



New Features Guide

New Features Guide

Copyright © 2011-2016 Vitech Corporation. All rights reserved.

No part of this document may be reproduced in any form, including, but not limited to, photocopying, translating into another language, or storage in a data retrieval system, without prior written consent of Vitech Corporation.

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7013.

Vitech Corporation

2270 Kraft Drive, Suite 1600
Blacksburg, Virginia 24060
540.951.3322 FAX: 540.951.8222
Customer Support: support@vitechcorp.com
www.vitechcorp.com



is a trademark of Vitech Corporation and refers to all products in the GENESYS software product family.

Other product names mentioned herein are used for identification purposes only, and may be trademarks of their respective companies.

Publication Date: January 2016

TABLE OF CONTENTS

Overview of GENESYS 4.0	1
Connecting with the Work Environment	2
Excel Connector	2
PowerPoint Connector	3
Project Connector	4
OSLC Connector	4
Architecting New Solutions with Cross-Project Relationships	5
Accelerating Requirements Capture	6
Expanding our Toolkit of System Representations	7
Specifying System Structures Using the Class Diagram	7
Defining Constraints with the Constraint Block Definition Diagram and the Parametric Diagram	8
Increasing the Value of the Sequence Diagram	9
Sizing and Auto-sizing Diagram Content	10
Additional Diagram Content and Decorations	10
Enabling the Engineer	11
Increasing Project Visibility with Teamview	13
Expanded Project Navigation	14
Audit Logging	15
A Myriad of Additional Refinements	15
Enhanced Performance	15
Setting Attributes for Multiple Objects	15
Changing an Entity's Class	16
Connecting Operational and System Simulation	16
Renaming via New Convenience Commands	16
Additional Reports	16
Additional Scripts	17
Additional User Preferences	17
Comprehensive List of GENESYS 4.0 Schema Changes	18



CUSTOMER RESOURCE OPTIONS

Supporting users throughout their entire journey of learning model-based systems engineering (MBSE) is central to Vitech's mission. For users looking for additional resources outside of this document, please refer to the links below. Alternatively, all links may be found at www.vitechcorp.com/resources.



[Webinars](#)

Webinar archive with over 40 hours of premium industry and tool-specific content.



[Screencasts](#)

Short videos to guide users through installation and usage of Vitech software.



[A Primer for Model-Based Systems Engineering](#)

Our free eBook and our most popular resource for new and experienced practitioners alike.



[Help Files](#)

Searchable online access to Vitech software help files.



[Technical Papers](#)

Library of technical and white papers for download, authored by Vitech systems engineers.



[MySupport](#)

Knowledge Base, Exclusive Webinars and Screencasts, Chat Support, Documents, Download Archive, etc.

Our team has also created resources libraries customized for your experience level:

All Resources	Advanced
Beginner	IT / Sys Admin
Intermediate	Student

OVERVIEW OF GENESYS 4.0

GENESYS 4.0 is the new standard of systems engineering software, combining our exclusive fully-integrated system design environment, proven STRATA™ methodology, and industry-accepted systems engineering best practices. Leveraging the rich heritage and experiences gained from over two decades of engineering development through CORE™, GENESYS offers a modern systems engineering environment allowing the systems engineering team to connect and integrate with other applications across the greater project team and enterprise for analytical engineering, project management, and stakeholder communication.

The marriage of robust systems engineering capabilities and open architecture provides a unique and powerful model-based systems solution unmatched in the market today. It is now easier than ever to design and manage complex systems, providing full-lifecycle traceability from concept development to deployment, ensuring a delivery of a comprehensive, integrated, and defensible system architecture and design solution.

Key enhancements in GENESYS 4.0 include:

- **Expanded Microsoft® Office® Connectors plus OSLC Support:** Enable easy data sharing, communication, and enhanced collaboration with team members, management, and stakeholders with add-in connectors to Excel®, PowerPoint®, and Project® plus support for the Open Service for Lifecycle Collaboration (OSLC) data interface for end-to-end project lifecycle traceability.
- **Cross-Project Relationships:** Tackle your most complex systems issues by partitioning your model into multiple, relatable projects. Cross-project relationships provide flexibility for model management and control to address systems issues such as systems of systems and enterprise architectures.
- **Expanded System Representations:** Advance your system design by leveraging additional systems engineering representations offered in GENESYS 4.0. The class diagram provides a structured view into software-intensive systems, bridging the gap between systems and software engineering. The constraint block definition and parametric diagrams allow the engineer to identify and manage the system constraints, bridging from architecture into engineering analysis.
- **Document Parser:** Accelerate the initial requirements phase by parsing documents into GENESYS using key words to create requirements and other entities.
- **Model Assistant:** Lower the barrier to entry and leverage the power of truly integrated, model-based systems engineering through automated model assistance. Model Assistant allows the user to select from a host of best practices for the tool to enforce model consistency enabling you to focus on the substance of the design.
- **TeamView:** Enhance project visibility with the easy-to-browse static HTML representation of your project database, complete with diagrams and property sheets, generated with push-button ease direct from the system design for publication to an intranet or sharing with stakeholders.
- **Enhanced Change Management:** Efficiently track design changes with audit logging. This change management enhancement complements embedded versioning to clearly indicate all changes at the entity level. With a log of all revisions along with the corresponding timestamp and responsible user, all changes in your system design are accounted for ensuring complete control of your project information.
- **A Myriad of Additional Enhancements:** GENESYS 4.0 delivers refined diagramming capabilities to increase richness while enhancing the user experience, new reports and scripts for data output and manipulation, expanded project navigation options, convenience commands to simplify and streamline various activities, and additional user preferences for even greater user flexibility.

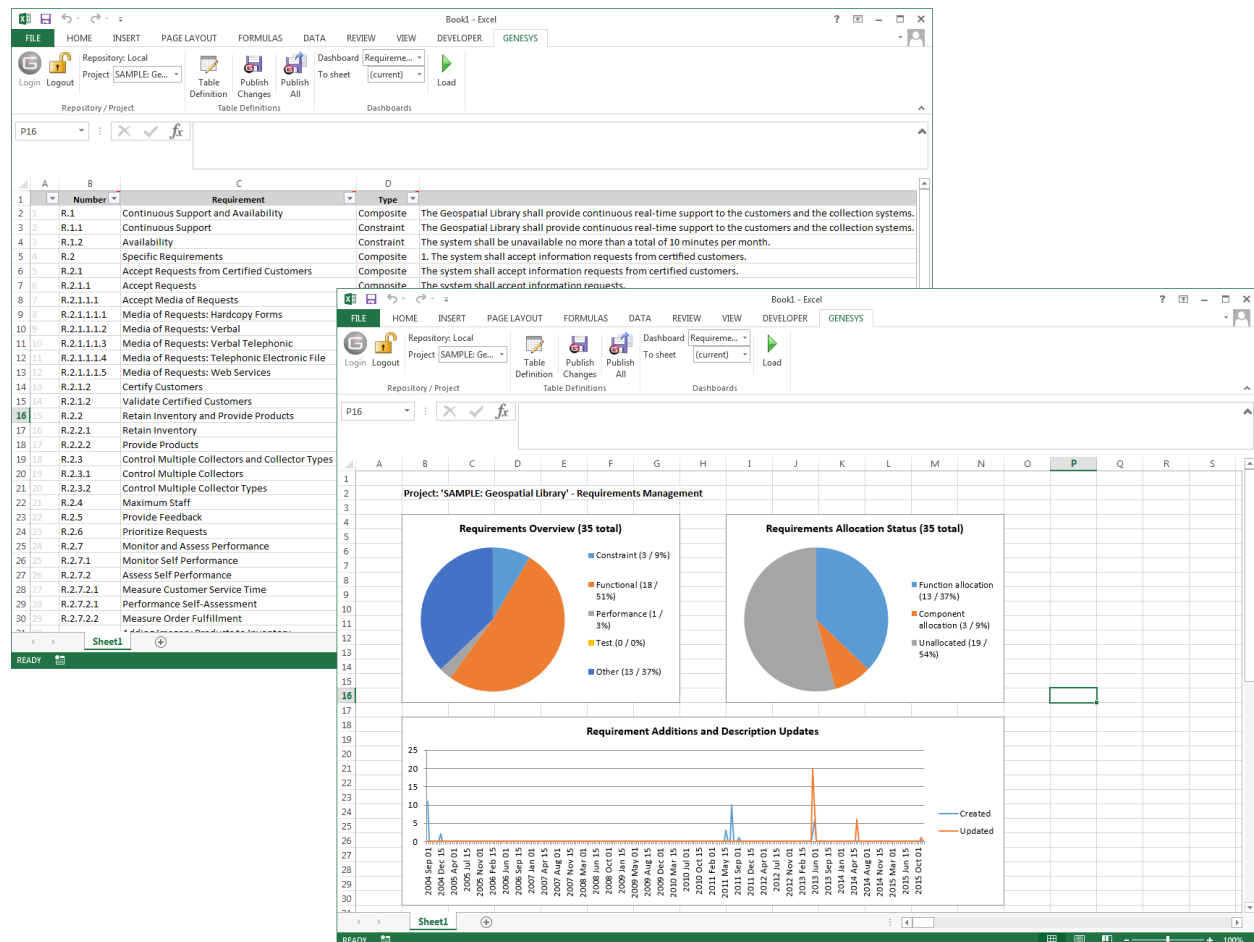
CONNECTING WITH THE WORK ENVIRONMENT

The most common supporting application in the systems engineer's environment is the Microsoft Office suite. This is particularly true when interacting with project and program management functions. Communications with stakeholders, management, and contractors are often delivered through PowerPoint presentations. Project planning and management offices frequently use Microsoft Project to project, track, and report project progress. Engineers and managers alike rarely deliver work products that aren't created in Excel, as it has become the multi-functional workhorse of the program suite, providing customizable spreadsheets to meet most communication needs.

In order to ease the interface with the Office environment, GENESYS 4.0 provides native connectors embedded within Excel, PowerPoint, and Project. These connectors allow Office users to log into GENESYS directly from Office and connect to the information of interest without ever leaving the environment within which they are comfortable – the ultimate in “your data your way.”

Excel Connector

GENESYS now provides a bidirectional interface to Excel, allowing data to be loaded directly into Excel – no configuration or data exchange files required. Conversely, Excel users can update data directly in GENESYS without ever leaving Excel. Of course, everything operates within the bounds of the permissions granted and controlled within the GENESYS project. Whatever data you are allowed to read or update in GENESYS, you can read or update directly in Excel. This bidirectional interface enables continuous data exchange and provides flexibility in communicating with all individuals on the project through the ubiquitous Excel medium.



New Features Guide

The connector is a separately installed .NET application that extends Excel (Office 2010 and later). When installed, it adds a tab to the Excel ribbon allowing the user to log in to, access, and update data in the GENESYS repository. Initially released with GENESYS 3.0, the Excel Connector has been vastly expanded, now allowing the display and update of arbitrarily complex tables of entities, relationships, attributes, and parameters. Import a set of requirements. Review the allocation of your functional architecture and revise as desired. Develop your requirements verification matrix and more. Though certainly not a replacement for the full power of GENESYS, the Excel Connector enables very rich interactions with traditional tabular views in a format and environment that is natural for individuals across the project team.

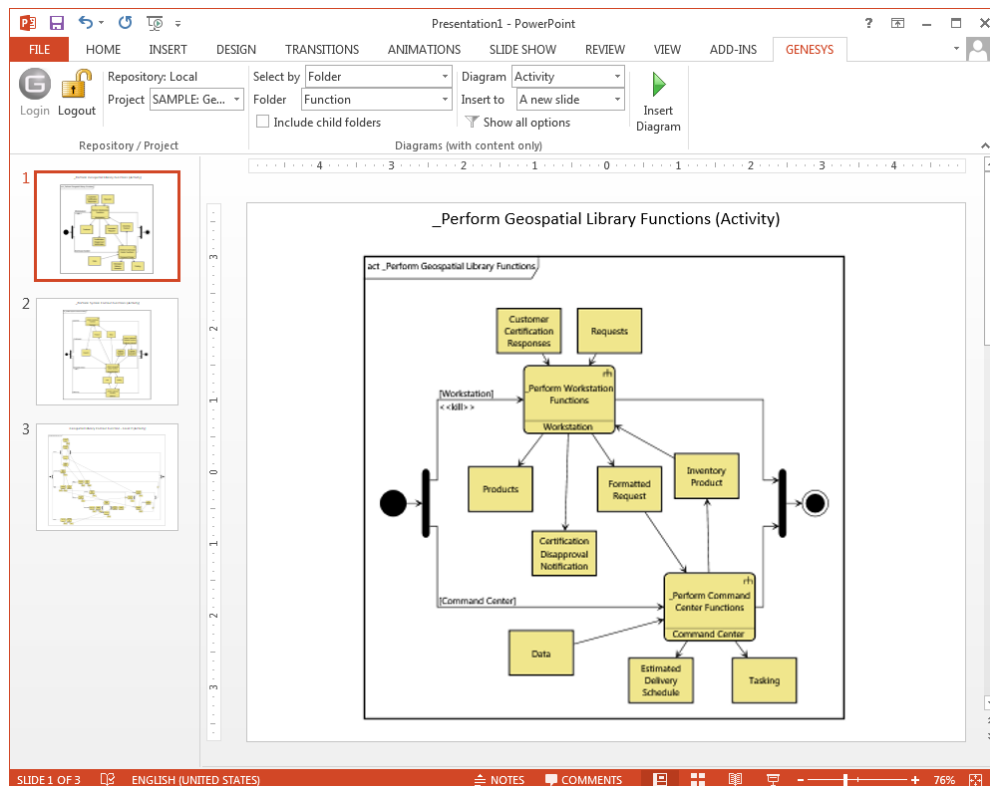
The connector includes a set of pre-defined table definitions for quick retrieval of commonly queried datasets enhanced by an interface allowing the user to create and customize these table definitions. A set of pre-defined dashboards give the user critical visualization of database statistics. Full directions for using the Excel Connector are provided in the Excel Connector Guide.

PowerPoint Connector

The PowerPoint Connector provides a seamless interface for developing presentation materials by transferring system representations directly from the GENESYS repository. No longer do you have to export specific diagrams as you prepare for design reviews, informal meetings, or simply need to allow a team member to view a specific representation. You can generate these presentations yourself, or you can allow team members to access the diagrams directly from within their PowerPoint environment.

The GENESYS add-in appears in PowerPoint as a new tab on the ribbon. Simply click the GENESYS icon to log in to GENESYS and allow the connector to retrieve data from the project repository. (Permissions still apply so users are only allowed to see that information to which they have been granted access). Select the various entities and diagrams you wish to see, and PowerPoint builds new slides pulling the graphics directly from GENESYS. You are then free to format the slides leveraging the rich visualization tools of PowerPoint, spending your time on added communication value rather than fundamental data transfer.

Directions for using the PowerPoint Connector are provided in the PowerPoint Connector Guide.

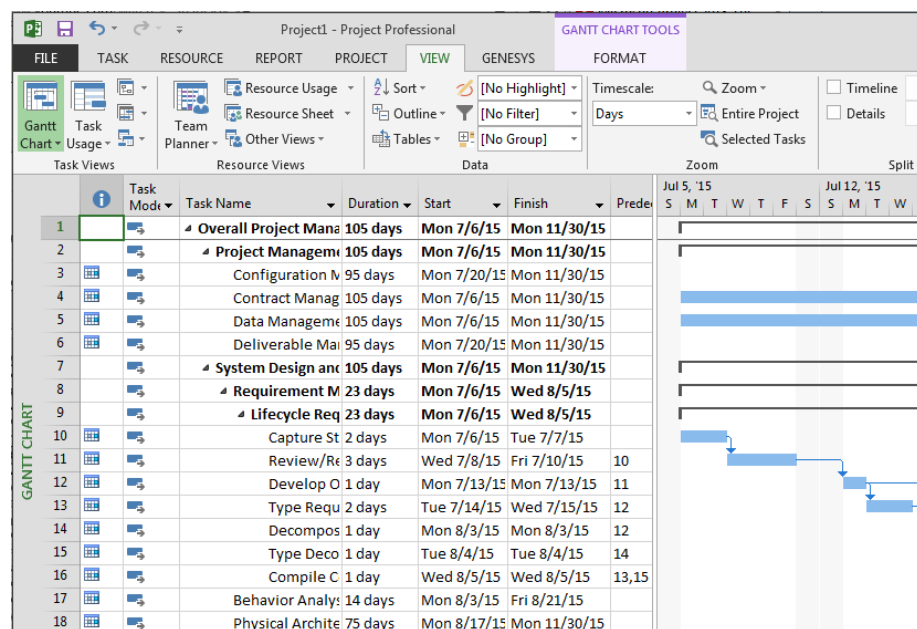


Project Connector

Closing the gap between program management and systems engineering is one of the key challenges highlighted in The Guide to Lean Enablers for Managing Engineering Programs, May 2012, published by the joint MIT-PMI-INCOSE Community of Practice on Lean Program Management. Systems engineers and program managers are both focused on effectively and efficiently meeting customer needs. Unfortunately, sometimes collaboration and coordination between those focused on technical merit and those focused on budget and schedule becomes challenging. The Project Connector helps bridge that gap.

A native plug-in that extends Project, the Project Connector allows project managers (and others working in Project) to extract program management information from the GENESYS systems engineering environment directly into Project. ProgramElements, ProgramActivities, and TestActivities can be extracted in order to align and synchronize the systems engineering and project management efforts. The result is increased coordination through seamless exchange of information with each individual able to see the information in their format in his or her environment.

Complete directions for using the Project Connector are provided in the Project Connector Guide.



OSLC Connector

Open Services for Lifecycle Collaboration (OSLC) is an open community focusing on developing specifications and standards to connect Application Lifecycle Management (ALM), Product Lifecycle Management (PLM), IT operations, and more. To help connect model-based systems engineering (MBSE) with the greater engineering and enterprise environment, GENESYS 4.0 now supports OSLC data exchange. The OSLC Connector provides the means to develop an interface with other applications to exchange engineering information with other tools across the project lifecycle.

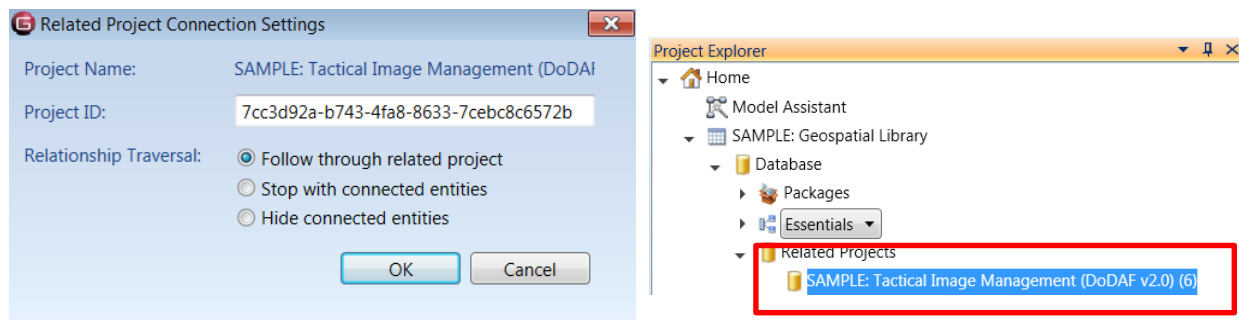
Effective data exchange and development of a connected enterprise typically leverages a blend of point-to-point connectors and open standards. The series of Microsoft Connectors reflect dedicated interfaces with clear, pre-defined mapping of information and a streamlined user experience. OSLC provides a framework by which organizations can define their own semantic mappings between any set of project information. Through the combination, GENESYS 4.0 delivers on the promise of a high fidelity, high power engineering environment for the systems engineer combined with the open architecture necessary to effectively and efficiently connect across the enterprise.

ARCHITECTING NEW SOLUTIONS WITH CROSS-PROJECT RELATIONSHIPS

As teams adopt model-based approaches, a commonly cited challenge is model management including effective navigation, collaboration, access control, and reuse. First introduced in CORE 9, cross-project relationships address these challenges allowing users to create relationships between entities in different projects. This provides project teams and enterprise efforts nearly infinite flexibility to partition their models as they see fit. Whether addressing process needs, reuse, or contractual obligations, cross-project relationships enable teams to architect, manage, baseline, and reuse their models in a highly modular manner.

With the same drag-drop ease of connecting one entity to another, you can now drag an entity from one project onto an entity in another project to relate them together. You can create as many or as few cross-project relationships as needed resulting in highly linked or loosely coupled projects.

Once you establish a relationship, GENESYS displays a new project in the Related Projects section of the Project Explorer. From this view, you can see all of the cross-project relationships and the associated entities. You can also quickly redefine the connection characteristic.



With cross-project relationships, you have more power than ever to tackle challenging problems including:

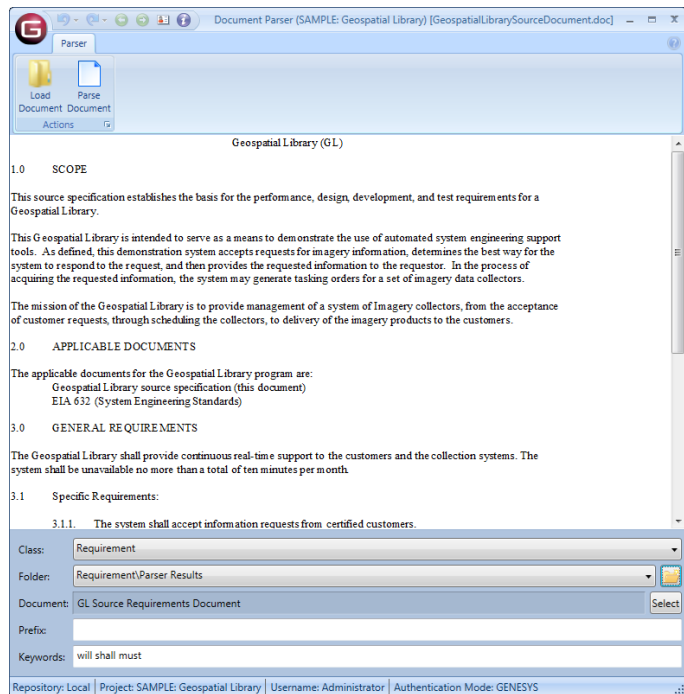
- **Enterprise Architecture.** In this scenario, a governing architecture has been developed and is owned by a given team. As deficiencies are identified and capabilities proposed, those solutions must be developed within the framework of the enterprise architecture. Ownership of the solution architecture is separate from the ownership of the enterprise architecture.
- **System of Systems.** In this scenario, the solution architecture is comprised of a federated system of systems – some existing, some under development. In a federated scenario, different organizations have ownership and authority over different systems, yet there is a solution architect trying to integrate these systems to deliver a new capability.
- **Contractor Boundaries.** In most systems, there are several levels of organizations involved, each with their own responsibilities and ownership. The customer may own the operational analysis and source requirements development. The prime contractor may own the system-level design. The subcontractors may own individual components. In many cases, to some degree, work will proceed in parallel across these groups. Regardless, visibility and permissions across the boundaries are tightly controlled.
- **Systems Integration.** Oftentimes, systems engineering is tasked with leveraging existing solutions – be that capabilities, systems, or components – and integrating them in different ways to deliver new value. The existing solutions are under different control and may often continue to evolve while the architect is required for developing the new capability.
- **Baselining / Product Revision.** As a system moves through the development lifecycle, there is often a need to baseline and lock prior work, though the work must remain available for extension. Related is the issue of iterative development when a product is incrementally released to the marketplace, either as successive versions or as related products in a product family.
- **Reuse.** Many times, there is a common seed that is reused by multiple projects. This seed is developed and owned by one organization with dependent projects owned by different organizations. The common seed can continue to evolve over time (a software library for reuse, a process framework, a common architecture).

ACCELERATING REQUIREMENTS CAPTURE

In almost every situation, the project team works with countless artifacts from others which must be integrated into the engineering environment. Most often, these are various requirements documents – concepts of operations, specifications, required standards, meeting minutes, and more. If the data has been organized in Excel, the Excel Connector can quickly ingest the information into GENESYS for analysis, management, and traceability. More often than not, the artifacts are documents. With the document parser introduced in GENESYS 4.0, you can accelerate the parsing of documents, focusing your valuable time on the analysis of the material.

The document parser accelerates the initial requirements phase by parsing electronic documents into GENESYS in seconds, allowing the engineer to be more efficient yet also mindfully choosing which information is pertinent to the current project. The parser is essential to organizations transitioning from a document-centric to model-centric approach, providing relationships from the source information to the engineering data, and ultimately, full traceability to the system design. Using the parser, you can open any traditional document format – .DOC, .DOCX, .HTM, .HTML, .RTF, .TXT – natively in GENESYS and then control how the information is parsed into your project repository:

- **Class** – the class of the entities to be created. Requirement is the default class, though any can be selected.
- **Folder** – the folder in which to place the parsed entities. A best practice is to place requirements in a separate folder for through review to avoid the garbage-in, garbage-out problem.
- **Document** – the Document entity that will be linked to the root entities parsed from the document. If no document is specified, parsing will continue and root entities will simply not trace to a document.
- **Prefix** – a string that will prefix the names of all entities created during the parsing process.
- **Keywords** – the collection of words the parser will key on to separate requirements from other content. The generally accepted set is shall, will, and must. When a keyword is detected, the parser treats the phrase as an entity, naming and relating it appropriately. When no keyword is detected, the phrase is treated as debris, an entity is created and generically named. Note that if no keywords are specified, you will be prompted for confirmation before the document is parsed. If you choose to proceed, all content will be identified as entities rather than debris.



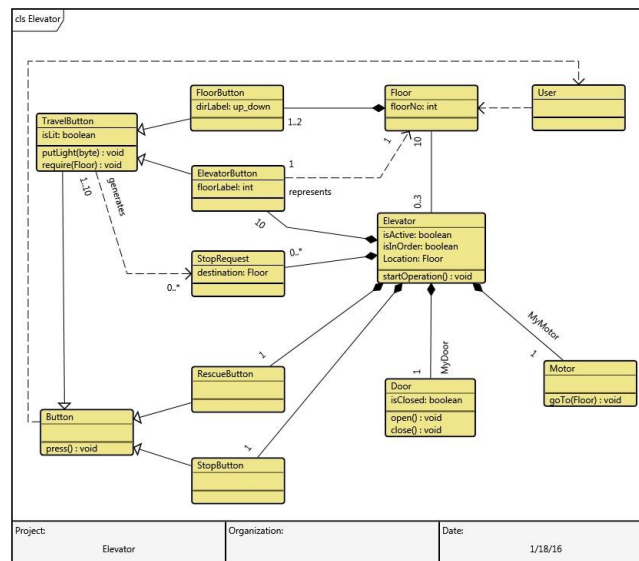
Once you specify the required settings, parsing the document directly create the corresponding entities in GENESYS. GENESYS interprets the document structure to create a hierarchy of entities mirroring the source document so that you can focus your time on thoughtful review and analysis.

EXPANDING OUR TOOLKIT OF SYSTEM REPRESENTATIONS

In our continuing efforts to provide the most comprehensive systems engineering solution, GENESYS 4.0 brings three new views to help enrich communication across the comprehensive engineering team, providing additional insight into the system design. In addition, the sequence diagram has been revised to deliver more flexibility, and numerous enhancements have been made across the diagram suite to improve both value and usability.

Specifying System Structures Using the Class Diagram

Bridging the gap between systems and software engineering, the class diagram displays the static structure and composition of a system. It shows the system's classes and optionally displays their attributes, operations, and relationships among the entities. Though most frequently used to reflect software design architecture, the flexibility of the class diagram makes it a useful addition to a library of integrated system representations.



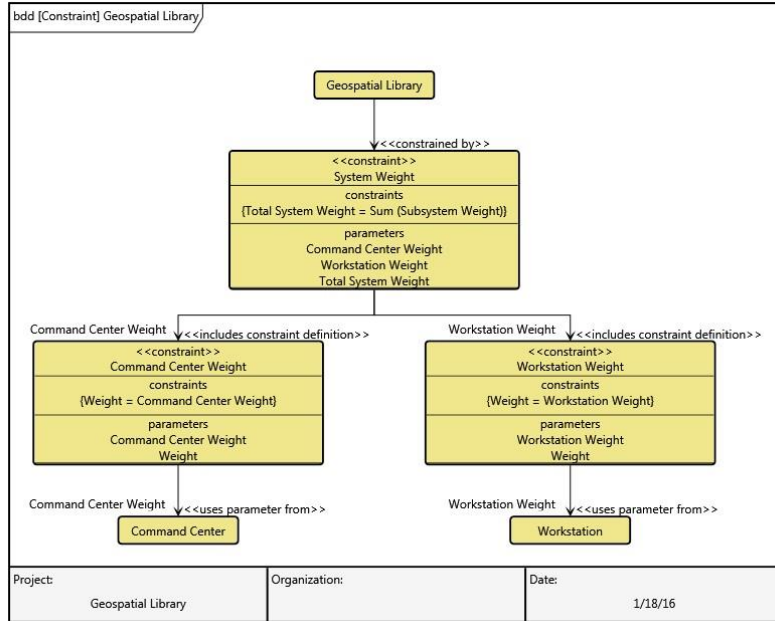
One of the primary benefits of this diagram is the ability to reflect different types of relationships between two or more entities within your system. The types of relationships include:

- **Dependency.** A dependency relationship exists between two entities if changes to the definition of one entity may cause changes to the other. Essentially, one entity requires, needs, or depends on the other for specification or implementation.
- **Association.** An association is an abstract term that can represent any relationship between two classes. An association can be either bidirectional or unidirectional.
- **Aggregation.** An aggregation shows the formation of a particular class as a result of one class being “aggregated” or built as a collection. For example, a course is an aggregation of students.
- **Composition.** A composition is similar to the aggregation, with the difference being the emphasis on dependence. For example, a house is a composition of rooms. However, if the house ceased to exist, the rooms would no longer exist, as their existence is dependent on the existence of the house.
- **Generalization.** A generalization relationship is a type of relationship that establishes an associated class as a child (subclass) of another by assuming inheritance of the functionalities of the parent (superclass). The child class is considered a specialization of the parent class. For example, a cat would be a specialization of an animal, where a cat is the child inheriting the functionality of the parent animal.

Defining Constraints with the Constraint Block Definition Diagram and the Parametric Diagram

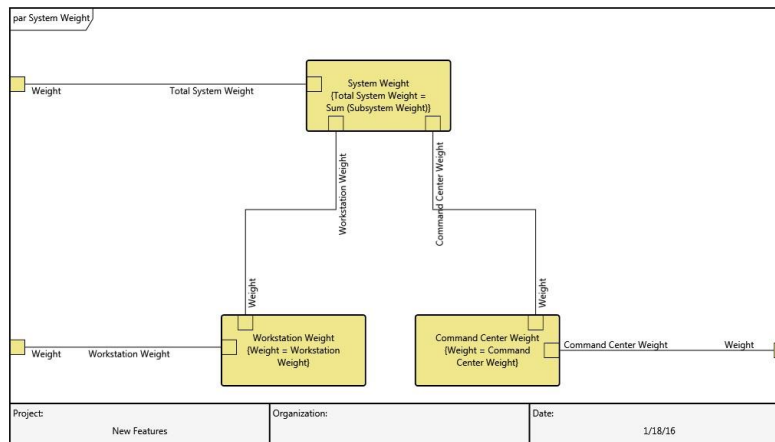
System constraints are an important, but often overlooked, aspect of the system design. System constraints specify assertions about valid system values in an operational system and provide insight to perform engineering analysis during the design phase of the lifecycle. GENESYS 4.0 leverages these SysML representations to better identify and manage the constraints on the system.

Constraint Block Definition Diagram (Constraint BDD). The constraint BDD is similar to the structure BDD and classification BDD. Rather than composition or classification, the constraint BDD shows the ConstraintDefinitions (constraint expression) used to constrain the subject entity (component) of your system. The constraint block will include the constraint expression (an expression that must evaluate to either true or false and is most likely an equation or inequality used to constrain value properties of blocks) and the independent and dependent variables that constitute the constraint parameters. The constraint parameters usually represent quantities, so they are typed – most often by value types. Constraint parameters receive their values from the value properties they are bound to or being constrained by. The constraint BDD does not convey which constraint parameters and value properties are bound to one another. This is expressed in the associated parametric diagram.



Parametric Diagram. The parametric diagram is created from a subject ConstraintDefinition (as developed in the Constraint BDD), which is the key entity on the parametric diagram. The parametric diagram expresses a set of constraints that determine the values that are valid in a system that is operating nominally. Similar to the internal block diagram in layout and general representation, the parametric diagram is a refinement of the constraint BDD. It is constructed from fundamentally the same subset of system design information and shows the usages of the blocks in the constraint BDD with a focus on the bindings between the value properties and the constraint parameters.

The representation of the bindings between the value properties and constraint parameters is unique to the parametric diagram. The depiction of the network of equations that constrains the properties of the subject block provides support for engineering analysis and can help identify the properties of the system that are critical to satisfying the requirements.



Increasing the Value of the Sequence Diagram

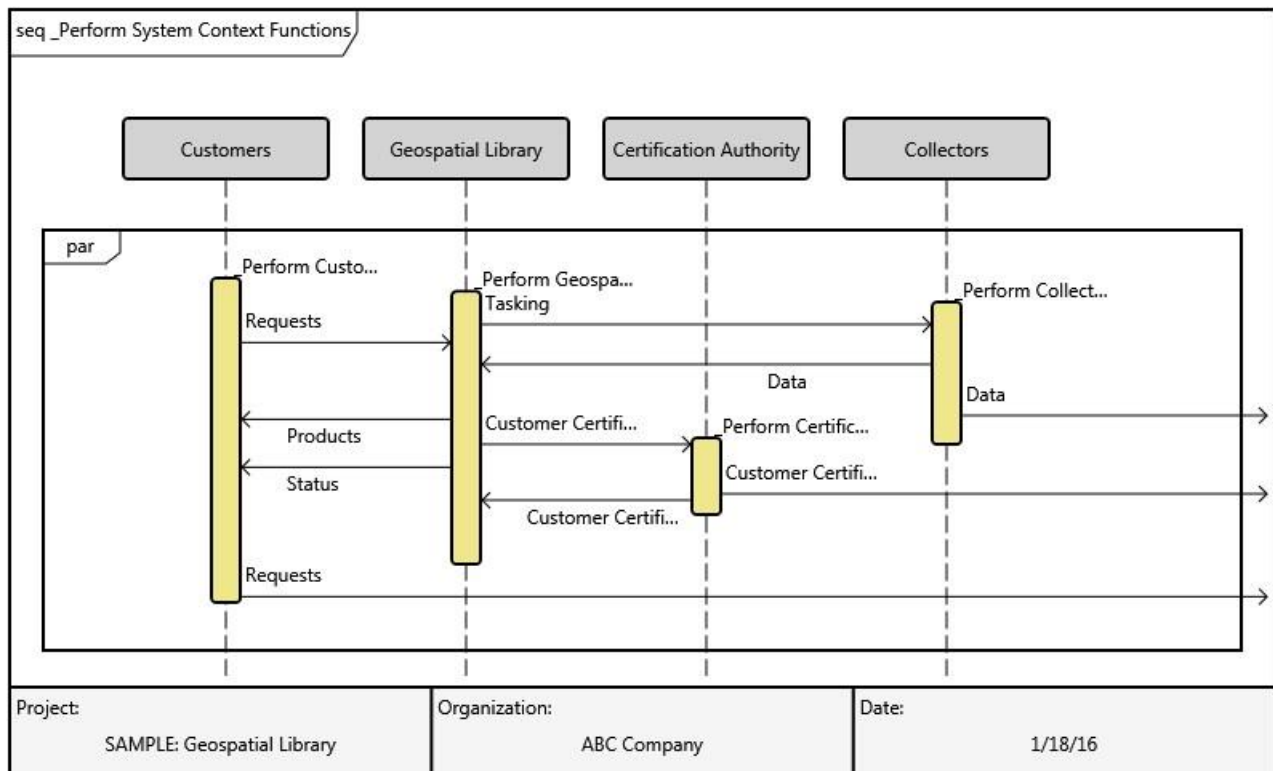
Introduced in GENESYS 3.0, the sequence diagram has undergone notable revision in GENESYS 4.0 drastically increasing its flexibility, utility, and communication value.

Part of the magic of the GENESYS sequence diagram is that it is automatically computed from the structure of the underlying model. Rather than allowing you to accidentally misrepresent the sequencing and flows, GENESYS performs a static analysis of the behavioral model and computes the position of the activities on the lifelines based upon this analysis. While highly valuable, this embedded intelligence was also the weakness of the sequence diagram. The resulting layout was constrained, a bit awkward, and definitely non-traditional...until GENESYS 4.0.

Now you get the best of both worlds. The fundamental sequencing is still computed, and you cannot position the nodes out of order. However, you have great flexibility within those computational boundaries to lay the diagram out in a way that better communicates the interactions within your system.

- Activity nodes can be slid up and down their lifeline bounded only by the adjacent nodes on the lifeline and any containing constructs.
- Item arrows can be repositioned, either by grabbing and dragging the line or by grabbing the arrowhead.

The net result is a diagram that both represents the engineering truth of your system design and the visualization necessary to understand, analyze, and communicate.



Sizing and Auto-sizing Diagram Content

With the ability to manually resize individual icons on a diagram, you now have the ability to size specific nodes to display the needed content. Whether for a long function name on an activity diagram, a full description on a requirements diagram, or a picture of your system, you can now maximize the use of the available space to show the needed content.

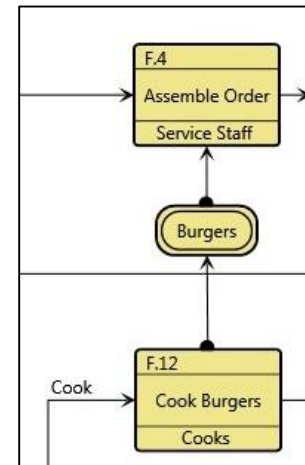
GENESYS 4.0 also includes two different auto-sizing options:

- **Auto-size.** This command (available from the menu and the toolbar) adjusts both the width and the height to display the desired content. Starting with the icon width specified at the diagram level, the algorithm checks each fixed height field, increasing the width as required to display all content. Once the width is determined, the icon height is adjusted (up or down) until there is space to display the full content of all dynamically-sized fields. Note that the maximum icon dimension is still 500x500.
- **Auto-size Height.** This command (available from the menu or by holding down the control key while issuing the Auto-size command) keeps the icon width the same but adjusts the height to display the full content of all dynamically-sized fields.

Both commands can be applied either to the selected objects (nodes, labels, and diagram decorations) or to the entire diagram. If no diagram objects are selected, you are prompted for confirmation before all diagram content is resized.

Additional Diagram Content and Decorations

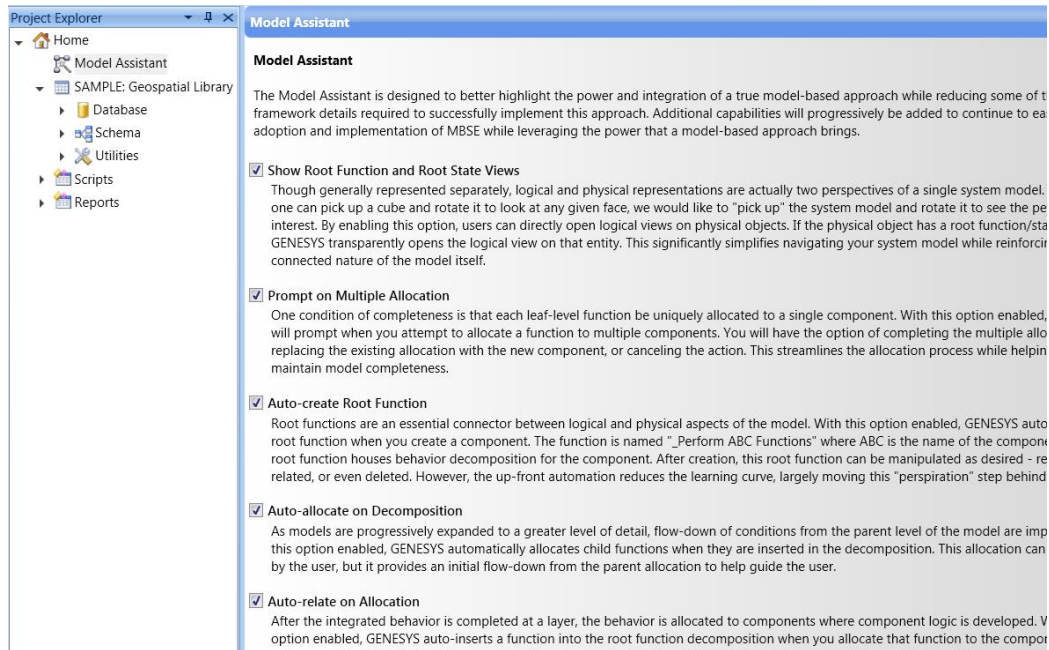
- Resources are now optionally displayed on EFFBDs. Resources are drawn with a double border. Consumes and produces relationships are indicated with a half circle and arrowhead decoration to reflect the direction of the flow. Captures relationships are shown with arrowheads on both ends. Resources have long been the behavioral aspect hidden from the behavioral views. With this revision, the EFFBD now provides a comprehensive visual of the entities and relationships that influence behavior. (Note that resources are not shown on activity diagrams as they are not part of the SysML definitions.)
- The block definition diagram has been extended to display the part multiplicity per the SysML standard. The part multiplicity is an attribute on the parent-child built from relation.
- The requirements diagram has been extended to display the containment decoration. The containment decoration (a small circle with a plus sign) indicates the partitioning of a complex requirement without adding meaning (for example, breaking a compound requirement joined by the word “and” into two child requirements). This differs from the <<deriveReq>> decoration which reflects a more detailed requirement arrived at via analysis.



ENABLING THE ENGINEER

Leveraging the insights gained over 20 years of practice as well as the prototype implementation in CORE, the GENESYS model assistant moves beyond simple data capture to analytic assistance. As a step towards heuristics, the model assistant helps the engineer avoid certain classes of errors (i.e., getting allocation out of sync) while streamlining operations – once again, freeing the engineer through the power and integration of a true model-based approach to focus on analysis and engineering rather than bookkeeping.

The individual model assistant rules are enabled or disabled per-user via the Model Assistant item in the Project Explorer tree. By default, all options are enabled.

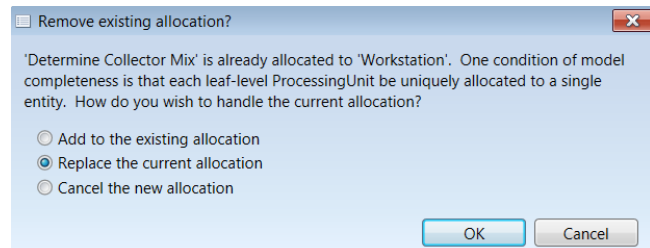


- **Show Root Function and Root State Views.** Though generally represented separately, logical and physical representations are actually two perspectives of a single system model. Much as one can pick up a cube and rotate it to look at any given face, we would like to "pick up" the system model and rotate it to see the perspective of interest. By enabling this option, users can directly open logical views on physical objects. If the physical object has a root function/state, GENESYS transparently opens the logical view on that entity. This simplifies navigating your system model while reinforcing the connected nature of the model itself.

To see this capability in action, open a sample project and click through the Component class. For components with root functions, you can see the many logical views. For components with root states, the State Transition view will become active.



- **Prompt on Multiple Allocation.** One condition of completeness is that each leaf-level function be uniquely allocated to a single component. With this option enabled, GENESYS will prompt when you attempt to allocate a function to multiple components. You will have the option of completing the multiple allocations, replacing the existing allocation with the new component, or canceling the action. This streamlines the allocation process while helping to maintain model completeness.



New Features Guide

- **Auto-create Root Function.** Root functions are an essential connector between logical and physical aspects of the model. To those who understand the methodology, their role is clear. To those just learning the way, they are odd artifacts of the approach that muddy the system definition effort.

GENESYS can automatically create a root function when you create a component. The function is named “_Perform ABC Functions” where ABC is the name of the component. This root function houses behavior decomposition for the component. After creation, this root function can be manipulated as desired – renamed, related, or even deleted. However, the up-front automation reduces the learning curve, largely moving this “perspiration” step behind the scenes.

- **Auto-allocate on Decomposition.** As models expand to a greater level of detail, flow-down of conditions from the parent level of the model are important. With this option enabled, GENESYS automatically allocates child functions when they are inserted in the decomposition. You can change this allocation, but it provides an initial flow-down from the parent allocation to serve as a guide.
- **Auto-relate on Allocation.** This option enables the allocation of behavior where the component logic is developed. Previously, the allocation and behavior were disjoint steps, subject to cross-check via consistency scripts but without automated support.

GENESYS automatically inserts a function into the root function decomposition when you allocate that function to the component that performs the root function. This only occurs if neither the function nor one of its ancestors is already on the root function decomposition. Again, this automation helps reinforce the alignment, integration, and synchronization of the models. It does not create the behavior structure, but it does ensure all necessary functions are present and considered.

- **Auto-create Root States.** Root states are an essential connector between logical and physical aspects of the model. With this option enabled, GENESYS auto-creates a root state when you create a component. The state is named “_Exhibit ABC States”, where ABC is the name of the component. This root state houses state decomposition for the component. After creation, this root state can be manipulated as desired – renamed, related, or deleted. This included automation reduces the learning curve, largely moving this “perspiration” step behind the scenes.
- **Auto-inclusion on Decomposition of States.** As models mature to a greater level of detail, flow-down of conditions from the parent level of the model are important. With this option enabled, GENESYS automatically creates an exhibited by relation between the child states and related component, when the child states are inserted into the decomposition. If needed, this relationship can be changed by the user, but it provides an initial flow-down from the parent state to help ensure model consistency.
- **Auto-relate on Inclusion.** After the integrated state decomposition is completed at a layer, the decomposition is related to the components where component logic is developed. With this option enabled, GENESYS automatically inserts a state into the root state decomposition when you create the exhibited by relation to the component. This only occurs if neither the state nor one of its ancestors is already included in the root state decomposition. This capability helps reinforce the alignment, integration, and synchronization of the model, ensuring all component states are present and considered.
- **Auto-maintain Connector Names.** A standard naming convention for interfaces, links, needlines, transitions and UML associations is created as “Entity Name 1 / Entity Name 2” to reflect the names of the two connected entities. With this option enabled, GENESYS automatically generates and maintains the names for these connector entities. If a default entity name is used, GENESYS will automatically rename the entity.

INCREASING PROJECT VISIBILITY WITH TEAMVIEW

GENESYS delivers a rich integrated set of representations for users to leverage in developing, analyzing, and communicating their system design. Those outside of the immediate systems engineering effort are often times left piecing together a complete systems understanding from a series of separate artifacts – diagrams, tables, informal reports, formal specifications. TeamView allows project teams to deliver a full hyperlinked representation of the design – generated with push-button ease directly from the project repository – for use by a wide array of stakeholders, customers, and partners.

TeamView allows those focused on understanding the system design (rather than developing the system design) to see the current state of the system design in native HTML format. Whether at baseline points, on a regular schedule, or on demand, the GENESYS team can leverage TeamView to generate a static HTML representation of the complete project – texts and graphics combined – for publication on a corporate internet or for distribution to customers and partners. All of the data – text and diagrams alike – is hyperlinked, enabling viewers to navigate the project design in the same way they would navigate a website. The result is highly efficient and effective communication in a format comfortable for all and generated with ease.

GENESYS 4.0 TeamView
Project: SAMPLE: Tactical Image Management Architecture
Powered By GENESYS. Generated 18 January 2016.

Requirement / Modification Requirements
View: Property sheet

Attributes	Parameters
Name	Modification Requirements
Number	MR
Description	The requirements for the Tactical Image System are to: 1) Maintain existing tactical user support without affecting existing processes or capabilities. 2) Provide a mechanism to support delivery of newly-generated products to JTF users.
Type	Composite
Key Performance Parameter	False
Performance Parameter Type	nil
Origin	Operational
Objective	nil

Function / _Perform TIM System Context Functions
View: Sequence

seq_Perform TIM System Context Functions

```

graph TD
    CO[CO Cmd Intel Terminal] --> COReq[CO Intel Request...]
    BGE[BGE Intel Terminal] --> BGEReq[BGE Intel Reques...]
    JTF[JTF Tactical Intel Cell Terminal] --> JTFReq[Perform JTF Tac...]
    COReq --> BGEReq
    BGEReq --> JTFReq
  
```

Project: SAMPLE: Tactical Image Manage... Organization: ABC Systems Company Date: 1/18/16

EXPANDED PROJECT NAVIGATION

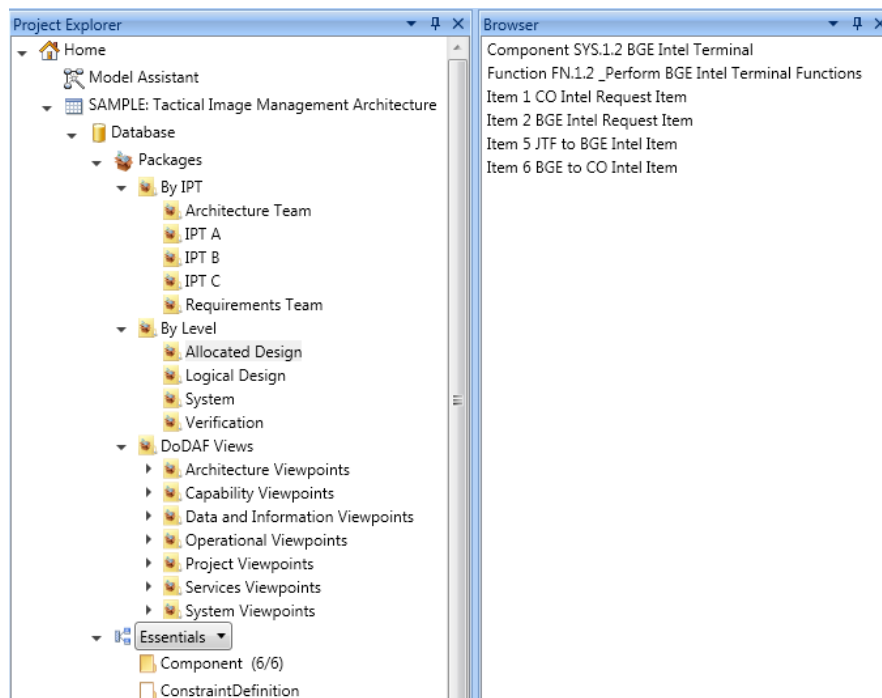
Traditionally, navigation within GENESYS has been database-centric. To access the entity of interest, you started by drilling into the specific class and then down into folders until you found what you were looking for. For those with a database-centric frame of reference and command of the MBSE language, wonderful! Those without these pre-requisites could easily struggle.

Previously in CORE, we raised the question of alternate navigation approaches, not to replace the database-centric paradigm but to complement it. We hypothesized role-based structures, level-based structures, and view-based structures (to name just a few). To enable all of these approaches, GENESYS now leverages the SysML approach of packages.

In SysML, packages are fundamentally tools for model management. They are a type of container that can hold diagrams, entities, and other packages (creating a nested structure). As packages are created, they are shown in the project list pane, enabling you to implement whatever packaging and navigation approaches are desired.

Of particular note:

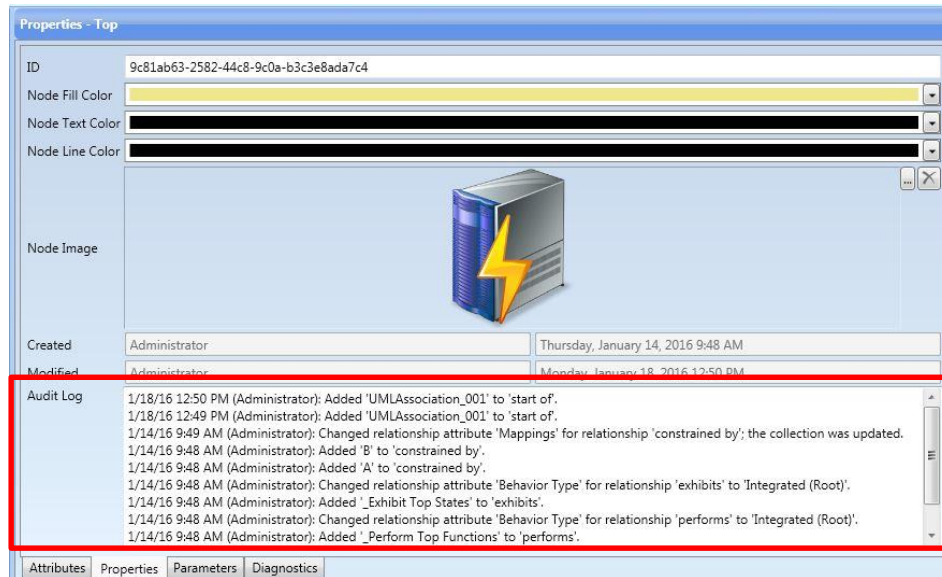
- Packages are shown as a segment under the Database item in the project list. This locates them alongside the existing ability to navigate the database via facility / class.
- Under the Packages item, GENESYS lists all root packages. These are the package entities that are not included in any other package.
- Packages and their contents can be manipulated directly from the Package section of the navigation list or by manipulating the native GENESYS entities in the Package class.
- Entities can be included in more than one package. This allows you to establish multiple package-based navigation schemes – by IPT, by level, ...
- As with entity folders, packages can be set to show sub package contents. This allows you to look all the way down the package hierarchy in a single, integrated list. Entities included in sub packages are shown in italics.
- Package contents can span multiple classes. This enables you to apply operations across entities from multiple classes (such as setting access control, purging and baselining, etc.).



AUDIT LOGGING

GENESYS 4.0 introduces the ability to track all modifications of the project data at the entity level. Once enabled, audit logging maintains a textual log of all data changes, along with the timestamp and the responsible user. These are stored in a read-only Audit Log attribute on the Properties tab of the property sheet. Combined with the highly detailed attribute versioning, audit logs provide insight into the evolution of the design.

As with versioning, commands exist at the entity, category, package, and project level to clear audit logs. Clearing the audit log requires entity administrator privileges and creates a corresponding entry in the audit log history. While audit logging is less appropriate at the beginning of a project as engineering data is still in flux, it is essential as the project matures and controlled evolution becomes critical.



A MYRIAD OF ADDITIONAL REFINEMENTS

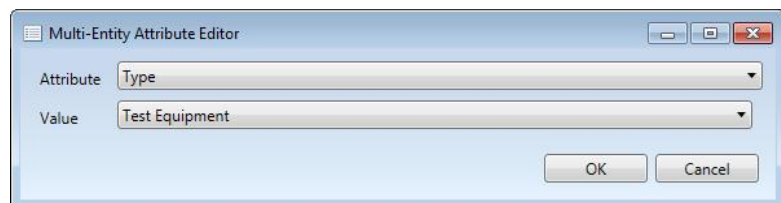
GENESYS 4.0 includes a wide range of new capabilities and refinements throughout the integrated systems engineering environment. Following are highlights on several items of note, but don't be surprised when you find extensions and refinements throughout to bring new power and an enhanced user experience.

Enhanced Performance

Through improvements in data handling, caching, and other refinements behind the scenes, the performance profile of GENESYS 4.0 has been considerably enhanced. This delivers the power and confidence of a Microsoft SQL database with improved performance, allowing for smoother, faster response as you develop and analyze your system definition.

Setting Attributes for Multiple Objects

Often times, there is a need to set a specific value for a range of entities – most commonly, setting the type for requirements, components, etc. The Set Attribute command allows you to change an attribute value for one or one hundred entities simultaneously. It can be applied to a homogenous set of entities in a single class or a heterogeneous set of entities in a package or on a diagram. Simply select the entities in question (via shift or control+clicking as appropriate) and click the Set Attribute command. The resulting dialog keys off the first entity selected to present the list of possible attributes to set. Select the desired attribute from the drop-down and the value pane will change



to best represent the attribute type and to show the current value. When you click OK, that value will be set for all selected entities.

Changing an Entity's Class

Sometimes during the design process, you realize that you selected the wrong class for an entity or perhaps the design has simply evolved. For example, perhaps you initially created an OperationalActivity but as the design progresses, you realize that it should be a Function. The Transform Class command allows you to quickly change the class for one or more selected entities. Any existing attribute values, parameters, and relation specifications that are appropriate for the new entity class are maintained, and those attributes and relations that are not defined for the new class are discarded.

Connecting Operational and System Simulation

For those using Capability Architect Development (formally DoDAF) schema, historically your operational activity and functional architecture have been separated with an implemented by relationship connecting operational to system models. GENESYS 4.0 now bridges these together and delivers a combined simulation. When the simulator reaches the bottom of the operational activity hierarchy, it automatically traversed the implements relationship and executes the corresponding functions (unless you set the attribute to not follow decomposition.)

Renaming via New Convenience Commands

Renaming objects is a frequent activity during the design effort, sometimes to reflect new concepts, sometimes to align with team vocabulary and style guides. Objects can always be renamed by opening the corresponding property sheet and updating the name field, but this requires multiple clicks. To streamline the operation, GENESYS now includes the following convenience commands:

- **Rename Entity.** Rename an entity from the right-hand pop up menu, rather than requiring use of the property sheet.
- **Rename Object.** While in schema mode, you can now choose to rename an object through the right-hand pop up menu, rather than requiring use of the property sheet.

Additional Reports

GENESYS 4.0 includes six new reports to expand communication with customers and stakeholders, assess model maturity, and assist in managing the system model. The new reports are:

- **System/Segment Design Description (SSDD).** This document defines the design of the system or subsystem and its operational and support environments.
- **Interface Requirements Specification (IRS).** This report outputs the requirements for all interfaces of a system/component or for a list of interfaces.
- **Risk Report.** This report outputs a color-coded matrix of risk severity with the number of risks in cell, a table of risks by severity, and a listing of risks with all their details. This report is used in conjunction with the Risk Rating script.
- **Project Compare Report.** This report documents the differences between the current project and a user-selected base project. This allows you to compare independent work as well as clearly present all changes made to a system design since a previous baseline.
- **Attribute Version Report.** This report documents the history of user-selected attributes in a tabular format for all entities of the database, selected package, or selected folder.
- **Schema Definition Report.** This report outputs the definition of all classes in a user-selected facility.

Additional Scripts

GENESYS 4.0 includes two additional scripts designed to support the engineering effort:

- **Risk Rating Script.** The Risk Rating Script automatically sets the 'riskRating' attribute for one or more Risk entities. The script will enhance the visualization by setting the fill and text color for the entities, based on the values of the 'consequence' and 'likelihood' attributes. Running the script allows you to select the project, folder, or package on which to set the risk rating.
- **Get Entity by ID Script.** This utility script allows the user to search for an entity by its ID.

Additional User Preferences

To provide individual users with even more control over their specific environment, GENESYS now includes the following new user preferences:

- **Change Pane Color when Filtered** - controls whether the background color of the entities pane shifts to a light green when a filter is applied. This visual cue serves as a valuable indicator that a filter is currently active.
- **Set Font** - controls the font displayed in all text windows and diagrams with the exception of formatted text fields.
- **Show Subfolder Entities** - controls whether entities in subfolders are shown by default when a project explorer is opened. Once a project explorer has been opened, you can use the Show Entities in Subfolders command to toggle the display of subfolder entities on or off.

COMPREHENSIVE LIST OF GENESYS 4.0 SCHEMA CHANGES

With each release, Vitech continues to enhance the underlying schemas to reflect our lessons learned and best practices in the application of MBSE.

A complete list of all schema changes is included below. At the highest level, the key schema changes for include:

- Renaming the DoDAF schema to Capability Architecture Development to reflect its broad use and applicability
- ConstraintDefinition class to support the constraint BDD and parametric diagrams
- UMLAssociation class to support the class diagram
- New classes
 - ConstraintDefinition
 - UMLAssociation
- New relation pairs
 - constrained by / constrains
 - end of / ends at
 - included in constraint definition / includes constraint definition
 - parameter used in / uses parameter from
 - start of / starts at
- Class Changes
 - ConnectingUnit
 - Added constrained by relation to target class ConstraintDefinition
 - Added parameter used in relation to target class ConstraintDefinition
 - ImplementationUnit
 - Added constrained by relation to target class ConstraintDefinition
 - Added contained by relation to target class UMLAssociation
 - Added start of relation to target class UMLAssociation
 - Added end of relation to target class UMLAssociation
 - Added parameter used in relation to target class ConstraintDefinition
 - InformationUnit
 - Added constrained by relation to target class ConstraintDefinition
 - Added parameter used in relation to target class ConstraintDefinition
 - ProcessingUnit
 - Added constrained by relation to target class ConstraintDefinition
 - Added parameter used in relation to target class ConstraintDefinition
 - Requirement
 - Added attribute performanceParameterType
- Facility Changes
 - Essentials
 - Added class ConstraintDefinition
 - Systems Engineering
 - Added class ConstraintDefinition
 - Architecture Development (DoDAF)
 - Added class ConstraintDefinition



Vitech Corporation
2270 Kraft Drive, Suite 1600
Blacksburg, Virginia 24060
540.951.3322 FAX: 540.951.8222
Customer Support: support@vitechcorp.com
www.vitechcorp.com