| **Name** | **Description** | **Type** | **User Group** | **Issue ID** |
| --- | --- | --- | --- | --- |
| swAdmin2: Last Login filtering and tools | This fixes intermittent issues with sorting and filtering user accounts by Last Login. | Bug fix | Administrators | 13097, 14597 |
| ExtensionsAPI: DescriptionGenerator | There is now a method available to create simple formatted descriptions in our extensions API. This is an experimental version of the functionality and may be subject to changes, or may be removed. | Improvement | API user | 14495 |
| VS Templates: Installing for VS 2019 | We now provide a .vsix file with our SystemWeaver templates package that can be used to install our templates in VS 2019. Just run the .vsix as administrator, and they should be automatically copied to the Visual Studio installation directory and registered with the Visual Studio development environment. See [Getting Started With swExplorer Extensions](https://support.systemweaver.se/support/solutions/articles/31000133772-getting-started-with-swexplorer-extensions) for more information. | Improvement | API User | 13696 |
| Path query: New method Mother on nodes | We now have the method **Mother** valid for nodes which switches context from the current (context) node to the corresponding node of a tree with one level shallower, i.e., where the current Owner node would not be. The Mother can be seen as the template or generatrix for the current node structure.    In the *Node Reference Example* above, the *Mother* of the nodes are according to the table below (Note that top nodes do not have a Mother)   |  |  | | --- | --- | | Node | Mother | | N1 | - | | N2 | N4 | | N3 | N5 | | N4 | - | | N5 | N6 | | N6 | - | | Improvement | Architect | 14710 |
| Reports: Support for <BUS> in GenerateGraph | There is now support for the use of <BUS> in GenerateGraph in reports. Example: | Bug fix | Architect | 14542 |
| Create test: New meta model pattern supported | The Create Test view has been updated to support a new meta model pattern using Abstract types for Test Scope, Test Specification and Test case to enable some more freedom when working with the Test meta model.  Old meta model that is still supported:    New meta model: | Improvement | Architect | 14682 |
| Coverage (mapping item): Support for parameters | There is now support for parameters in the Coverage (mapping item) view which makes it possible for a user to use the view in a local context rather than having to load a much larger model than needed.  The Parameters tag is added, as usual, directly at the top of the individual xml configuration and gives parameter values available globally for the entire view.  Depending on which type of grid or tree you use, you get access to the parameters as follows:  <CoverageMappingItemConfig id= "unique\_idd2p">  <ViewSettings>  <Caption>Requirement mapping</Caption>  <RibbonGroup>Mapping</RibbonGroup>  </ViewSettings>  <TopItem itemType="BIDM"/>  **<Parameters>**  <Parameter name="p1" caption="Parameter">  <Values>  <ForEach select="/REQS">  <AddValue/>  </ForEach>  </Values>  </Parameter>  **</Parameters>**  <Mappings>  <PathFromMainToMappingContainer select="**$p1**/DEPA/AIML" />  ....  </Mappings>  <LeftSide>  <Grid>  <Columns>  ....  </Columns>  <ForEach select=**"$p1"**>  <Row/>  </ForEach>  </Grid>  <StructureTree>  <TopItemPath>**$p1**/IGRF</TopItemPath>  </StructureTree>  <ContextTree>  <Context>  <AddParts owner="main" sid="PM1P" part="PrimaryMappings" defobj="PrimaryMapping"/>  <AddParts owner=**"$p1"** sid="IGRP" part="PrimaryMappings" defobj="PrimaryMapping"/>  </Context>  </ContextTree>  </LeftSide>  <RightSide>  ....  </RightSide>  </CoverageMappingItemConfig> | Improvement | Architect | 14661 |
| Script Language: Support for more data types in parameters | The existing functionality of Parameters has been extended to support, not only items, but also, e.g., strings, integers and boolean.  There is now an optional *as* attribute in the Parameter definition. To use for strings or integers, you must use *as.* Example:  <Parameters>  **<Parameter caption="Section" name="p1" as="Integer" >**  <Values>  <ForEach select="[1, 2, 3]">  <AddValue/>  </ForEach>  </Values>  </Parameter>  </Parameters>  <Parameters>  **<Parameter caption="Section" name="p1" as="String" >**  <Values>  <ForEach select="['One', 'Two', 'Three']">  <AddValue/>  </ForEach>  </Values>  </Parameter>  </Parameters>  <Parameters>  **<Parameter caption="Section" name="p1" as="Boolean" >**  <Values>  <ForEach select="[true, false]">  <AddValue/>  </ForEach>  </Values>  </Parameter>  </Parameters>  For backwards compatibility, if you do not have an *as* defined, it will default to "Item".  Only parameters of type Item are added to contexts. Other parameter values are only available through variables. | Improvement | Architect | 14546 |
| swArchitect: Path for computed attribute not saved | This fixes an issue in the swArchitect client where the value set in **Path**, when a computed attribute was being created, was not being saved.  In addition, the **Default value** option has been removed from the Create Attribute Type dialog when creating Computed attributes since it is not applicable. | Bug fix | Architect | 14615 |
| Script Language: Added support for /axis::\* | When working with axises, it is now possible to use an asterisk, "\*" to follow all part types.  It is valid for the following axises:   * part * back * back-to-part * refobj-back-to-part * issueref * objref | Improvement | Architect | 14457 |
| Reports: DescriptionIsEmpty filter does't work when deleting description | This fixes an issue where DescriptionIsEmpty filter did not indicate a description was empty after clearing the description content. | Bug fix | Architect,User | 14234 |
| swExplorer: 3rd party notices | A complete list of third-party libraries or other resources is now available for viewing on the About tab in the swExplorer and swArchitect clients. | Improvement | IT Admin | 14684 |
| Mirror: In memory option | We now offer the option to run a mirror server without having a local cache file on the machine running the mirror.  It works by having the mirror server load all data from the main server at startup.  This might be suitable for smaller installations, i.e., with a small database size, or in a QA environment where you have less requirements on the uptime of the main server.  It is not suitable for larger databases, i.e., > 1 Gb because the main server will stop responding to any other requests while the in-memory mirror server is loading data. The server will stop for about the same time as it it takes for the main server to start.  Journaling still has to be enabled on the main server for this to work.  However, we always recommend running the mirror with a local cache if possible, since this gives all the advantages of the full mirror server solution including fast startup time, and with load on the main server. | Improvement | IT Admin | 14540 |
| REST API: Move to .NET Core 3.1 | The SystemWeaver Rest API has moved from .NET Framework to .NET Core version 3.1. | Improvement | REST API user | 14626 |
| REST API: Add issue types to project endpoint | There is now issue reference type and attribute type information on the project endpoint to give required information for issue creation. | Improvement | REST API user | 14567 |
| REST API: Issue filtering endpoint | For the issue endpoint, it is now possible to request issues and filter them using an XML-filter in the same way XML filters can be used in the swExplorer. | Improvement | REST API user | 14446 |
| REST API: Better error message for property values | The error message "The value has invalid JSON" has been improved to better describe the expected input. | Improvement | REST API user | 14487 |
| REST API: Add workflow handle to issue type response | A workflow handle attibute has been added to the issue type response so that a workflow can be matched to an issue type. | Improvement | REST API user | 14506 |
| REST API: Add get/post/patch for relationships | * When executing Get for a Project, the response will include IssueRefType and AttributeType information * When executing Get for an Issue, the response the response will include reference objects and reference issues * When executing Post or Patch for an Issue, commands for add, updating and deleteing reference objects are implemented * A new endpoint for retrieving an item with reference issues has been added | Improvement | REST API user | 14632, 14631, 14633, 14510 |
| REST API: Get all Users in User endpoint | A new endpoint for retrieving all users has been added.  restapi/users | Improvement | REST API user | 13972 |
| REST API: Read access denied not handle on Item | This fixes an unhandled exception thrown when the RestAPI tried to read an Item without having read access when server Security Level is set to 3. | Bug fix | REST API user | 14613 |
| REST API: Unable to create a part between items with single multiplicity | This fixes an issue where users could not create a part between two items that have single multiplicity. | Bug fix | REST API user | 14600 |
| TestCaseManager: Multiselect of requirements when creating test cases | In the Test Case Manager view, it is now possible for users to multi-select requirements and create Test Cases for each of the selected requirements.    An individual test case will be created for each selected requirement. Each test case will be named after its requirement, and can be modified afterwards if needed. | Improvement | User | 14512, 13731 |
| Document: Improved table format when saving to RTF | Some improvements have been made to the .rtf output from the Document view for both Attribute tables and Change log. Users are still informed, however, that some formatting will not transfer to the RTF file.  It is important to note that although an RTF output of a document can be subsequently modified in MS Word, it is not ideal to modify documents outside of the SystemWeaver system since then you cannot trust or know that a released document in the system is actually what was delivered. By design and to maintain data integrity, there is no way within SystemWeaver to modify a generated PDF. By not manipulating the output, it secures that you know what was generated and what was delivered. | Bug fix | User | 14547 |
| Projects: Optimized CPU usage in issue views | We have optimized CPU usage when multi-selecting issues. | Bug fix | User | 14563, 14566 |
| Issues: Print and print preview options | To clarify printing options, Print and Print Preview buttons have been added to the Sand Box dialog for printing individual issues. | Improvement | User | 14635 |
| Configurable graph: Various improvements | * The scaling of the component shapes listed below has been improved so as to fit all interfaces/ports and minimize the overlapping of ports and connection names on the components.   + Chevron   + Ellipse   + Parallelogram   + Rectangle * The NodeGroup header will now always display as one row in height. In this way, sub-nodes will not cover any part of the header. * Should a NodeGroup header be too long, it will now be truncated with "...". A new tooltip is provided when hovering over a graph object to allow viewing of the entire header. The graph objects that support the tooltip are:   + Port (tooltip = the port name)   + Node/node inside node (tooltip = the node name)   + Edge (tooltip = the edge name)   + Group (tooltip = the node group name) * When selecting a GroupNode, its immediate sub-nodes will now also be displayed in the Neighborhood pane. | Improvement | User | 14293, 14250, 14681 |
| Issues: Change log for attachments | It is now possible to add a comment when adding, modifying, and removing an attachment file to an issue in a project.    It is also now possible to view the change log for not only a single attachment to an issue, but also for all added, modified, and/or removed attachments to an issue.    Information about the attachment events (e.g., adds/changes/deletions) can be exported to Excel or CSV:    The attachment events along with the comments can also be printed using <AttachmentChangeNoteHeader/>: | Improvement | User | 13267 |
| Enum Attributes: Improved support for clearing values | This enhancement relates to enum attributes with descriptions.    There is now a clear button  available for clearing values in a grid (if configured to allow editing).  If the attribute allows empty values, the cell will appear white.  If it is not allowed, it will appear red as before.    The clear option is also available in the Part Attributes "grid" in the Attributes view. | Improvement | User | 14497 |
| Workflow restrictions: Access violation | This fixes an access violation error that sometimes occurred when the issue editor was open and a "null" issue was in the issue list when you right-clicked on the page. | Bug fix | User | 14468 |
| Coverage (mapping item): Unnecessary coupling of editable conditions | This fixes an issue with editability on the right side of the mapping view when the primary side editable parts (LeftStructureTreeEditableParts) had a blank value. | Bug fix | User | 14675 |
| Coverage (mapping item): Activate/deactivate options based on mapping container | This fixes an issue where the view options (buttons) were not always activated correctly as they should based on the meta model and determining item status. Given this, to keep users from mapping, the mapping container would need to be released or frozen, and not the top item. The fix also includes an improved error message. | Bug fix | User | 14715, 13994 |
| System Signals: Gray out Include system signal option | This fixes an issue in the System Signals built-in view where an access violation error was thrown when using the available Include system signal option when a selected signal did not have a "Sending ECU" and/or "Receiving ECU". Because the option is not valid if either of these are null, the option has been grayed out in this scenario. | Bug fix | User | 14653 |
| Float attribute: Unable to manually enter negative value in grids | This fixes an issue where the user was unable to enter a negative value (with minus symbol) for a float attribute on a part. | Bug fix | User | 14651 |
| Extensions: Hydra focus problem | This fixes a focus problem introduced with the upgrade to Hydra 6.2.101.1237 causing users to sometimes have to click twice to interact with the extension view. | Improvement | User | 14472 |
| Date Attribute: Empty value throws error message | This fixes an issue where an empty Date attribute threw an unexpected date type error. | Bug fix | User | 14406 |
| Issue Attributes: Problem with computed attributes | This fixes a problem where Computed attributes for issues didn't work when you tried to use values from other attributes on issues. | Bug fix | User, Architect | 13762 |
| TcpSubServer: Change to mirror calls | This fixes an issue where a few read calls were still going directly to the main server instead of the mirror which resulted in slowness for the user during those operations. | Improvement | User, IT Admin | 14709, 14683 |