

## Niagara<sup>AX</sup> Lexicon Guide

---

Information and/or specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein are covered by one or more U.S. or foreign patents. This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of those products, subject to the contracts that authorize such distribution. It may not otherwise, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc. Complete confidentiality, trademark, copyright and patent notifications can be found at: <http://www.tridium.com/galleries/SignUp/Confidentiality.pdf>. © 2013 Tridium, Inc.

JACE, Niagara Framework, Niagara AX Framework and the Sedona Framework are trademarks of Tridium, Inc.

This document provides information on NiagaraAX Lexicons in AX-3.7 and AX-3.8, and contains the following main sections:

- “Lexicon overview” on page 2
  - “About nested lexicon values” on page 2
  - “Summary of changes to lexicons in AX-3.7” on page 2
- “Lexicon components” on page 3
  - “Lexicon display options” on page 3
  - “Search Lexicon” on page 4
- “Lexicon plugins (views)” on page 4
  - “Lexicon Report view” on page 4
  - “Lexicon Editor view” on page 5
  - “Lexicon Module Builder view” on page 6
- “Common tasks” on page 8
- “Lexicon Tool Component Summaries” on page 11
- “Lexicon Tool Plugin (view) Summaries” on page 11
- “Document change log” on page 11

## Lexicon overview

Starting in Workbench 3.7, standard lexicons are distributed as *modules* (niagaraLexiconXx.jar) that are included in the Workbench installation. Each lexicon module contains one or more lexicon files (*moduleName.lexicon*). Lexicon files are simple text files that map various entity “keys” (such as interface names or error and status values) to localized language character values. Often, mapped characters use special encoding.

NiagaraAX provides *non-English* language support with the use of *lexicons*. Lexicons are identified by Java *locale codes*, such as “fr” (French) or “de” (German). When installing NiagaraAX on your PC, the full complement of lexicon modules are installed in the `!modules` folder.

**Note:** *In some cases, lexicon editing provides utility even in English language scenarios. For more details, see “About English (en) lexicon usage” on page 8.*

For example, the following 3 lines are a “snippet” from the `alarm.lexicon` file in a French (fr) lexicon set:

```
commands.reset.label=Remettent \u00E0 l'\u00E9tat initial
messages.connectionError=N'a pas pu se relier \u00E0: {0}
portal.wizard.title=Ajoutez La Console D'Alarme
```

**Note:** *The factory supplied language lexicons typically require review and editing before they can be used on an actual job. Text file based customizations of the standard lexicons are saved under a “!lexicon” folder in your Niagara directory.*

To edit lexicons, you typically use the Workbench Lexicon Tool (**Tools > Lexicon Tool**). The “Lexicon Editor” view not only provides edit features, but also shows the default (English) value for each line entry. Any changes you make to a lexicon using the Lexicon Editor are saved as *lexicon files*, under the appropriate `!lexicon` subfolder for the target locale. For example, changes to the French lexicon are saved in the `!lexicon/fr` subfolder.

**Note:** *The !lexicon folder does not exist in a new Workbench 3.7 or 3.8 installation. The folder is created the first time you edit a lexicon and save it.*

The lexicon tool also provides a “Lexicon Report” view, useful to see various details about a lexicon and its contained files. Additionally, the Lexicon Report view lists all lexicons installed on your Workbench PC. A third view added in Workbench 3.7, the “Lexicon Module Builder” view, facilitates collecting multiple lexicons in a module for ease of distribution. For more details, see “[Lexicon Module Builder view](#)” on page 6. Once reviewed and edited, you typically install any needed lexicon into all JACE platforms. For more details, see the “Lexicon Installer” and “Software Manager” sections in the *Platform Guide*.

### About nested lexicon values

AX-3.6 and later versions support “nested lexicon values”. This means the value of a lexicon key can reference another key in either the same module’s lexicon file, or a different module’s lexicon file. Multiple levels of nesting are supported, as needed. This feature may be most used by NiagaraAX developers who are creating new modules that need lexicons.

The syntax for nested lexicon values is as follows:

`{lexicon:moduleName:key}` (if referencing a key in a different lexicon file), or

`{lexicon:key}` (if referencing a key in the same lexicon file)

For a multi-level example within the same (module) lexicon file, consider these `key=value` pairs in the “test” module’s lexicon:

```
fum.text={lexicon:foe.text} Fum
foe.text={lexicon:fie.text} Foe
fie.text={lexicon:fee.text} Fie
fee.txt=Fee
```

In the example above, the value for `fum.text` resolves to “Fee Fie Foe Fum”.

This could be seen in a Px label widget using BFormat scripting syntax: `%lexicon(test.fum.text)%`

### Summary of changes to lexicons in AX-3.7

If you are already familiar with AX-3.6 lexicons, the following summary of changes may be helpful in using the AX-3.7 or AX-3.8 version.

- Standard lexicons are distributed as *modules* (niagaraLexiconXx.jar), where each lexicon module contains one or more lexicon files (*moduleName.lexicon*). The full complement of standard lexicons are included in the Workbench installation. The lexicon modules are located in the `!modules` fold-

- er of your Niagara directory.
- The `!lexicon` folder is the location where lexicon overrides are saved. This is where you can manage your *custom* lexicons.
- The *Lexicon Tool* contains three views: Lexicon Report, Lexicon Editor and Lexicon Module Builder.
- The Lexicon Report and Lexicon Editor views each have the added ability to filter by *Type*. Selecting this filter excludes keys that end with “.displayName” or “.icon”.
- Also, a Search capability has been added to the Lexicon Report and Lexicon Editor views. You can search for specified text on either the lexicon property Key, the property Default, or the custom Value.
- The new view, Lexicon Module Builder, lets you bundle lexicon files into modules for ease of distribution.
- The platform Commissioning Wizard has been changed to support loading lexicon modules or files for selected locales.

## Lexicon components

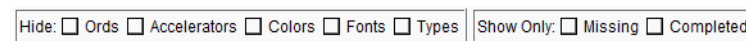
The following components are in the Lexicon Tool:

- [Lexicon display options](#)
- [Search Lexicon](#)

### Lexicon display options

Both [Lexicon Report view](#) and [Lexicon Editor view](#) use a type of tabular presentation with the standard table features and options, as described in the *User Guide* section “Table controls and options”. Lexicon-specific controls and options are displayed at the top of both the Lexicon Editor and Lexicon Report. These controls and options are shown in [Figure 1](#) and described below:

**Figure 1** Lexicon display options

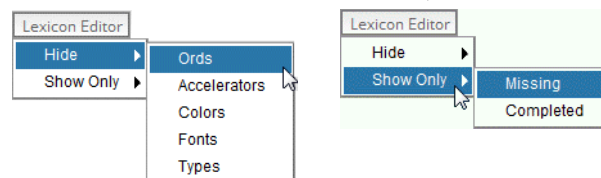


- Hide Ords**  
If enabled, it filters out any referenced based entries, such as `modules://icons/x16/alarm.png`
- Hide Accelerators**  
If enabled, it filters out entries that define keyboard accelerators (entries end with “.accelerator”).
- Hide Colors**  
If enabled, it filters out entries that define colors (entries end with “.bg” or “.fg”), such as those defined with hexadecimal code `#FFFF0000`.
- Hide Fonts**  
If enabled, it filters out entries that define fonts.
- Hide Types (Starting in Workbench 3.7)**  
If enabled, it filters out entries that end with “.displayName” or “.icon”.
- Show Only Missing**  
If enabled, this shows only entries without any assigned values—entries will be the “red” orb ones only.
- Show Only Completed**  
If enabled, this shows only entries that do have assigned values—entries will be the “green” orb ones only.

**Note:** The two “Show Only” settings are mutually exclusive.

In addition, the Lexicon display options can be accessed under the **Lexicon Editor** pull-down menu, as shown in [Figure 2](#). You can clear both of the “Show Only” options, but set only *one*.

**Figure 2** Lexicon Editor menu bar display options



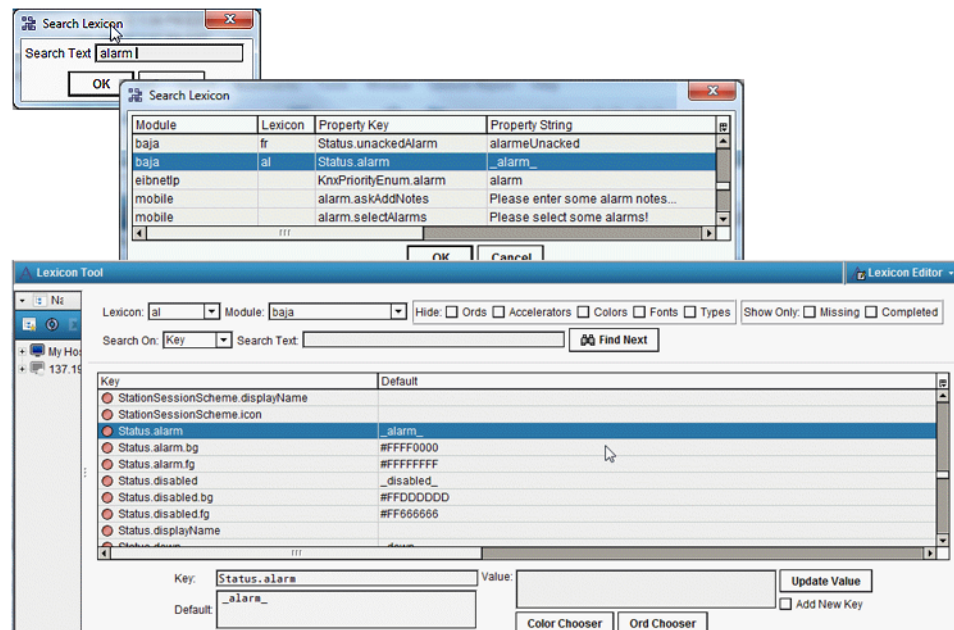
**Note:** The default settings for these display options are set in the **Options** dialog box, as described in the *User Guide* section “Lexicon editor options”. However, these controls allow you to override those Workbench defaults.

## Search Lexicon

Starting in Workbench 3.7, a “Search Lexicon” feature is available in both the Lexicon Report view and Lexicon Editor view. The search capability lets you search for specific text on either the lexicon property Key, the property Default, or the custom Value. Search results are displayed in a table that shows the Module, Lexicon, Property Key, and Property String for each lexicon property that matched the search.

Rows in the search results table are selectable. If you select a row containing only a Module value, then click OK, the corresponding row is selected in the lexicon report table. If you select a row that contains Module and Lexicon values, and then click OK, the view will switch to the [Lexicon Editor view](#). In the Lexicon Editor view, the Lexicon and Module values are set, the properties are loaded for that combination and the row containing the selected key/value pair from the search is selected, as shown in [Figure 3](#). The selection is also applied to the Lexicon Report, so that if you go back to the Lexicon Report view, the corresponding Lexicon list and Module table entries are selected

**Figure 3** Search functionality in Lexicon Tool



## Lexicon plugins (views)

For general information on lexicons, see [“Lexicon overview”](#) on page 2. For details on installing (edited) file-based lexicons in remote JACE platforms, see the *Platform Guide* section “Lexicon Installer”. For details on installing module-based lexicons in remote JACE platforms, see the *Platform Guide* section “Software Manager”.

The following views are found in the lexicon tool:

- [Lexicon Report view](#)
- [Lexicon Editor view](#)
- [Lexicon Module Builder view](#)

### Lexicon Report view

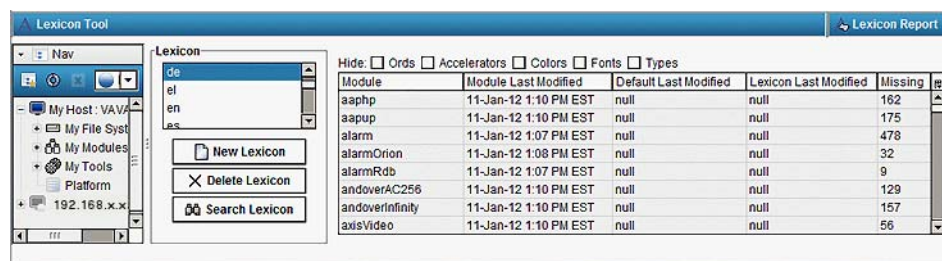
When selecting the **Lexicon Tool** from the **Tools** menu, the default view is the “Lexicon Report,” as shown in [Figure 4](#). The *left* side of this view shows a list of all module-based and file-based lexicon locales installed on your Workbench PC. Each lexicon module (one module per language) contains text-based lexicon files (moduleName.lexicon).

**Note:** *It is possible to have lexicon files in the !lexicon folder, and not have a lexicon module. You can deploy custom lexicons in any file-based and/or module-based combination.*

A lexicon module is conventionally named niagaraLexiconXx and includes a two-letter language designation, such as “De” for German or “Fr” for French (niagaraLexiconFr.jar is the French lexicon module). As needed, you click a lexicon to select it.

**Note:** To load the default English lexicon in the Lexicon Report view, click the **New Lexicon** button and type in “en” in the text field of the **New Lexicon** dialog box. Click the **Ok** button and the lexicon displays in the Lexicon Report view. After this initial display, the “en” name is always listed in the lexicon pane unless you use the **Delete Lexicon** button to remove it.

**Figure 4** Lexicon Report view



Selecting a lexicon in report view shows various status parameters about each lexicon (broken down by module) in the table columns on the right side. These status parameters include the following:

- **Module**  
Each Niagara module that contains a lexicon file (*moduleName.lexicon*).
- **Module Last Modified**  
Timestamp of when the lexicon in the module (JAR file) was last modified.
- **Default Last Modified**  
Timestamp of when the default lexicon file (*!modules/moduleName.jar/moduleName.lexicon*) for that module was last modified. This field is blank if there is no lexicon file for that module. However, you can still edit that module’s lexicon and Save, which creates the lexicon file and updates this timestamp field.
- **Lexicon Last Modified**  
Timestamp of when any module-based or file-based lexicon file for the selected lexicon was last modified. This field is blank if there is no lexicon file for that module. However, you can still edit that module’s lexicon and Save, which creates the lexicon file and updates this timestamp field.
- **Missing**  
Number of keys that are missing values (in the lexicon for that module).

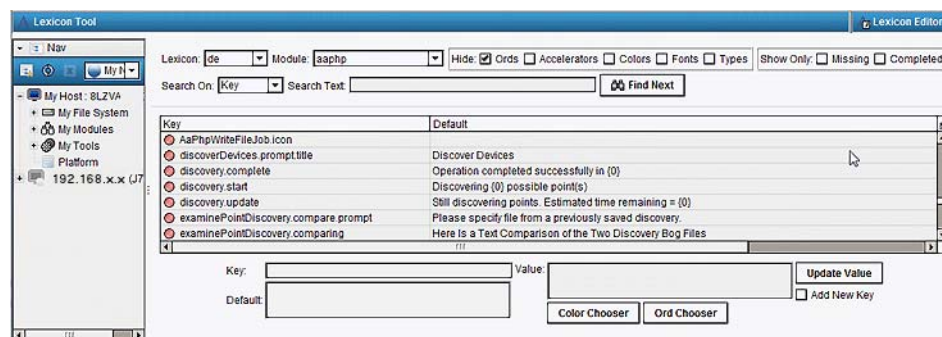
**Note:** Check boxes at the top of report view allow you to “hide” values that could affect “missing” status counts, as described in “[Lexicon display options](#)”.

In this view, you can also add (new) lexicons, delete unwanted lexicons, or search all available lexicon property values that contain a given text. Additionally, from this view you can launch the [Lexicon Editor view](#) by double-clicking any module row, which shows the contents of that lexicon file.

## Lexicon Editor view

The Lexicon Editor lets you view and edit the contents of lexicon files within the lexicon modules on your Workbench PC. You can then install the edited lexicon files in a remote JACE using the platform Lexicon Installer view. Also, you can create (or update) a lexicon module with those edited lexicon files, using the new Lexicon Module Builder view. You can install the module in a remote JACE using the platform Software Manager view.

**Figure 5** Lexicon editor



Typically, you access the editor from the [Lexicon Report view](#) with a lexicon selected on the left side, by double-clicking a module in the selection table on the right side



The lexicon editor displays a table listing the defined keys for that module, with columns for mapping to a localized value override for the default value, as follows:

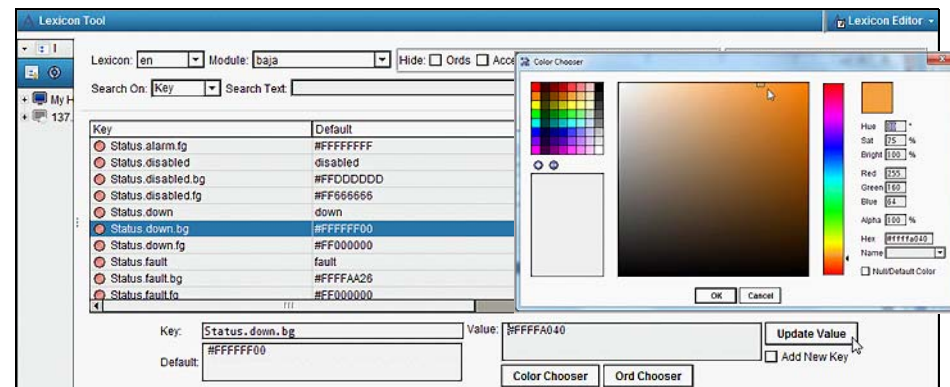
- **Key**  
Niagara interface entity, constant in that module from lexicon to lexicon.
- **Default**  
Default character value, used if no local value is defined for that key (English equivalent).
- **Value**  
Localized character value (overrides default value).

**Note:** Check boxes at the top of the view allow you to “hide” values that could affect “missing” status counts, or filter the view to show only those keys “missing” localized character values or only those keys with “completed” localized character values, as described in [“Lexicon display options”](#).

At the top of this view, you can search for the specified text on either the lexicon property Key, the property Default, or the custom Value. As needed, you can resort the table columns to find information needed, and simply click any row to select that key. Clicking any row in the table populates the **Key** field.

At the bottom of the Lexicon Editor view, the **Add New Key** button enables the Key field. The **Value** text field allows you to modify the value of the selected key or add a value for a new key. Starting in Workbench 3.7 the **Value** field has associated **Color Chooser** and **Ord Chooser** buttons. These options allow you to browse for a specific color or ORD element. Update the table with the the Value field entry by clicking the **Update Value** button. Write changes to the lexicon file by clicking the **Save** button on the toolbar.

**Figure 6** Use color chooser to enter hex code for background color value for Status Alarm down



## About Adding Key/Value Pairs

As mentioned above, the **Add New Key** option allows you to add a new key/value pair to a lexicon file. If a particular key is not included in a lexicon file, you can use this option to add the key. For example, if you want to globally replace the “Elapsed Active Time” and “Elapsed Active Time Numeric” property descriptors with “Runtime” and “Runtime Numeric”, you must add those keys to the lexicon file. This is sourced in the control module. So, you must add the new keys in the control module.

Refer to [“Common tasks”](#) on page 8 for more details on adding key-value pairs.

## Lexicon Module Builder view

Starting in Workbench 3.7, a Lexicon Module Builder view is available. The lexicon module builder allows you to bundle multiple lexicon files into a module for ease of distribution. The lexicons available for use are those lexicon files (modulename.lexicon) located in the appropriate !lexicon subfolder for the target locale. For example, !lexicon/fr is the subfolder which would contain French lexicon files. You can build new lexicon modules or replace existing ones. In this view there are text fields for defining module information, a **Browse** button that allows you to locate existing lexicon modules, a selection table of available lexicon files that can be selected for inclusion, a check box to delete source files after the build is completed, and a **Build Module** button to initiate the build, as shown in [Figure 7](#).

**Figure 7** Lexicon module builder

The text fields in this view allow you to provide basic information for the module. Several of these fields are required. Leaving a required field empty will result in a failed build. The required fields are:

- **Module Name**  
Recommended naming conventions for new modules are: “customLexicon<Xxxx>” or “<Vendor>Lexicon<Xxxx>” or “<Brand>Lexicon<Xxxx>”.
- **Preferred Symbol**  
An abbreviation (1 to 8 alphabetic characters) of the module name .
- **Vendor Name**  
A text string to describe the vendor.
- **Version**  
An integer or integers separated by periods, for example 1 or 1.4.2.

**Brand Pattern**, an optional text field, allows you to specify a “brand” attribute for this module. The Lexicon system refers to this field at runtime, comparing it to the brand for the current system (JACE or Workbench). If you provide a brand name in this field, the set of lexicon files in your module will be available for loading only if this value matches the runtime brand identifier defined as part of the license.

**Note:** *Providing a Brand Pattern value is optional. A lexicon module with no brand pattern will always be available for use.*

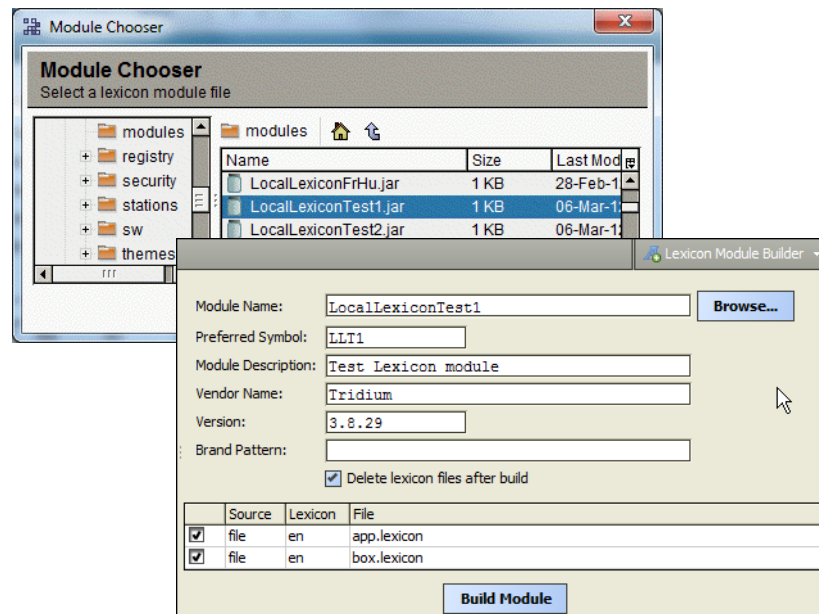
Located just below the brand pattern text field is a check box option to delete lexicon files after the build is completed. If you select this option, when you click the **Build Module** button you will be prompted to confirm that you want to delete files in the !lexicon folder.

As mentioned earlier, the selection table in the lower half of the view lists the lexicons available for use. Click a check box in the left column of a table row to mark a lexicon file for inclusion in the module. Double-clicking an entry in the selection table will switch to the [Lexicon Editor view](#) for the selected language-module combination. Switching between the lexicon views will retain your selections (text box values, selected table rows, check box values) unless a module was loaded from the Module Chooser. In that case, the module must be re-loaded in order to re-populate the selection table with the lexicons within that module.

If rebuilding an existing lexicon module, click the **Browse** button to open a Module Chooser dialog box. Once you select a module in the Chooser and click **OK**, it fills in the context of that module in the available text fields in the Lexicon Module Builder view and the module’s existing lexicon files are added to the selection table, as shown in [Figure 8](#). Clicking the **Cancel** button in the Chooser closes the dialog box. When you click the Build Module button, you will get separate confirmation dialog boxes to overwrite the existing module, to delete the lexicon files, and indicating the module was rebuilt successfully.

**Note:** *Lexicon modules can be rebuilt multiple times.*

**Figure 8** Browse to select existing lexicon modules



**Note:** A lexicon module edited with the Lexicon Module Builder must adhere to the following restrictions:

- Module **may not contain** Java class files (compiled Java code).
- Module **may not contain** default lexicon modules provided by Tridium (*niagaraLexicon<xx>.jar* files in the `!modules` folder).
- Source files selected for inclusion in the module must be uniquely named. For example, selecting a file-based lexicon and a module-based lexicon both of which are named `control.lexicon` will result in an error message.

On completion of the build, you will see a confirmation message indicating that the module was constructed and placed in the `!modules` folder.

## Common tasks

This section provides instructions on how to use the Lexicon Tools to perform the following common tasks:

- [About English \(en\) lexicon usage](#)
  - [Editing the default English \(en\) lexicon](#)
  - [Adding key-value pairs to a lexicon module](#)
  - [Editing an existing key-value pair in a lexicon module](#)
  - [Creating of a new lexicon module for distribution](#)

### About English (en) lexicon usage

An important use of lexicons is for non-English language support. However, you can also modify the English (en) lexicon to provide a “global default” change for display of items seen in workbench, or served by a running station to client web browsers. This could be a possible alternative to working from the slot sheet for multiple points to make the same type of interface or “display name” change, or from making similar changes using the **Batch Editor** view of the ProgramService.

The English language serves as the default Workbench lexicon. As such, it is not included as a separate lexicon “en” distribution subfolder with `module.lexicon` files. However, using the [Lexicon Editor view](#) in Workbench you can still edit and add lexicon keys in different modules for English usage. These act as “default overrides”, saved as `module.lexicon` files in a `!lexicon/en` subfolder. After restarting Workbench, such changes are used when opening any station (including one on your Workbench PC, if applicable). Typically, you also install this *modified* en lexicon into remote JACE platforms, such that stations running on them serve up the same changed descriptions to browser clients.



## Editing the default English (en) lexicon

This example describes how to add the English (en) lexicon to the list of currently installed lexicons in the Lexicon Report view.

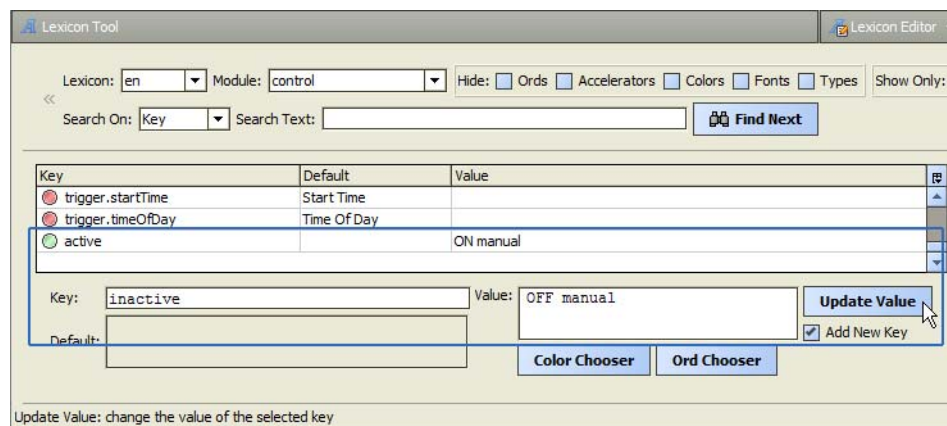
To edit the English (en) lexicon:

- Step 1 From the Tools menu, select **Lexicon Tool**.  
The **Lexicon Report** view (Figure 4) displays all lexicons currently installed.
- Step 2 If an “en” lexicon is not listed in the left pane, click the **New Lexicon** button at the bottom.  
In the **New Lexicon** dialog box, type “en” in the **Name** field and click **OK**.  
An “en” entry is added to the bottom of the list of lexicons.
- Step 3 Select the “en” lexicon, which populates the right pane with the different modules in which you can change or add lexicon keys.

## Adding key-value pairs to a lexicon module

This procedure shows how to edit an existing lexicon to add key-value pairs. In this example, you edit the `control` module in the English (en) lexicon to add two key-value pairs in order to change the default right-click descriptors for the actions, “active” and “inactive”.

- Step 1 In the **Lexicon Report** view, select the “en” lexicon in the left pane, which populates the right pane with the different modules in which you can change or add lexicon keys.  
**Note:** *If the English (en) lexicon is not listed, you can add it by following the procedure for “Editing the default English (en) lexicon” on page 9.*  
In this case, you need to work in the `control` module.
- Step 2 Double-click the `control` module, to load a list of existing key-value pairs in the **Lexicon Editor**.  
By default, there are no existing keys shown for the actions, `active` and `inactive`, so they must be added.
- Step 3 Select the “**Add New Key**” check box at the bottom, far right in the view, which enables the **Key** field.
- Step 4 In the **Key** field, enter the *slot name* (in this case, type slot name `active`), and in the **Value** field enter the desired text, for example: `ON manual`  
Click the **Update Value** button to add the new key-value pair.
- Step 5 Repeat the previous step to add another key-value pair, editing text in the **Key** field to: `inactive` and editing the desired text in the **Value** field, for example: `OFF manual`, as shown here.



Deselect the “Add New Key” check box after adding this second key-value pair.

- Step 6 Click the **Save** button on the toolbar.  
This creates (or updates) the `control.lexicon` file located in your `!lexicon/en` folder.
- Step 7 Restart Workbench, open a station and right-click a writable point to issue actions.  
The new default values (`ON manual` and `OFF manual`) replace the old defaults (`Active` and `Inactive`).  
**Note:** *If working on a Supervisor (hosting a station), note that following a station restart, such lexicon changes become effective defaults to web browser clients connecting to the station.*

*Note also that any “display name” changes for slots on a component override any lexicon values, including any modified in this manner. For example, if a BooleanWritable has the Display Name of its “active” action slot edited to “Start Pump”, that command text will always appear when issuing this action, whether from client browser access or from Workbench access.*

### Editing an existing key-value pair in a lexicon module

This procedure shows how to change the alarm module **Priority** key value from “Priority” to “Importance”.

To edit the alarm module in the English (en) lexicon:


- Step 1 In the **Lexicon Report** view, select the “en” lexicon in the left pane, which populates the right pane with the different modules in which you can change or add lexicon keys.

**Note:** *If the English (en) lexicon is not listed, you can add it by following the procedure for “Editing the default English (en) lexicon” on page 9.*

In this case, you need to work in the alarm module.

- Step 2 Double-click the alarm module, to load a list of existing key-value pairs in the **Lexicon Editor**. Scroll or Search to find the **priority** key.

- Step 3 In the **Value** field type in the desired text, for example: Importance  
Click the **Update Value** button to replace the previous value (Priority) for this key.

- Step 4 Click the **Save**  button on the toolbar.  
This creates (or updates) the alarm.lexicon file under your !lexicon/en folder.

- Step 5 Restart Workbench, open a station and right-click a writable point to issue actions.  
The new default value (Importance) replaces the old default (Priority).

**Note:** *If working on a Supervisor (hosting a station), note that following a station restart, such lexicon changes become effective defaults to web browser clients connecting to the station.*

*Note also that any “display name” changes for slots on a component override any lexicon values, including any modified in this manner. For example, if a BooleanWritable has the Display Name of its “active” action slot edited to “Start Pump”, that command text will always appear when issuing this action, whether from client browser access or from Workbench access.*

### Creating of a new lexicon module for distribution

This procedure shows how to bundle two edited lexicon files (alarm, control) in a module for easy distribution.

**Note:** This method can be used to bundle legacy lexicon files (from AX-3.6 and earlier) for distribution on an AX-3.7 or AX-3.8 installation.

To create the new lexicon module:

- Step 1 From the Tools menu, select **Lexicon Tool**.  
The **Lexicon Report** view (Figure 4) displays all lexicons currently installed.
- Step 2 Click the **View** dropdown menu and select **Lexicon Module Builder**. The Lexicon Module Builder view displays populating the selection table with any subfolders and files currently located under your !lexicon folder.
- Step 3 Define the new module information by filling in the text fields as shown in this example:
- Module Name: abcLex
  - Preferred Symbol: abc
  - Module Description: Modified En alarm and control
  - Vendor Name: Tridium
  - Version: 1.1
- Step 4 Click the check box to select the following Source files in the En lexicon:
- alarm.lexicon
  - control.lexicon
- Step 5 Click the check box to **Delete lexicon files after build** to remove these source files from the !lexicon/en folder.
- Step 6 Click the **Build Module** button.  
This creates the new module under your !modules folder, in this case the new module is abcLex.jar.
- Step 7 Restart Workbench.

In this example, navigate to the !modules folder and scroll to confirm the new module was created. You can install this new lexicon module on remote JACE platforms (using platform Software Manager), such that stations running on them serve up the same changed descriptions to browser clients.

**Note:** *If working on a Supervisor (hosting a station), note that following a station restart, such lexicon changes become effective defaults to web browser clients connecting to the station.*


*Note also that any "display name" changes for slots on a component override any lexicon values, including any modified in this manner. For example, if a BooleanWritable has the Display Name of its "active" action slot edited to "Start Pump", that command text will always appear when issuing this action, whether from client browser access or from Workbench access.*

## Lexicon Tool Component Summaries


The following sections describe Lexicon Tool components:

- [Lexicon Tool](#)
- [Lexicon Display Options](#)
- [Search Lexicon](#)


### Lexicon Tool

 Starting in AX-3.7, the Lexicon Tool component is available under the Workbench Tools menu. The lexicon tool is the container for the three lexicon views that enable viewing lexicon modules and files installed on your Workbench PC, editing the contents of lexicon files within lexicon modules installed on your Workbench PC, and building new lexicon modules. For detailed information, see "[Lexicon overview](#)" on page 2.

### Lexicon Display Options

 Lexicon-specific controls and options are displayed at the top of both the Lexicon Report view and the Lexicon Editor view. For detailed information, see "[Lexicon display options](#)" on page 3.

### Search Lexicon


 Starting in Workbench 3.7, a "Search Lexicon" feature is available in both the Lexicon Report view and Lexicon Editor view. The search capability lets you search for specific text on either the lexicon property Key, the property Default, or the custom Value. For detailed information, see "[Search Lexicon](#)" on page 4.

## Lexicon Tool Plugin (view) Summaries


The following sections describe plugins (views) available in the lexicon tool:

- [Lexicon Report](#)
- [Lexicon Editor](#)
- [Lexicon Module Builder](#)


### Lexicon Report

 The Lexicon Report is the default view displayed when you select the Lexicon Tool. Lexicon Report view lists all module-based and file-based lexicon locales installed on your Workbench PC. For detailed information, see "[Lexicon Report view](#)" on page 4.

### Lexicon Editor

 The Lexicon Editor lets you view and edit the contents of lexicon files within the lexicon modules installed on your Workbench PC. For detailed information, see "[Lexicon Editor view](#)" on page 5.

### Lexicon Module Builder

 Starting in Workbench 3.7, a Lexicon Module Builder view is available. The lexicon module builder allows you to bundle multiple lexicon files into a module for ease of distribution. For detailed information, see "[Lexicon Module Builder view](#)" on page 6

## Document change log

- Revision date: November 6, 2013  
Minor corrections throughout, including:
  - a statement in the section on Lexicon Module Builder view of where lexicon files are located.

- updated a screen captures in figures 4 and 5.
- added emphasis to user tasks in the Common tasks section.
- Publish date: May 16, 2012