"</Include>  </Part>  </CloneParts>  SW Architects can refer to *Configure Clone item view in* the application Help for all configuration details. | Improvement | User | 12807 |
| Configurable graph: Confirmation dialog for redirect to tree | When working in **Edit** mode with manual layout, users will now receive a **Proceed** dialog should they mistakenly double-click on a node in the graph which redirects to the node's item in the structure tree. This action previously resulted in losing any unsaved manual layout changes. | Improvement | User | 16833 |
| Configurable graph: Ports sometimes remain selected | This fixes an issue with the selection of ports in the graph when the graph was locked. | Improvement | User | 16924 |
| Configurable graphs: Overlapping ports | This fixes many cases of ports overlapping making it difficult to read and follow the connections.  Before:  After:  After upgrade to Bagaregården (R42): If you are currently using the system default layout and have overlapping ports, consider, using the **Relayout** option in **Edit** mode to potentially lessen overlapping ports. Use **Save layout** to save the changes.  Note: The logic takes into account any *maxHeight* settings for Shapes. | Bug fix | User | 16346 |
| Coverage (item-item): Faulty mapping w parameters in right-side grid | This fixes an issue when using parameters to map items when a Grid is configured for the right-side pane. The Faulty mapping list was not updating/refreshing properly. | Bug fix | User | 16126 |
| Description editor: Data size bar-excess indicated in wrong color | This fixes the displayed color of excess data size in the Image Editor. | Bug fix | User | 16824 |
| Document Print: Prompt to Refresh and not printing message | This fixes some cases where a user is prompted to refresh for no apparent reason. A "Printer is not currently printing" message has also been removed in cases where it was not necessary, and when clicking Print and then clicking elsewhere in the client. | Bug fix | User | 16709 |
| Document swExport: Setting changes | When using the swExport to export Documents from SystemWeaver, note that the settings have changed. All settings now begin with a lower case letter. In addition, there is a new, required setting called *type*. The required value is 'document'. These are breaking changes for document export using the swExport tool. To view the new settings, visit the support portal. | Improvement | User | 16934, 16770, 16787, 16648 |
| FaultTreeGraph view: Fails to load | This fixes an issue where the FaultTreeGraph extension view did not load if the Failure rate (SID=FTAR) attribute was empty on any item in the FTA structure. | Bug fix | User | 15735 |
| Grid swExport: Introduction of grid export | With this release, we are introducing support for exporting a grid to MS Excel via command line using the swExport tool. Using this application, you can, e.g., implement your own program that utilizes the swExport.exe based on your organization's needs.  This tool's grid export functionality has no required meta model, or item structure. It is suitable for all configurable grid use cases, and provides support for specifying parameters.  Because swExport can now be used for both document and grid export, it is important to set the *type* attribute to "grid" for grid exports. See the Support Portal for more details.  **Example Command Lines**  Using Username and Password:  swExport serverName=sys7 serverPort=1345 serverUserName=john.doe serverPassword=test type="grid" itemID=x04000000000C5AD4 configItem=BIVD:ECU filename="C:\Work\ECU\_allocation.xlsx"  Using Network Authentication:  swExport serverName=sys7 serverPort=1345 type="grid" itemID=x04000000000C5AD4 configItem=BIVD:ECU filename="C:\Work\ECU\_allocation.xlsx"  With parameters:  swExport serverName=sys7 serverPort=1345 type=grid itemID=x04000000000A76C1 configItem=SREL:Release\_Summary parameter="(p1;x0400000000051B65)" parameter="(p2;'Iteration')" parameter="(p3; true)" filename="C:\temp\release\_summary.xlsx“  **Example Result**    **Known Behavior**  • Export is of plain text only, just as it is with all **Export to Excel** operations in the client. | New | User | 16934, 16770, 16787, 16753, 16648 |
| Incorrect access rights flash on in view/dialog | This fixes an issue where a "read-only" type of message flashed in the Coverage (item>item) view and the New item dialog even when the user had edit rights. | Bug fix | User | 13781 |
| Issue Editor: Copy button | There is now a **Copy issue** button in the Issue editor. Clicking the button puts the issue's SystemWeaver URL in your clipboard. This is useful to use when you want to copy the URL of an issue that is not displayed in the issue view that you currently have loaded.  You can then paste the URL as needed. | Improvement | User | 16801 |
| JIra2: Support for special characters in password | There is now support in our Jira extension view for Jira passwords containing most special characters (e.g. !&%,@). Unicode characters are not supported due to a Jira limitation. | Improvement | User | 16269 |
| MetaModelGraph: Default attributes not included in graphml output | Default attributes are now included in .graphml output. | Bug fix | User | 16393 |
| Print to PDF: Handle filename/paths too long for Window PDF printer | For some PDF print jobs, users received a message that the file path was too long for the pdf printer. Now, those print jobs will print to the selected location. | Improvement | User | 15263 |
| Project workflow: Assertion failure no statuses | This resolves an assertion failure that appeared after a workflow status deletion. | Bug fix | User | 15305 |
| References: Access violation sorting w node references | This fixes an access violation error thrown when sorting by Part Type when there are node references. | Bug fix | User | 16133 |
| Snapshot view: Various improvements | The following improvements have been made to the Snapshot view.  It is now possible to edit item Descriptions by double-clicking on the Description to open the editor.  In addition, Description scrollbars have been removed, and the Description will auto-size based on the data.  The view now also includes two new options - Lock and Refresh.   * **Lock** enables you to lock the view on the selected item. * **Refresh** will refresh the contents of the view.   In addition, any images included in an item Description will now appear in the view.  **Note:** There is no support for viewing values for RVF, Calcuated, Custom, Array of External Reference, and XML type attributes at this time. | Improvement | User | 16803, 16808, 16909 |
| Structure tree: Secondary item view selection | This fixes a problem where the selection of Secondary item view via the right-click menu in the structure tree appeared to remain activated, even after the view was closed. | Bug fix | User | 15701 |
| swExplorer: Unresponsive after using alt+tab and Chinese language | This fixes a recently introduced problem with using the swExplorer client while performing the Windows Alt+Tab shortcut with Chinese set as the Windows Language. | Bug fix | User | 16888 |
| SystemSignalGenerator extension | The System Signal generator extension view has been updated to support multiple senders of the same signal allowing for platform architecture design:  (The above example shows functionality. Name uniqueness is supported.) | Improvement | User |  |