TaskGuide™

Authors' Guide

Version 2.0 Beta

Stottler Henke
Smarter Software Solutions

SBIR Data Rights
Contract #: W91WAW-07-C-0067
Stottler Henke Associates, Inc.
951 Mariner's Island Blvd., Suite 360
San Mateo, CA 94404
Expiration of SBIR Data Rights Period: July 15, 2016

Stottler Henke Associates, Inc.
951 Mariner's Island Blvd., suite 360
San Mateo, CA 94404

Tel: 650.931.2710
Fax: 650.931.2701
Web: www.stottlerhenke.com
Email: info@stottlerhenke.com

11/7/2011

Table of Contents

Chapter 1 - Introduction

The TaskGuide™ software system provides a graphical authoring tool and execution tool for creating and running interactive applications called *procedures* which display a series of screens to share information with users, derive information, generate recommendations, and exchange data and commands with other systems and databases. This document describes how *procedure authors* can use the TaskGuide Procedure Editor to create, edit, and deploy these procedures.

The *TaskGuide Users' Guide* introduces TaskGuide's capabilities and describes how to run procedures using the TaskGuide Procedure Execution Tool and the Procedure Execution Applet. It is recommended that you review the *TaskGuide Users' Guide* before reading this document.

The *TaskGuide Programmer's Guide* describes how *software developers* can embed TaskGuide software within other software applications and create extensions to the TaskGuide Procedure Editor.

Chapter 2 - Getting Started

PC Configuration Requirements

TaskGuide runs on PCs running the Windows XP, Windows Vista, and Windows 7 operating systems which satisfy the following configuration requirements:

RAM	2 GBytes
Free disk	11 MBytes (this does not include disk space needed for Java runtime system which is installed separately)
Display monitor resolution	1024 x 768 pixels or higher
Display monitor colors	8 bit color or higher

Table 1 – TaskGuide Configuration Requirements

Java Run-Time Environment

TaskGuide is written in the Java programming language and requires installation of the Java 6 or Java 7 run-time environment (JRE). You can download Java from:

http://www.java.com/

If a 64-bit version of the Windows operating system is installed on your computer, download and install the 64-bit version of Java. To test your installation of Java, type "java –version" into a Windows command window. If Java has been installed successfully, the command window will display the version number of the currently installed version of Java.

3rd Party Software

TaskGuide uses the following third party software:

- **Commons Codec** software for encoding and decoding data. License: open source. Web: http://commons.apache.org/codec/. Software file: commons-codec-1.3.jar.
- **HTML Parser** for parsing HTML text. License: open source. Web: http://htmlparser.sourceforge.net/. Software file: htmllexer.jar.
- **JLayer** software for playing MP3 audio files. License: GNU Lesser General Public License (LGPL). Web: http://www.javazoom.net/javalayer/javalayer.html). Software file: jl1.0.1.jar.
- **Sferyx JSyndrome** HTML editing software. License: commercial, Sferyx Srl. Web: http://www.sferyx.com/. Software file: HTMLEditorEnterprise.jar.

Uninstalling TaskGuide

Before installing a new version of TaskGuide, uninstall any previous versions of TaskGuide. To uninstall the TaskGuide software, select the *Add or Remove Programs* menu choice from the Windows Control Panel. In the list of programs, select the TaskGuide program to be removed and press the *Change/Remove* button. For example, to uninstall TaskGuide version 1.1, *select TaskGuide Version 1.1* from the list of programs in the *Add or Remove Programs* dialog and press the *Change/Remove* button.

Installing TaskGuide

The TaskGuide Authoring Tool installation program file (e.g., taskguide_author_2.0_beta.exe) installs TaskGuide Procedure Editor and TaskGuide Execution Tool software and documentation.

By default, the installation program installs the files in a folder named *c:\Program Files\Stottler Henke\TaskGuide 2.0 beta*. In this document, this folder is called the *installation folder.*

Launching the TaskGuide Procedure Editor

The TaskGuide Procedure Editor is a software application written in the Java programming language.

To launch the TaskGuide Procedure Editor, select *TaskGuide/Procedure Editor* from the *Windows Start/All Programs* menu. Selecting this menu item runs a Windows command file named *runTaskGuideEditor.bat* in a folder named *software* in TaskGuide installation folder. This command file starts the Java run-time system and launches the TaskGuide Procedure Editor.

Using the TaskGuide Procedure Editor, you can create and edit TaskGuide procedures. In addition, the Editor enables you to run the procedure you are editing to test its behavior. If you run procedures that use classes and methods defined by other Java libraries, you should edit file *runTaskGuideEditor.bat* to include the library (.jar) files.

Chapter 3 – Types of TaskGuide Applications

This chapter describes how you can use TaskGuide to create training tutorials, training simulations, debriefing systems, and task support systems.

Training Tutorials

A TaskGuide training tutorial displays a series of screens that:

- Present direct (didactic) instruction to students,
- Pose questions and accept and evaluate each student's response,
- Provide hints and feedback on each student's responses,
- Branch to the appropriate next screen based on each student's response, and
- Tracks the student's responses and scores using files or a SCORM learning management system.

TaskGuide procedures can use response-specific hints, feedback, and branching to provide instruction that adapts to the student's background and responses during the tutorial.

TaskGuide procedures can pose two types of questions:

- Text input questions that prompt the student for free-text responses, and
- Multiple choice questions that prompt the student to select one or more choices.

TaskGuide provides specialized support for authoring multiple choice items that include response-specific hints and feedback. This feature is described in *Chapter 9 – Multiple Choice Item Extension.*

Training Simulations

You can create TaskGuide procedures that implement simple training simulations. For example, in a branching scenario, each step in the procedure presents a scene or situation using text, graphics, and/or audio and prompts the student for a decision or action. After the student enters a decision or action, TaskGuide advances to the procedure step that represents the next situation. TaskGuide procedures can determine the next situation based on the student's last decision or action and the values of procedure variables that store the state of the simulation and previous decisions and actions.

In addition, you can embed Java graphical user interface objects within a screen. In this way, you can create training systems that interleave custom training simulations with didactic instruction and other types of training.

Debriefing Systems

You can create TaskGuide procedures that implement debriefing systems that discuss the student's performance soon after an interactive exercise such as a simulated scenario. The debriefing can assess and critique noteworthy instances of student performance. It can also help students reflect on their performance by asserting relevant facts, providing hints, asking probing questions about the student's analysis and decision-making, and delving more deeply with follow-up facts and questions.

TaskGuide procedures that implement debriefing systems contain groups of steps that discuss all of the possible topics that might be worthwhile to discuss after a training simulation exercise. The TaskGuide author can specify assessment and branching logic in the debriefing procedure that identifies the discussion topics that are relevant to each student's performance and executes only the groups of steps that support those topics. The assessment and branching logic can select relevant discussion topics based on the values of TaskGuide procedure variables that store:

- The student's actions and decisions during the simulation,
- Assessments of the student's knowledge and skills, based on the student's actions and decisions in the context of the simulation events and state, and
- The student's responses to questions and other prompts presented earlier in the debriefing.

Performance Support Systems

TaskGuide procedures can be used to implement performance support systems that present step-by-step instructions and guidance to help users carry out procedural tasks, analyze data, assess situations, operate equipment and software, troubleshoot problems, select and configure products or systems, develop plans, or make decisions.

Procedures can include branching logic to display only the steps that are relevant to the user's current situation, goals, and decisions made so far. TaskGuide procedures can incorporate calculations that query, filter, and calculate data and configure data displays that are tailored to the user's goals or situation. These calculations can even generate possible options and make recommendations.

TaskGuide supports the gradual incorporation of automation into procedures. For example:

1. An initial version of a procedure might present manual steps that simply provide instructions and guidance. If it is necessary to use an external system to retrieve information or execute operations, the TaskGuide procedure could provide specific instructions on how to operate an external system to carry out the specific task. The procedure could also provide guidance on how to analyze the information or make a particular decision.
2. Using pre- and post-calculations, a TaskGuide procedure could exchange data and commands with external systems or databases. That is, instead of telling the user how to operate the user interface, the TaskGuide procedure could

communicate with the external system directly to query data, present this data to the user, prompt the user for decisions, and then send one or more commands to the external system that execute the user's decisions. In this way, TaskGuide procedures can provide a more convenient task-oriented user interface to the external system that reduces the need for the user to find and operate the appropriate user interface controls provided by the external system.

3. The procedure could be further automated with algorithms that analyze information, recommend decisions, and present these recommendations to the user to confirm their execution or override them. As confidence in a recommendation algorithm solidifies, the semi-automated step could be replaced by a fully automated step so that no confirmation by the user is required.

Adaptive Questionnaires

TaskGuide procedures can be used to implement adaptive questionnaires that select or generate questions to ask based on the user's answers to previous questions and other relevant information retrieved from databases. Adaptive questionnaires can assess the user's situation and recommend actions more efficiently by focusing on the information that is relevant to each particular situation.

TaskGuide branching logic can be used to decide whether to display or skip each set of screens in the procedure. TaskGuide's *conditional inclusion* feature can be used to select which text to display in a particular screen, based on information previously entered by the user, queried from databases, or computed by TaskGuide calculations.

Chapter 4 – Sample Procedure Editing Session

Editing a Task Support Procedure

If you have not already done so, run the sample Procedure Execution session described in the *TaskGuide Users' Guide* to become familiar with the AHM Reset procedure.

The example TaskGuide procedures are stored in a folder named *examples* in the TaskGuide installation folder which, by default, is *c:\Program Files\Stottler Henke\TaskGuide 2.0 beta.* On Windows 7, files in the *Program Files* folder are read only and cannot be written. To edit the following example TaskGuide procedure, copy the entire *examples* folder to another location outside of the *c:\Program Files* folder and set the folder to be writable as follows: Using Windows Explorer, right-click over the *examples* folder icon and select *Properties* from the context menu. Click on the *General* tab and uncheck the check box labeled *Read-only* and press *OK*. When prompted in the next dialog box, select the radio button labeled *Apply changes to this folder subfolder and files* and press *OK*.

The following sample procedure editing session should take about 5 minutes to complete.

1. Start the TaskGuide Procedure Editor by selecting *TaskGuide/Procedure Editor* from the Windows *Start/All Programs* menu.
2. Select top-level menu choice *File/Open* to open procedure file AHM_Reset.tg in the TaskGuide *examples/satellite* folder.
3. Select top-level menu choice *File/Save As…* to save this file with a new name such as *AHM_Reset_new.tg* in the *examples* folder.
4. Left-click on the *Procedure* tab in the left pane to see the *Procedure Summary*, a hierarchical summary of the procedure's step nodes and group nodes. (This summary may already be displayed when you first open the procedure file.)
5. Click on *AHM Reset* at the top of the Procedure Node Summary to select the top-level procedure node.
6. Click on the *Preview* tab in the Details Pane on the right side of the Procedure Editor Window. This Preview display shows how the top-level procedure node is displayed by the Procedure Execution Tool or Applet to the operator.
7. Click on the *Summary* tab in the right-hand side Details Pane to see a summary of the procedure as a whole.
8. Click on the *Graphical HTML Editor* sub tab.
9. Click and drag your mouse to select the phrase "Do not perform passplan earlier than six hours after eclipse exit."
10. Press the italics button, on the toolbar. This operation italicizes the selected text.

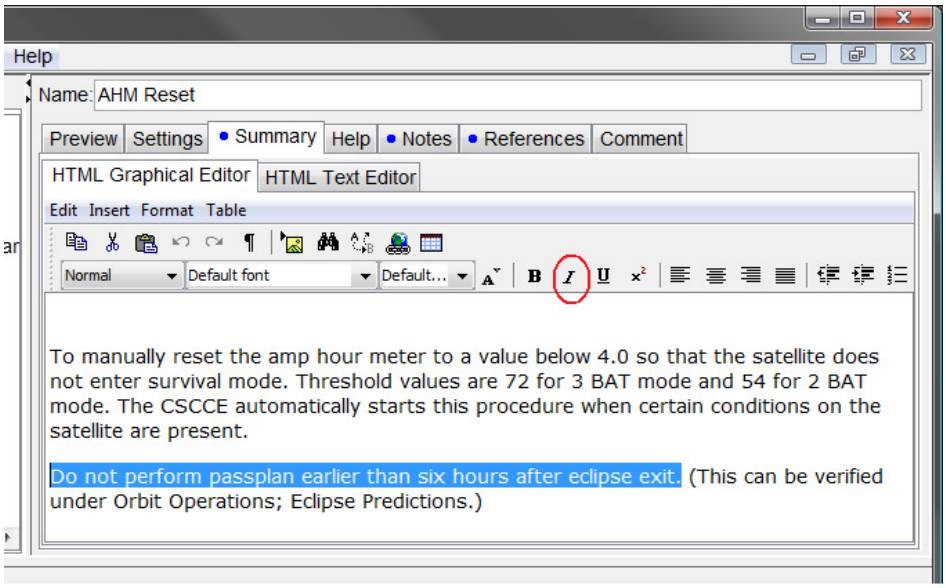


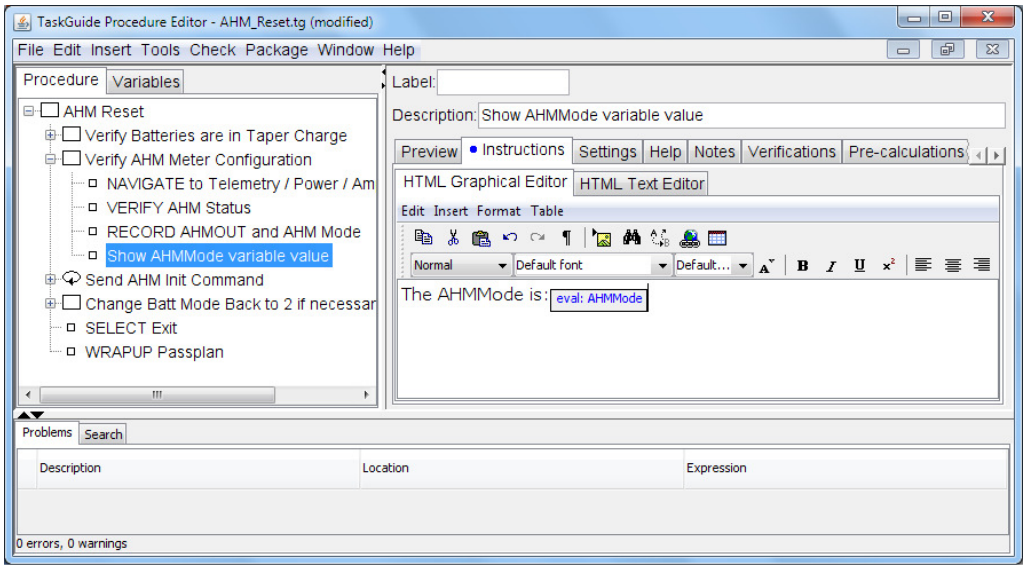
Figure 1 – Use the TaskGuide Procedure Editor's HTML Editor to select text to italicize

11. In the *Procedure* tabbed window on the left, open the second group node labeled *Verify AHM Meter Configuration* by double-clicking on its icon. You should now see the three step nodes that belong to this group.
12. Select the third step node in this group labeled *RECORD AHMOUT and AHM Mode* by clicking on the group node's label.
13. Click on the *Instructions* tab.
14. Click on the Graphical HTML Editor sub tab.
15. Press the left mouse button after "2 batt" and type "eries".
16. Press the left mouse button after "3 batt" and type "eries".
17. Select the group node labeled *Verify AHM Meter Configuration.*
18. Select menu choice *Insert/Step Node* to add a new simple step node to the end of this selected group node. The right-hand side Details Pane prompts you for the details of this new simple step.
19. Type "Show AHMMode variable value" in the text input control labeled *Description*.
20. Click on the *Instructions* tab.
21. Click on the *Graphical HTML Editor sub* tab.
22. Click over the text editing area in the right-hand side pane and type:

    `The AHMMode is:`
23. With the cursor positioned after the colon, press the right mouse button and select Special Tags/ *Embedded Expression* from the context menu. The Editor inserts a gray box that contains "eval: edit expression". This gray box is a placeholder that indicates an embedded expression. When the procedure is run, TaskGuide calculates the value of this expression and displays this value in the step's instructions.

24. Double-click on the gray box to edit the embedded expression.

25. In the popup dialog titled "Edit Expression", type "AHMMode" (without the quotation marks) in the text box that is below "Expression".
26. Press OK. The figure below shows how the Editor should appear at this point.



27. Select top-level menu choice *Check/Check for Syntax Errors* to check the procedure for errors.
28. Select top-level menu choice *File/Save* to save the procedure file.
29. Test the procedure using the TaskGuide Procedure Execution Tool by selecting *File/Run Procedure* from the TaskGuide Editor main menu.
30. Click on the top-level procedure node labeled *AHM Reset*. Press the green *Next Step* button at the bottom of the right hand side pane. TaskGuide advances to **step 1**.
31. At **steps 1 through 4**, press the *Next Step* button to advance to the next step.
32. At **step 5**, notice that the radio buttons display your new radio button labels *2 Batteries* and *3 Batteries*. Enter *10* in the text input control and select the radio button labeled *2 Batteries*. Press the green *Next Step* button to indicate that you have finished entering data and are ready to advance to the next step.
33. At **step 6**, notice that the instructions reports the value of variable AHMMode by displaying "The AHMMode is 2". During procedure execution, TaskGuide replaced the embedded expression "AHMMode" with its value, 2.
34. Select the top-level *File/Exit* menu choice to exit from the Procedure Execution Tool.

Chapter 5 – Overview of the Authoring Process

The TaskGuide Procedure Authoring Process is comprised of four high-level steps:

1. Create a *TaskGuide package folder* that stores procedure files and media files,
2. Assemble media files referenced by the procedures,
3. Create, edit, and test each procedure,
4. Prepare and test the procedure package for distribution to users.

Step 1 - Create a Package Folder

A *TaskGuide package folder* is a Windows folder that stores all of the files that will be distributed to users to run a set of related procedures. For example, a package folder could implement a training course where each procedure implements a training tutorial or training simulation. Or, a package folder could contain a set of related task aids, one TaskGuide procedure per task.

You can create a package folder using Windows Explorer as you would create any other Windows folder. Using Windows Explorer, you should also create a *procedure subfolder* for each procedure in the procedure package. Each procedure subfolder stores the TaskGuide procedure file and other media files that are used exclusively by the procedure. For example, you might create three subfolders named *tutorial1, tutorial2,* and *tutorial3* that store procedure files and media files for each of three training tutorials.

Using Windows Explorer, you can also create one or more *shared media subfolders* that store media files that are shared by two or more procedures. For example, you might create a single subfolder named *shared_media* that stores all media files that are shared across procedures.

You can create TaskGuide procedures that launch lower-level procedures. This feature enables you to create a top-level procedure that users can use to browse and run each of the other procedures provided by the procedure package.

Step 2 - Assemble Media Files

Steps within TaskGuide procedures can include HTML image tags (`<img>`) and hyperlink tags (`<a>`) that refer to image files and hyperlinked documents in formats such as (but not limited to) HTML, Adobe Acrobat (PDF), and Microsoft Word. These files can be stored within subfolders in the procedure package folder along with other files that are distributed to users of the procedures.

To include an image or document file within the procedure package, copy the file to the appropriate procedure-specific or shared subfolder in the package folder. For example, if

a PDF document is hyperlinked from the introductory tutorial, you might copy the PDF file to subfolder named *intro* that stores all files for this tutorial.

Although TaskGuide enables you to specify the height and width of image to resize it automatically, the image will look better if you use third party software to reformat the image file so that it has the desired height and width.

In addition, procedures can refer to external image files and hyperlinked document files that are provided by web servers. These external files are not stored in the package folder. Instead, they are referenced within procedures by their full web address, beginning with "`http://`"). Since those external files are not distributed to users and they may be changed or relocated, you should use references to external files only if:

- You are confident that those files will remain at those web addresses, or
- It is not practical or legal to copy those files into the TaskGuide package folder and distributed to users.

Step 3 – Create, Edit, and Test Each Procedure

1. Use the TaskGuide Procedure Editor's *File/New* menu item to create a new procedure.
2. Use the *File/Save As* menu item to save the procedure in its own *procedure subfolder*.
3. Edit the procedure by adding and configuring step nodes and group nodes.
4. Use the *Check/Check for Syntax Errors* and *Check/Check URLs* menu items to check for syntax errors and references to invalid media files and web addresses. TaskGuide reports errors in the *Results Pane* at the bottom of the window.
5. Run the *File/Run Procedure* menu item to test that the procedure runs correctly. If your procedure contains branching logic that executes different steps in response to user inputs, test your procedure using these various inputs. It is a good practice to check, save, and test your procedure periodically.
6. You can enter comments in step nodes, group nodes, and the top-level procedure node. Procedure-level comments are useful for storing a log of changes that have been to the procedure as a whole.

Tip: you can continue to copy additional image and document files after you have started to create and edit procedure files. However, you should copy each image and document file to a subfolder within the package folder before you refer to the file from within your procedure

Step 4 – Prepare and Test the Procedure Package for Distribution

After you have created all of the procedure files and assembled all of the media files needed by your procedure package, you can prepare your procedure package for distribution.

If the TaskGuide Execution Tool has already been installed on each user's computer, you can simply distribute the package folder and files. Users can use the TaskGuide Execution Tool to open and run the TaskGuide procedure files in the package folder. Or, you can include in the package folder all of the software files needed to run the procedures in the package. To simplify the distribution of these files, you can use WinZip or similar file utility to create an easily distributed archive file that contains all of the files in the procedure package.

If some of the procedures in the procedure package use additional Java library (.jar) files or require special Java launch parameters, you should create and distribute custom command (.bat) files that specify these Java libraries and launch parameters. The TaskGuide Procedure Editor generates these command files when you select *Generate Command Files* from the *Package* menu.

TaskGuide provides two Java applet classes that execute procedures:

- The standard TaskGuide Procedure Execution Applet – executes procedures that are not managed by a SCORM learning management system (LMS), and
- The TaskGuide SCORM Procedure Execution Applet – executes procedures that implement SCORM Sharable Content Objects (SCOs) and are launched by a SCORM LMS.

If your users will run procedures using the standard Procedure Execution Applet, it will be necessary to create HTML files that contain appropriately-configured applet tags. The TaskGuide Procedure Editor generates these HTML files when you select *Generate Applet HTML Files* from the *Package* menu.

If your users will run procedures as SCORM Sharable Content Objects (SCOs) served by a SCORM-compliant learning management system (LMS), you can select *Generate SCORM Files* from the *Package* menu to generate the files needed to import the TaskGuide package as a SCORM package into a SCORM-compliant LMS. These files include a SCORM manifest file, one SCORM Applet HTML file for each SCO, and SCORM schema definition (.xsd) files. TaskGuide places these files in the package folder, one-level above the subfolders that contain procedure files and supporting media files.

Chapter 6 – Procedure Editing Concepts

Presenting Information

The main purpose of most steps in TaskGuide procedures is to present information to the user. In a training tutorial, these steps present subject matter to the student. In a task support system, these steps present relevant data, recommendations, and instructions that the user should follow. In a training simulation, each step describes a scene or situation within the scenario. In a debriefing, these steps critique the student's performance during a simulation exercise.

TaskGuide steps can present information using formatted text, images (including animated GIF files), and embedded Java user interface objects. In addition, TaskGuide can play an optional audio file while displaying each step.

Text formatting and embedding of images are specified using the HyperText Markup Language (HTML). Section *HTML Graphical* Editor in *Chapter 7 – Using the Procedure Editor*, page 42, describes how to edit the instructions in each step using the HTML editor within the Procedure Editor.

Variables

You can create one or more variables that store values that have been:

- Entered by the user via a user interface control (also known as a form field), or
- Computed within TaskGuide *calculations* from other variable values or constants.

Variable values can be:

- Updated by and displayed within user interface controls within step instructions,
- Used by pre- and post-calculations to compute other variable values or used as arguments to calls to Java methods (*Pre- and Post-Calculations*, page 15),
- Recomputed and updated by calculations (*Pre- and Post-Calculations*, page 15),
- Checked by TaskGuide branching test conditions to determine the next step to execute (Looping and Branching, page 15),
- Referenced by conditional inclusion tags (*If/Endif Tags*, page 52) and embedded expressions (*Embedded Expression Tag*, page 53), in step instructions to generate instructions dynamically, and
- Tracked, using TaskGuide's tracking feature (page 16).

Figure 2 below shows the various ways in which variables can be queried and updated. A two-headed arrow indicates that the TaskGuide software component, such as a graphical user interface (GUI) control, queries and updates variable values. A one-headed arrow

indicates that the software component, such as a branching test condition, only queries variable values. The various ways in which variables can be queried and updated are described in detail in the rest of this chapter.

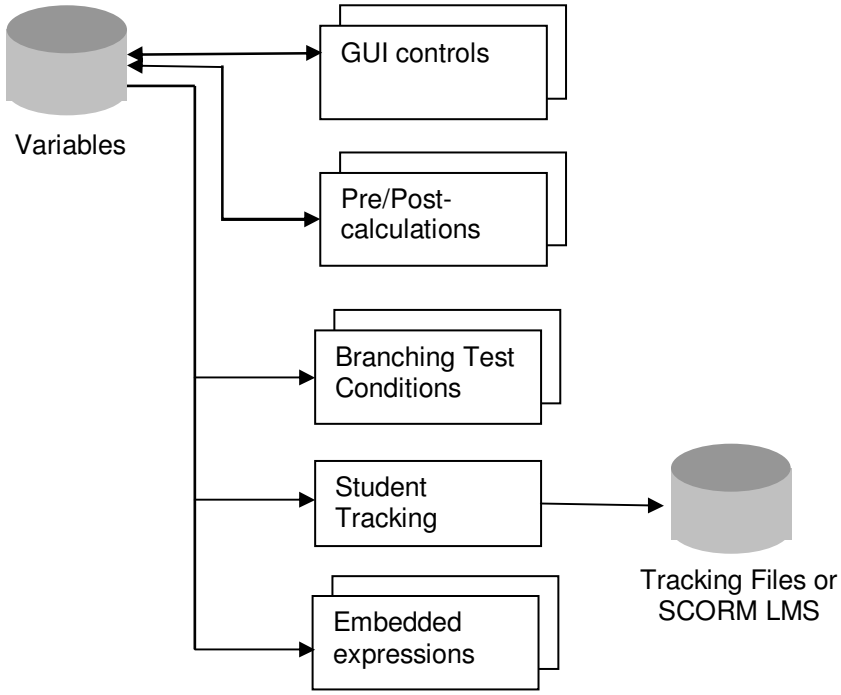
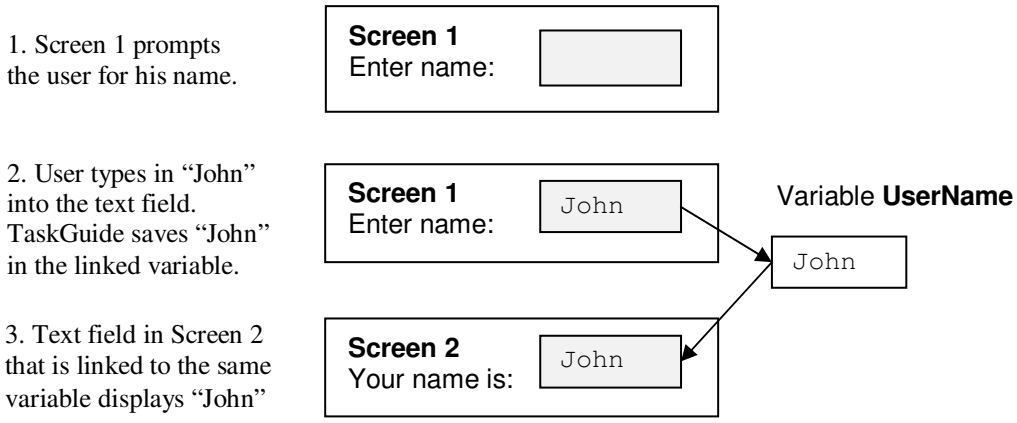


Figure 2 – GUI controls and Pre/Post-calculations query and update variable values. The other components query variables values.

User Input

Step instructions can contain graphical user interface (GUI) controls such as text fields, check boxes, and radio buttons that prompt the user for input. Each GUI control is linked to a TaskGuide variable that stores information entered by the user via the GUI control. In addition, each GUI control also displays the value already stored in its linked variable. In the example shown below, after the user types in "John" into a text field in Screen 1 and presses the *Next Step* button to indicate completion of the step, TaskGuide stores this value in the associated variable named *UserName*. Because Screen 2 contains a text field that is linked to the same variable, its text field initially displays "John" as well. The user can type in a new value into this text field.



TaskGuide also provides a Multiple Choice Item object that stores prompts, options that the user can select, the correct set of options, and response-specific hints and feedback. Once you configure a Multiple Choice Item object with this information, you can quickly create steps in the procedure that present these prompts, options, hints, and feedback.

Pre- and Post-Calculations

Steps can contain *calculations* that evaluate expressions containing constants, variables, and calls to Java methods. Calculations can save the values of these evaluated expressions in procedure variables. These variable values can be used within calculations in other steps that are executed later. *Pre-calculations* execute at the beginning of each step before the step's instructions are presented to the user. This is useful for querying and/or computing variables values that are displayed in GUI controls or within expressions embedded within the step's instructions. *Post-calculations* execute at the end of the step, after the user has followed the step's instructions, entered data into GUI controls, and pressed the Next Step button to indicate completion of the step. Post-calculations are useful for checking, interpreting, processing, saving, or acting upon the user's inputs. In addition, you can create invisible steps that are not displayed to the user. Their sole purpose is to execute pre- and/or post-calculations.

Looping and Branching

By default, TaskGuide displays each step in linear sequence, starting with the first step. However, a TaskGuide group node can be configured with a test condition that is calculated from the values of variables and/or constants. For example, the test condition:

```
number_of_iterations > 2
```

is true only if the value of numeric variable *number_of_iterations* is greater than 2. If the test condition is true, the steps within in the group node are executed. If the test condition is false, the steps within the group node are skipped.

If a group node is configured to *loop*, steps within the group are executed repeatedly until either:

- The value of the group node's test condition is false, or
- An exit step within the group is executed and the value of its exit condition is true.

Providing Hints and Feedback

The Multiple Choice Item object enables you to specify response-specific hints and feedback for each possible option that the student might select. Once this information has been entered into the object, you can select a menu items that create steps that provide hints and feedback based on the options selected by the student.

If you require more flexible selection of hints or feedback, you can use other types of branching logic supported by TaskGuide. For example, you could specify a branching group node that displays a hint or feedback only if:

- the user carries out a particular sequence of actions (vs. a single action), or
- the user carries out a particular action *and* certain variables representing simulation state have specified values.

Tracking Student Performance

TaskGuide enables you to track:

- *Progress* – the set of tracked group nodes that the user has completed,
- *Interactions* - input values entered by the user via user input controls, and
- *Scores* – that quantify the student's performance with respect to a learning objective.

You can specify whether a group is tracked by using the group node's *Settings* tabbed window. You can use a step node's *Settings* tabbed window to specify the tracking of interactions or scores.

TaskGuide provides methods for saving tracking information:

- *File-based* – progress, interactions, and scores are saved in a file
- *SCORM-based* – progress, interactions, and scores are reported to a SCORM-compliant learning management system (LMS).

To enable tracking, insert an invisible step node that contains a pre-calculation that starts the tracking process. When the TaskGuide Execution Tool runs the procedure using file-based tracking, this step node should execute after the student name has been assigned to a pre-defined variable named *_userName.* A context menu item named *Tracking/Insert Start Tracking Mode* in the *Procedure* pane simplifies the insertion of this special step node (Procedure Tab, page 19).

If the procedure runs within the TaskGuide SCORM Procedure Execution Applet and is launched by a SCORM LMS, TaskGuide automatically sets the *_userName* variable to the SCORM *cmi.student_name* parameter provided by the LMS, so the procedure does not need to prompt for the user name.

Grouping Nodes

You can create group nodes that contain step nodes and lower-level group nodes. There are several reasons why group nodes are useful:

- By grouping together related step nodes and subgroups and organizing them hierarchically, it is easier to keep track of many step nodes in a procedure.
- A group node can be configured as a branching group. This enables TaskGuide to skip an entire group of steps, based on the values variable values that are included in the group node's Boolean (true/false) test condition.
- A group node can be configured to loop, so TaskGuide can execute a set of steps repeatedly. For example, a procedure might prompt the user to select an operation at the beginning of each iteration of the loop and then use branching logic to execute just the steps that carry out the selected operation.
- A group node can be tracked so that completion of the group node is saved in a tracking file or reported to a SCORM learning management system. If the user suspends execution of the procedure and then re-runs the procedure at a later time, TaskGuide resumes execution as the first step that follows the last tracked group that the user completed.

Dynamic Instructions

You can author procedures so that they generate instructions for some steps dynamically, based on values of variables that are entered by the user, received from external systems and databases, or computed from expressions that refer to other variables when the procedure is executed. The values of expressions that refer to these variables can be inserted into dynamically-generated instructions by **embedding expressions** within the instruction's HTML text. When a step is executed, TaskGuide generates the step's instruction by evaluating each embedded expression and replacing the expression with its value.

TaskGuide also supports **conditional inclusion of HTML text.** This feature enables you to specify that a block of HTML text should be included and displayed within a step's instructions only if a particular Boolean condition is satisfied. Typically, this test condition evaluates the values of certain variables which are set during procedure execution.

Compared to static instructions, dynamically-generated instructions can filter information to present instructions that are more succinct and targeted to the situation. They can also compute default values for input parameters, generate recommended actions, and provide tailored hints and instructional feedback to students.

Chapter 7 – Using the Procedure Editor

This chapter describes how to operate the TaskGuide Procedure Editor user interface.

The TaskGuide Procedure Editor enables the procedure author to create, edit, and save files that contain TaskGuide procedures. The Procedure Editor provides a multiple document interface, so you can edit more than one procedure at time. This multi-document capability is useful for comparing procedures and for copying parts of one procedure into another procedure. The TaskGuide Procedure Editor also provides utilities for generating other support files, such as dictionary files, and for preparing procedure packages for deployment.

The Procedure Editor displays three window panes: the Procedure Catalog Pane on the left, the Details Pane on the right, and the Results Pane at the bottom, as show in

Figure 3. You can click and drag the dividers that separate the panes to adjust their sizes.

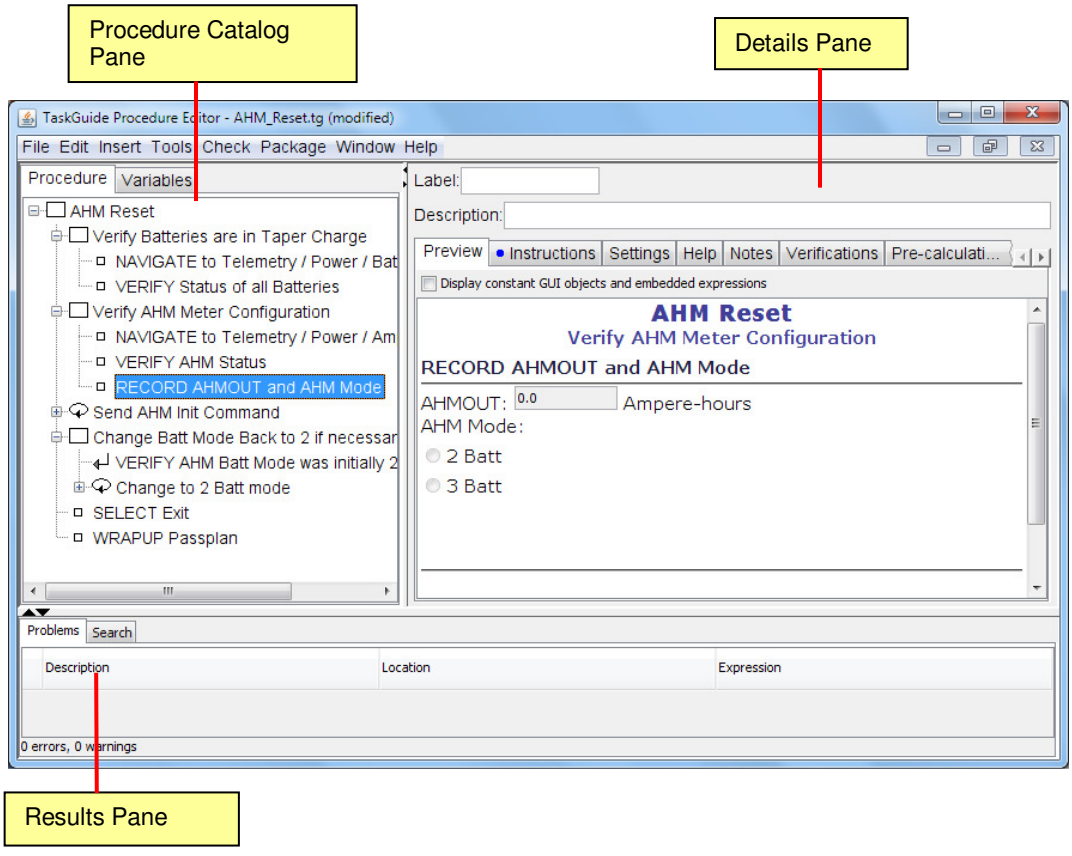


Figure 3 – Procedure Editor user interface.

Procedure Catalog Pane

The Procedure Catalog Pane displays three tabs that enable you to select a part of the procedure to display and edit.

Procedure Tab

Click on the *Procedure* tab to display a Procedure Summary, a hierarchical list of the procedure's group nodes and step nodes. This is similar to the Procedure Summary Pane displayed by the Procedure Execution Tool. This interactive display enables you to:

- View and scroll through an overview of the procedure's group nodes and step nodes,
- Insert, cut, copy, paste, and delete nodes,
- Click on a node to select it and edit its details in the Details Pane, and
- Group, ungroup, and move nodes within the procedure.

To **select a node** and display its properties in the Details Pane, left-click on the node in the Procedure Summary.

Press the right mouse button to display the **context menu**. This menu provides access to the following editing operations:

Cut - Removes the selected procedure node and copies it to the clipboard.

Copy - Copies the selected procedure node to the clipboard.

Paste - Inserts the procedure node into the procedure that was previously copied to the clipboard. If a destination group node is selected, the node in the clipboard is added to the end of the group. If a destination step node is selected, the node in the clipboard is inserted after selected step node.

Delete - Removes the selected procedure node without copying the node to the clipboard.

Insert Step Node - Creates a new, empty step node and inserts it into the procedure. hierarchy as the new last node. If a destination group node is selected, the new node is added to the end of the group. If a destination step node is selected, the node in the clipboard is inserted after selected step node.

Insert Exit Step Node - Creates a new exit step node and adds it to the procedure at the end of the selected group node or after the selected step node.

Insert Conditional Branches Node – Creates a new conditional branches node and adds it to the procedure at the end of the selected group node or after the selected step node.

Insert Group Node - Creates a new, empty group node and adds it to the procedure at the end of the selected group node or after the selected step node.

Move Node - Displays a cascading menu that enables you to select one of the following submenu items to move the selected node within its group:

Up	Ctrl-U
Down	Ctrl-D
Top	Ctrl-T
Bottom.	Ctrl-B

You can also drag and drop a node within the Procedure Summary to move it to a new location within the procedure. If you drop a dragged node onto a destination group node, the dragged node will be added to the end of the group node. If you drop a node onto a step node, the dragged node will follow the step node.

Group – Creates a new group node that contains the selected nodes as children.

Ungroup - Replaces a group node containing zero or more children nodes with the children node.

Tracking – Displays a cascading menu that provides submenu items for inserted steps that support procedure tracking.

Insert Start Tracking Node – Inserts a step node with a pre-calculation that starts tracking.

Insert Prompt for User Name – Inserts a step node that prompts for the user name.

Multiple Choice Item - If a Multiple Choice Item object has been selected in the Variables tabbed window, this context menu item is enabled. Selecting this menu item causes the Editor to display the following submenu items:

Insert Step to Show Question and Select Options

Insert Step to Provide Hints

Insert Step to Show Question, Show Reference Options, and Provide Feedback

Insert Group to Conditionally Provide Feedback

The items in this submenu are described in *Inserting New Steps and Groups* on page 70 in *Chapter 9 – Multiple Choice Item Extension.*

Variables Tab

Click on the Variables tab to display the Variables Summary, a list of all of the variables in the procedure, as shown in Figure 4.

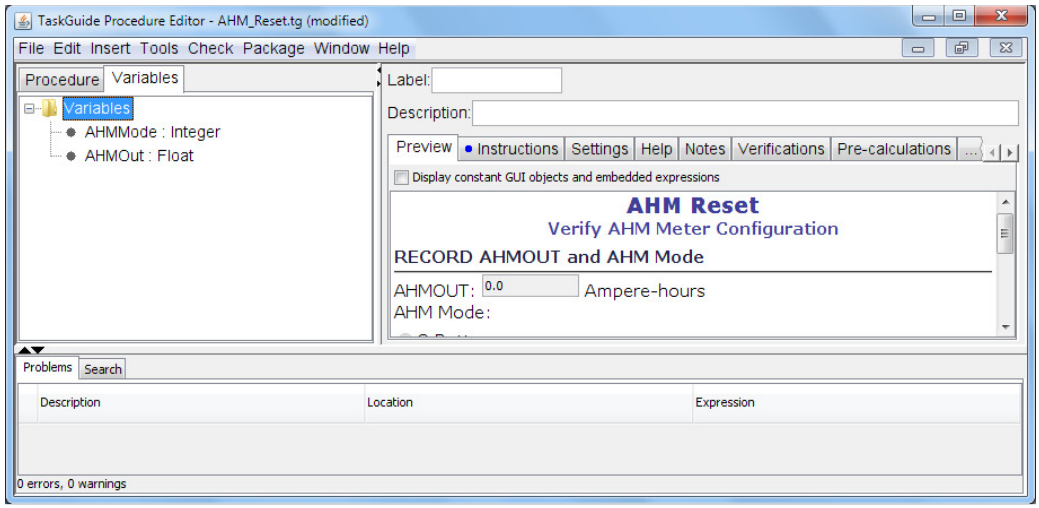


Figure 4 – The Variables tabbed window lists all of the procedure's variables.

To select a variable, click the left mouse button over its icon or name. TaskGuide highlights the selected variable using a blue background.

Click the right mouse button to display a **context menu** that provides the following options:

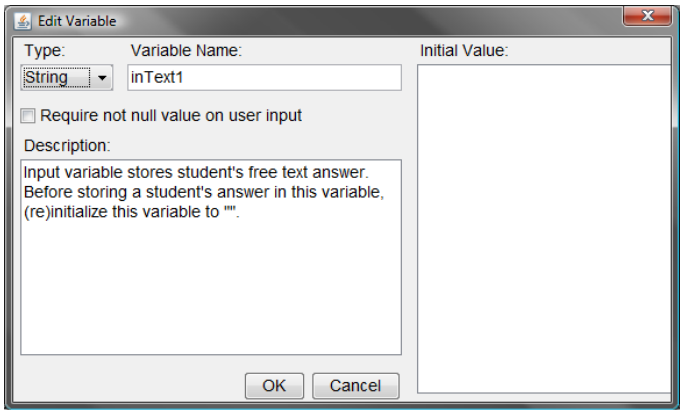
Cut - Removes the selected variable and copies it to the clipboard.

Copy - Copies the selected variable to the clipboard.

Paste - Pastes the variable in the clipboard (if any) into the Variables Summary. If the Variables Summary already contains a variable with the same name, the pasted variable will be assigned a new name by pre-pending "Copy of " to the variable's original name.

Delete - Removes the selected variable without copying it to the clipboard.

New Variable - Creates a new variable. The Editor will display the Edit Variable dialog that prompts you for the variable's name, type, and optional initial value.



Variable types include Integer, Double, String, one-dimensional List, two-dimensional Table, and Java Object. If the variable stores a Java Object, the Editor will prompt you for the full Java class name of the Object. If you check *Require Non-Empty Value on User Input*, TaskGuide will require the user to enter a value in all user interface input controls that are linked to this variable.

You can drag a variable name from the list of variables in the Variables tabbed window

and drop it into text fields in the Details Pane. For example, you can drag and drop a variable name into a table cell in the *Expression* column in the Pre-calculations and Post-calculations tabbed windows to specify calculations more quickly.

In addition to variables that you create, TaskGuide automatically creates the following variables:

_audioPlayer – stores a Java object defined by the AudioPlayer class. You can call methods of this object within pre- and post-calculations to invoke the more advanced audio playback features such as playing multiple audio files in quick succession or dynamically selecting the audio file to play when the procedure is executed. To play a single audio at a step, specifying the name of the audio file in the step's Setting tab is sufficient.

_audioPlayerPanel – stores a GUI object that provides buttons to play and stop playing audio files. If you select context menu item Audio Player / Insert Player Control from the Instructions tabbed window, the TaskGuide Editor inserts an embedded GUI object tag that is linked to this variable.

_childrenExecutionMode – If a procedure launches a sub-procedure, the value of the pre-defined _executionMode variable in the sub-procedure, described below, is set to the value of the _childrenExecutionMode of this procedure.

_executionMode – The value of this variable is set automatically by TaskGuide to one of the following String values: *"normal", "review",* and *"browse*". If this variable equals "*review"* or "*browse"*, tracking is automatically disabled. If this variable equals "*normal"*, TaskGuide tracks the execution of the procedure if the procedure is authored to track execution. Procedure tracking is described in *Tracking Student Performance* (page 16).

The TaskGuide Execution Tool runs procedures in "*normal"* mode. If the procedure is executed by the TaskGuide SCORM Procedure Execution Applet and implements a SCORM SCO, the execution mode is set by the SCORM LMS.

_nextStep – The Conditional Branches node advances to the step whose label equals this variable if none of its branch conditions are true. This feature enables you to use pre- or post-calculations to determine the next step.

_parameters – A procedure can be called with a list of parameters that provide information about how the procedure should be executed. This list is automatically stored in variable named *_parameters*. The procedure can call Java methods to extract the size and contents of this list. For example:

_parameters.size() = the number of parameters in the list.

_parameters.get(0) = the 1st parameter in the list

_parameters.get(i) = parameter i+1 in the list

_procedureURL – This variable stores the URL of the folder that contains the TaskGuide procedure currently being executed. This variable can be supplied as a

parameter to Java methods such as URL.URL() to specify a base URL.

_tgForm – The values of HTML form fields within a step's instructions are copied to their associated TaskGuide variables only after the user presses the *NextStep* button. The `_tgForm` variable provides access to the values of HTML form fields in a step's instructions *before* the user has pressed the *NextStep* button. Unlike the other TaskGuide variables, this variable can only be referenced within JavaScript expressions that are attributes of the push button tag (page 51) and the JavaScript expression tag (page 58). To access form field values, call the `get()` method and supply the name of the form field's variable as the argument. For example, suppose the step's instructions contain a text input field that is bounded to a variable named `inputText`:

```
<input name="inputText" size="10" type="text" />
```

A JavaScript expression in a JavaScript expression tag or push button tag can access the value of this text field as:

```
_tgForm.get("inputText")
```

To set the value of an HTML form field in a step's instructions, call the *set* method:

```
_tgForm.set(formFieldVariableName, newVariableValue)
```

_this – This variable stores the TaskGuideExecutionFrame Java object that contains the TaskGuide user interface and execution engine. This value is supplied as a parameter to some Java methods that are included in the TaskGuide application programming interface.

_userName – If the procedure implements a SCORM SCO and is launched by a SCORM LMS, TaskGuide automatically sets the value of variable `_userName` to the SCORM parameter named `cmi.student_name`. If a procedure launches a sub-procedure, the sub-procedure automatically inherits the value of the `_userName` variable in the calling procedure.

Results Pane

The Results Pane contains two tabbed windows. The *Problems* tab displays a table containing one row per problem detected by the TaskGuide Procedure Editor. The *Check for Syntax Errors* operation, selected from the *Check* menu, checks for problems such as invalid expressions or labels that refer to non-existent steps or groups. This operation does not check for undefined variables or functions. Instead, those types of problems can be identified when you run the procedure. The *Check URLs* operation checks that every file that is referenced by its relative filename or full web address (URL) can be accessed.

When you select *Find* or *Find in All Files* from the *Edit* menu, the *Search* tabbed window displays a table containing one row per occurrence of the search string anywhere in the procedure. If you double-click on a row, the TaskGuide Editor will display the relevant part of the procedure to show the selected problem or occurrence of the search string.

Details Pane

The Details Pane enables you to display and edit the details of each node. A text field at the top of this Pane displays and inputs the *Name* of each node. Another text field displays and inputs an optional *Label* that uniquely identifies the group node or step node. Labels of nodes are used by exit step nodes, conditional branch nodes, and the tracking feature to identify the node to exit from, branch to, or track, respectively.

The Details Pane contains tabbed windows that enable you to display, enter, or edit additional information about each type of node. The TaskGuide Procedure Editor displays different tabbed windows for each type of node, as shown in Table 2.

Tab Label	*Procedure*	*Group*	*Simple Step*	*Exit Step*	*Conditional Branches Node*
Preview	●	●	●	●	●
Settings	●	●	●	●	●
JavaScript	●				
Summary	●	●			
Instructions			●	●	●
Help	●	●	●	●	●
Notes	●	●	●	●	●
Verifications			●	●	●
Pre-calculations			●	●	●
Input Validations			●	●	●
Post-calculations			●	●	●
References	●				
Comments	●	●	●	●	●

Table 2 – Tabs in the Procedure Details Pane for each Type of Node

Preview Tab (All Nodes)

Click on the *Preview* tab to display the node's information as it would be presented to the operator in the Details Pane of the Procedure Execution Tool or Applet.

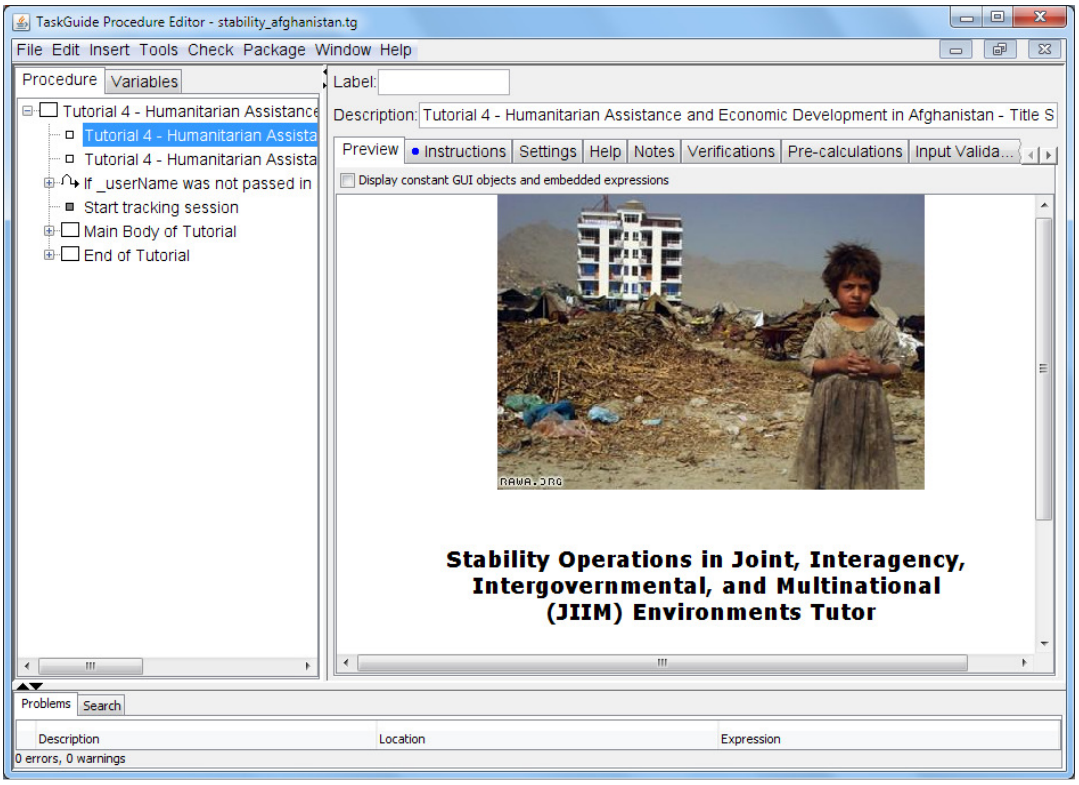


Figure 5 – The Preview Pane shows a step as it will be seen when the procedure is executed.

If the *Evaluate GUI objects and constant embedded expressions* checkbox is checked, the TaskGuide Editor evaluates the values of any expressions and Java graphical user interface objects that are embedded within the step's instructions, based on the *initial values* of any procedure variables that are referenced. It then inserts these values into the instructions. However, the values of some embedded expressions and GUI objects might be calculated when the procedure executes and not known when authoring. In this case, leave this checkbox unchecked, so that the Editor displays a placeholder for each embedded expression or embedded GUI object.

Settings Tab (Procedure)

This tabbed window enables you to specify settings for the procedure as a whole.

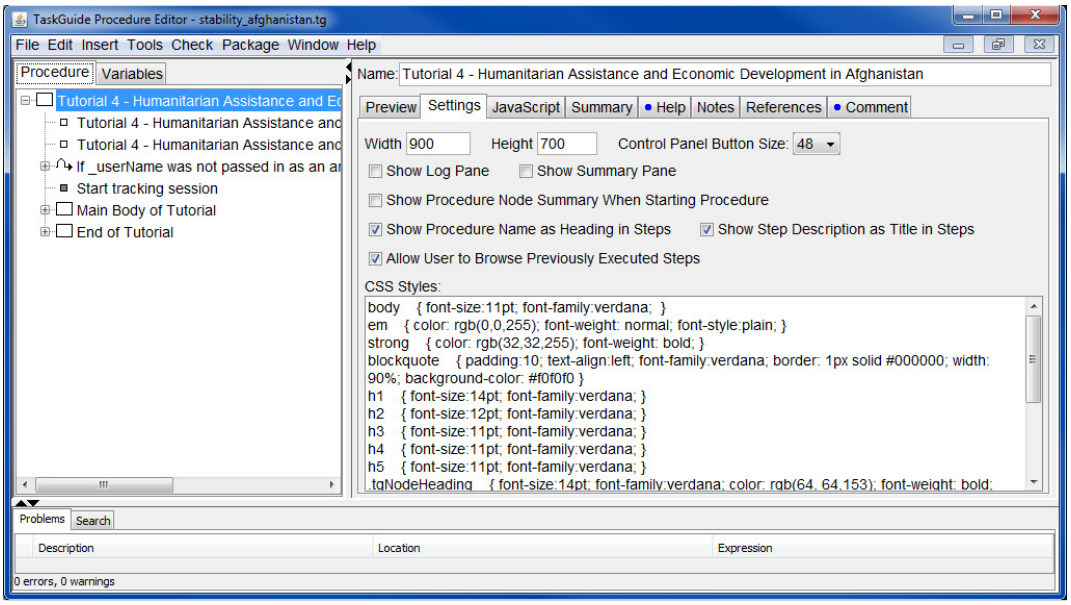


Figure 6 – Settings tabbed window for the top-level procedure node.

Width – of the Procedure Execution Tool when running this procedure.

Height – of the Procedure Execution Tool when running this procedure.

Control Panel Button Size – Size of the procedure navigation buttons in the Procedure Execution Tool's control panel, in pixels.

Show Log Pane – If checked, show the Log Pane in the lower left corner of the Procedure Execution Tool.

Show Summary Pane – If checked, show the Summary Pane in the upper left corner of the Procedure Execution Tool.

Show Procedure Node Summary When Starting Procedure – If checked, the Summary specified for the procedure node is displayed when the procedure is started. Otherwise, the first step in the procedure is shown.

Show Procedure Name as Heading in Steps – If checked, the procedure name is shown as the heading in any step in which the step's *Show Procedure Name as Heading* property is also true.

Show Step Description as Title in Steps – If checked, the description of any step is shown as the step's title if the step's *Show Step Description as Title* property is true.

Allow User to Browse Previously Executed Steps – If checked, the TaskGuide Execution Tool remembers previously-executed steps, so the user can press the Back button to review them.

CSS Styles – This property enables you to specify Cascading Style Sheet style rules that control the appearance of various HTML tags within the Step Details pane. By default, new TaskGuide procedures contain CSS style definitions for the following standard

HTML tags:

<body>	<h1>	<h4>
	<h2>	<h5>
	<h3>	
<blockquote>		

In addition, TaskGuide procedures contain definitions for the following CSS classes that are specific to TaskGuide:

.tgNodeHeading	Heading displayed at the top of the window. The heading is the name of the TaskGuide procedure.
.tgNodeSubheading	Subheading displayed below the heading. The subheading is the name of a TaskGuide module that contains the current step.
.tgNodeTitle	Title displayed below the subheading. The title is the description of the current step, specified in the step's Settings tab.
.tgRunProcedure	Run Procedure tag
.tgTerm	Word or phrase with an associated dictionary term
.tgComment	Temporary authoring comment about the procedure step that is embedded in a step's instructions. These comments augment comments entered in the *Comments* tab. This format can be applied using the *Format / Text Formatting / Authoring Comment* menu item in the HTML Graphical Editor.
.tgUserPrompt	Text that prompts the user or student for input. This format can be applied using the *Format / Text Formatting / User Prompt* menu item in the HTML Graphical Editor.
.tgJavaScript	Hyper-linked text in a JavaScript expressions tag.

TaskGuide generates HTML tags that refer to these CSS classes. You can also specify additional CSS classes and refer to these classes within custom HTML formats and templates. See *Appendix B – Adding Formats and Templates to the HTML Graphical Editor.*

JavaScript Tab (Procedure)

JavaScript Files – You can specify an optional, comma-separated list of files, separated by semicolons, that contain JavaScript function definitions. TaskGuide loads each of these files at the beginning of each procedure run. TaskGuide procedures can call these JavaScript functions from within TaskGuide expressions (e.g., within pre-conditions, post-conditions, test expressions within exit step nodes and conditional branches nodes, embedded expression tags, JavaScript expression tags, push button tags, and conditional inclusion if/then/endif tags.)

Import Java Classes – You can specify Java classes to be imported when the procedure is first run. If a Java class is imported, your TaskGuide procedure can refer to the Java class by its short name. Otherwise, if the Java class is not imported, your procedure can only refer to the class by its fully-qualified name which contains the Java package name.

Import Java Packages – You can specify a list of Java packages to import. Java classes within these imported packages can be referenced without their Java package name. Your

procedures will start execution somewhat faster if you minimize the number of imported packages, especially when running TaskGuide within an applet.

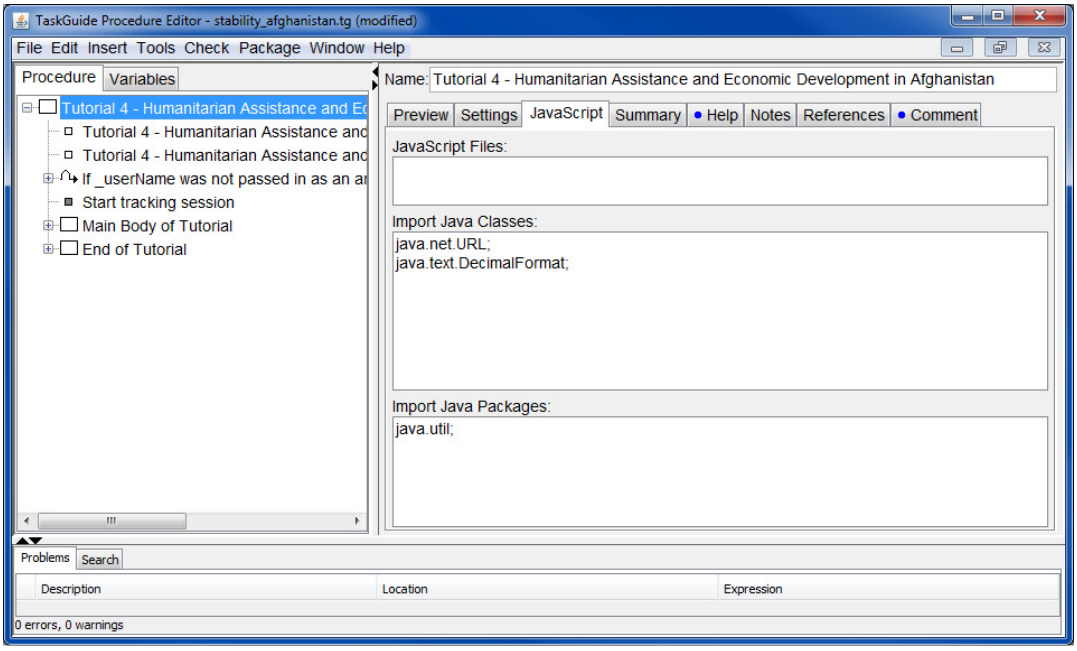


Figure 7 – JavaScript tabbed window for the top-level procedure node.

Settings Tab (Group)

Click the Settings tab to display and edit the looping, branching, tracking, and subheading properties of a group.

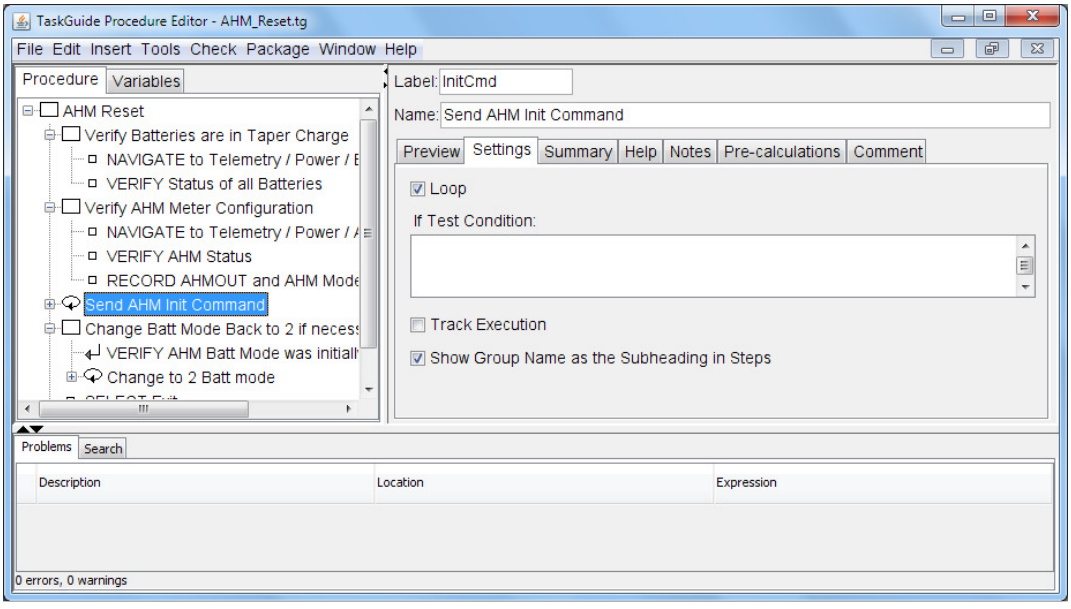


Figure 8 – Settings tabbed window for group nodes.

If Test Expression - To configure the group node as a branching group, enter a Boolean test expression. During execution, the Procedure Execution Tool evaluates this expression when the group node is entered. If the value of the expression is true, execution advances to the first step within the group. Otherwise, the Procedure Execution tool skips the group, and execution advances to the first step that follows the group.

Loop – Check the check box labeled *Loop* to configure the group node as a looping group that repeatedly executes the steps and subgroups within the group. If an *if test Condition* has also been entered, it is evaluated at the beginning of each iteration. TaskGuide executes the steps in the group as long as the test condition is true. When the test condition becomes false, TaskGuide advances to the step that follows the group.

Track Execution - Select *Track Execution* to track the execution of this group node. When a group node is tracked and the procedure is run within an applet, TaskGuide reports to the SCORM-compliant LMS when the group node has been executed. When a group node is tracked, and the procedure is run within a Java application, TaskGuide saves this information in a progress tracking file in the *tracking* subfolder in the procedure package folder. TaskGuide considers a group node to have been executed when the group has been entered and exited during execution. If the group node contains exit step nodes or other types of branching, some steps in the group may be skipped, so it is possible to execute a group without executing every step within the group.

Show Group Name as the Subheading in Steps - Check this checkbox to specify that the name of this group should be displayed as the subheading of any step contained within the group in which:

- The step's *Show Group Name as Subheading* property true (checked), and
- The step is not enclosed by a lower-level group node within this group that is also configured to *Show Group Name as the Subheading in Steps* If such a subgroup node does exist, the name of that subgroup, rather than this group, is displayed as the subheading of the step.

In Figure 9, TaskGuide displays the name of Group A as the heading of Steps A1 and B1. It displays the name of Group C as the subheading of Step C1

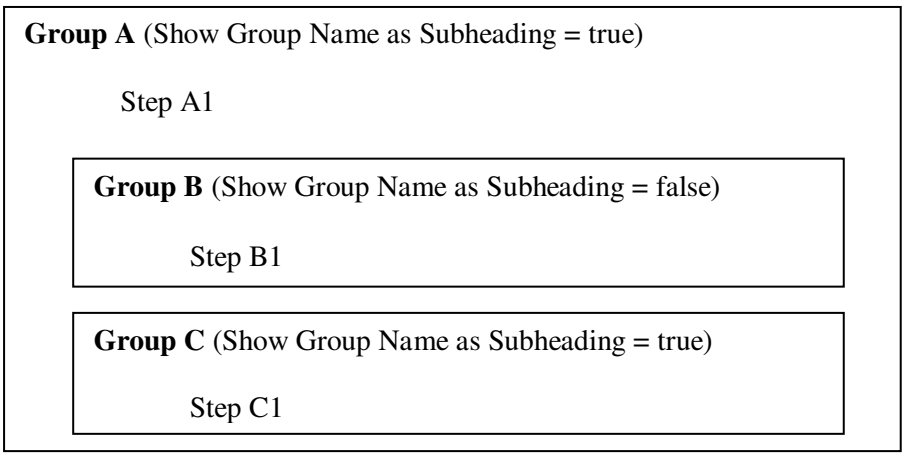


Figure 9 – Group Names shown as subheadings of steps

Settings Tab (Step)

The Settings tabbed window enables you to edit the following properties of step nodes:

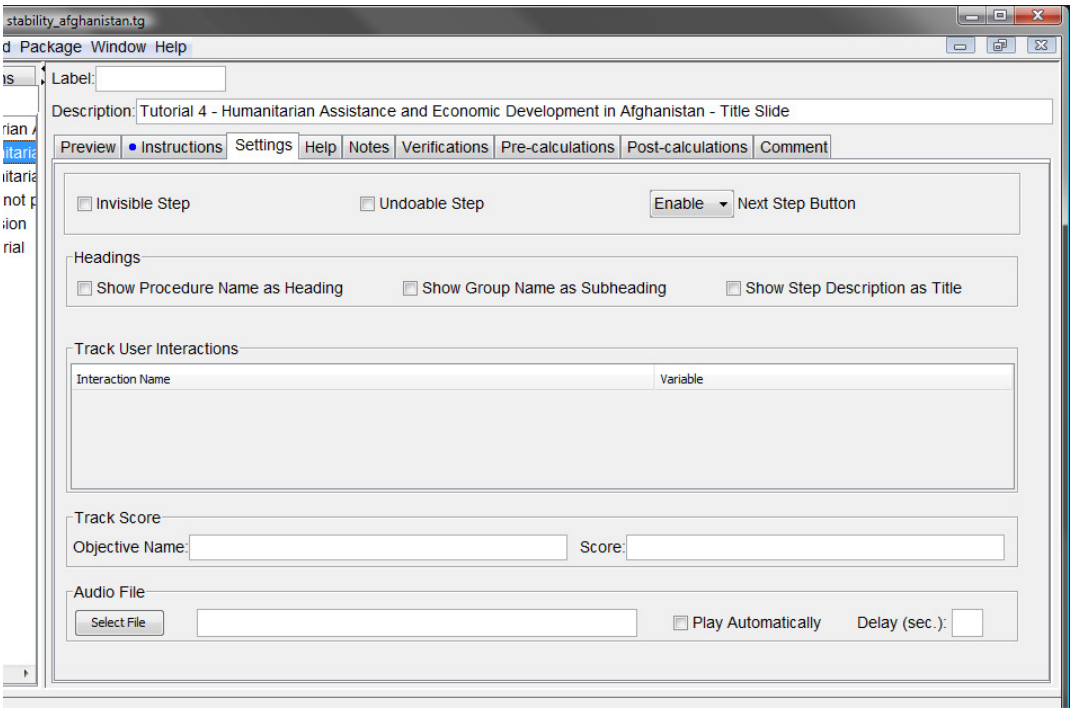


Figure 10 – Setting tab for step nodes

Invisible Step - Check this box to run the step's pre- and post-calculations without displaying instructions to the user.

Undoable Step (advanced feature) - Check this checkbox to specify that the step is undoable. The Undoable Steps feature is usually used when creating procedures that implement task support systems. Undoing a step is different from browsing previously executed steps. When users browse backward, they can only *view* a previously-displayed step. When they undo a step, they can re-enter values into the user input controls of those steps.

The TaskGuide Procedure Execution Tool enables the *Undo Step* button (so that it is clickable) if the previously-executed visible step was specified in the procedure as un-doable. When the operator presses this button, TaskGuide re-displays the instructions for this previous step. For example, assume that un-doable interactive step #1 is followed by automated step #2, and then by interactive step #3. When the operator presses the Next Step button at step #1, TaskGuide binds values of input control variables in step #1's instructions to variables, executes step #1's post-calculations, executes pre- and post-calculations for automated step #2 without displaying instructions, executes pre-calculations for step #3, and then displays step #3's instructions. Because step #1 was un-doable, the *Undo Step* button will be enabled when TaskGuide displays step #3's instructions. If the operator presses the *Undo Step* button, TaskGuide will reset the current step to step #1 and re-display step #1's instructions.

TaskGuide does not reverse the side-effects of any pre- or post-calculations when it undoes an action. It is the responsibility of the procedure author to ensure that the step with pre- or post-calculations can be undone safely. Procedure authors should carefully review the post-calculations of the un-doable step (step #1), the pre-calculations of the next interactive step (step #3), and all pre- and post-calculations in automated steps that lie between the two interactive steps. In some situations, there may be more than one possible step that follows the un-doable step, determined by the procedure's branching and looping logic. When this is true, authors should review all possible sets of pre- and post-calculations that might be executed after the undoable step is executed.

Enable/Disable/Hide Next Step Button - Select *Enable* (default) to enable the *Next Step* button. This enables users to press this button after they have completed the step and are ready to advance to the next step. Select *Disable* to disable his button so that it is grayed out and does not respond to mouse clicks. Select *Hide* to hide this button. These last two options are useful if you are using custom Java software to enable or show the button at the appropriate time. For example, a pre-calculation might display a custom user interface that enables the *Next Step* button only after the user has finished interacting with the user interface. The example procedure file *enableAdvanceNextStep.tg* in the *examples/integration* folder in the TaskGuide Editor installation folder shows examples of using these options. Run this example procedure by launching command file *runEnableAdvanceNextStep.bat*. The *TaskGuide Programmer's Guide* provides additional information about embedding TaskGuide within another program.

Show Procedure Name as Heading - If checked, the name of the procedure is displayed as the step's heading at the top of the screen. The font family and style of the heading are specified by the CSS style named *.tgNodeHeading*.

Show Group Name as Subheading - If checked, TaskGuide searches for the inner-most group node whose *Show Group Name as Subheading in Steps* property is set to true. The name of this group node is displayed as the step's subheading, below the step's heading. The font family and style of the heading are specified by the CSS style named *.tgNodeSubheading*.

Show Step Description as Title - If checked, the description of the step is displayed as the step's title, below the heading and subheading. The font family and style of the heading are specified by the CSS style named *.tgNodeTitle*.

Tracking and Reporting User Interactions and Objective Scores

To save or report tracking data (tracked user interactions and objective scores), the step must be included in a tracked module (Settings Tab (Group), page 28). TaskGuide tracks and saves tracking data for completed groups. That is, TaskGuide saves or reports tracking data when the tracked module that contains this step is completed. If the student exits from the procedure before

If tracking is enabled and the TaskGuide Execution Tool (Java application) is used to run the procedure, tracked interactions and objective scores are stored in files. A folder named *tracking* in the procedure package folder contains one subfolder for each student who has run a procedure. Each student's folder contains two files for each procedure that the student has started. One file saves the labels of tracked group nodes in the procedure that the student has completed. The name of this progress tracking file starts with

"Progress_" and ends with the name of the procedure file. For example, if the procedure file is named *proc1.tg,* the progress tracking file will be named *Progress_proc1.txt.* A second file keeps track of the interactions and objectives that have been saved for the student and procedure. The name of this results file starts with "Results_" and ends with the name of the procedure file.

If tracking is enabled and the TaskGuide SCORM Procedure Execution Applet (Java applet) is used to run the procedure, TaskGuide reports this data to a SCORM-compliant learning management system by calling the appropriate SCORM JavaScript function.

Track User Interactions

The Track User Interactions table enables you to specify that some or all of the values entered by the user via GUI controls in this step should be tracked. Tracked input values can be reported to a SCORM learning management system as SCORM *interactions*, or they can be saved in TaskGuide tracking files. The table displays a row for each TaskGuide variable that is linked to a GUI control in the step's instructions. To track a variable, enter a name in the column named *Interaction Name*. TaskGuide will save or report this interaction name along with the value entered by the user (the interaction value).

Track Score

You can specify the name of a learning objective and the student's score with respect to that objective. TaskGuide will save or report this **objective name** along with the value of the objective's **score**. You can use pre- or post-calculations to calculate scores and assign this value to a TaskGuide variable. For example, you can call the numReferenceOptionsSelected() method of the Multiple Choice Item object to retrieve the number of reference options the student selected. If the item contains one reference (correct) option, this method returns 1 if the student selected the correct option, and 0 if the student selects an incorrect option. You can compute a score by adding the number of correct answers selected by the student for a set of items.

Audio File

Select File – When you press this button, the TaskGuide Editor displays a file selection dialog that enables you to select an audio file to play when this step is executed. Audio files should be included in the TaskGuide package folder.

Play Automatically – Check this checkbox to specify that the audio file should be played automatically when the step is executed. If you leave this box unchecked, you should provide audio playback controls in the step's instructions that enable the user to command TaskGuide to play the audio file. To insert audio playback controls into a step's instructions, click the *Instructions* tab and the *HTML Graphical Editor* sub tab. Then, right-click and select *Insert Player Control* from the *Audio Player* context menu item.

Delay (sec.) – Enter a delay in seconds. When TaskGuide displays the step's instructions, it will wait this many seconds before playing the audio file. If this field is left blank, TaskGuide will play the audio file as soon as the step is displayed.

Settings Tab (Exit Step)

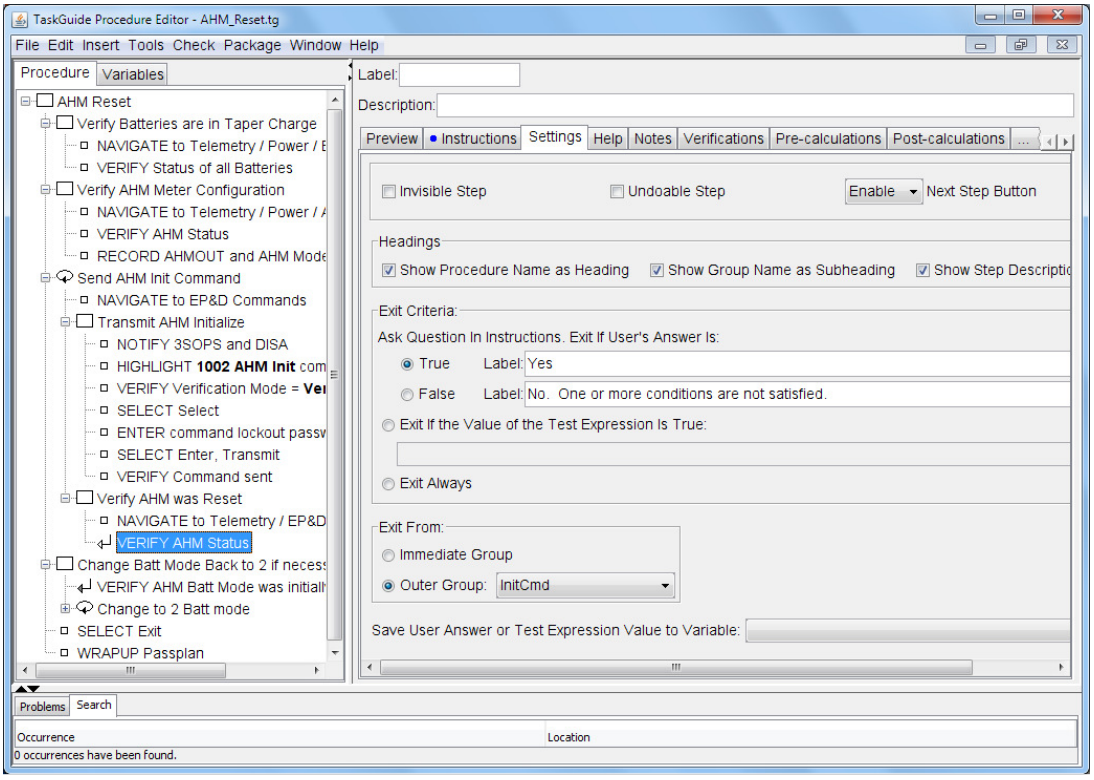


Figure 11 – The Settings tabbed window shows the exit node's exit criteria and behavior

The Settings tabbed window enables you to specify the following attributes of the Exit Step node, as described above in *Settings Tab (Step)*, page 30.

- Invisible Step
- Undoable Step
- Enable/Disable/Hide Next Step Button
- Show Procedure Name as Heading
- Show Group Name as Subheading
- Show Group Name as Subheading
- Show Step Description as Title

In addition, the Settings tabbed window enables you to display and specify when and how TaskGuide breaks out of a group node when it executes an exit step node, as shown in Figure 11.

When configuring each exit step node, you can select one of the following three **Exit Criteria:**

Ask question in instructions - Include a yes/no or true/false question in the step's instruction text, using the Instructions tab. Then, select either the radio button labeled *true* or the one labeled *false* to specify whether an exit should occur when the operator selects true or false, respectively. You can override the default labels displayed next to the radio buttons.

Exit if the value of the text expression is true - TaskGuide evaluates the exit node's test expression when it executes the exit step. If the value is true, TaskGuide exits from the immediate or outer group.

Exit always – When this exit step is executed, TaskGuide exits from the immediate our outer group in all cases. This option is usually used only in exit step nodes that lie within a conditional group node.

By default, if the exit step's exit criterion is satisfied, the Procedure Execution Engine **exits from** the group that immediately encloses the exit step node, and execution advances to the first step that follows this group. You can also specify exiting from an **outer group** (vs. the immediate group) that contains the exit step, identified by the group's label.

In Figure 12 below, if Exit Step C1 is configured to exit from its immediate group and its exit condition is true, TaskGuide advances to Step B2 because it Step B2 follows Group C. If Exit Step C1 is configured to exit from an outer group identified by label *Group B* and the exit condition is true, TaskGuide advances to Step A1 because A1 follows Group B.

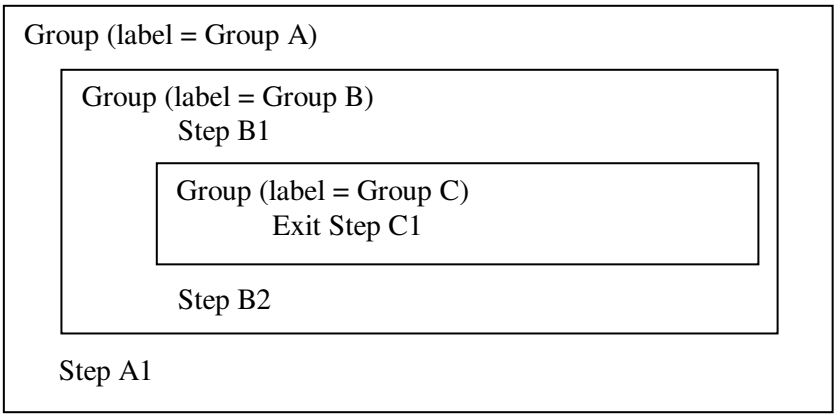


Figure 12 – Nested Groups and Exit Step C1

You can also select a variable in which TaskGuide saves the user's answer to the exit criteria question or the value of the test expression. This makes it possible for calculations or embedded expressions to refer to the user's answer.

Settings Tab (Conditional Branches Node)

The Settings tabbed window enables you to specify the following attributes of the Conditional Branches node, as described above in *Settings Tab (Step)*, page 30.

- Invisible Step
- Undoable Step
- Enable/Disable/Hide Next Step Button
- Show Procedure Name as Heading

- Show Group Name as Subheading
- Show Step Description as Title

The Settings tabbed window also enables you to display and edit the conditional branches and the default branch of a conditional branches node, as shown in Figure 13.

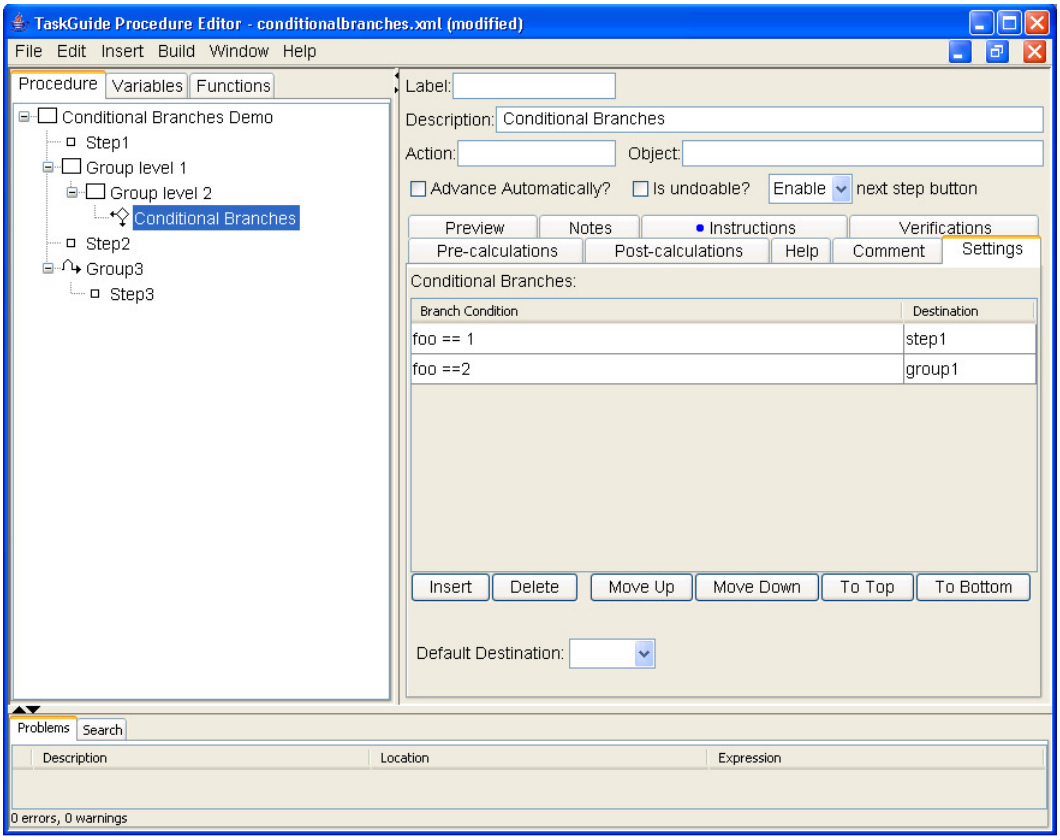


Figure 13 – The Settings tabbed window shows the conditional branches node's conditional branches and default branch

For each conditional branch:

- Specify a Boolean test expression as the condition to be tested, and
- Select the label of a step node or group node label from the drop down list as the destination if the test condition is true.

TaskGuide executes a conditional branches node by evaluating each branch condition in order. If a branch condition is true, TaskGuide advances to the destination node specified for the conditional branch. If none of the conditional branches are true, TaskGuide advances to the step node whose label is specified as the default destination. If no default branch has been specified, execution advances to the step whose label equals the value of variable `_nextStep`. If the value of this variable does not equal a valid step label, a run-time error occurs.

Summary Tab (Procedure, Group)

Click on the *Summary* tab to display and edit the summary of a procedure node or group node using the HTML Editor as described in section *HTML Graphical* Editor *on page 42*. TaskGuide displays the summary of a procedure node at the beginning of the procedure execution if the procedure node's *Show Procedure Node During Execution* property is checked. TaskGuide does not display the summary of group nodes during execution.

Instructions Tab (Step, Exit Step, Conditional Branches Node)

The Instructions tabbed window enables you to specify the content and format of the information displayed to the user when a step is executed. Instructions can also contain input controls that prompt the user for information. Information and input controls are specified using the HyperText Markup Language (HTML). In addition, instructions can contain **special tags** that provide additional functionality. Special tags are described in *Special Tags*, page 51.

The Instructions tabbed window contains two sub-tabs that provide different methods for editing the step's instructions:

- **HTML Graphical Editor** enables you edit the HTML-formatted instructions while viewing them roughly as they would be displayed during execution. This Editor is the easier and more convenient method of editing the step's instructions.

- **HTML Text Editor** displays a text editor for viewing and editing the raw HTML. This advanced feature is provided for people who understand HTML and need to view or edit HTML source code directly.

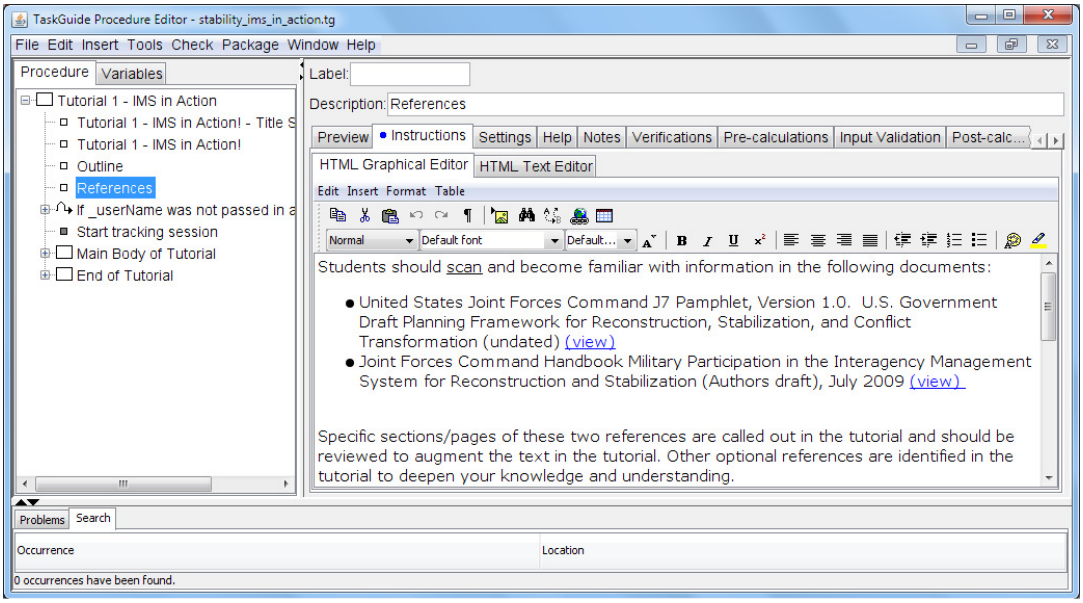


Figure 14 – The HTML Graphical Editor sub-tab in the Instructions tabbed window.

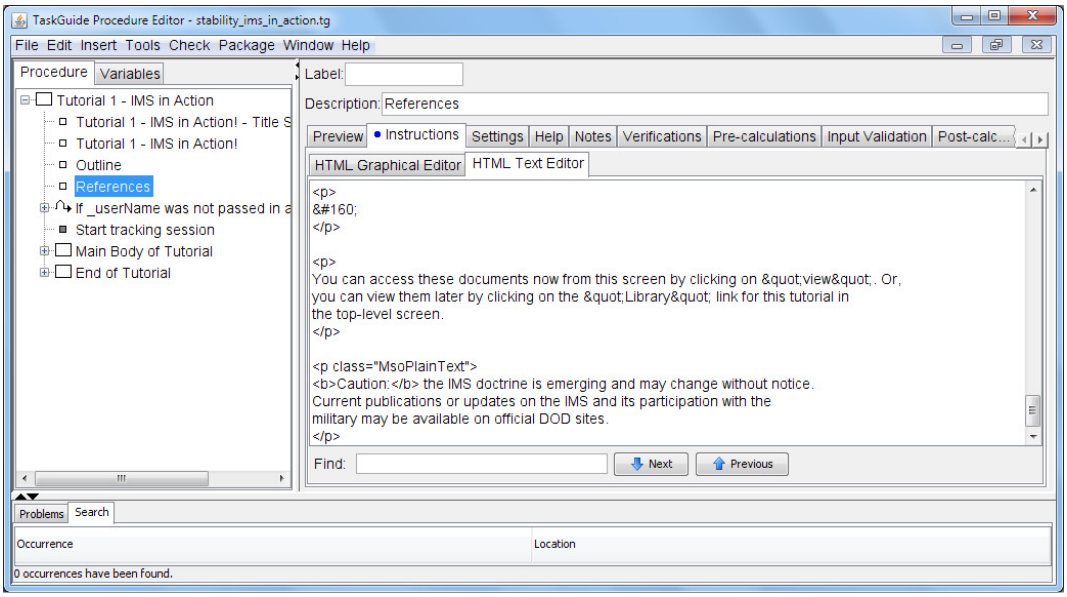


Figure 15 – The HTML Text Editor sub-tab in the Instructions tabbed window.

Help Tab (All Nodes)

Click on the Help tab to display and edit a step node's help information using the HTML Editor as described in section *HTML Graphical* Editor. When the user presses the Help button while the step's instructions are displayed, TaskGuide displays help information specified for the procedure as a whole (entered via the procedure node's Help tab), followed by help information specified for the step node. Currently, the TaskGuide Execution Tool does not display help text specified for group nodes.

Notes Tab (All Nodes)

The Notes feature is usually used when creating procedures that implement task support systems. A note specified for a step node is displayed when that step node is executed. A note specified for a group node is displayed when any step node within the group is executed.

Click on the Notes tab to display and edit one or more notes associated with any type of node. This tabbed window displays a table that contains one row for each note associated with the node. To insert a new note, press the *Insert* button. The Procedure Editor adds a row to the table. Click on a cell in the Type column to select one of three types of notes from the pull-down menu: Information (blue icon), Caution (yellow icon), or Warning (red icon). Type in the text of the note in the *Content* column.

To delete a note, click on a row to select the note. Then, press the *Delete* button. Use the *Move Up, Move Down, To Top,* and *To Bottom* buttons to re-order the selected note.

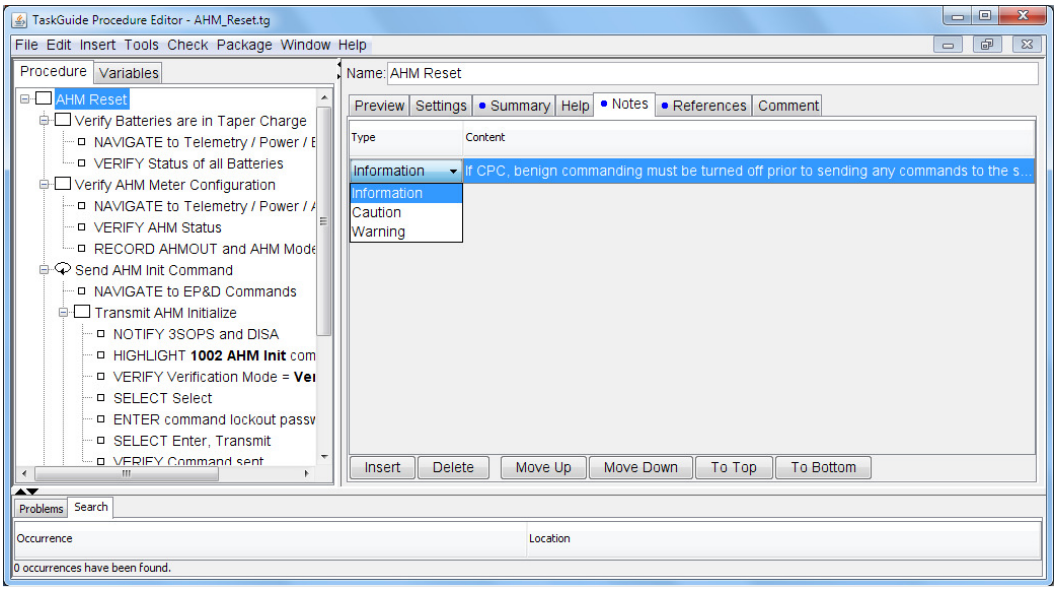


Figure 16 – The Notes Window enables you to display, enter, and edit notes associated with step nodes, group nodes, and the entire procedure.

Verifications Tab (Step)

Verifications are usually used when creating procedures that implement task support systems.

Click on the Verifications tab to display and edit a step node's verifications using the HTML Editor as described in section *HTML Graphical* Editor on page 42. Verifications describe to the user how to verify that a step was completed successfully.

Pre-Calculations Tab (Group, Step, Exit Step, Conditional Branches Node)

The Pre-Calculations tab enables you to display and edit the set of pre-calculations for a group or any kind of step node.

Pre-calculations specified for a group node are executed when the group is first entered during execution, before any steps in the group are executed and before the group's test condition is evaluated.

Pre-calculations specified for a step node are executed at the *beginning* of each step's execution, before the step's instructions are displayed. Pre-calculations are useful for calculating and assigning values to variables that are linked to GUI controls in the step's instructions or are contained within embedded expressions in the step's instructions. Pre-calculations are also useful for instantiating and configuring embedded GUI objects that are displayed by the step.

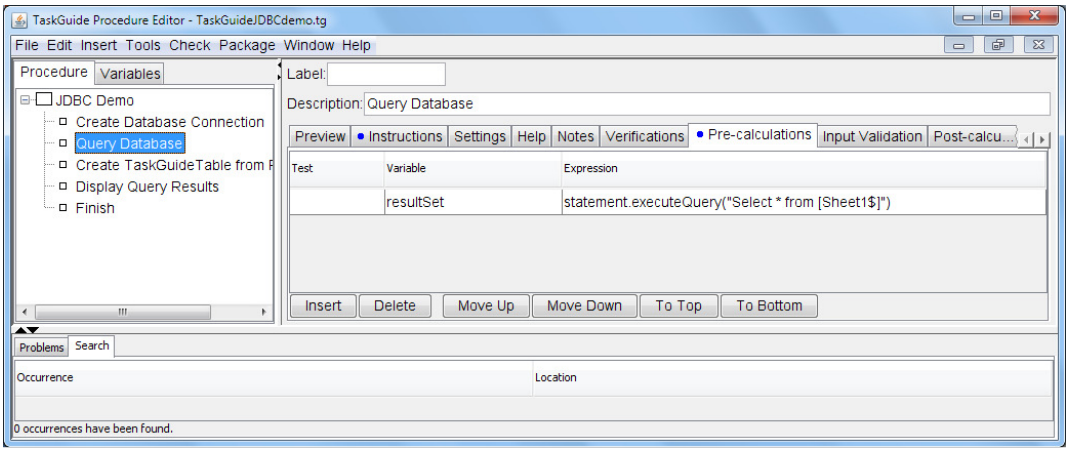


Figure 17 – The pre-calculations Tabbed Window SSS

The Pre-Calculations tabbed window displays a three-column table. Each row specifies one pre-calculation.

Test (optional) - If a test condition is entered in the left-most column labeled *Test*, the pre-calculation is evaluated only if the value of the test condition is true.

Expression - If there is no test condition or if the test condition is true, TaskGuide executes the pre-calculation by evaluating the expression specified in the *Expression* column. *Appendix A – TaskGuide Expression Syntax* (page 74) specifies the syntax of these expressions.

You can specify the expression by typing it into the *Expression* column. In addition, you can drag a variable name from the *Variables* tabbed window and drop it into the *Expression* column. If the expression is long, you can edit it in a pop-up dialog. To display this pop-up dialog, click once to select the row. TaskGuide will highlight the entire row. Then, double-click on the expression while depressing the *Shift* key.

If you type the first few letters of a variable name, TaskGuide displays a yellow pull-down menu that lists all of the variable names that begin with those letters. Press the up and down arrow keys to select a variable name, and press the *Return* key to insert the selected variable name into the expression.

If you enter a variable name whose value is a Java object, followed by a period, TaskGuide displays a yellow pull-down menu that displays the names of all of the object's fields and methods. If you type the first few letters of the field or method name, TaskGuide displays a filtered menu list that contains only the fields and methods that begin with those letters. Use arrow keys to select a field or method, and press the *Return* key to insert the selected field or method into the expression.

Variable (optional) – If the pre-calculation is executed and a variable is specified in the *Variable* column, the value of the expression is assigned to the TaskGuide variable. You can specify the variable for each pre-calculation by clicking in the *Variable* column and selecting a variable from a pull-down list.

You can right-click over this pane to display a **context menu** that provides operations

that streamline the insertion of common pre-calculations.

The **User Input / Clear Input Variable** context menu item enables you to add a pre-calculation to the currently-selected step that clears the value of a variable before displaying its value in an HTML input field linked to the variable. When you select this context menu item, the TaskGuide editor displays a *Select Input Variable* dialog that prompts you to select the variable to be cleared.

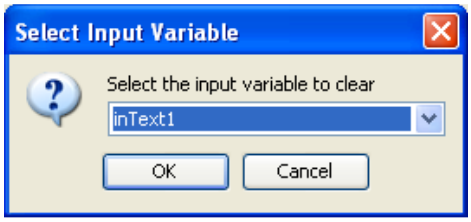


Figure 18 – This dialog prompts you to select the variable to be cleared.

Input Validation Tab (Step, Exit Step, Conditional Branches Node)

This tabbed window enables you to specify validation rules that check whether the user's input is valid. These rules are executed in sequence when user presses the *Next Step* button or when custom software instructs TaskGuide to advance to the next step using the TaskGuide application programming interface. If any of rules detect invalid user input, TaskGuide presents feedback to the user, based on the first rule that failed, and TaskGuide remains at the current step, instead of advancing to the next step, so the user can re-enter the input values.

Each input validation rule specifies three items:

- **Variable (optional)** – If a particular variable is specified and the rule's test condition is not satisfied, the GUI control(s) associated with that variable are highlighted.
- **Test** – This Boolean expression is evaluated when each rule is run. If the expression is true, the user's input is valid. If the expression is false, TaskGuide displays an error message and optionally highlights the relevant GUI controls to re-prompt the user for input.
- **Error Message** – This field contains a text string that is displayed to the user if the rule's test condition is false. If the first character of this field is an equals sign (=), the rest of the field is treated as a JavaScript expression, and the value of this expression is displayed to the user. This feature enables you to specify error messaged that are generated dynamically, based on the user's input and other variable values. For example, if the user incorrect enters 5 in a text field bound to variable *inputValue* and the Error Message field equals:

  ```
  = 'Invalid value: ' + inputValue
  ```

  TaskGuide will display to the user:

  ```
  Invalid value: 5
  ```

  Note: You can also configure a variable so that TaskGuide ensures that any input field in any step that is linked to the variable requires user

input (*Variables Tab*, page 20). If you configure a variable in this way, you do not need to check each individual input field for user input.

Post-Calculations Tab (Step, Exit Step, Conditional Branches Node)

Post-calculations are executed after the user has completed review and execution of a step's instructions, including data entry. They are useful for computing and saving variable values or sending data and/or commands to external systems and databases, based upon data entered into the instruction's input controls by the operator.

You can enter and edit post-calculations the same way you enter and edit pre-calculations, as described in the previous section, *Pre-Calculations Tab (Group, Step, Exit Step, Conditional Branches Node).*

You can right-click over this pane to display a **context menu** that provides operations that streamline the insertion of common post-calculations.

References Tab (Procedure Node)

Click on the References tab to display and edit a list of references for the procedure. This list is displayed to the user if the procedure has been configured to show top-level procedure summary at the beginning of the procedure.

Comments Tab (All Nodes)

Click on the Comments tab to display or edit comments associated with the node. Comments are displayed only within the Procedure Editor for use by procedure authors. They are not displayed by the Procedure Execution Tool to the user when the procedure is executed.

Tip: When authoring a procedure, you may find it useful to mark open issues or outstanding tasks by typing "TODO" in the Commands tabbed window, followed by a description of the issue or task. You can then use the TaskGuide Procedure Editor's search capability, invoked by selecting *Find* from the *Edit* menu, to find all comments describing open issues or tasks.

HTML Graphical Editor

HTML Graphical Editor tab enables you display and edit the HTML-formatted instructions while viewing them approximately as they would be displayed during execution.

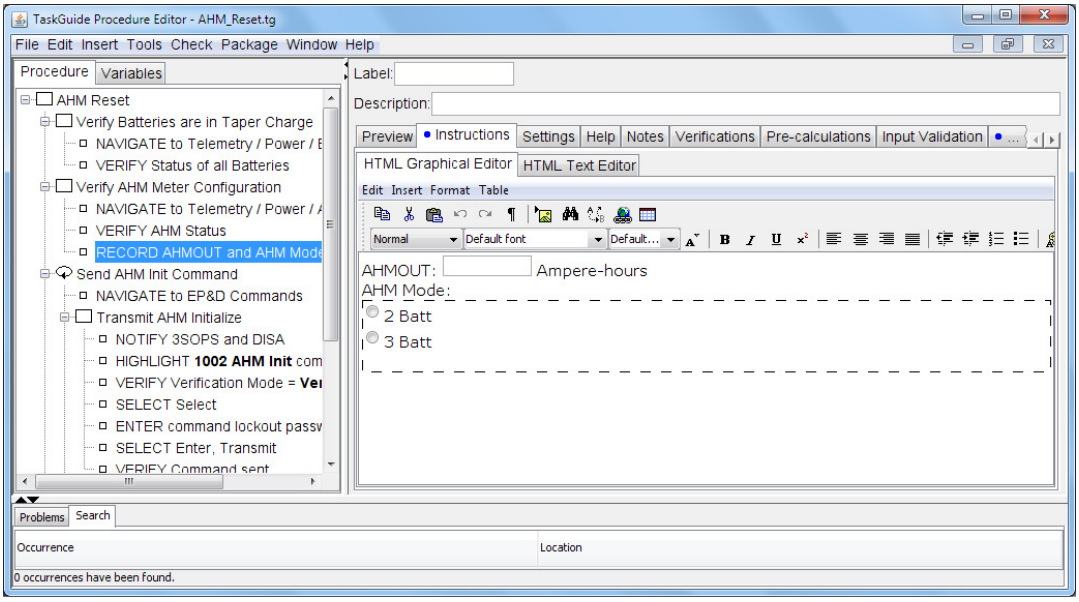


Figure 19 – The HTML Graphical Editor

You can type text into the HTML Graphical Editor. Or, you can paste text that has been copied from another application such as Microsoft Word.

The HTML Graphical Editor provides three ways to invoke editing operations:

- Select menu items from the top-level menu,
- Press buttons on the tool bar that is immediately below the top-level menu bar, and
- Select menu items from the context menu. Display the context menu by pressing the right mouse button.

Top-Level Menu

The **Edit Menu** provides the following menu items:

Undo – the last HTML editing operation.

Redo – the last HTML editing operation.

Copy – HTML text selected in the step's instructions to the Windows clipboard.

Cut - HTML text selected in the step's instructions and copy it to the Windows clipboard.

Paste - HTML text in the Windows clipboard into the step's instructions at the current text insertion point.

Paste Without Formatting – HTML text in the Windows clipboard into the step's instructions at the current text insertion point and remove any formatting of the pasted text. This operation is useful when you want the text in the step's instruction to follow the CSS style defined for *body*, rather than retain its original formatting in the document from which the text was copied.

Select All – HTML text in the step's instructions.

Find – a text string, searching the step's instructions.

Replace - a text string with a new text string in the step's instructions.

The **Insert Menu** provides menu items that enable you to insert HTML tags: Break `<br>`, Paragraph `<p>`, Horizontal Rule `<hr>`, Hyperlink `<a>`, Picture `<img>`, and Table `<table>`. URLs referenced within hyperlink tags can specify a file name that is relative to the procedure file, or they can specify full URLs beginning with "http://". In addition, URLs can specify destinations such as HTML anchors. For example, *webpage.htm#a1* specifies an anchor named *a1* within HTML file *webpage.htm.*

The **Insert / Input Form** menu enables you to insert input form fields: Single-line text box, Multiple-line text box, Check box, Radio button, or Dropdown menu. After you insert the input form, link it to a TaskGuide variable as follows:

1. Right-click on the input form to display the context menu,
2. Select Form Field Properties from the context menu,
3. In the pop-up window, type the name of the linked variable in the text field labeled *Field name.* If the input form accepts a single value (e.g., a text box, individual check box, set of radio butons, or dropdown menu configured not to accept multiple selection) , the variable can be a String, Integer, or Float. If the input form accepts multiple values (e.g., multiple check boxes linked to a single variable or a dropdown menu configured to accept multiple selections), the variable must be a List.

   If you insert a dropdown menu, you should also add Items to the menu. The Choice column displays the text displayed in the dropdown menu. The Value column contains the values that are assigned to the variable when the choice is selected. In the figure below, if the user select "Three" from the dropdown menu, TaskGuide assigns a value of 3 to variable selectANumber.

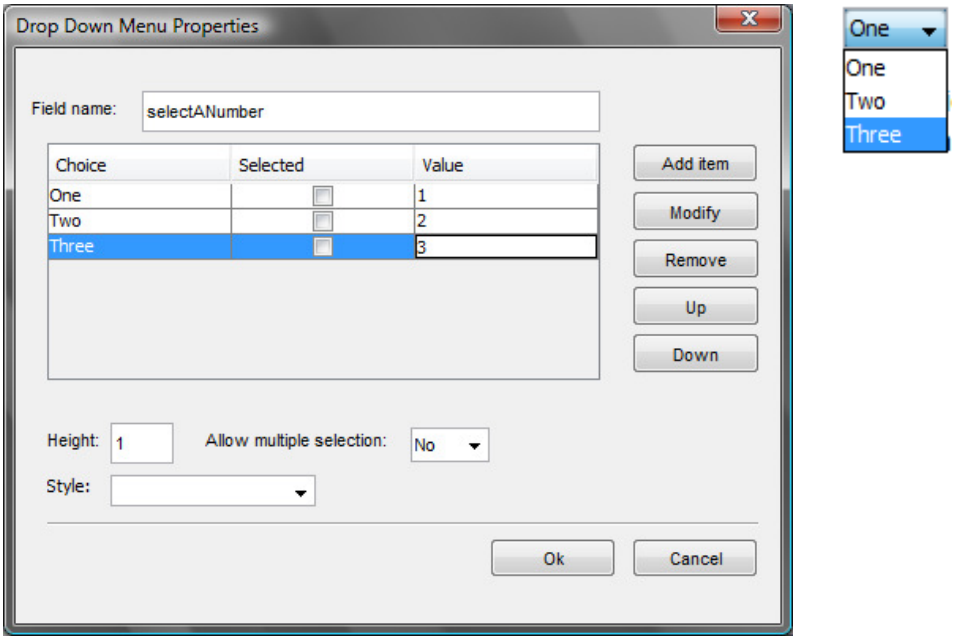


Figure 20 – Editing a dropdown menu

If a **Multiple Choice Item** has been recently selected in the Variables tabbed window, the Insert menu contains a menu item whose label is "Multiple Choice Item: ", followed by the name of the most recently selected multiple choice item object. A submenu enables you to select one of the following options:

- Insert Tag for Question
- Insert Tag to Select Options
- Insert Tag to Provide Hints
- Insert Tag to Compare Selected Options to Reference Options
- Insert Tag to Provide Feedback
- Insert HTML for Question
- Insert HTML to Select Options
- Insert HTML to Provide Hints
- Insert HTML to Compare Selected Options to Reference Options
- Insert HTML to Provide Feedback

These Multiple Choice Item operations are described in more detail in *Inserting HTML into an Existing Step* (page 71).

The **Insert / User Input** menu provides three items named *Insert Text Input Field, Insert Radio Buttons, and Insert Check Boxes.*

When the **Insert Text Input Field** item is selected, the Procedure Editor displays the following popup dialog:

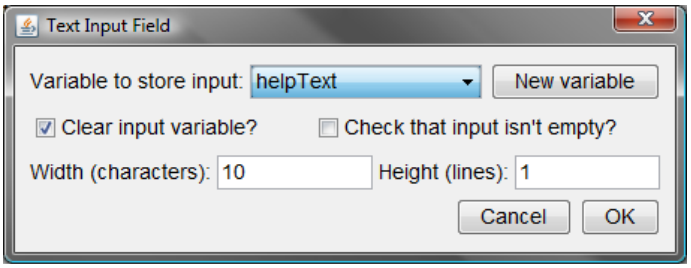


Figure 21 – The Text Input Field dialog enables you to specify an input field and linked variable.

After you fill in the dialog box and press *OK*, the Editor inserts a text field that is linked to a variable to store the user's input. This variable can be an existing variable you selected from the pull-down list or the new variable you created by pressing the *New variable* button. If you check *Clear input value?*, the new step contains a pre-calculation that clears the value of the input variable. If you check *Check that input isn't empty*, the step contains an input validation that checks that the user entered a value in the text field.

When the **Insert Radio Buttons** item is selected, the Procedure Editor displays the following popup dialog:

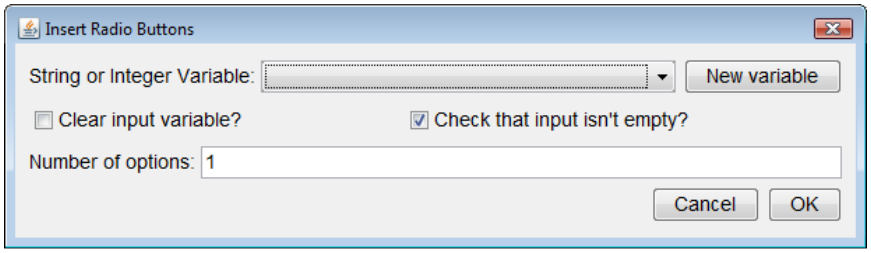


Figure 22 – Insert Radio Buttons dialog

After you fill in the dialog box and press *OK*, the Editor inserts a set of radio buttons that are linked to the variable you selected or created to store the user's input. Right click over each radio button and select *Form Field Properties* from the context menu to reset the value associated with each radio button.

When the **Insert Check Boxes** item is selected, the Procedure Editor displays the following popup dialog:

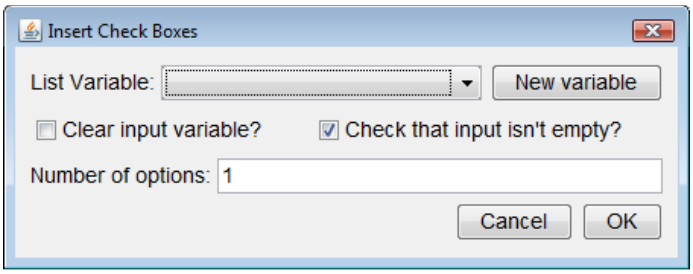


Figure 23 – Insert Check Boxes dialog

After you fill in the dialog box and press *OK*, the Editor inserts a set of check boxes that are linked to the List variable you selected or created to store the user's input. Right click over each check box and select *Form Field Properties* from the context menu to reset the value associated with each radio button.

The **Insert / Audio Player** menu item provides a submenu that enables you to *Insert Player Control* in the step's instructions. These controls enable the user to play or stop playing the audio file specified in the Settings tab. Optionally, pre- or post-calculations can be used to play one or more audio files during the step, using the AudioPlayer Java class in Java package *com.stottlerhenke.taskguide.media.*

When you select the **Insert / Special Tags** menu item, TaskGuide displays a submenu that lets you select the type of special tag to insert:

- If/Endif
- If/Else/Endif
- Embedded Expression
- Run Procedure
- Dictionary Term
- Java GUI Object

These special tags are described in section *Special Tags* (page 51).

The **Insert / HTML Template** menu item provides submenus that enable you to insert pre-defined snippets of HTML. For example, if you select the *Summary/Detail Table* submenu, TaskGuide inserts an HTML template for a two-column table. The left column contains placeholders for summary points, formatted using the HTML `<strong>` emphasis tag. The right column contains placeholders for detailed information.

The **Insert / Format** menu provides the following menu items:

- *Text Properties* - Set font properties of the selected HTML text,
- *Text Formatting* - Insert text formatting tags before and after the selected text. For example, if you select submenu item *Emphasis*, the `<em>` tag will be inserted before the selected text and `</em>` will be inserted after it.
- *Paragraph Properties* – Set paragraph properties of the selected text.
- *Paragraph Formatting* – Insert paragraph formatting tags before and after the selected text. . For example, if you select submenu item *Centered Block Quote*, `<center><blockquote>` will be inserted before the selected text, and `</blockquote></center>` will be inserted after it.

- *Remove Formatting* – Remove HTML formatting from the selected text.
- *List* - Set the selected HTML text to a numbered or bulleted list.
- *Increase Indent* - Decrease the indentation of the selected text.
- *Decrease Indent* - Decrease the indentation of the selected text.
- *Align* - Left, Right, Center, or Justify the selected text.

The **Table** menu provides the submenu items for inserting a new HTML table and for editing the properties of existing HTML tables.

Tool Bars

The **upper tool bar**, immediately below the main menu, contains toolbar buttons that enable you to copy to clipboard, cut to clipboard, paste from clipboard, undo, redo, show paragraphs, insert image, insert hyperlink, or insert table. When you hover the mouse over a toolbar button, the TaskGuide Editor displays a tool tip that describes the button's function.

The **lower toolbar** that is below the upper tool bar provides pull-down lists and buttons that enable you to set the font family and size, set font properties, set the selected text to bold, italics, or underlined, align text (left, center, right, justify), decrease and increase indentation, convert selected text to a numbered or bulleted list, and set the foreground and highlight color of the selected text.

Context Menu

When you press the right mouse button over the Graphical HTML Editor, TaskGuide displays a context menu that enables you to Cut, Copy, Paste, Paste Without Formatting, Select All, and Remove Formatting from selected HTLM text. Additional menu items enable you to set Text, Paragraph, and Hyperlink properties of the selected HTML text. The context menu provides the same Multiple Choice Item, User Input, Audio Player, and Special Tags menu items that are provided by the *Insert* menu on the main menu. If you click on an input form field, the context menu also contains a Form Field Properties menu item. Select this menu item to display a pop-up dialog that enables you to edit the properties of the input form field. In particular, you can use this dialog to view or change the name of the procedure variable that is linked to the input control.

Adding User Interface Controls to Procedure Steps

This section describes the types of user interface controls that can be inserted into a step's instructions. In addition, the Multiple Choice Item object, described in *Chapter 9 – Multiple Choice Item Extension* (page 66), simplifies the creation of steps that display check boxes and labels to prompt students for selections and provide hints and feedback.

Text Input Fields

A text input field:

- *Prompts* the user to enter a new value by typing its value in the text input field. This value is stored in an Integer, Float, or String variable that is linked to the input field. If the input field is linked to an Integer or Float variable and the user enters a non-integer or non-numeric value, TaskGuide displays an error message and re-prompts the user for a value.
- *Displays* the current value of the associated variable.

The *User Input / Insert Text Field* operation, selected from the HTML Graphical Editor's context menu, simplifies the insertion of text fields into a step's instructions.

To **edit** the text input field's properties after you have created it, right-click over the text input field and select *Form Field Properties…* from the context menu. Edit the *Field name* property to change the variable associated with this text input field. Edit the *Size in characters* property to reset the width. TaskGuide ignores the *Value/label*, *Field type*, and *Style* properties.

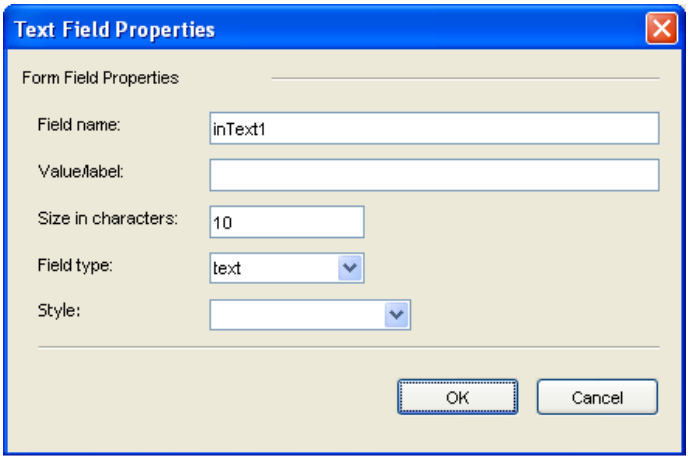


Figure 24 – The Text Field Properties dialog enables you to edit properties of a text form field.

To **delete** the text input field, click your mouse immediately after the text input field and then press the *Backspace* key.

To **cut** or **copy** the text input field, select the input field by press your mouse immediately before the text input field, move your mouse to the position immediately to the right of the text input field, and then release the mouse button. Then, select *Cut* or *Copy* from the *Edit* menu.

Radio Buttons

Radio buttons:

- Display the current value of the string variable associated with the radio buttons by displaying one of the radio buttons as selected, and

- Prompt the user to assign a new value for an associated String variable by clicking on one of the radio buttons.

Each radio button belongs to a set of radio buttons that are associated with the same String variable. When the user clicks on a radio button, TaskGuide sets the value of the associated variable to the value you assigned to the user-selected radio button.

The *User Input / Insert Radio Buttons* operation, selected from the HTML Graphical Editor's context menu, simplifies the insertion of a set of radio buttons into a step's instructions.

To **edit** the radio button's properties after you have created it, right-click over the radio button and select *Form Field Properties…* from the context menu. Edit the *Group name* to set the integer or string variable that will receive the value of the selected radio button. Edit the *Value/label* to change the value that is stored in the variable when this radio button is selected. In general, you should pick a value that is reasonably short and unlikely to change.

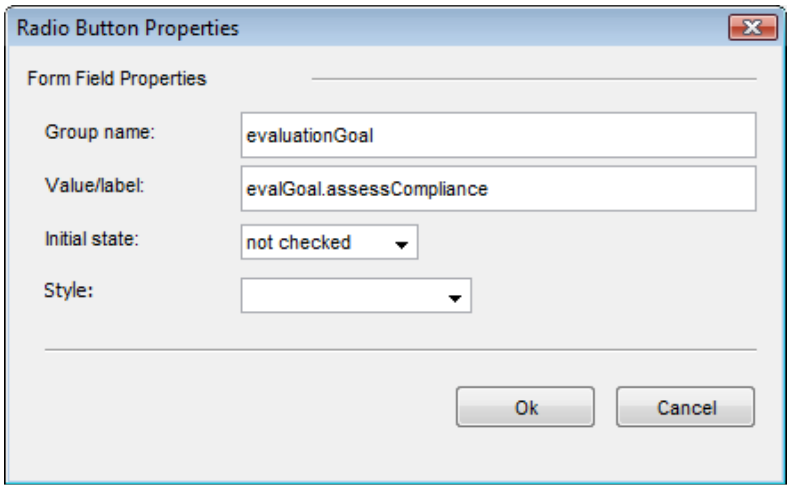


Figure 25 – Radio Button Properties dialog

For example, if the value of the first radio button is "one" and the value of the second radio button is "two", and the Group name of the two radio buttons is "selection", when the user clicks on the first radio button, TaskGuide will set the value of the variable named "selection" to "one".

Check Boxes

Check boxes:

- Display the values of a variable by displaying each check box as either checked or unchecked, and
- Prompt the user to select zero or more options by clicking on the check boxes. The value of each check box is assigned to an associated variable.

TaskGuide provides two methods of setting TaskGuide variable values based on the operator's selection of check boxes. The first method uses one Boolean (true/false)

variable for each check box. Each variable is set to true or false depending upon whether the operator checked the check box associated with the variable.

The second method uses a single List variable to store one String value for each check box checked by the operator. For example, if the user selects one check box with value *apple* and a second check box with value *orange*, the associated List variable will contain two string values: *apple* and *orange*.

TaskGuide provides two methods for entering and configuring check boxes. You can use the Multiple Choice Item extension, described in *Chapter 9 – Multiple Choice Item Extension.* This method enables you to create a data object that stores the question, option labels and values, and optional hints and feedback. After you create this data object, you can insert steps or HTML within existing steps to present the question, prompt the user for a response, store the response in a List variable, assess the student's response, and provide hints and feedback. This method is more convenient if you need to assess the correctness of the student's selection or provide hints or feedback.

The second method enables you to enter check boxes in step instructions without using the Multiple Choice Item extension. To insert a set of check boxes into a step's instructions that are associated with a single List variable, select the *User Input / Insert Check Boxes* item from the HTML Graphical Editor's top-level menu or context menu. To insert one or more check boxes that are associated with different Boolean variables, select *Insert / Input Form / Check box* from the HTML Graphical Editor's top-level menu.

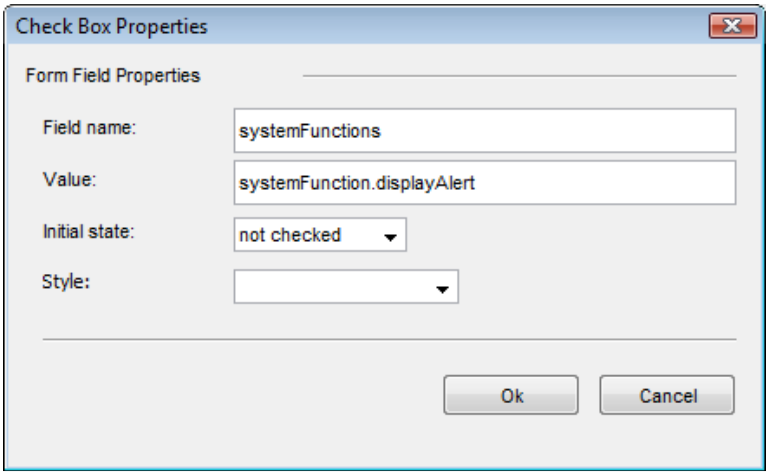


Figure 26 – Check Box Properties dialog

To **edit** the check box's properties after you have created it, right-click over the check box and select *Form Field Properties…* from the context menu. Edit the *Field name* to set the List or Boolean variable that will receive the value of the selected radio button. Edit the *Value/label* to change the value that is stored in the variable when this check box is selected. If you are using a different Boolean variable for each check box, the value should be *true*. If you are using a single List variable for a set of check boxes, the value for each check box should be different so that when that value is added to the List variable, it identifies which check box was checked. In general, you should pick a value that is reasonably short and unlikely to change.

HTML Text Editor (Advanced Feature)

HTML Text Editor tab enables you display and edit the raw HTML that is displayed within each step's instructions. Normally, you would use the HTML Graphical editor to edit each step's formatted instructions. However, the HTML Text Editor is available for use in unusual situations when it is helpful to review, and possibly edit, the raw HTML.

Values of attributes of HTML tags should be enclosed within double quotes. If an attribute value is a JavaScript expression which contains a string constant, it is simplest to enclose the string constant within single quotes.[1] For example, in the embedded expression tag below, the values of the *show_doc, expr, is_constant,* and *decimal_format* attributes are enclosed in double quotes. The value of the *expr* attribute is a JavaScript expression which contains a string constant, `'The answer is '`, enclosed in single quotes, concatenated with the value of a variable named *answer*.

```
<eval show_doc="false" expr="'The answer is ' + answer" doc=" "
is_constant="true" decimal_format=" " />
```

Push Buttons (Advanced Feature)

Using the HTML Text Editor, you can insert an HTML button that evaluates a JavaScript expression when the user pushes it. Like other expressions in TaskGuide procedures, a push button expression can refer to TaskGuide procedure variables and call Java methods. In addition, push button expressions can refer to a special variable called *_tgForm* to access the values of form fields *before* the user has pressed the *NextStep* button.[2] For example, the following HTML specifies a button that, when pressed, calls a JavaScript function named *runOnClick*. This function accepts the value of the form field associated with variable *var1*.

```
<input type="button" name="button1" value="button1"
        onclick="runOnClick( _tgForm.get('var1') )" />
```

The value of the *type* attribute must equal "button". TaskGuide ignores the *name* and *value* attributes of this HTML tag.

Special Tags

This section describes special tags supported by TaskGuide to support dynamic instructions and other features.

[1] If the value of an HTML tag attribute is a JavaScript expression that contains a double quote character, that character should be encoded in the raw HTML as *&quot*;

[2] Normally, when the user presses the *Next Step* button, TaskGuide stores the values of form fields such as text fields or check boxes in TaskGuide variables that are bound to those fields. The step's post-calculations can then refer to these variables. The *_tgForm* object enables push button expressions to access the values of form fields *before* the user presses the Next Step button.

- **Conditional inclusion tags** - specify HTML text that are included in the Instructions if the value of the tag's boolean test condition is true or false (*If/Endif Tags, If/Else/Endif Tags*, page 52).
- **Embedded expression tags** - specify expressions that are evaluated after the step's pre-calculations are executed. When the procedure step is displayed, each embedded expression tag in the step's instructions is replaced by the value of the embedded expression (*Embedded Expression Tag*, page 53).
- **Run procedure tags** – specify special kinds of hyperlinks that, when clicked, instructs TaskGuide to run another procedure in a separate window (*Run Procedure*, page 54).
- **Dictionary term tags** – associate a definition with a word or phrase. The definition is displayed in a rollover window when the user's mouse hovers over the word or phrase (*Dictionary Term*, page 55).
- **Java GUI object tags** - specify variables that store Java GUI objects. When the procedure step is displayed, each embedded GUI tag in the Instructions is replaced by the tag's Java GUI object (*Java GUI Object Tag (Advanced Feature),* page 57).
- **JavaScript expression tags** – associate a JavaScript expression with a hyperlink. When the user clicks on the hyperlink, TaskGuide evaluates the expression.

If/Endif Tags

If/Endif tags enable you to conditionally include blocks of HTML text within a step's instructions. The `<if>` tag contains a Boolean test condition that usually contains the names of one or more procedure variables. After TaskGuide executes the step's pre-calculations, it evaluates the *if* tag's test condition. If the condition is true, the HTML text between the `<if>` and `<endif>` tags is included when the step's instructions are displayed to the user. This text can contain HTML tags and special tags. If the test condition is false, the HTML text is excluded.

To embed the if/endif conditional inclusion tags within Instructions and Verifications, right-click over the HTML Graphical Editor tabbed window and select *Special Tags/ If/Endif* from the context menu. TaskGuide will insert an *if* tag and and *endif* tag. Double-click on the gray placeholder of the <if> tag to edit it. TaskGuide will display the following pop-up dialog:

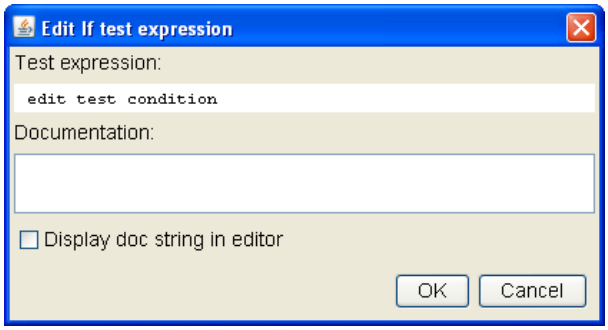


Figure 27 – The Edit If Tag dialog prompts you for attributes of an <if> tag

Type a test expression that evaluates to a Boolean value. Optionally, type in a short

documentation string that describes the expression. If you select *Display doc string in editor*, TaskGuide will display the first few words of this documentation string in the tag's placeholder.

If/Else/Endif Tags

You can also insert an `<else>` tag between the `<if>` and `<endif>` tags. If the *if* tag's test condition is true, the HTML text between the `<if>` and `<else>` tags is included when the step's instructions are displayed to the user. If the test condition is false, the HTML text between the `<else>` and `<endif>` tags is included.

To insert if/else/endif conditional inclusion tags into a step's Instructions or Verifications left-click the mouse in the HTML Graphical Editor to set the current cursor position. Then, right-click and select *Special Tags /If/Else/Endif* from the context menu. TaskGuide will insert an `<if>` tag, an `<else>` tag, and an `<endif>` tag. Click on the gray placeholder of the `<if>` tag to edit it. TaskGuide will display the *Edit If Test Expression* dialog, as shown in Figure 27.

Embedded Expression Tag

The embedded expression tag specifies a JavaScript expression, described in *Appendix A – TaskGuide Expression Syntax* (page 74). After TaskGuide executes the step's pre-calculations, it evaluates the JavaScript expression specified by this tag. It then replaces the tag in the step's instructions with the value of this expression. This feature enables you to dynamically generate some or all of a step's instructions, based on the values of variables whose values are computed when the procedure is executed.

To embed Embedded Expression tags within Instructions or Verifications at the current cursor position, right-click over the HTML Graphical Editor tabbed window and select *Special Tags / Embedded Expression* from the context menu. TaskGuide inserts a gray placeholder with a label that starts with `"eval:",` followed by the expression. This gray placeholder represents the embedded expression tag. Double-click on this gray box to edit the tag. TaskGuide displays the following dialog:

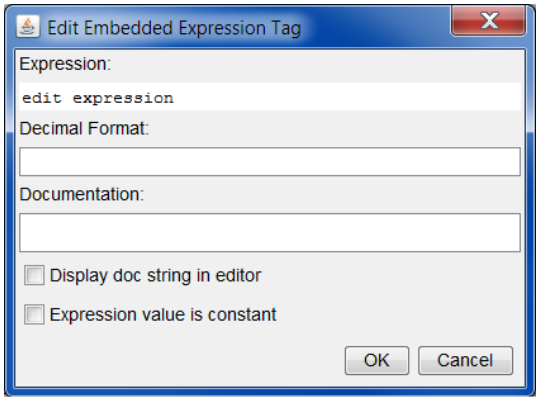


Figure 28 – The Edit Embedded Expression Tag dialog SSS (Decimal Format)

Type an expression that evaluates to a text string value. This expression might be as

simple as a variable name, or it might contain operators and/or calls to Java methods.

If the value of the expression is numeric, you can specify a format to control how the expression value is displayed. For example, a format of `"#.00"` forces the display of 2 decimal digits. `"#.00E0"` displays the number in exponential format with two digits to the right of the decimal point. This feature relies on the Java DecimalFormat class.

You can also type in a short documentation string that describes the expression. If you check *Display doc string in editor*, TaskGuide will display the first few words of this documentation string in the tag's placeholder.

After the TaskGuide Procedure Execution Tool or Applet computes each step's pre-calculations, it evaluates each expression embedded within the step's instructions and verifications and replaces it with its value. For example, if the expression is `temperature + 10`, and the value of variable *temperature* is `200`, the embedded expression tag will be replaced by the text string `"210".`

If the value of the expression is known when the procedure is authored, and the value does not change during execution, you can check *Expression value is constant?* If you do so:

- If you check *Evaluate GUI objects and constant embedded expressions* when previewing this step, the TaskGuide Procedure Editor will display the value of the expression rather than a gray labeled rectangle that serves as a placeholder for the expression, and
- When the user browses this step using the Procedure Execution Tool, the Execution Tool will display the expression's value rather than its placeholder in the Details Pane.

You can use pre- and post-calculations to compute HTML text strings that contains dynamically-generated instructions, assign this HTML text to a variable, and then include this variable within an embedded expression tag. TaskGuide provides a Java class named *HTMLGenerator* which provides methods that generate HTML text dynamically (at procedure execution time) from data stored TaskGuide lists and tables.

> **Tip:** To trace the logic of your procedure, you may find it useful to create steps that contain instructions with embedded expressions to display variable values. By doing so, you can check that the variables are being set to correct values.

Run Procedure

The Run Procedure tag specifies a TaskGuide procedure to run if the user clicks on a word or phrase that serves as a label for the procedure in the step's instructions. The tag specifies:

- the file name or URL of the procedure to run and
- the label that is displayed in the step's instructions.

To insert a *Run Procedure* tags into a step's instructions or verifications, left-click the mouse in the HTML Graphical Editor to set the current cursor position. Then, right-click

and select *Special Tags / Run Procedure* from the context menu. Double-click on the gray placeholder of the tag to edit it. TaskGuide will display the *Edit Run Procedure Expression*, as shown below.

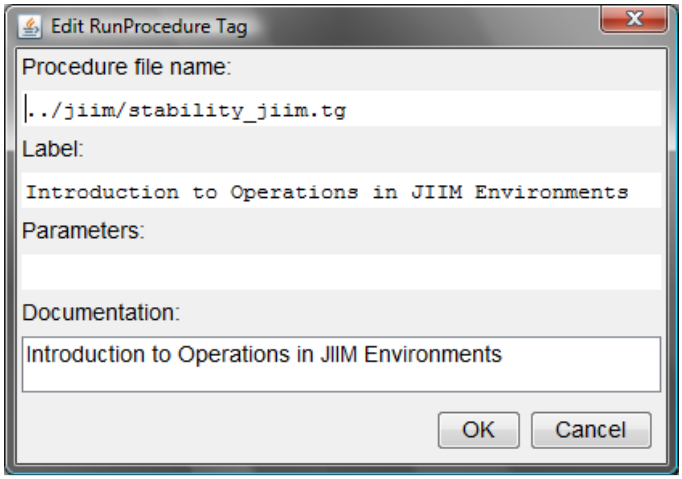


Figure 29 – The Edit Run Procedure Tag dialog

The *Procedure File Name* usually specifies the name of a file, relative to the procedure being edited, that contains the *called procedure* (i.e., the procedure that is run when the user clicks on a procedure label in the step instructions.) For example, if the currently edited procedure file is in a subfolder named *procedure1* in the package folder, and the procedure file being called is named *proc2.tg* in a parallel subfolder named *procedure2* in the package folder, the *Procedure File Name* field should be set to:

../procedure2/proc2.tg

You can also specify the full web address of a procedure file that is accessed from a web server over the Internet. The web address should begin with "`http://`".

The *Label* field specifies the text that is displayed in the step's instructions that the user can click on to run the procedure. By default, the label is underlined and displayed in bold face in dark blue. You can control how the label is displayed by specifying the *.tgRunProcedure* CSS style.

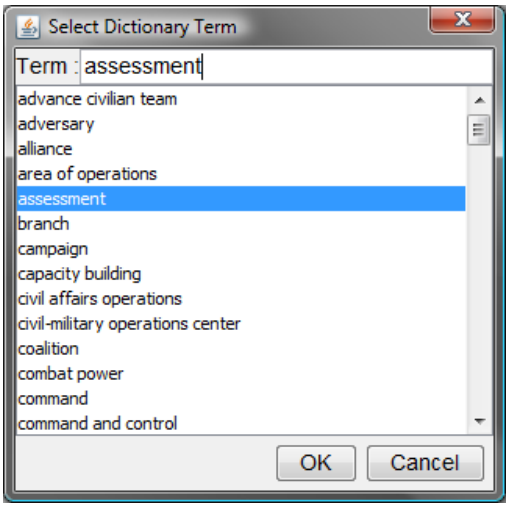
The *Parameters* field contains an optional comma-separated list of expressions that are evaluated and passed as parameters to the procedure being run. This comma-separated list will be converted to a TaskGuide list and assigned to a pre-defined variable named `_parameters` in the procedure that is run.

Dictionary Term

TaskGuide enables you to associate a definition with a word or phrase in a step's instructions. When the user hovers their mouse over this text, TaskGuide displays the associated definition in a small window. By default, TaskGuide displays the text in blue italics to indicate that a definition is available. You can control how the text is displayed by specifying the *.tgTerm* CSS style. Also, when running a TaskGuide procedure, users

can press control-G or select *Show Dictionary* from the context menu to display all of the definitions in the dictionaries used by the procedure.

Figure 30 – TaskGuide displays a definition associated with text in a step's instructions



To associate a definition with text in the step's instructions (or verifications), click and drag your mouse to select and highlight the text. Then, right-click and select *Special Tags / Dictionary Term* from the context menu. The TaskGuide Editor will then display a pop-up dialog that enables you to select the dictionary term to associate with the selected text. When you press OK, the TaskGuide Editor will insert special hyperlink tags that surround the selected text and associate it with the dictionary term. For example, if you select "IMS" in the step's instructions and associate it with the dictionary term "Interagency Management System", the Editor will insert a special tag into the step's instructions that associates the word or phrase (in this case, "IMS") with the dictionary term, as shown below:

```
<a href="term:Interagency Management System"
class="tgTerm">IMS</a>
```

Creating and editing dictionaries is described in *Tools Menu,* page 63.

Java GUI Object Tag (Advanced Feature)

TaskGuide enables you to embed a tag that specifies the name of a TaskGuide procedure variable that stores a Java graphical user interface (GUI) object. When TaskGuide displays a step's instructions, it replaces the embedded GUI object tag with the Java GUI object.

You can embed Java graphical user interface objects within a step's instructions at the current cursor position, by:

- Using pre- and post-calculations to instantiate and configure the Java graphical user interface (GUI) object, and
- Embedding an embedded GUI object tag within the step's instructions.

To embed this tag within Instructions and Verifications, right-click over the HTML Graphical Editor tabbed window and select *Special Tags / Java GUI object.* TaskGuide displays a pop-up dialog that prompts you for the details of the tag:

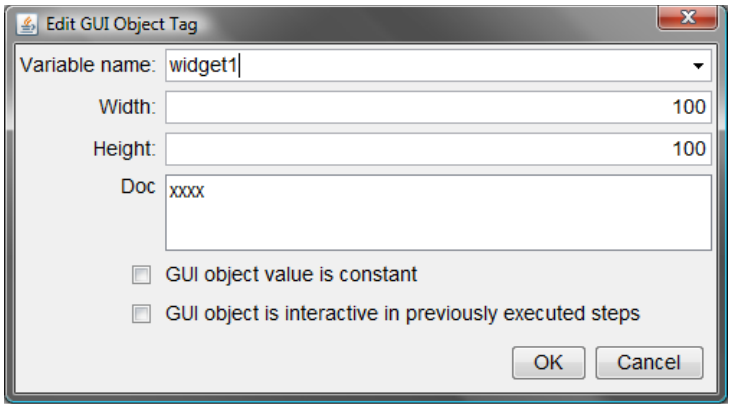


Figure 31 – Edit GUI Object Tag

Using this dialog, select the variable that stores the Java GUI object and specify the GUI object's width and height in the step instructions, in pixels.

If you specify an optional **documentation** ("Doc") text string, that string will be displayed in the tag's placeholder and tooltip when the tag is displayed in the *Instructions* tabbed window.

If:

- the value of the GUI object is initialized in the declaration of the variable that stores the GUI object, and
- this value does not change when the procedure is executed,

you can check *GUI Object Value is Constant*. If you do so, when the user browses the step while running the procedure by clicking on the step's icon in the Summary Pane, the Procedure Execution Tool will display a non-interactive version of the GUI object in the Details Pane. Otherwise, if unchecked, the Details Pane will show a gray placeholder for the GUI object.

The property named *GUI Object is Interactive in Previously Executed Steps* controls whether an interactive or non-interactive version of the GUI object is displayed when the Procedure Execution Tool user clicks on a previously-executed step in the Execution Log Pane. In most situations, you would leave this property unchecked if the GUI object provides user interactions that enable the user to execute commands.

JavaScript Expression Tag (Advanced Feature)

A JavaScript expression tag associates a JavaScript expression with a hyperlink. When the user clicks on the hyperlink, TaskGuide evaluates the JavaScript expression.

The JavaScript expression tag is an HTML hyperlink (`<a>`) tag with an `href` attribute value that begins with "`javascript:`". To create a JavaScript expression tag, select text in the step's instructions to be hyperlinked. Then, select *Special Tags / JavaScript Expression* from the context menu. The TaskGuide Editor will insert `<a href="javascript:" class="tgJavascript">` before the selected text, and it will

insert `</a>` after the selected text. Select the *Hyperlink Properties* context menu item and type a JavaScript expression after "`javascript:`" in the URL property.

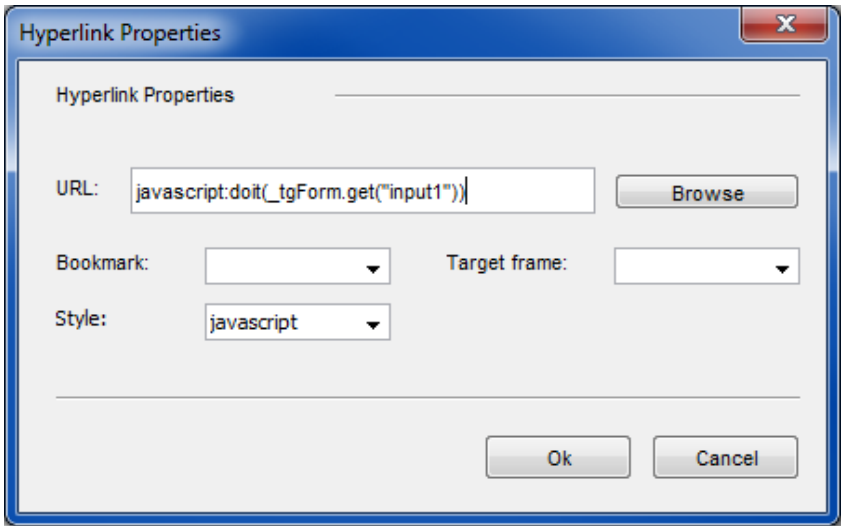


Figure 32 – When specifying a JavaScript tag, enter the JavaScript expression after "javascript:" in the URL property.

This expression can refer to the TaskGuide variable named `_tgForm` to query and set the values of HTML form fields in the step's instructions. For example, in Figure 32, the URL attribute specifies that when the user clicks on the hyperlink, TaskGuide calls a JavaScript function named `doit`. TaskGuide passes the current value of the HTML form field linked to variable `input1` as the argument to doit.

```
<a href="javascript:doit(_tgForm.get("input1"))" class="tgJavascript">
```

You can define the CSS style named `tgJavascript` to control how these hyperlinks are displayed. Ignore attributes *Bookmark* and *Target frame.*

Extensions

The TaskGuide Editor can be augmented with *extensions* that provide specialized authoring capabilities. For example, a Multiple Choice Item extension simplifies the authoring of multiple choice questions with automated assessment and response-specific hinting and feedback.

Each extension adds menu choices to some or all of the following parts of the TaskGuide Editor user interface:

- The *Tools* menu in the top-level menu bar,
- The context menus of the *Procedure* tab and the *Variables* tab in the Catalog Pane, and
- The context menus of the *Instructions, Pre-calculations,* and *Post-calculations* tabbed windows in the Details Pane.

Please refer to the documentation for each extension for detailed descriptions of the extension's capabilities and menu items. *Chapter 9 – Multiple Choice Item Extension* describes how to operate the extension that simplifies the authoring of multiple choice items. The *TaskGuide Programmers' Guide* describes the Java application programming interface that enables software developers to create extensions.

Chapter 8 – Procedure Editor Menu Reference

File Menu

New (Ctrl+N) – TaskGuide creates and displays a new, empty procedure.

Open (Ctrl+O) - TaskGuide displays a file selection dialog that prompts you to select a file containing a procedure. It then opens the selected file and displays the procedure. TaskGuide provides a multiple document interface, so you can open more than one procedure at a time. This feature is useful for copying information from one procedure and pasting it into another.

Save (Ctrl+S) - TaskGuide saves the currently selected procedure to a file.

Save As… - TaskGuide displays a file selection dialog that prompts you to select the name of the file in which to save the procedure. TaskGuide procedure file name end with a *.tg* extension.

Exit - For each procedure that has been edited since it was created or opened, TaskGuide asks if you would like to save the file. TaskGuide saves these files, as specified, and then exits.

Edit Menu

Undo (Ctrl+Z) – Undo the most recent editing operation.

Redo (Ctrl+Y) – Redo the most recent editing operation that was undone.

Move - When this menu choice is selected, the Procedure Editor displays a cascading menu containing the following choices that enable you to move the selected procedure node within its group:

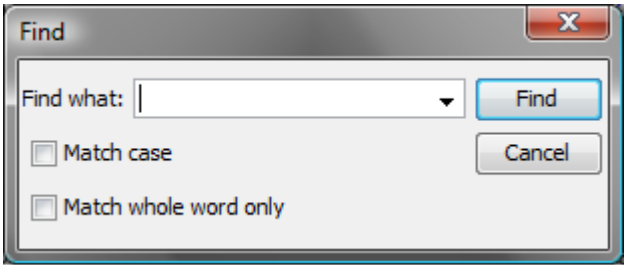
Up	Ctrl-U
Down	Ctrl-D
Top	Ctrl-T
Bottom	Ctrl-B

Group - Replaces one or more selected procedure nodes with a new group node that contains the selected nodes.

Ungroup - Replaces a group node containing zero or more children nodes with the children node.

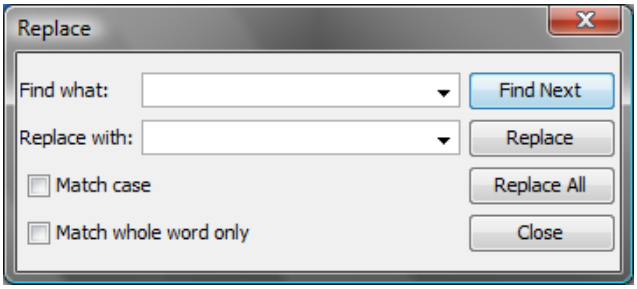
Find (Ctrl+F) - TaskGuide prompts you for a text string and searches for all occurances

of this string in the currently-edited procedure. Each occurrence of the string is displayed in a row in the Search tabbed window in the Results Pane. Double-click on the row to navigate to the location in the procedure that contained the string.



Find in All Files - TaskGuide prompts the operator for a text string and searches for all occurrences of this string in all open procedures.

Replace - TaskGuide displays a dialog that enables you to replace occurances of one string with another string either one at a time or all at once.



Insert Menu

Step Node (Ctrl+Shift+S) - Create a new, empty step node and inserts it into the procedure hierarchy.

Exit Step Node (Ctrl+Shift+X) - Create a new, empty exit step node and inserts it into the procedure hierarchy

Conditional Branches Node (Ctrl+Shift+B) - Create a new, empty conditional branches step node and inserts it into the procedure hierarchy.

Group Node (Ctrl+Shift+G) - Create a new, empty group node and inserts it into the procedure hierarchy.

The currently selected procedure node determines where new nodes are inserted. If the currently selected node is a group node, the new node is inserted at the end of this selected group node. If the currently selected node is a step or exit step node, the new node is inserted as the last node in the group node that contains the selected node.

Variable (Ctrl+Shift+V) – Displays a pop-up dialog that enables you to declare a new variable. This dialog is described in the *New Variable* operation in section *Variables Tab* that starts on page 20.

Tools Menu

Dictionary Editor – TaskGuide dictionary files contain definitions that you can associate with words or phrases in step instructions. Select *Dictionary Editor* from the *Tools* menu to display the Dictionary Editor. The Dictionary Editor enables you to add and delete dictionary files that can be used by the procedure, and add, edit, and delete the terms defined in each dictionary. A dictionary file can be shared by more than one procedure.

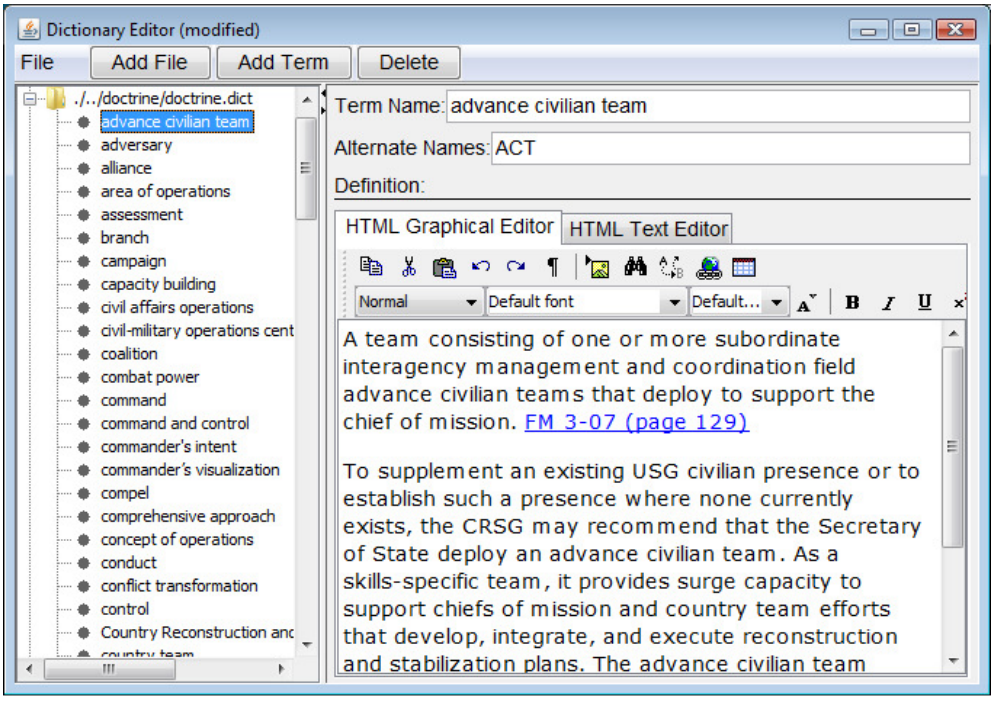


Figure 33 – The Dictionary Editor

Figure 33 shows the Dictionary Editor. The overview pane displays a yellow folder icon for each dictionary file that the procedure can access. It displays a bullet item for each term in the dictionary. The Dictionary Editor provide the following operations:

- To **add a dictionary file** to this procedure, press the *Add File* button. The TaskGuide Authoring Tool will display a file selection dialog that enables you to select a dictionary file. Dictionary files should be stored in the package folder or subfolder.
- To **add a term** to a dictionary, click on an icon in the overview pane that represents a dictionary file to select it. Then, press the *Add Term* button.
- To **delete a dictionary file** from the procedure, click on the icon representing the dictionary file to select it, Then, press the *Delete* button.
- To **delete a term** from the dictionary, click on the icon representing the term to select it. Then, press the *Delete* button.

You can also invoke these operations from the context menu. To display the context menu, right-click over the overview pane.

In the right pane, enter an identifier for the dictionary term in the field labeled Term Name. Use the HTML Graphical Editor to edit a definition of this term. You can include

formatted text, images, and hyperlinks that link to files that are contained in the procedure package folder or are stored on web servers, accessed over the Internet.

View All Procedure Steps – Select this menu item to display all of the steps in the procedure in a pop-up window. This operation is useful for reviewing your procedure to check for errors. The pop-up window also displays a *Save* button which you can press to save the display as an HTML file in the folder that stores the procedure file. You can use a web browser such as Internet Explorer or Firefox to view or print this HTML file.

> **Tip:** You might find it useful to use a PDF print driver to generate a listing of the procedure steps in PDF format for distribution and review. There are several free PDF print drivers.

New Multiple Choice Item – Select this menu item to add a new variable that stores a Multiple Choice Item object. This operation is described in *Creating a Multiple Choice Item Object* in *Chapter 9 – Multiple Choice Item Extension* on page 66

Apply *Multiple Choice Item name* – Select this menu item to apply the Multiple Choice Item that was most recently selected in the *Variables* tabbed window. The name of this multiple choice item will be displayed in the menu item label. This menu item displays submenu items that enable you to Pose a Question, Provide Hints, and Provide Feedback. This operation is described in Applying a Multiple Choice Item Object in *Chapter 9 – Multiple Choice Item Extension* on page 70.

User Input – Select this menu item to display a submenu that enables you to:

- Insert Step with Text Input Field
- Insert Text Input Field in Selected Step
- Insert Pre-calculation to Clear Input Field
- Insert Post-calculation to Check that Input Variable is Not Empty

Check Menu

Check for Syntax Errors - TaskGuide checks for certain types of syntax errors in all expressions in all nodes in the procedure. Syntax errors are displayed the *Problems* tabbed window in the Procedure Editor's Results Pane.

Check URLs – Select this menu item to check that the files that are referenced via relative file name or full URL in your procedure can be accessed. This feature is useful for checking that you have specified the correct file names or URLs in your procedure. Files that cannot be accessed are listed in the Procedure Editor's Results Pane.

Package Menu

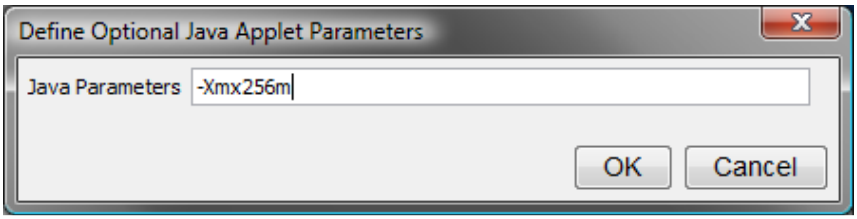
This menu provides operations that generate files used to deploy a procedure package. These file generate utilities assume that each TaskGuide procedure file is stored in its own subfolder within the package folder.

Generate Software Files - TaskGuide generates the software files needed to run

procedures. It places these files in a subfolder named *software* in the procedure package folder. If your procedures use additional, custom Java libraries, you should copy the library (.jar) files to the *software* folder.

Generate Command File – Generates a Windows command file that launches the currently edited procedure. If your procedures use additional, custom Java libraries, you should edit the command file to include these library files.

Generate Applet HTML File – Generates an HTML file that contains an applet tag that invokes the non-SCORM TaskGuide Execution Applet to run the currently edited procedure. The TaskGuide Editor will prompt you for optional Java parameters to include in the applet tag, as shown below:



If your procedures use additional, custom Java libraries, you should edit the applet tag in the HTML file to include these library files.

Generate SCORM Files – Prompts the author to specify all TaskGuide procedure files in the procedure package. Generates a SCORM manifest file, SCORM XML schema (.xsd) files, and an HTML file for each each procedure that implements a SCORM sharable content object (SCO). Each HTML file contains an applet tag that invokes the SCORM TaskGuide Execution Applet. If your procedures use additional, custom Java libraries, you should edit the applet tag in the HTML file to include these library files.

TaskGuide uses a SCORM Procedure Execution Applet when running procedures as SCORM SCOs. This version of the TaskGuide Procedure Execution Applet exchanges data with a SCORM learning management system.

Window Menu

When editing more than one procedure at a time, each in their own window, select the Window Menu to control which procedure window is displayed on top and how the procedure windows are arranged.

Tile - Display all procedure windows so that they abut one another.

Cascade - Display the procedure windows as a stack, each window slightly displaced from the next window.

Help Menu

About - TaskGuide displays its version number, copyright notice, and Stottler Henke's web address, www.stottlerhenke.com.

Chapter 9 – Multiple Choice Item Extension

Overview

The Multiple Choice Item extension helps TaskGuide authors create procedures that pose multiple choice questions and compare the student's selections to a reference answer to display response-specific hints and feedback.

This extension provides a set of user-selectable commands and a Java class named *MultipleChoiceItem*. Each Multiple Choice Item object stores the following information:

- the question,
- the answer options that the student can select,
- the reference answer that specifies which answer options are correct or preferred, and
- response-specific hints and feedback that can be presented to the student.

Once a Multiple Choice Item object is created and configured to store this information, authors can select menu items in the Procedure Editor to create steps or reconfigure existing steps to pose the question, prompt the student for an answer, compare the student's answer with the reference answer, and provide hints and feedback. The author can also specify the minimum and maximum number of options that the student's answer should include.

Because questions, hints, and feedback are specified using HTML, they can contain text formatting, images, and hyperlinks, just like step instructions.

Creating a Multiple Choice Item Object

There are two ways to create a Multiple Choice Item object:

- Select *New Multiple Choice Item* from the *Tools* menu on the top-level menu bar, or
- Click on the *Variables* tab in the Catalog pane, right-click to display the context menu, and select *New Multiple Choice Item.*

The TaskGuide Procedure Editor will display a dialog that prompts you for the name of a new procedure variable that will store the Multiple Choice Item object, as shown below:

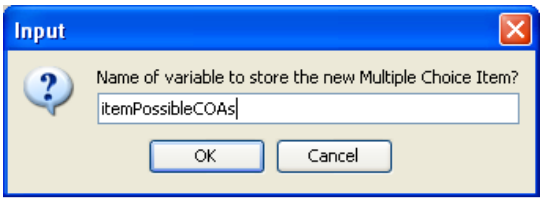


Figure 34. Dialog to Name a new Multiple Choice Item

Enter a unique variable name that not already used by an existing variable and press the *OK* button. TaskGuide will display another window that enables you to configure your Multiple Choice Item, as described in *Editing a Multiple Choice Item Object* on page 67.

> **Tip:** You may find it helpful to use a naming convention for your variables. For example, you could start the name of every Multiple Choice Item variable with "item". Naming conventions help you see more quickly what kind of information is stored in each variable.

Editing a Multiple Choice Item Object

The Multiple Choice Item Editor window enables you to display and edit the contents of a Multiple Choice Item object.

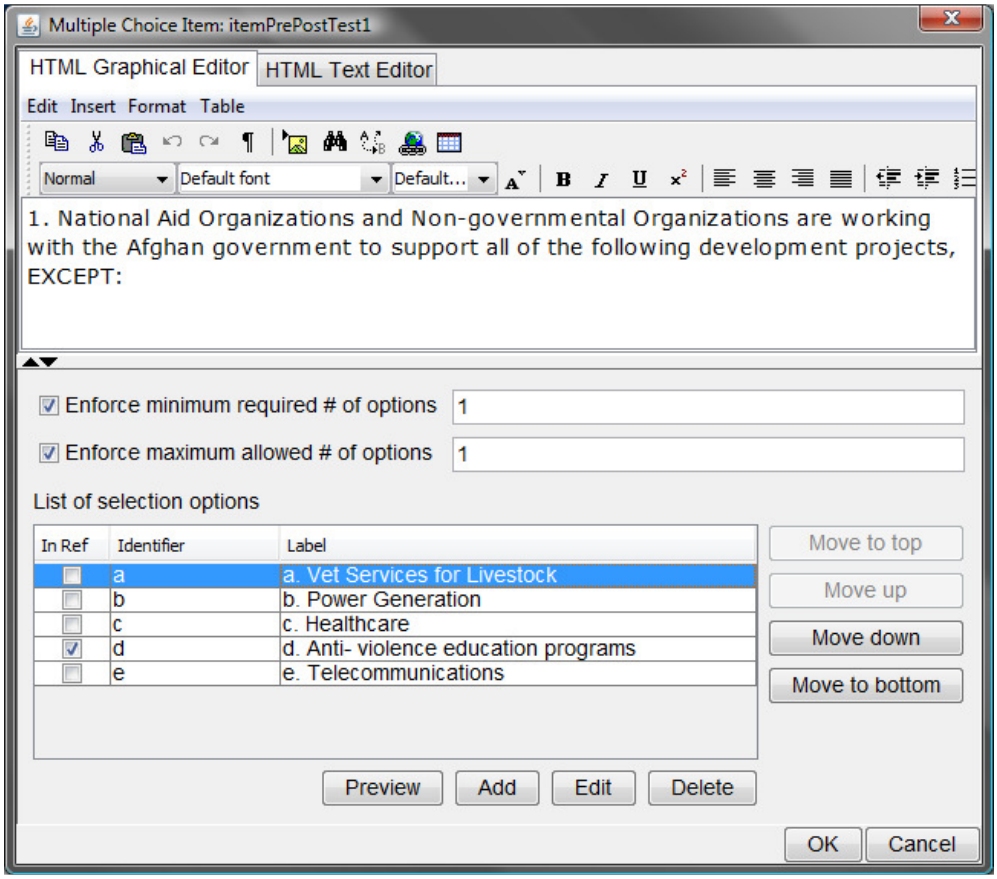


Figure 35. The *Multiple Choice Item* Editor dialog.

TaskGuide displays this window when you:

- Create a new Multiple Choice Item object, as described in *Creating a Multiple Choice Item Object* on page 66, or
- Edit a variable that stores a Multiple Choice Item object. When you double-click on a Multiple Choice Item variable in the *Variables* tabbed window, TaskGuide displays

the *Edit Variable* dialog. Then, press the *Edit Instance* button in the *Edit Variable* dialog.

Enter the item's prompt at the top of this dialog. Then, check the check boxes to specify whether TaskGuide should enforce that the user select a minimum and/or maximum number of options. If so, enter the minimum and/or maximum number in the text fields. In Figure 35, the Multiple Choice Item is configured to require that users select at least 1 option. If the student does not select the required number of options during procedure execution, TaskGuide re-prompts the student to select an acceptable number of options.

To **add a new option**, press the *Add* button. TaskGuide will add a row to the list of selection options. If an existing option is selected when the *Add* button is pressed, the new option is added after the selected option. Otherwise, the new option is added at the end of the list of options. Double-click on a table cell and then type the identifier or label. If you use TaskGuide's tracking feature to save this value or report it to a SCORM learning management system, you should pick an identifier that is reasonably short and unlikely to change. Check the checkbox in the left-most column to indicate that the option is one of the reference or preferred options.

To **edit an existing option**, click on the option in the options list to select it. Then, press the *Edit* button. TaskGuide will display an *Option Editor* window that enables you to specify hints and feedback for the option.

To **delete an option**, select the option in the options list and press the *Delete* button.

To **preview the multiple choice item**, press the *Preview* button. The TaskGuide Editor will display a pop-up window that displays all of the item's configuration details.

During procedure execution, TaskGuide displays options to students in the order they are specified in the options list. To **reorder options**, click on an option to select it and press one of the *Move…* buttons to move the selected option up or down or to the top or bottom of the list.

Editing an Option's Hints and Feedback

The Multiple Choice Item *Option Editor* window, shown in Figure 36 enables you to specify hints and response-specific feedback associated with an option. If your item does not provide hints or response-specific feedback, you do not need to use this window. TaskGuide displays this window when you select an option and press the *Edit* button.

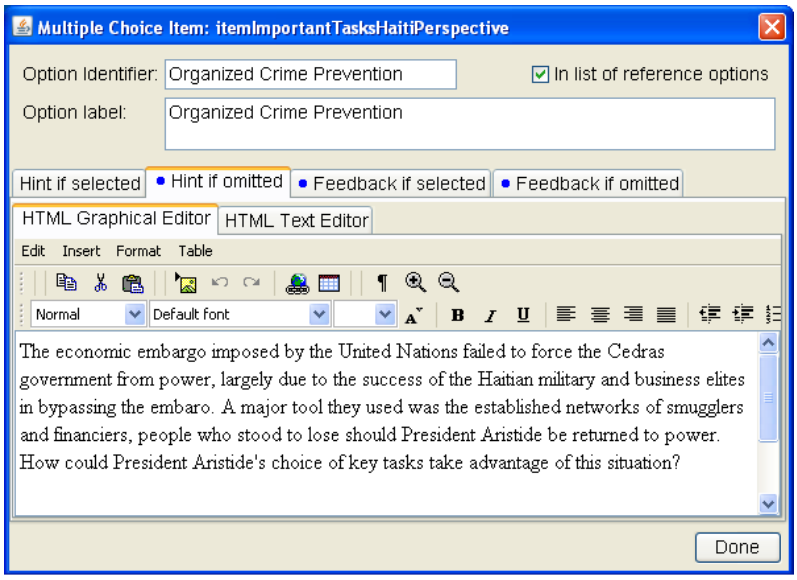


Figure 36. Sample Option, with "Hint if omitted" text.

The top-most text box labeled *Option Identifier* prompts you for a word or phrase that is used by the TaskGuide software to identify this option. During procedure execution, TaskGuide stores a list of the identifiers of all selected options in a List. An object method called *selectedOptions* returns this List so that it can be used in procedure calculations, saved in a tracking file, or reported to a SCORM learning management system. This field is required.

The text box labeled *Option label* prompts you for an optional label that is displayed to the user or student next to the option's check box during execution. Typically, the label is a phrase or a sentence. If no label is entered, the option's identifier is used as the label.

Check the check box labeled *In list of reference options* if this option is one of the reference options.

The four tabbed windows prompt you for optional response-specific hints and feedback:

Hint if selected	This text is included in the hints shown to the student if 1) the option *is not* one of the reference options and 2) the student's selections *include* this option.
Hint if omitted	This text is included in the hints shown to the student if 1) the option *is* one of the reference options and 2) the student's selections *do not include* this option.
Feedback if selected	This text is included in the feedback shown to the student if the student's selections *include* this option.
Feedback if omitted	This text is included in the feedback shown to the student if the student's selections *do not include* this option.

Press the *Done* button to complete editing of the option.

Applying a Multiple Choice Item Object

After you have created and editing a Multiple Choice Item object, you can *select* and *apply* the object to create and modify steps in the procedure to pose questions, prompt for selections, and provide hints and feedback.

To select a Multiple Choice Item object to apply:

1. Click on the *Variables* tab, and
2. Click on the variable that stores the Multiple Choice Item object to apply.

You can apply the selected Multiple Choice Item object by inserting steps, groups, HTML text into a step's instructions, and post-calculations to perform the following functions:

- **Pose the question** by displaying the prompt specified for the Multiple Choice Item.
- **Prompt for selections** by displaying a labeled check box for each option in a step's instructions. After the user completes the step, usually by pressing the *NextStep* button, TaskGuide sets the *selectedOptions* field of the Multiple Choice Item object to a list of identifiers of the selected options.
- **Provide hints** based on whether each option was selected. TaskGuide inserts these hints in the step's instructions and encloses the hints within if/else/endif tags. These tags contain test expressions that check if the user selected the option. If so, the option's *Hint if selected* is conditionally included in the step's instructions. Otherwise, the *Hint if omitted* is conditionally included.
- **Compare selected options with reference options.** A labeled checkbox is displayed for each option. Labels of reference options are displayed in blue, and selected options are checked.
- **Provide feedback** based on whether each option was selected. The *Feedback if selected* and *Feedback if omitted* for each option are enclosed between if/else/endif tags. If the option was selected, the *Feedback if selected* is conditionally included in the step's instructions. Otherwise, the *Feedback if omitted* is conditionally included.

Inserting New Steps and Groups

When a Multiple Choice Item variable is selected, the context menu of the *Procedure Summary* window contains a menu item whose label equals "Multiple Choice Item:" followed by the name of the selected variable. For example, if a author has selected a Multiple Choice Item variable named "itemImportantTasks", the Procedure Summary context menu contains a menu item labeled:

Multiple Choice Item: itemImportantTasks

If you select this menu item, TaskGuide displays the following submenu:

- **Insert Step to Show Question and Select Options** – This operation inserts a new step that contains instructions that pose the question, display the options using labeled check boxes, and prompt for selections. In addition, the new step contains a post-calculation that checks that an acceptable number of options were selected.
- **Insert Step to Provide Hints** - This operation inserts a new step whose instructions contain response-specific hints associated with each option.

- **Insert Step to Show Question, Show Reference Options, and Provide Feedback** - This operation inserts a new step that displays the original question, compares the student's selected options to the reference options, and provides response-specific feedback associated with each option.
- **Insert Group to Conditionally Provide Hints** – This operation inserts a branching group that tests if there are any hints to display, based on the user's selections. If so, a step within the branching group displays these hints.

These inserted steps contain tags which are placeholders for multiple choice item information. When you run the procedure, the tag will be replaced by the appropriate information stored in the multiple choice item. For example, if you insert a tag for the question portion of a multiple choice item, TaskGuide will insert the item's question into the instructions when the procedure runs. Click on the *Preview* tab in the Details Pane to see what the step's instructions will look like when the procedure runs.

If a group or procedure node was selected in the *Procedure* tab, the new step node is inserted as the last node in the group or procedure. If a step node was selected, the new node is inserted after the selected step node.

Inserting HTML into an Existing Step

When a Multiple Choice Item variable has been selected, the context menu of the HTML Graphical Editor window contains a menu item whose label equals "Multiple Choice Item:" followed by the name of the selected variable.

- **Insert Tag to Pose Question.** This operation inserts a tag into the instructions of the selected step to pose the question,
- **Insert Tag to Select Options.** This operation inserts a tag into the instructions of the selected step to display labeled check boxes that list the options and prompt for selections.
- **Insert Tag to Provide Hints.** This operation inserts a tag into the instructions of the selected step to display response-specific hints.
- **Insert Tag to Compare Selected Options to Reference Options.** This operation inserts a tag into the instructions of the selected step to help users compare their selections with the reference options.
- **Insert Tag to Provide Feedback.** This operation inserts a tag into the instructions of the selected step to display response-specific feedback.
- **Insert HTML to Pose Question.** This operation inserts HTML into the instructions of the selected step to pose the question,
- **Insert HTML to Select Options.** This operation inserts HTML into the instructions of the selected step to display labeled check boxes that list the options and prompt for selections.
- **Insert HTML to Provide Hints.** This operation inserts a tag into the instructions of the selected step to display response-specific hints.
- **Insert HTML to Compare Selected Options to Reference Options.** This operation inserts HTML into the instructions of the selected step to help users compare their selections with the reference options.
- **Insert HTML to Provide Feedback.** This operation inserts HTML into the

instructions of the selected step to display response-specific feedback.

The first five menu options insert multiple choice item **tags**. As described in *Inserting New Steps and Groups*, page 61, when you run the procedure, the tag will be replaced by the appropriate information stored in the multiple choice item. Click on the Preview tab in the Details Pane to see what the step's instructions will look like when the procedure runs.

The second set of five menu options insert multiple choice item **HTML**. This approach is useful in those special (and less common) situations when you need to change the formatting of the multiple choice item information. In these situations, you can insert HTML instead of inserting a tag and then reformat the inserted HTML using TaskGuide's HTML editor. However, if you insert HTML and later edit the multiple choice item object, you will need to delete the old HTML and re-insert new HTML to reflect the new contents of the multiple choice item.

Inserting Pre-Calculations

Whenever a Multiple Choice Item variable has been selected, the context menu of the Pre-calculations tabbed window contains a menu item whose label equals "Multiple Choice Item:" followed by the name of the selected variable.

If you select this menu item , TaskGuide displays the following submenu:

- **Initialize Selected Options.** Whenever TaskGuide prompts the user to select options using a Multiple Choice Item, the user's selections are stored in the Item. By default, if TaskGuide re-prompts the user to select options a second time using the same Item, the user's previous selections will be checked in the options' check boxes. You can initialize selected options to uncheck all of the options selected by the user. Then, if the options are offered to the user a second time, the previous selections will be erased.

Using the Tools Menu

In addition to selecting Multiple Choice Item operations from context menus, you can also select these operations from the top-level *Tools* menu. When a Multiple Choice Item variable is selected, the *Tools* menu contains a menu item whose label equals "Apply " followed by the name of the selected variable. For example, if the author has selected a Multiple Choice Item variable named "itemImportantTasks", the Tools menu contains a menu item labeled:

Apply itemImportantTasks

You can select Multiple Choice Item operations from submenus to insert steps, groups, HTML, and post-calculations.

Before you insert a new step or group, specify its location within the procedure by using the *Procedure* tab to select either:

- the group that should contain the new step or group, or

- the step that the new step or group should follow.

Before you insert instructions text or a pre-calculation into an existing step, use the *Procedure* tab to select the existing step.

Appendix A – TaskGuide Expression Syntax

TaskGuide enables you to specify expressions written in the JavaScript programming language that calculate numeric, string, and boolean values. Expressions can be included within the following parts of a TaskGuide procedure:

- Test conditions in Exit Step and Conditional Branches nodes. You can specify these test conditions in the Settings tabbed window when editing these nodes.
- Test conditions within input validations,
- Test conditions and expressions in pre- and post-calculations,
- Embedded expression (`<eval/>`) tags, conditional inclusion (`<if/>`) tags, push button (`<input>`) tags, and JavaScript expression tags (`<a href="javascript: xxx">`) tags within step instructions and verifications.

Within TaskGuide procedures, JavaScript expressions can contain:

- Boolean (true/false) values, text strings (enclosed in single or double quotes), and numbers,
- Names of TaskGuide variables
- Numeric, string, or logical operators such as +, -, or &&.
- Calls to JavaScript and Java object methods and object constructors. When launching TaskGuide, include the necessary Java .jar files in the Java class path.

Data Types

You can declare TaskGuide variables to store the following types of values:

TaskGuide Data Type	Example Values	Java Class Used to Store These Variables
Boolean	`true, false`	java.lang.Boolean
Integer	`3, -4`	java.lang.Integer
Float	`3.5, 0.04`	java.lang.Float
String	`'Quick brown fox'`	java.lang.String
List	One-dimensional array of values.	Java.util.Vector
Table	Two-dimensional array of values.	com.stottlerhenke.TaskGuide.TaskGuideTable
Object	A Java Object.	java.lang.Object

The JavaScript programming language stores all numbers as floating point values, so TaskGuide integers are automatically converted to floating point numbers when they are included within JavaScript expressions. If you call a Java method that expects an argument of type *int* or *java.lang.Integer*, use the *Math.round()* Java method to ensure that numeric values included in JavaScript expressions are converted to integers.

Values of TaskGuide String variables that are included within JavaScript expressions are converted from their original form as java.lang.String objects to JavaScript strings when the expressions are evaluated. Thus, within JavaScript expressions, you can invoke JavaScript string methods, rather than java.lang.String class methods.

Operators

The JavaScript expression language includes the following operators:

	Operator	**Operands (Number, Type)**	**Example**

`==`	equal to	2 numbers, Booleans, strings, or objects	`'4' == '3' → false`
			`4 == 3 → false`
			`4 == 4 → true`

`!=`	not equal to	2 numbers, Booleans, strings, or objects	`'4' != '3' → true`
			`4 != 3 → true`
			`4 != 4 → false`

`+`	addition	2 numbers	`4 + 3 → 7`
`-`	subtraction	2 numbers	`4 - 3 → 1`
`*`	multiplication	2 numbers	`4 * 3 → 12`
`/`	division	2 numbers	`12 / 3 → 4`
`+`	string concatenation	2 text strings	`'4' + '3' → '43'`
`<`	less than	2 numbers	`4 < 3 → false`
`<=`	less than or equal to	2 numbers	`4 < 3 → false`
`>`	greater than	2 integers or floats	`4 > 3 → true`
`>=`	greater than or equal to	2 numbers	`4 >= 3 → true`
`!`	logical negation	1 Boolean	`!false → true` `!(4 != 3) → false`
`&&`	logical AND	2 Booleans	`false && (4 != 3) → false`
`||`	logical OR	2 Booleans	`false || (4 != 3) → true`

`? :`	if-then-else	First argument is a Boolean	`(true ? 1 : 2 ) → 1`
			`(false ? 1 : 2 ) → 2`
			`((1==0) ? 1 : 1+1 ) → 2`

Java Method Invocation

You can call a Java constructor method using the following syntax:

new *ClassName (arglist)*

You can call an object's method using the following syntax:

ObjectVariableName.methodName (arglist)

For example, if a TaskGuide variable named *v1* is bound to a Java java.lang.Vector object, you can add a string value of `'hello'` as a new element to this Vector by using its add method as follows:

```
v1.add ('hello')
```

You can call a static Java method using the following syntax:

ClassName.methodName (arglist)

TaskGuide Java Classes

TaskGuide provides several Java classes that you can use within pre/post-calculations embedded expressions, and test (conditional) expressions. These classes and their methods are documented in Javadoc format in the *javadoc* subfolder of the TaskGuide installation folder.

AudioPlayer

The Settings tab of the Step Node enables you to specify the playing of a single audio file when the step is displayed. The AudioPlayer class provides methods for more advanced audio playback capabilities such as the playing of several audio files in succession. The _audioPlayer pre-defined variable is an instance of this class. Java package: com.stottlerhenke. taskguide.media.

HTMLGenerator

The HTMLGenerator class provides methods that generate HTML text from TaskGuide Tables and other types of variables. The return value of these methods can be included in expressions embedded within TaskGuide step instructions. Java package: com.stottlerhenke. taskguide.utils.

Procedures

This class provides methods for running procedures and managing variable values. Java package: com.stottlerhenke. taskguide.utils.

SessionTrackingMgr

This Java class provides static methods for tracking the student's answers and objective scores. A context menu item named *Tracking/Insert Start Tracking Mode* in the *Procedure* pane simplifies the insertion of a step node that starts tracking (Procedure Tab, page 19). Java package: com.stottlerhenke. taskguide.utils.

TaskGuideTable

TaskGuide Tables are stored in instances of the Java class TaskGuideTable. This class provides methods that help you query and update data stored in these objects. Java package: com.stottlerhenke.taskguide.engine. (***)

VariableBindings

This class defines objects that store variable names and values that can be extracted from a procedure and saved to or loaded from a file. Java package: Java package: com.stottlerhenke. taskguide.utils.

TaskGuide JavaScript Functions

TaskGuide provides one JavaScript function that you can use within pre/post-calculations.

tgMakeList()

This function accepts one or more arguments and returns a TaskGuide list that contains these argument values. The list is an instance of the Java java.util.Vector class.

Appendix B – Adding Formats and Templates to the HTML Graphical Editor

You can add menu items to the HTML Graphical Editor that enable authors to apply pre-defined formats and insert templates more easily. To add these menu items, create configuration files that specify the menu labels and their associated formats and templates. You can add format and template menu items to the following HTML Graphical Editor menus:

Insert / HTML Template – This menu contains one submenu for each pre-defined HTML template that you can insert. Each line of text in file *html_templates.txt* specifies two fields: the name of the submenu and the pre-defined HTML template, separated by a tab character.

Format / Text Formatting – This menu contains one submenu for each method of formatting text that the author has selected in the Graphical HTML Editor by inserting HTML text formatting tags. When a submenu is selected, HTML tags are inserted before and after the selected text. Each line of text in file *html_text_formatting.txt* specifies three fields: the name of the submenu, the HTML start tags to insert before the selected text, and HTML end tags to insert after the selected text, separated by tab characters.

You can specify custom CSS text styles by using the `<span>` tag and the *class* attribute. For example, the Authoring Comment submenu item surrounds the selected text with `<span class="tgComment">` and `</span>`. You can use the procedure-level *Settings* tab to add a CSS style rule that specifies the appearance of the tgComment class or other new classes.

Format / Paragraph Formatting - This menu contains one submenu for each method of formatting text that the author has selected in the Graphical HTML Editor by inserting HTML paragraph formatting tags. When a submenu is selected, HTML tags are inserted before and after the selected text. Each line of text in file *html_paragraph_formatting.txt* specifies three fields: the name of the submenu, the HTML start tags to insert before the selected text, and HTML end tags to insert after the selected text, separated by tab characters.

You can specify custom CSS paragraph styles by using the `<div>` tag and the *class* attribute. You can use the procedure-level *Settings* tab to add a CSS style rule that specifies the appearance of new classes.

Customizing HTML Editor Menus for Specific TaskGuide Packages

This configuration files are stored in the top-level TaskGuide authoring tool folder (e.g., C:\Program Files\Stottler Henke\TaskGuide 2.0). These configuration files customize the three Graphical HTML Editor menus when editing any TaskGuide procedure.

You can configure these menus for a particular TaskGuide package as follows:

1. Select the TaskGuide *Package / Generate Software Files* menu to generate software files for the package. This operation copies the three configuration files to the package's *software* subfolder.
2. Edit any of the three configuration files in the package's *software* subfolder to add, remove, or change submenus. These changes will apply only to that package.