Introduction

Welcome to the ACT! Localization Development Kit (LDK). The LDK includes tools and the Alchemy Catalyst Translation Toolkits (TTKs) that are used to localize the ACT! application using Alchemy Catalyst 6.

This Help file covers the localization process for ACT! and ACT! Premium for Web. Localizing ACT! Premium for Web builds requires additional steps and tools in some cases.

Before you begin

- You should be familiar with ACT! contact management software, including how to edit supplemental files such as layouts, templates, reports, envelopes, and labels. For more information, see the *ACT! Help*.
- You should be familiar with localizing American software for other cultures.
- You should be generally familiar with how a software package is put together, including the various file formats such as .DLL, .EXE and .XML.

Roles

Generally, two roles are engaged in producing a localized ACT! build:

• The **LDK User**:

- Prepares the computer where the LDK will be installed (see the following requirements).
- Prepares the resources (Application TTKs, Database TTK, and ACT! supplemental files) for translation.
- Validates the translated resources.
- Creates the localized build of ACT!.

• The **Translation User**:

- Translates TTKs using Alchemy Catalyst.
- Translates ACT! supplemental files using ACT!.
- Translates other string and auxiliary files using a text or XML editor as needed.

Software requirements

	IDV	TT1 4*
Software	LDK User	Translation User
Alchemy Catalyst 6 SP1 or later	X	X
ACT! (version must match what you are localizing)	X	X (If translating ACT! supplemental files)
Macrovision InstallShield 12	X	
Macrovision DemoShield	X	
Microsoft VisualStudio .NET 2005 or later	X (if translating ACT! plug-ins)	
Text or XML editor		X
Image editor capable of editing and exporting PSD files		X
IIS 6 (Web LDK only)	X	
ACT! Premium for Web package (Web LDK only) - access from the LDK package and add the path to pftwwiz.exe as	X	

an environment	
variable.	

Next steps

See the <u>Overview of the ACT! localization process</u> for steps for translating resources and finalizing a localized build.

Overview of the ACT! localization process

With each ACT! and ACT! Premium for Web release, an LDK will be distributed. The LDK contains the necessary components to localize ACT!. The LDK for each product version includes the following:

- ACT! Application Translation Toolkits (TTKs)
- ACT! Database TTK
- LDK configuration profile (.ldk)
- LDK Process Manager
- Build media report
- ACT! LDK Getting Started
- ACT! LDK Help

The ACT! localization process is as follows:

- 1. Install the LDK.
- 2. <u>Translate</u> TTKs and other components.
- 3. Use the LDK Process Manager to <u>copy ACT! build files</u> to your LDK directory.
- 4. Use the LDK Process Manager to <u>finish localizing files</u>.

Installing the LDK

The LDK installation CD contains all of the necessary files to create a localized version of ACT!. See the Installation catalog for an overview of files that install on your computer.

To install the LDK

- 1. Prepare the system where you are installing the LDK with the system requirements in the Introduction.
- 2. Insert the LDK CD and follow the instructions on the screen to install the files.

After installing the LDK, you must configure Alchemy Catalyst to work properly with the LDK files.

To configure Alchemy Catalyst

- 1. In Catalyst, in the **Tools** menu, point to **Options**, and then click **.Net Framework**.
- 2. Specify a **Key file** of C:\program files\ACT\ACT LDK\MercuryPublicKey.snk.
- 3. Select Extract only modified resources to satellite assemblies.

Related topics

<u>Installation catalog</u>

Installation catalog

The installation catalog tells you what files the LDK copies to your local system. You look into the appropriate folders to find the resources that you need to translate before Finalization.

The LDK CD installs the following:

Folder Name	Description
XXSource	Contains the ACT! assemblies for the current LDK release.
CatExtract	Contains language-specific folders where Catalyst extracts TTKs.
New Media	Contains localized builds when you have <u>run all processes</u> .
Plugins	Contains ACT! plug-ins.
Resources\	
ActXDemo-database files	Contains the ACT! demo database for the current release.
ActAuxFiles\	Contains ACT! supplemental files.
Bitmaps\	Contains LDK-specific graphics.
C1Spell	Contains the Component One Dictionary Utility.
CDBrowser\	Contains CD browser files.
Database	Contains ACT! and SQL database files.
DNBXML	Contains .NET XML controls.
OtherFiles	Contains miscellaneous files for the LDK.
TTKs	Contains the TTK files for the current LDK release.

About localization and translation

After you install the LDK, you are ready to begin the localization and translation process. Some elements of the application require an overall localization procedure, and some elements are simply files to translate. You may translate files internally, or work with a translation vendor.

Elements to localize include:

- Quote files
- ACT! plug-ins
- Dictionaries
- ACT! Link for Palm OS
- ACT! Network Synchronization Service

Elements to translate include:

- Application TTKs
- Database TTK
- ACT! supplemental files
- CD browsers
- SQL1
- Plug-ins
- ACT! installer
- AdminPlugin (Web only)
- Web installer including the Outlook Integration installer (Web only)
- Additional files

Other tasks:

- <u>Determining documentation conditions</u>
- Integrating online Help

Translating TTKs

The ACT! Localization Build Kit includes the following TTKs:

- Application TTKs ACT.UI TTKs (vary in number)
- Database TTK ACT.Database.ttk

Most of ACT!'s user interface (UI) resources are contained in the Application TTKs. You use Alchemy Catalyst to localize the TTKs and generate the translated output. The Application TTKs each contain application binaries (.dlls and .exes). The Database TTK contains the strings in the ACT! database.

Translating other build components

The ACT! build contains a number of other elements that you may translate, including ACT! supplemental files, CD browsers, splash screens, the SQL1 file, and plug-ins.

You translate ACT! supplemental files in the ACT! application. Additionally, you may also translate other text and XML-formatted files using a text or XML editor. If you are translating user documentation, see the readme accompanying the documentation source files for specific information.

You can create and manage custom word lists and spelling dictionaries using the Component One Dictionary Utility that comes with the LDK.

Translating the Application TTKs

The Translation User uses Alchemy Catalyst to translate strings and dialogs in the Application TTKs. Open each TTK in Catalyst and translate the appropriate resources.

If you are localizing an ACT! Premium for Web build, the LDK package includes one or more additional TTKs that are exclusive to the Web product. These TTKs contain strings you translate for the Web product.

To translate the Application TTKs

- 1. In Catalyst, open a TTK and examine the resources you will translate.
- 2. Import your translation glossaries and apply them to the TTK, or translate the resources manually by typing directly into Catalyst.

Tip: Save a local copy of each TTK as you work through the localization project.

3. Repeat steps 1-3 for each Application TTK.

For help using Catalyst, see the Catalyst online Help or Experts. The next step in the localization process is to <u>translate the Database TTK</u>.

Translating the Database TTK

The Translation User uses Alchemy Catalyst to translate strings in the Database TTK. You must open the TTK in Catalyst and translate the appropriate resources.

To translate the Database TTK

- 1. In Catalyst, open the TTK and examine the resources you will translate.
- 2. Import your translation glossaries and apply them to the TTK, or translate the resources manually by typing directly into Catalyst.

Tip: Save a local copy of each TTK as you work through the localization project.

For help using Catalyst, see the Catalyst online Help or Experts.

The next step in the localization process is to <u>translate ACT! supplemental files</u>.

Translating the ACT! supplemental files

There are a number of ACT! supplemental files, such as layouts and reports, that you translate during the localization process. The Translation User uses the ACT! application to translate ACT! supplemental files. The following sections contain steps for translating these files.

Translating ACT! layouts

A person with an understanding of database concepts and is knowledgeable in using the ACT! Layout Designer should perform layout translation.

To translate ACT! layouts

1. Open each of the English layouts from the most current English ACT! version in the ACT! Layout Designer of the localized ACT! product.

The database fields should pre-populate correctly for the localized language given that the database schema is localized.

2. Edit each label or tab name as preferred for dialect and/or language.

Language dependent resizing and rearranging of labels and fields may be necessary due to length of strings and preferred address format.

- 3. Set the tab stops and enter stops as desired.
- 4. Save the layout.
- 5. Test the layout to assure correct data and complete labels appear.
- 6. If any field data is incorrect, delete and then replace the field.

Translating reports

A person who is knowledgeable in basic reporting principles, database concepts, and XML should perform report translation.

To translate reports

1. Open each of the English reports from the most current ACT! English version in the ACT! Report Designer of the localized ACT! product.

The database fields should pre-populate correctly for the localized language given that the database schema is localized.

2. Edit each label as preferred for dialect and/or language.

Language dependent resizing and rearranging of labels and fields may be necessary due to length of strings and preferred address format.

- 3. Save the report.
- 4. Test the report to assure correct data and complete labels are displayed

Note: To test properly, you must enter all variations of data in the ACT! application for Contacts, Groups, Companies, Activities, Opportunities, History, and Notes.

- 5. If any field names are incorrect, delete and then replace the field.
- 6. Custom Visual Basic scripting may be required to optimize appearance of address and footer formats.
- 7. Certain field names included in custom scripting need to be translated directly from the report script using a UTF-8 text editor (Examples: True/False, Open, Close-Won, Closed-Lost).

Translating other supplemental files

To translate other supplemental files

- 1. In the ACT! desktop application, open the localized ACT! database.
- 2. Translate each supplemental file as needed.

For help using the ACT! desktop application, see the online Help.

The next step in the localization process is to <u>translate any additional files</u> with a text or XML editor.

Translating CD browsers

You can translate the ACT! CD browsers by integrating localized images into the CD browser panels and translating the browser panel links. You receive the CD browser images in PSD format with the LDK.

Input

Image files and URLs

Output

Translated CD Browsers.

To translate CD Browsers

1. Edit the English-language PSD images with an image editor.

Note: The PSD images correspond to every browser panel except the closing bitmap, which you edit later in this procedure.

- 1. Export the images to JPG format.
- 2. For each image to replace with a localized verion, right-click the image, point to Properties, point to Fill Styles, and then click Import.
- 3. For each panel with links to translate, right-click the link, point to Properties, click Actions, and edit the link.
- 4. Localize the file Closing.BMP with an image editor, and replace the existing closing image using the instructions in step 2.

Translating the SQL1 file

You translate the SQL1 file to ensure that the localized build contains the appropriate currency and language name.

Input

Transations and locale codes

Output

SQL statements that the LDK runs during a build

To translate SQL1

- 1. Browse to and open the SQL1.sql file, located in the database folder.
- 2. Find the CONFIGVALUE of CULTURE NAME and set the locale code. For example, "es-ES".
- 3. Find the DESCRIPTION of CULTURE NAME and set text for the locale you want. For example, "Español (Spain)".
- 4. Find the CONFIGVALUE of CURRENCY SYMBOL and set the currency symbol. For example, "€".
- 5. Save the SQL1.sql file in the same location that it was installed.

The LDK uses this file to update the database during the build process.

Translating plug-ins

You use Visual Studio.Net 2005 to translate plug-in projects.

Input

CS and RESX files

Output

Plug-in assemblies

To translate plug-ins

- 1. Open the projects and translate strings that are displayed in plug-ins.
- 2. Translate strings in the .resx files.
- 3. Translate constant strings in the .cs files.

Keep the projects where they were installed so that the LDK process can build and copy them to the correct location in the build.

Related topics

Localizing ACT! plug-ins

Translating the ACT! installer

Use Installshield to open the installer project and translate the strings within.

Input

String table

Output

Installer executable files

To translate the installer

- 1. Open the InstallShield project and export the string table.
- 2. Translate the string table (send to translator).
- 3. Import the translated string table to the InstallShield project.
- 4. Go to each release project, and make sure the .NET framework and Language Pack is the language of the project.
- 5. Find Localization.ini installed in Other Files under the LDK.
- 6. Make sure that the LocaleFolder and LocaleNetSymbol are set to the project locale.

This file will be copied into the media with the LDK.

Translating the Outlook Integration installer

Use Installshield to open the installer project and translate the strings within.

Input

String table

Output

Installer executable files

To translate the installer

- 1. Open the InstallShield project and export the string table.
- 2. Translate the string table (send to translator).
- 3. Import the translated string table to the InstallShield project.
- 4. Go to each release project, and make sure the .NET framework and Language Pack is the language of the project.
- 5. Find Localization.ini installed in Other Files under the LDK.
- 6. Make sure that the LocaleFolder and LocaleNetSymbol are set to the project locale.

This file will be copied into the media with the LDK.

Translating additional files

The LDK contains additional text and XML-formatted files that you translate using a text or XML editor. These are generally any files listed in your <u>task list</u> other than the TTKs and ACT! supplemental files. Check your task list to indentify these files.

To see the steps for translating the Import Synonyms XML files, click <u>here</u>. The next step in the localization process is to <u>finalize your files</u>.

Translating the Import Synonyms XML Files

The ACT! Import Wizard references several XML files containing synonyms (aliases) for default mapping of company and contact field names between ACT! and other source files, such as Microsoft Excel, tab-delimited (.TXT), and comma-separated value (.CSV) files.

These import synonyms XML files are named:

- CompanyFieldSynonyms.xml
- ContactFieldSynonyms.xml

Note: A person with an understanding of XML should perform the translation.

To translate the Import synonym files

1. Make a copy of the file before you open it.

Caution: Do not rename the file. The ACT! Import Wizard references the files as named above.

- 2. Open the file using an XML or text editor.
- 3. Translate the text that appears in the opening and closing brackets. See graphic below.

The text that appears between the opening and closing set of brackets is the ACT! field name the synonym is mapped to. In the example below, the text is Address 1.

4. Save the xml file.

Managing dictionaries

The ACT! LDK includes two dictionary files:

Main.dct is the primary word list the ACT! spelling checker uses.

Custom1.dic is where the application places words users add to the dictionary.

Through the LDK, you can provide a language-specific spelling dictionary that replaces the Main.dct file. You first create or obtain a word list in the desired language, then you use the Component One Dictionary Utility provided with the LDK to create the dictionary file. You can also add special words to the custom dictionary file, Custom1.dic.

Creating a text word list

The Dictionary Utility uses a text file (.TXT format) word list formatted as follows:

- One word per line.
- Each line must end with a carriage return.
- The last character of the file must be an end-of-line character (ASCII code decimal 26 or CTRL-Z).
- The words do not have to be in alphabetical order.
- The words are not case-sensitive.

Creating a spelling dictionary (Main.dct)

To provide a language-specific dictionary in a localized build, you use the Dictionary Utility to convert the word list into a DCT file that ACT! can use.

To create a spelling dictionary

- 1. Open the Dictionary Utility by browsing to C:\ACT\ACT LDK\Resources\C1Spell and double-clicking **Dictutil.exe**.
- 2. Under **Text file**, click **Browse** (...) button and select a text word list.
- 3. Under Dictionary file, type or browse to the path and filename of the new dictionary file to create.

Note: The file must be named "Main.dct" to be recognized by the ACT! application. If the file already exists, the utility will overwrite it.

4. Click **Build dictionary from text file**.

Modifying the custom dictionary

You can modify the custom dictionary if you want to include special words without replacing the entire spelling dictionary.

To modify the custom dictionary

• Open the Custom1.dic file in a text editor.

The file should be formatted as follows:

- The name always must be Custom1.dic.
- The file should remain in ANSI format, unless special characters require UTF-8 or UTF-16 format.
- One word per line.
- Each line must end with a carriage return.
- The last character of the file must be an end-of-line character (ASCII code decimel 26 or CTRL-Z).
- The words do not have to be in alphabetical order.
- The words are not case-sensitive.

Documentation conditions

If you translate user documentation for a localized build, you receive documentation source files. Since user documentation is single-sourced to generate multiple versions, much of the source text has conditions applied by the authoring tool. Each delivery of source files includes a readme file that details the conditions used and the manner in which they apply.

Localizing ACT! plug-ins

In the ACT! localization files area, auxiliary files for each locale are kept in folders that are canonically named for the version and target language. The Spanish file set for ACT! 8.02, for example, is:

\\sagelocobuild1\Resources\802_LDK\ES\ES802

Each such folder contains a Visual Studio solution (here ES802.sln) that presents the auxiliary file set as a series of projects. These projects organize the files into related sets, such as print templates. Projects IEPluginProj contains the files for the Internet Explorer plug-in, and OfficePluginProj contains the files for the MS Office plugin.

Plug-in localization does not take place in the TTKs. Any references to plug-in code in the current set of TTKs should be removed. In releases after Mercury, the TTK mechanism will be extended to include these plug-ins in a special TTK of their own. These instructions hold until then.

Make all localization changes directly in the source as given below. Do not compile source in this project. These files will be included with the other projects in this solution, then compiled as part of the localization build. Start all localizations by copying the ENU base for the current Mercury version, which will contain the original English strings to be manually translated, then changing the version and target language names to form a new aux files tree for the locale. Because the file links in the solution are relative, no changes to the solution and project structure need be made unless files are added or subtracted for a new version.

Localization Changes to Plugins Source

Changes are indicated are indicated:

- In .cs files by comments of the form // *Example* Localize -->
- In .resx files by comments of the form <!-- *Example* Localize -->

All modifiable source files for each plugin are included in these projects, but not all source contains changes. The changes in each file are given below, it is good practice to check for new occurrences of a comment containing *Example*. This comment precedes each set of strings to be translated.

Project **IEPluginProj**:

CAttachFile.cs

```
const string REMOTING_CONNECT_ERROR = "Translation" const string MENU_TEXT = "Translation" const string MENU_STATUS_BAR = "Translation" const string BUTTON_TEXT = "Translation"
```

CAttachFile.resx

```
Value for <data name="DEFAULT_VISIBLE">
Value for <data name="MENU_TEXT">
Value for <data name="MENU_STATUS_BAR">
Value for <data name="BUTTON_TEXT">
Value for <data name="REMOTING_CONNECT_ERROR">
```

Project **OfficePluginProj**:

${\bf Add Template Field Dialog.cs}$

```
ret = "Translation"
ret = "Translation"
ret = "Translation"
```

AddTemplateFieldDialog.resx

```
Value for <data name="label1.Text">
```

```
Value for <data name="add.Text">
Value for <data name="close.Text">
Value for <data name="$this.Text">
```

Connect.cs

Constants.cs

ExcelAdapter.cs

ExcelAdapter.resx

```
Value for <data name="MENU_ACT_CAPTION">
Value for <data name="MENU_ATTACH_CAPTION">
Value for <data name="MapToContactButtonCaption">
Value for <data name="MapToCompanyButtonCaption">
Value for <data name="MapToGroupButtonCaption">
Value for <data name="RefreshButtonCaption">
Value for <data name="CouldNotConnectMessage">
Value for <data name="MapToActMessageTitle">
Value for <data name="CouldNotConnectUnknownMessage">
Value for <data name="CanNotAddPermissionMessage">
Value for <data name="SerializationExceptionMessage">
Value for <data name="MapFieldsContactLabelText">
Value for <data name="MapFieldsCompanyLabelText">
Value for <data name="MapFieldsGroupLabelText">
Value for <data name="MESSAGE_DOC_NOT_SAVED">
Value for <data name="MESSAGE TITLE TEXT">
Value for <data name="MESSAGE_WORKBOOK_ATTACHED">
```

IMFieldsDialogHost.cs LibraryTabConnect.cs MapFieldsDialog.cs MapFieldsDialog.resx OfficeAdapterBase.cs

```
attachButtonItem.Caption = "Translation"
```

Office A dapter Base.resx

```
Value for <data name="MENU_ACT_CAPTION">
Value for <data name="MENU_ATTACH_CAPTION">
```

Of fice Adapter Factory. cs

OfficeConnect.cs

```
if (string.Compare(installTarget, "Translation", true) == 0)
```

PowerPointAdapter.cs

PowerPointAdapter.resx

```
Value for <data name="MESSAGE_TITLE_TEXT">
```

Value for <data name="DOC_NOT_SAVED_MESSAGE">

Value for <data name="CouldNotConnectMessage">

Value for <data name="CouldNotConnectUnknownMessage">

Value for <data name="CanNotAddPermissionMessage">

Localizing quote files

The ACT! Quotation feature depends on maintenance of a special relationship between two files in the Specials section:

Quote.adt – A Word 2000-format file containing an embedded Excel spreadsheet that displays the quotation detail line for each product in the quotation. The spreadsheet maintains its own grand totals.

Quote.xlt – The embedded spreadsheet component.

Although Quote.adt contains an embed of Quote.xlt and does not link to it, some ACT! Features need to refer to Quote.xlt directly. Others use Office Automation to refer to and operate the embedded spreadsheet in Quote.adt.

For each locale, translate this file pair as follows:

- 1. In Excel 2000, translate and save Quote.xlt.
- 2. In Word 2000, translate and save Quote.adt.
- 3. In Quote.adt, in the **Reveal Codes** view, locate the spreadsheet embed block {EMBED Excel, 8} and carefully delete it, precisely maintaining the cursor position.
- 4. Re-embed the translated Quote.xlt file in the same place by clicking **Insert**, pointing to **Object**, and then clicking **From File**, leaving both special-handling check boxes cleared to create a new embedded, not linked, object.

This should result in an identical EMBED statement as you deleted.

- 5. In the **Reading** page view, select the visible spreadsheet embed, and then examine the object properties with "Format Object...".
- 6. In the **Web** tab, insert the object codebehind name **Quote** into the empty box. This is the name by which the quotation feature finds the spreadsheet component.
- 7. Return to **Normal** page view.
- 8. Save and exit.
- 9. In an ACT! Installation of the target locale, verify that you can:
 - Click Other Files... from the Write menu, select Quote.adt, and view the translated quotation form and blank spreadsheet embed.
 - Add an Opportunity to a contact, create at least one product with nonzero values for it, click Quote, and see the same translated quotation form with spreadsheet. The values you entered for each

product of the opportunity should be filled in the spreadsheet and summed. Verify that all columns are visible.

Integrating online Help

If you translate and include ACT! online Help in your localized build, you produce an uncompiled Web Help file from your translated Help system. When integrating the Web Help file, you must include all files and folders.

About the LDK Process Manager

There are two essential LDK elements handled by the LDK Process Manager: *tasks* and *processes*.

Tasks are operations the LDK Process Manager performs on your translated resources. An example of a task is copying files from one folder to another.

Processes are sets of tasks, grouped together for a common purpose. An example of a process is compiling, digitally signing, and placing files into a localized build.

The LDK Process Manager lets you view and customize the tasks and processes that will produce the localized build. The LDK Process Manager uses .LDK files provided with the LDK installation to retrieve the processes necessary for the localized build. When you run the processes, the tool outputs results to an XML file named ACT.ldr.

You use the LDK Process Manager to:

- Select a localization profile.
- Review and edit the processes in the localization profile.
- Run the processes.

To start the LDK Process Manager, double-click the LDK Process Manager icon on your desktop.

The ACT! LDK currently provides two localization profiles. The first, CopyGoldBuild.ldk, <u>copies ACT! build files</u> from a network location or CD to your local LDK directory. The second, ACT90.ldk, performs the <u>finalization tasks</u>.

How do I...?

Select a localization profile

Copy ACT! build files

View the resources to process

Selecting a localization profile

The LDK Process Manager uses *localization profiles* to load settings and customizations. Each localization profile is specific to the version of ACT! it is designed for. When you start the LDK Process Manager, you open a localization profile provided with the LDK.

The ACT! LDK currently provides two localization profiles. The first, CopyGoldBuild.ldk, <u>copies ACT! build files</u> from a network location or CD to your local LDK directory. The second, ACT90.ldk, performs the <u>finalization</u> tasks.

To select a localization profile

- 1. Start the LDK Process Manager.
- 2. Click **Open Config**, browse to the .ldk file to open, and then click **Open**.

Copying ACT! build files

The ACT! LDK includes a localization profile that copies ACT! build files from a network location or CD to your local LDK directory. You run the processes in this profile before proceeding to finalization.

The default location to copy files from is a network share containing gold ACT! builds. You can change the metadata in the process tasks to copy files from another location, such as a CD.

To copy ACT! build files

- 1. Start the LDK Process Manager and <u>select</u> the CopyGoldBuild localization profile.
- 2. (Optional) In any task, <u>edit the metadata</u> to specify a new SourceDirectory to copy files from a CD or other location.
- 3. Click **Run All Processes**.

Viewing the resources to process

The LDK Process Manager features several visual tools for viewing and managing the processes in your <u>localization profile</u>. A process is set of tasks in the profile that must be validated and executed in order to produce a localized build. The number of processes in your localization profile depends on the information and resources provided in each LDK version. When you load the localization profile, the Configuration tab is selected by default and the profile name appears in the task window.

You can expand the profile name by clicking the plus sign next to it, which displays a list of the processes in the localization profile. You can then expand each process to see the individual tasks that make up that process.

Once you are familiar with the processes in your localization profile, you validate and run each process in the <u>finalization</u> stage.

About finalization

When all TTKs, ACT! supplemental files, and additional files have been translated, you are ready to finalize the files using the LDK Process Manager. Finalization consists of:

- Making any necessary changes to metadata for each task in each process.
- Validating each process.
- Correcting any metadata errors.
- Running each process.

You use the LDK Process Manager for each of these operations.

How do I...?

Make changes to task metadata
Validate a process
Correct metadata errors
Run a process

Editing task metadata

Each task in each process of your localization profile contains metadata that defines certain parameters of the task. You can edit the metadata as needed. For example, one of the tasks in the process of localizing your Application TTKs might be log into Alchemy Catalyst. In the task for logging into Catalyst, you could change the username the LDK Process Manager uses to log in.

To edit metadata in a single task

- 1. Select a task to edit.
- 2. In the Metadata window, click a field to edit and enter text.

-or-

Click the **Find and Replace** tool to edit multiple instances of a word or phrase in the selected task.

To edit metadata in all tasks

• Click the **Find and Replace** tool and enter the text to replace.

The next step is to <u>validate</u> the metadata changes you made.

Validating task metadata

After translating a resource, you use the LDK Process Manager to validate it. The validation process checks the resource to ensure that the LDK Process Manager can compile and put the resource in a localized build correctly.

To validate a task or process

• In the Task window, click a task or process, and then click the **Validate** tool.

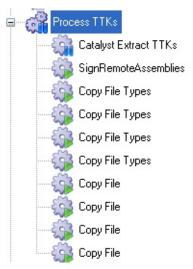
Note: Validating a process will validate the metadata in every task in that process.

Results of your validation appear in two places: the Process/Task window and the Log window.

Viewing results in the Process/Task window

After validating an item, an indicator appears that represents the validation result. If you validate an individual task, the indicator appears on that task. If you validate an entire process, an indicator appears on every task under the process, as well as on the process name itself, as follows:

- A blue "pause" icon on a *task* or *process* indicates a metadata error.
- A green "play" icon on a *task* indicates the task metadata is valid.
- A blue "check" icon on a *process* indicates the metadata is valid for every task in the process.



If you have a validation error, see <u>Correcting metadata errors</u> for more information.

Viewing results in the Log window

A record of the metadata validation also appears in the Log window. A log entry appears when you start the validation process, and for the results. A green "check" icon and accompanying log message indicates successful validation; a yellow "warning" triangle and accompanying log message indicates one or more validation errors.

How do I...?

Correct metadata errors

Correcting metadata errors

If you encounter metadata errors while validating tasks and processes, you must correct them before you can run your processes. You can view validation results in the Validation Messages window. For example, if you entered metadata for a task that instructed the task to place files in a directory that does not exist, an error message would appear in the window.

To view and correct metadata errors

- 1. Click a task that displays an error icon.
- 2. Review the error message in the **Validation Messages** window.
- 3. Make the necessary corrections and <u>validate the item</u> again.

Running processes

The final stage in producing a localized build is running the processes you have validated. All processes must be <u>validated</u> before running, ensuring that their parameters are correct. Processes have a variety of functions, such as:

- Copying files.
- Processing TTKs.
- Digitally signing files.
- Building plug-ins.
- Adding resources to a localized ACT! build via InstallShield.

To run a single process

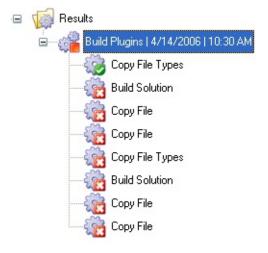
• Select a process to run, and then click the **Run Selected Process** tool.

To run all processes

• Click the **Run All Processes** tool.

Viewing results

When you run a process, the Results tab appears automatically with a list of processes. Successfully run processes display a green check, and unsuccessful processes display a red "X". You can expand each process to view individual tasks, and then click a task to see any log or error messages.



When you have run all processes successfully, the LDK process manager creates your <u>localized build</u>. Generally, if one task in a process fails, the subsequent tasks will also fail because they are dependant on the previous tasks. In the example above, the task "Build Solution" failed, causing the remaining tasks to fail.

How do I...?

Correct process errors
Find my localized build

Correcting process errors

If you encounter errors while running processes, you must correct them before you complete your localized build.

To correct process errors

- 1. On the Results tab, under the failed process, click the first failed task.
- 2. Review the information in the Log Messages and Error Messages windows.
- 3. Make the necessary corrections and run the process again.

Once you correct any errors and run the processes successfully, <u>find your</u> localized build.

Finding your localized build

When you successfully run each process in your localization profile, your localized build is ready. Browse to \\Program Files\ACT\ACT LDK\New Media to locate the build.

Glossary

Satellite Assembly

.NET strategy for localized resources.

Tier

Tier refers to the product level of ACT! installation. Tiers include ACT! by Sage and ACT! by Sage Premium.

TTK

Translation Toolkit. A TTK is an Alchemy Catalyst project file and contains various application resource binaries for translation using Alchemy Catalyst 6.

Localizing ACT! Link for Palm OS

- 1. Edit the ActPalmLinkSetup.exe.config file and set "installedLanguage" to the desired locale.
- 2. Add the satellite assembly folder (for example, en-gb) to the palmOne Folder that is installed with the Link application.
- 3. Add copies of the sync satellite assemblies into the above directory.

After following these steps the config dialogs will be translated.

Localizing the ACT! Network Synchronization Service

- 1. Ensure that Act.Framework.Synchronization.Service.UI.exe.config is installed in the ACT Network Sync directory. It is found in the ACT! install directory currently.
- 2. Edit Act.Framework.Synchronization.Service.UI.exe.config and set "installedLanguage" to the desired language.
- 3. Create a TTK for Act.Framework.Synchronization.Service.exe and translate to generate satellite assembly.
- 4. Sign the new satellite assembly.
- 5. Create a satellite assembly folder for the desired language in the ACT Network Sync folder.
- 6. Place the new satellite assembly in the above directory.