Syndeia Features

Syndeia 3.1

Table of Contents

Syndeia is packed with hundreds of features. The listing below provides a tabulation of Syndeia features.

1   Syndeia and DOORS-NG

DO1	**Connect to one or more DOORS-NG repositories and view contents** Connect to one or more DOORS-NG repositories (servers) in your organization by supplying the URL and your credentials.
DO2	**View contents of DOORS-NG repositories** View the contents of DOORS-NG repositories, such as projects, folders, requirements, requirement collection structures, multi-level modules, and links.
DO3	**Search DOORS-NG repositories** Search DOORS-NG repositories for requirements, collections, or modules by specifying their unique identifiers.
DO4	**Create Reference connections between SysML model elements and DOORS-NG elements** *Drag-n-drop* a SysML model element to any DOORS-NG repository element (e.g. project / folder / requirement / collection / module), or vice versa, to create a reference connection. A reference connection is the most basic form of traceability between SysML model elements and DOORS-NG repository elements. A reference connection can be used to check if the version of the DOORS-NG artifact has changed.
DO5	**Generate DOORS-NG requirements and requirement collections from SysML requirements, and create Data Map or Model Transform connections** *Drag-n-drop* a SysML requirement to a DOORS-NG project or folder to generate DOORS-NG requirements or requirement collections. This also creates connections between the SysML requirements and DOORS-NG requirements or collections that can be used for compare and bi-directional sync operations. Data Map connections can be used to compare/sync requirement attributes while the Model Transform connections can be used to compare/sync requirement attributes and structure.
DO6	**Generate SysML requirement structures from DOORS-NG requirements, collections, or modules, and create Data Map or Model Transform connections** *Drag-n-drop* a DOORS-NG requirement or collection or module to a SysML package to generate SysML requirement structure. This also creates connections between the SysML requirements and DOORS-NG requirements / collections / modules that can be used for compare and bi-directional sync operations. Data Map connections can be used to compare/sync requirement attributes while the Model Transform connections can be used to compare/sync requirement attributes and structure.
DO7	**Compare SysML requirement structure and DOORS-NG requirement / collection / module** Invoke the *Compare* operation on connections between SysML requirements and DOORS-NG requirements / collections / modules to generate a difference table that shows the in-sync and out-of-sync aspects of the two sides.
DO8	**Sync SysML requirements --> DOORS-NG requirements / collections** Invoke the _Sync SysML --> Target_ operation on connections between SysML requirements and DOORS-NG requirements / collections to compare and update the DOORS-NG requirements / collections based on the SysML requirement properties and structure.
DO9	**Sync DOORS-NG requirements / collections / modules --> SysML requirement structures** Invoke the _Sync Target --> SysML_ operation on connections between SysML requirements and DOORS-NG requirements / collections / modules to compare and update the SysML requirement structure based on the DOORS-NG requirement / collection / module.

DO10	**Connect existing SysML requirements and DOORS-NG requirements / collections / modules**
	Drag-n-drop an existing SysML requirement to a DOORS-NG requirement / collection / module (or vice versa) to create connections for downstream compare/sync.

2   Syndeia and GitHub

G1	**Connect to one or more GitHub servers** *Connect* to one or more GitHub servers by supplying the URL and your credentials.
G2	**View contents of GitHub servers** View the contents of GitHub servers, such as public/private software repositories owned by the user and organization, branches, tags, commits, files and folders.
G3	**Create Reference connections between SysML model elements and GitHub elements** *Drag-n-drop* a SysML model element to any GitHub repository element, e.g. file, branch, commit, or tag, or vice versa, to create a reference connection. A reference connection is the most basic form of traceability between SysML model elements and GitHub repository elements. A reference connection can be used to check if the version of the GitHub object (file, folder, commit) has changed.

3   Syndeia and Jama

JA1	**Connect to one or more Jama repositories and view contents** Connect to one or more Jama repositories (servers) in your organization by supplying the URL and your credentials.
JA2	**View contents of Jama repositories** View the contents of Jama repositories, such as projects, folders, requirements, test cases, and other Jama items.
JA3	**Search Jama repositories** Search Jama repositories for any Jama item, such as requirements.
JA4	**Create Reference connections between SysML model elements and Jama elements** *Drag-n-drop* a SysML model element to any Jama repository element (e.g. project / folder / requirement / test case), or vice versa, to create a reference connection. A reference connection is the most basic form of traceability between SysML model elements and Jama repository elements. A reference connection can be used to check if the version of the Jama artifact has changed.
JA5	**Generate Jama requirements and requirement structure from SysML requirements, and create Data Map or Model Transform connections** *Drag-n-drop* a SysML requirement to a Jama project or folder to generate Jama requirements and/or requirement structures. This also creates connections between the SysML requirements and Jama requirements that can be used for compare and bi-directional sync operations. Data Map connections can be used to compare/sync requirement attributes while the Model Transform connections can be used to compare/sync requirement attributes and structure.
JA6	**Generate SysML requirement structures from Jama requirements and sets, create Data Map or Model Transform connections** *Drag-n-drop* a Jama requirement (which may have child requirements) or a set to a SysML package to generate SysML requirement structure. This also creates connections between the SysML requirements and Jama requirements that can be used for compare and bi-directional sync operations. Data Map connections can be used to compare/sync requirement attributes while the Model Transform connections can be used to compare/sync requirement attributes and structure.
JA7	**Compare SysML requirements and Jama requirements** Invoke the *Compare* operation on connections between SysML requirements and Jama requirements to generate a difference table that shows the in-sync and out-of-sync aspects of the two sides.
JA8	**Sync SysML requirements → Jama requirements** Invoke the _Sync SysML → Target_ operation on connections between SysML requirements and Jama requirements to compare and update the Jama requirements based on the SysML requirement properties and structure.
JA9	**Sync Jama requirements → SysML requirements** Invoke the _Sync Target → SysML_ operation on connections between SysML requirements and Jama requirements to compare and update the SysML requirements based on the Jama requirements.
JA10	**Connect existing SysML requirements and Jama requirements** Drag-n-drop an existing SysML requirement to a Jama requirement (or vice versa) to create connections for downstream compare/sync.

4 Syndeia and JIRA

J1	**Connect to one or more JIRA repositories** *Connect* to one or more JIRA repositories (servers) in your organization by supplying the URL and your credentials.
J2	**View contents of JIRA repositories** View the contents of a JIRA repository, such as projects, assigned issues, and details directly from the Syndeia Dashboard.
J3	**Search JIRA repositories** Search for JIRA issues using an easy-to-use query builder or directly entering JQL queries from the Syndeia Dashboard.
J4	**Create Reference connections between SysML model elements and JIRA elements** *Drag-n-drop* a SysML model element to any JIRA repository element (e.g. project, issue), or vice versa, to create a reference connection. A reference connection is the most basic form of traceability between SysML model elements and JIRA repository elements. A reference connection can be used to check if the version of the JIRA issue has changed.
J5	**Generate JIRA issues from SysML blocks (stereotyped as JIRA issue), and create Data Map or Model Transform connections** *Drag-n-drop* a SysML block to a JIRA repository to generate a single issue (or an issue structure from block structure) with attributes populated from block value properties. This also creates connections between the SysML blocks and JIRA issues that can be used for compare and bi-directional sync operations. Data Map connections can be used to compare/sync block-issue attributes while the Model Transform connections can be used to compare/sync block-issue attributes and structure.
J6	**Generate SysML blocks (stereotyped as JIRA issue) from JIRA issues, and create Data Map and Model Transform connection** *Drag-n-drop* a JIRA issue to a SysML package to generate a block structure with value properties (based on issue attributes). This also creates connections between the SysML blocks and JIRA issues that can be used for compare and bi-directional sync operations. Data Map connections can be used to compare/sync block-issue attributes while the Model Transform connections can be used to compare/sync block-issue attributes and structure.
J7	**Seed JIRA issues from SysML block, requirement, or activity structure** Drag-n-drop a SysML block / requirement / activity to generate JIRA issue structure from block / requirement / activity structure. This also creates reference connection between the SysML elements and linked JIRA issues that can be used for checking if the issues have been updated.
J8	**Compare SysML block structure with value properties and JIRA issue structure with attributes** Invoke the *Compare* operation on connections between SysML blocks and JIRA issues to generate a difference table that shows the in-sync and out-of-sync aspects of the two structures.
J9	**Sync SysML block structure with value properties --> JIRA issue structure with attributes** Invoke the _Sync SysML --> Target_ operation on connections between SysML blocks and JIRA issues to compare and update the JIRA issue structure and attributes based on the SysML block structure and value properties.
J10	**Sync JIRA issue structure with attributes --> SysML block structure with value properties** Invoke the _Sync Target --> SysML_ operation on connections between SysML blocks and JIRA issues to compare and update the SysML block structure based on the JIRA issue structure.
J11	**Review JIRA project status from SysML** Select a package in the SysML model and view all issues connected to elements in that package

	(recursively), and get JIRA statistics such as number of: (1) connected issues, (2) open issues, (3) issues assigned to a specific user, or (4) issues reported by a user. Download as an Excel report.
J12	**Connect existing SysML block and JIRA issues** Drag-n-drop an existing SysML block to a JIRA issue (or vice versa) to create connections for downstream compare/sync.

5 Syndeia and Local Models (NX, Creo, Simulink, Stateflow, Excel).

L1	**Connect to one or more local file system repositories** *Connect* to folders in your local file system or shared drives and view their contents directly from the Syndeia Dashboard launched from your SysML modeling tool.
L2	**Create Reference connections between SysML model elements and files/folders** *Drag-n-drop* a SysML model element to a folder / file on your local file system / shared drive (or vice versa) to create a reference connection. A reference connection is the most basic form of traceability between SysML model elements and model files / folders created in other tools.
L3 - Siemens NX (CAD)	
L3.1	**View details of NX CAD models** Expand NX part files (.prt) to view the NX part/assembly structure. Create reference connections between SysML model elements and NX model elements.
L3.2	**Generate SysML block structure from NX part structure** *Drag-n-drop* a NX part/assembly structure to generate SysML block structure. This also brings mass properties (such as mass, volume, surface area, density, center-of-gravity, and bounding box dimensions) of NX parts to SysML, and provides options to generate part features and feature expressions/attributes in SysML. It also creates a connection between SysML blocks and NX parts for downstream compare and sync operations.
L3.3	**Compare and sync SysML blocks and NX parts** Invoke compare and sync services on Block – NX part connections to compare mass property values between SysML and NX models. Sync mass property values from NX -> SysML model.
L3.4	**Generate NX parts with features from SysML block structure** *Drag-n-drop* a SysML block structure with definitions of primitive features, such as cuboids, spheres, cylinders, cones, planes, lines, and points, to generate a NX part with primitive features. Communicate system level geometric requirements such as bounding box and keep-out zones as geometry to mechanical/electrical designers. This also creates a connection between SysML blocks and NX parts for downstream compare and sync operations.
L3.5	**Connect existing SysML block and NX CAD model (part/assembly)** Drag-n-drop an existing SysML block to a NX part/assembly (or vice versa) to create connections for downstream compare/sync.
L4 - PTC Creo (CAD)	
L4.1	**View details of Creo CAD models** Expand Creo assembly files (.asm) and part files (.prt) to view the part/assembly structure. Create reference connections between SysML model elements and Creo model elements.
L4.2	**Generate SysML block structure from Creo model** *Drag-n-drop* a Creo assembly model to generate SysML block structure. This also brings mass properties (such as mass, volume, surface area, density, center-of-gravity, and bounding box dimensions) of Creo models to SysML, and provides the option to generate model attributes in

	SysML. It also creates a connection between SysML blocks and Creo models for downstream compare and sync operations.
L4.3	**Compare and Sync SysML blocks and Creo models** Invoke compare and sync services on Block – Creo model connections to compare mass property values between SysML and Creo models. Sync mass property values from Creo -> SysML model.
L4.4	**Connect existing SysML block and Creo CAD model (part/assembly)** Drag-n-drop an existing SysML block to a Creo part/assembly (or vice versa) to create connections for downstream compare/sync.
	L5 – MathWorks Simulink
L5.1	**View Simulink model structure** Expand Simulink model file (.slx or .mdl) and view the model structure. Create reference connections between SysML model elements and Simulink model elements.
L5.2	**Generate Simulink models from SysML internal block structure** *Drag-n-drop* a SysML block to a folder to generate a Simulink model. This will create a Simulink model structure based on the SysML internal block structure, and also create connections between the SysML blocks and Simulink models that can be used for compare and sync operations.
L5.3	**Compare SysML internal block structure and Simulink model structure** Invoke the *Compare* operation on connections between SysML blocks and Simulink models to generate a table that shows the SysML block and Simulink model elements that are in-sync and out-of-sync.
L5.4	**Sync SysML internal block structure → Simulink model** Invoke the *Sync SysML → Target* operation on connections between SysML blocks and Simulink models to compare the two model structures, and update the Simulink model from the SysML block.
L5.5	**Sync Simulink model → SysML internal block structure** Invoke the *Sync Target → SysML* operation on connections between SysML blocks and Simulink models to compare the two model structures, and update the SysML internal block structure from the Simulink model.
L5.6	**Generate SysML internal block structure from Simulink models** *Drag-n-drop* a Simulink model to a SysML package to generate a SysML block structure. This will create SysML blocks, part properties, ports, and connectors from the Simulink model structure, and create connections between the SysML blocks and Simulink models that can be used for compare operations.
L5.7	**Generate Simulink models from SysML activity structure** *Drag-n-drop* a SysML activity to a folder to generate a Simulink model. This will create a Simulink model structure based on the SysML activity structure, and also create connections between the SysML activities and Simulink models that can be used for compare and sync operations.
L5.8	**Compare SysML activities structure and Simulink model structure** Invoke the *Compare* operation on connections between SysML activities and Simulink models to generate a table that shows the SysML activity and Simulink model elements that are in-sync and out-of-sync.
L5.9	**Sync SysML activity → Simulink model** Invoke the *Sync SysML → Target* operation on connections between SysML activities and

	Simulink models to compare the two model structures, and update the Simulink model from the SysML activity.
L5.10	**Sync Simulink model → SysML activity** Invoke the *Sync Target → SysML* operation on connections between SysML activities and Simulink models to compare the two model structures, and update the SysML activity from the Simulink model.
L5.11	**Generate SysML activity model structure from Simulink models** *Drag-n-drop* a Simulink model to a SysML package to generate a SysML activity structure. This will create SysML activities, call behavior actions, activity parameter nodes, and object flows from the Simulink model structure, and also create connections between the SysML activities and Simulink models that can be used for compare operations.
L5.12	**Connect existing SysML block/activity to a Simulink model** Drag-n-drop an existing SysML block/activity to a Simulink model (or vice versa) to create connections for downstream compare/sync.
L6 – MathWorks Stateflow	
L6.1	**View Statechart model structure** Expand Statechart subsystems in Simulink model (.slx) and view the model structure. Create reference connections between SysML model elements and statechart model elements.
L6.2	**Generate statechart from SysML state machine** *Drag-n-drop* a SysML state machine to a folder to generate a Simulink model with a statechart corresponding to the SysML state machine. This will also create a connection between the SysML state machine and statechart that can be used for compare and sync operations.
L6.3	**Compare SysML state machine and Simulink / Stateflow statechart model** Invoke the *Compare* operation on connections between SysML state machine and Simulink/Stateflow statechart to generate a table that shows the differences between the structures of the two models.
L6.4	**Sync SysML state machine → Simulink / Stateflow statechart** Invoke the *Sync SysML → Target* operation on connections between SysML state machines and Simulink/Stateflow statecharts to compare the two model structures, and update the statechart model from the SysML state machine.
L6.5	**Sync Simulink/Stateflow statechart → SysML state machine** Invoke the *Sync Target → SysML* operation on connections between SysML state machine and Simulink/Stateflow statecharts to compare the two model structures, and update the SysML state machine from the statechart model.
L6.6	**Create connections between existing SysML state machine and Simulink/Stateflow statechart model** that can be used for downstream compare and sync operations.
L7 – Excel	
L7.1	**View Excel workbooks and worksheets** Expand an Excel workbook file (.xlsx or .xls) to view all spreadsheets, named references, tables, and rows.

L7.2	**Generate Excel tables from SysML blocks** *Drag-n-drop* a SysML block to a folder, or Excel workbook / worksheet to create a table. This also creates a connection between the block and Excel table that can be used for compare and bi-directional sync operations.
L7.3	**Generate Excel table rows from SysML block instances / specialized blocks** *Drag-n-drop* instances or specializations of the SysML block connected to an Excel table to generate rows in that table. This also creates connections between the block instances / specialized blocks and Excel table rows that can be used for compare and bi-directional sync operations.
L7.4	**Generate SysML blocks from Excel tables** *Drag-n-drop* Excel tables to a SysML package to generate SysML blocks corresponding to the table. This also creates connections between the Excel tables and SysML blocks that can be used for compare and bi-directional sync operations.
L7.5	**Generate SysML block instances / specialized blocks from Excel table rows** *Drag-n-drop* Excel table rows to a SysML package to generate SysML instances / specializations of the block connected to the table. This also creates connections between the Excel table rows and SysML block instances / specialized blocks that can be used for compare and bi-directional sync operations.
L7.6	**Compare SysML block instances / specialized blocks and Excel table rows** Invoke the *Compare* operation on connections between SysML block instances / specialized blocks and Excel table rows to generate a table that shows the specific values that are in-sync and out-of-sync between SysML elements and table rows.
L7.7	**Sync SysML block instances / specialized blocks --> Excel table rows** Invoke the _Sync SysML → Target_ operation on connections to compare and push values from SysML instances / specialized blocks to Excel table rows.
L7.8	**Sync Excel table rows --> SysML block instance / specialized blocks** Invoke the _Sync Target → SysML_ operation on connections to compare and push values from Excel table rows to SysML instances / specialized blocks.
L7.9	**Connect existing SysML block/instance and Excel table/row** Drag-n-drop an existing SysML block/instance to an Excel table/row (or vice versa) to create connections for downstream compare/sync.

6   Syndeia and MySQL

S1	**Connect to one or more MySQL repositories** *Connect* to one or more MySQL repositories (servers) in your organization by supplying the URL and your credentials.
S2	**View contents of MySQL repositories** View the contents of a MySQL repository, such as databases, tables, and rows, in the Syndeia Dashboard.
S3	**Search MySQL repositories** Search for data in MySQL repositories using an easy-to-use query builder or directly entering MySQL queries from the Syndeia Dashboard.
S4	**Create Reference connections between SysML model elements and MySQL elements** *Drag-n-drop* a SysML model element to a database / table / row in a MySQL repository, or vice versa, to create a reference connection. A reference connection is the most basic form of traceability between SysML model elements and data managed in a MySQL repository.
S5	**Generate MySQL tables from SysML blocks** *Drag-n-drop* a SysML block to a MySQL database element to create a table. This also creates a connection between the block and MySQL table that can be used for compare and bi-directional sync operations.
S6	**Generate MySQL table rows from SysML block instances / specialized blocks** *Drag-n-drop* instances / specializations of a SysML block connected to a MySQL table to generate rows in that table. This also creates connections between the block instances / specialized blocks and MySQL table rows that can be used for compare and bi-directional sync operations.
S7	**Generate SysML blocks from MySQL tables** *Drag-n-drop* MySQL tables to a SysML package to generate SysML blocks corresponding to the table. This also creates connections between the MySQL tables and SysML blocks that can be used for compare and bi-directional sync operations.
S8	**Generate SysML block instances / specialized blocks from MySQL table rows** *Drag-n-drop* MySQL table rows to a SysML package to generate SysML instances / specializations of the block connected to the table. This also creates connections between the MySQL table rows and SysML block instances / specialized blocks that can be used for compare and bi-directional sync operations.
S9	**Compare SysML block instances / specialized blocks and Excel table rows** Invoke the *Compare* operation on connections between SysML block instances / specialized blocks and MySQL table rows to generate a difference table that shows the specific values that are in-sync and out-of-sync between SysML elements and table rows.
S10	***Sync SysML block instances / specialized blocks --> MySQL table rows*** Invoke the _Sync SysML --> Target_ operation on connections to compare and push values from SysML instances / specialized blocks to MySQL table rows.
S11	***Sync MySQL table rows --> SysML block instance / specialized blocks*** Invoke the _Sync Target --> SysML_ operation on connections to compare and push values from MySQL table rows to SysML instances / specialized blocks.

7   Syndeia and Simulink

See the section Syndeia and Models in Local File System (NX, Creo, Simulink, Stateflow, Excel).

8   Syndeia and Stateflow

See the section Syndeia and Models in Local File System (NX, Creo, Simulink, Stateflow, Excel).

9   Syndeia and Teamcenter / NX

T1	**Connect to one or more Teamcenter repositories** *Connect* to one or more Teamcenter repositories (servers) in your organization by supplying the URL and your credentials.
T2	**View the contents of Teamcenter repositories** View contents of Teamcenter repositories, such as folders, items / parts / requirements / requirement specs (and their revisions), and trace links directly from the Syndeia Dashboard.
T3	**Search Teamcenter repositories** Search Teamcenter repositories for any item, such as item/part/requirement/spec, using item name or item id. View search results and drag-n-drop them to the SysML model.
T4	**Create Reference connections between SysML model elements and Teamcenter elements** *Drag-n-drop* a SysML model element to any Teamcenter repository element, such as item / part / requirement / requirement spec / paragraph (or their revisions), trace links, BOM lines, or folders, to create a reference connection. A reference connection is the most basic form of traceability between SysML model elements and Teamcenter repository elements. A reference connection can be used to check if the version of the Teamcenter item has changed.
T5	**Generate Teamcenter item structure from SysML block structure** *Drag-n-drop* a SysML block to a Teamcenter folder to generate an item structure (items and item usages – BOM lines) in Teamcenter from the SysML block structure (blocks and part properties). This also creates connections between the SysML blocks and Teamcenter item revision that can be used for compare and bi-directional sync operations.
T6	**Generate SysML block structure from Teamcenter item / part structure** *Drag-n-drop* a Teamcenter item / part (or their revisions) to a SysML package to generate a block structure (blocks and part properties). This also creates connections between the SysML blocks and Teamcenter items / parts that can be used for compare and bi-directional sync operations.
T7	**Compare SysML block structure and Teamcenter item / part structure** Invoke the *Compare* operation on connections between SysML blocks and Teamcenter items / parts to generate a difference table that shows the in-sync and out-of-sync aspects of the two structures.
T8	***Sync SysML block structure ⮂ Teamcenter item / part structure*** Invoke the _Sync SysML --> Target_ operation on connections between SysML blocks and Teamcenter items / parts to compare and update the Teamcenter item / part structure based on the SysML block structure, such as rename / addition / removal of item / part usages (BOM lines) in Teamcenter items / parts.
T9	***Sync Teamcenter item / part structure ⮂ SysML block structure*** Invoke the _Sync Target --> SysML_ operation on connections between SysML blocks and Teamcenter items / parts to compare and update the SysML block structure based on the Teamcenter item / part structure (BOM lines), such as rename / addition / removal of part properties in SysML blocks.
T10	**Generate Teamcenter requirement / requirement spec / paragraph structure and populate properties from SysML requirement structure and properties** *Drag-n-drop* a SysML requirement to a Teamcenter folder to generate a requirement structure (including requirement specs, paragraphs, and requirements) in Teamcenter from the SysML requirement structure. This also creates connections between the SysML requirements and Teamcenter requirement / specs / paragraphs that can be used for compare and bi-directional sync operations. Select the option to generate trace links in Teamcenter from SysML dependencies (e.g. satisfy or derive).

T11	**Generate SysML requirement structure and populate properties from Teamcenter requirement spec / paragraph / requirement structure and properties** *Drag-n-drop* a Teamcenter requirement spec / paragraph / requirement (or their revision) to a SysML package to generate a SysML requirement structure. This also creates connections between the SysML requirements and Teamcenter requirement / specs / paragraphs that can be used for compare and bi-directional sync operations. Select this option to generate dependencies in SysML from trace links in Teamcenter.
T12	**Compare SysML requirement structure and Teamcenter requirement spec / paragraph / requirement structure, including properties** Invoke the *Compare* operation on connections between SysML blocks and Teamcenter requirement spec / paragraph / requirement structure to generate a difference table that shows the in-sync and out-of-sync aspects of the two structures.
T13	**Sync SysML requirement structure and properties --> Teamcenter requirement spec / paragraph / requirement structure and properties** Invoke the _Sync SysML -> Target_ operation on connections between SysML requirements and Teamcenter requirement spec / paragraph / requirement to compare and update the Teamcenter requirement structure based on the SysML requirement structure, such as rename / addition / removal of child requirement specs / paragraphs / requirements in the Teamcenter requirement spec / paragraph / requirement.
T14	**Sync Teamcenter requirement spec / paragraph / requirement structure and properties --> SysML requirement structure and properties** Invoke the _Sync Target -> SysML_ operation on connections between SysML requirements and Teamcenter requirement specs / paragraphs / requirement to compare and update the SysML requirement structure based on the Teamcenter requirement spec / paragraph / requirement structure, such as rename / addition / removal of child requirements in the SysML requirement.
T15	**View NX CAD models in Teamcenter** View datasets associated with an item/part revision. Expand datasets to view files. Expand NX CAD models (prt files) to view the part-assembly structure. Create reference connections between SysML model elements, such as blocks, and part/components in the NX CAD model.
T16	**Generate SysML block structure from NX part structure in Teamcenter** *Drag-n-drop* a NX part/assembly structure from Teamcenter to generate SysML block structure. This also brings mass properties (such as mass, volume, surface area, density, center-of-gravity, and bounding box dimensions) of NX parts to SysML, and provides options to generate part features and feature expressions/attributes in SysML. It also creates a connection between SysML blocks and NX parts for downstream compare and sync operations.
T17	**Drag-n-drop an existing SysML block/requirement to a Teamcenter item / part / requirement / spec (or vice versa) to create connections for downstream compare/sync.**

10 Syndeia and Windchill / Creo

W1	**Connect to one or more Windchill repositories and view contents** *Connect* to one or more Windchill repositories (servers) in your organization by supplying the URL and your credentials.
W2	**View contents of Windchill repositories** View the contents of a Windchill repository, such as Products, Parts, Part Versions, Folders, and Baselines, directly from the Syndeia Dashboard.
W3	**Search Windchill repositories** Search Windchill parts by specifying the containing Product, part number or/and name, and baseline criteria. View search results and drag-n-drop them to the SysML model.
W4	**Create Reference connections between SysML model elements and Windchill elements** *Drag-n-drop* a SysML model element to any Windchill repository element (e.g. product/part), or vice versa, to create a reference connection. A reference connection is the most basic form of traceability between SysML model elements and Windchill repository elements. A reference connection can be used to check if the version of the Windchill part has changed.
W5	**Generate Windchill part structure and populate attributes from SysML block structure and value properties, and create Model Transform connections** *Drag-n-drop* a SysML block with value properties to a Windchill product or folder to generate a part structure (parts and part occurrences) with attribute values in Windchill. This also creates connections between the SysML blocks and Windchill part versions that can be used for compare and bi-directional sync operations.
W6	**Generate SysML block structure with value properties from Windchill part structure and attributes, and create Model Transform connection** *Drag-n-drop* a Windchill part (or part version) with attributes to a SysML package to generate a block structure with value properties. This also creates connections between the SysML blocks and Windchill parts that can be used for compare and bi-directional sync operations.
W7	**Compare SysML block structure with value properties and Windchill part structure with attributes** Invoke the *Compare* operation on connections between SysML blocks and Windchill parts to generate a difference table that shows the in-sync and out-of-sync aspects of the two structures.
W8	***Sync SysML block structure with value properties ⮀ Windchill part structure with attributes*** Invoke the *Sync SysML -> Target* operation on connections between SysML blocks and Windchill parts to compare and update the Windchill part structure and attributes based on the SysML block structure and value properties.
W9	***Sync Windchill part structure with attributes ⮀ SysML block structure with value properties*** Invoke the *Sync Target -> SysML* operation on connections between SysML blocks and Windchill parts to compare and update the SysML block structure based on the Windchill part structure, such as rename / addition / removal of part properties in SysML blocks.
W10	**View details of Creo CAD models associated with EPM document** Expand EPM document versions in Windchill to view Creo models associated with them. Create reference connections between SysML model elements and Creo model elements.
W11	**Generate SysML block structure from Creo models in Windchill** *Drag-n-drop* a Creo assembly model under an EPM document version to generate SysML block structure. This also brings mass properties (such as mass, volume, surface area, density, center-of-gravity, and bounding box dimensions) of Creo models to SysML, and provides the option to generate model attributes in SysML. It also creates a connection between SysML blocks and Creo models for downstream compare and sync operations.

W12	**Connect existing SysML block and Windchill parts**
	Drag-n-drop an existing SysML block to a Windchill part (or vice versa) to create connections for downstream compare/sync.

11 Syndeia Dashboard Features

The Syndeia Dashboard is the primary way to use the Syndeia application. It can be launched by right clicking on any SysML element in the model tree of the SysML modeling tool. The Syndeia Dashboard provides the following main features.

D1	**Repository Manager – Connect to multiple enterprise repositories** Connect to multiple enterprise repositories, such as PLM (Teamcenter, Windchill), ALM (GitHub), Requirements management (DOORS-NG), Issue tracking and Project management (JIRA), database (MySQL), and local/shared drives.
D2	**Connection Manager – Drag-n-drop to generate and connect models** *Drag-n-drop* a SysML model element to a repository (or vice versa) to generate models and create model-based connections. Multiple connection patterns are available – reference, data map, function wrap, model transform.
D3	**Connection Browser – View, filter, search, compare/sync connections** View connections organized by SysML model structure, filter and search connections, invoke compare/sync operations on connections, and open connected models
D4	**Connection Summary – View and filter, search, compare/sync connections** View a list of connections, filter and search connections, invoke compare/sync operations on connections, open connected models, and export to Excel.
D5	**Comparison Results – View in-sync and out-of-sync model elements after comparison** View results of model compare operations invoked on connections (e.g. between SysML and Simulink or PLM models). View in-sync (green) and out-of-sync elements (red).
D6	**Visualize all connections as a chord diagram and get model analytics** View connections between elements in different repositories as a chord diagram. Hover on connections/artifacts to view details, filter based on repository/connection/artifact type. View model analytics and search connected artifacts.
D7	**Visualize and explore neighboring artifacts as tree or auto-layout** View and recursively explore *inter-model* and *intra-model* connections for a selected artifact as a tree or auto-layout graph. Export connection map as an image. Connections between elements in different repositories (e.g. SysML and PLM/Simulink) are *inter-model connections*, and connections between elements in the same repository (e.g. dependency between two SysML elements) are *intra-model connections*.
D8	**View connected artifacts & launch visualization from SysML model tree/diagram** Query and view connected artifacts and launch visualization for a given SysML model element from the model tree or any diagram in the SysML modeling tool.
D9	**Settings – Configure Syndeia using various setting options** Configure Syndeia's behavior using various setting options available for different repository types.