Acumatica

Contents

Copyright

11235 SE 6th, Suite 140
Bellevue, WA 98004

Restricted Rights

The product is provided with restricted rights. Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in the applicable License and Services Agreement and in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (c)(2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable.

Disclaimer

Acumatica, Inc. makes no representations or warranties with respect to the contents or use of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Acumatica, Inc. reserves the right to revise this document and make changes in its content at any time, without obligation to notify any person or entity of such revisions or changes.

Trademarks

Acumatica is a registered trademark of Acumatica, Inc. HubSpot is a registered trademark of HubSpot, Inc. Microsoft Exchange and Microsoft Exchange Server are registered trademarks of Microsoft Corporation. All other product names and services herein are trademarks or service marks of their respective companies.

Software Version: 6

Last updated: October 7, 2016

Introduction

This document provides important information about fixes, enhancements, and key features that are available in Acumatica ERP 6. The document is of particular interest to those who install Acumatica ERP or develop applications for it. All users can benefit from reviewing this content to determine how they may benefit from the changes in this release.

To try new features and improvements, you can use the demo company, which you can easily deploy by selecting the *SalesDemo* item in the Acumatica ERP Configuration Wizard. For detailed information on deploying the demo company and working with the demo data, see *Demo Materials* on the Acumatica ERP partners' portal.

Installation and Upgrade Notes

For a detailed description of the general procedure of updating Acumatica ERP, see *Updating Acumatica ERP*.

We strongly recommend that before you update to a newer product version, you back up all configuration files and databases used by the application instances. An upgrade to Acumatica ERP 6 may cause issues with customizations and dashboards.

Dashboards that were created in previous versions of Acumatica ERP will be converted into the new format during upgrade. We recommend that you check all dashboards after upgrade and adjust them according to your needs.

Multiple changes have been made in Acumatica ERP 6 that may affect customizations and integrations. For details, see *Upgrade Procedure: Customizations* and *Upgrade Procedure: Screen-Based API Wrapper*.

If your Acumatica ERP instance was integrated with software provided by an independent software vendor (ISV) partners, we recommend that you consult with your partner about the compatibility of their products with Acumatica ERP 6.

Upgrade Policy

On the *Apply Updates (SM.20.35.10)* form, only minor updates for your current version of Acumatica ERP are available. You should upgrade your Acumatica ERP from previous versions of the system to Version 6 manually on the server (upgrade by using the web interface is not supported due to significant changes in customizations).

Upgrade Notes

Upgrades to Acumatica ERP 6 must be performed as described in the table below. The upgrade process differs depending on the version from which you are upgrading.

: Acumatica ERP 6 supports only Integrated mode of the Internet Information Services application pool. Classic mode is not supported. Before starting the upgrade, switch the application pool to Integrated mode. Otherwise, the upgrade will not be performed, and a corresponding error message will be displayed.

From Version	Upgrade Path
5.0 and later	Direct upgrade to Acumatica ERP 6 is supported.
4.2 (4.20.1036 and later)	Incremental upgrade is required. The following steps must be performed in the listed order: **1.** Upgrade to the latest 5.3X build available. **2.** Upgrade to Version 6.
4.2 (before 4.20.1036)	Incremental upgrade is required. The following steps must be performed in the listed order: **1.** Upgrade to Version 4.20.1036. **2.** Upgrade to the latest 5.3X build available. **3.** Upgrade to Version 6.

: We do not recommend that you upgrade from Acumatica ERP 5.3X to Acumatica ERP 6 if your build of Acumatica ERP 5.3X was released later than the build of Acumatica ERP 6 you are going to upgrade because in this case some Acumatica ERP 5.3X fixes might not be included in Version 6.

Upgrade Procedure: Customizations

Multiple changes were done since version 5.3 Update 13 (5.30.2347) which may affect reports, customizations and integrations that were built in the prior version.

To prevent breaking changes in your customizations, do the following:

1. Analyze your customizations.
2. Carefully read the list of breaking changes in the *Upgrade Procedure: Breaking Changes* section.
3. Replace changed objects with their alternatives.
4. If the objects used in your customizations were removed without a replacement, consider creating new customizations.

After an upgrade to Acumatica ERP 6, to detect the changes that break an existing customization, you can check the compatibility of the code included in all published customization projects with the original code. For details, see *Publishing a Customization* (the Checking the Compatibility of the Customization Code with the Original Code section).

: The validation that detects breaking changes is by default turned on in Acumatica ERP6. If you receive any errors during the validation, we strongly recommend that you read the *Resolving an Issue Discovered During the Validation* article.

The complete list of the changes with the suggested alternatives is provided in the Acumatica ERP 6 Technical Release Notes document.

Upgrade Procedure: Breaking Changes

Multiple changes have been made since Version 5.3 Update 13 (5.30.2347) that may affect reports, customizations, and integrations that were built in a prior version. The basic changes are listed in this section. The complete list of the changes with the possible alternatives is provided in the Acumatica ERP 6 Technical Release Notes document.

Removed Screens

No screens have been removed that were present in the prior versions.

Removed Tables

The following table lists the tables that have been removed from the database and possible replacements for these tables, where applicable.

Removed	Suggested Replacement
EPChangesetDetail	
EPActivityKvExt	
EPReminder	CRReminder
EPReminderKvExt	
AUNotificationHistory	
APBalances	
EPActivity	• CRActivity • SMEmail • PMTimeActivity

Removed Fields

The following table displays the fields that have been removed from the database and possible replacements for these fields.

Removed	Suggested Replacement
CRActivityRelation.RefTaskID	CRActivityRelation.RefNoteID
CRActivityRelation.TaskID	CRActivityRelation.NoteID
CRActivityStatistics.LastIncomingActivityID	CRActivityStatistics.LastIncomingActivityNoteID
CRActivityStatistics.LastOutgoingActivityID	CRActivityStatistics.LastOutgoingActivityNoteID
CSAnswers.EntityID	CSAnswers.RefNoteID
CSAnswers.EntityType	CSAnswers.RefNoteID
CSAttributeGroup.Type	CSAttributeGroup.EntityType
EPAttendee.EventID	EPAttendee.EventNoteID
EPAttendeeMessage.EventID	EPAttendeeMessage.EventNoteID
EPOtherAttendee.EventID	EPOtherAttendee.EventNoteID

Removed	Suggested Replacement
EPOtherAttendeeMessage.EventID	EPOtherAttendeeMessage.EventNoteID
EPTimeCardSummary.ParentTaskID	EPTimeCardSummary.ParentNoteID
GIDesign.PrimaryScreenAsPopup	GINavigationScreen.WindowMode
GIDesign.PrimaryScreenID	GIDesign.PrimaryScreenIDNew
GIResult.SuppressNav	GIResult.DefaultNav and GIResult.NavigateTo
ListEntryPoint.EntryScreenNodeID	ListEntryPoint.EntryScreenID
ListEntryPoint.ListNodeID	ListEntryPoint.ListScreenID

Modified Field Types

No data types have been modified in comparison to the prior versions.

Removed Data Access Classes

The following table lists the data access classes (DACs) that have been removed from the database.

Namespace	Class Name
PX.SM	EMailAccountMaint+EPActivity
PX.Data	PXFieldScreenToSiteMapAddHelper<T>+SiteMapInternal
PX.Objects.EP	EmployeeActivitiesEntry+EPActivityFilter
PX.Objects.EP	NavigationFilter
PX.Objects.EP	EmailEnq+Box
PX.Objects.EP	EmailEnq+MailBoxFilter
PX.Objects.EP	EmailEnq+EmailFilter
PX.Objects.EP	EPChangesetDetail
PX.Objects.EP	EPReminder
PX.Objects.CR	SimpleCase
PX.Objects.CR	MyCasesByCustomerMaint+Filter
PX.Objects.CR	MyCasesByCustomerMaint+CaseInfo
PX.Objects.CR	MyCaseDetailsMaint+EPActivityWithOwner
PX.Objects.CR	IMessage
PX.Objects.CR	EPMessage
PX.Objects.CS	CSAnswers2

Activity Record Redesign

If an integration uses the `EPActivity` data access class and has been implemented in Acumatica ERP 5.3 or any earlier version, it has to be reconsidered and adjusted for use with Acumatica ERP 6; integrations are not compatible across versions. Also, you may need to review integrations and customizations that involve time cards and time activities.

Acumatica partners that have integration with Acumatica Self-Service Portal implemented should be aware that upgrading the Self-Service Portal to Version 6.0 may cause it to malfunction.

As a result of the redesign of this data access class, any type of activity (event, task, email, or activity) created in Acumatica ERP 6 is stored as a `CRActivity` record or one of the following combinations of different types of records:

- `CRActivity` and `PMTimeActivity`
- `CRActivity` and `SMEmail`
- `CRActivity`, `SMEmail`, and `PMTimeActivity`

`PMTimeActivity` and `SMEmail` records can represent self-sufficient entities: An `SMEmail` record can be an abridged version of an email (system email), and a `PMTimeActivity` record can be created on the *Employee Time Card (EP.30.50.00)* or *Employee Time Activities (EP.30.70.00)* form, with no related `CRActivity` record created.

During an upgrade to Version 6.0, custom fields and database indexes used in the older version are preserved in the `Obsolete_EPActivity` table. A custom field is likely an add-up to the new `PMTimeActivity` table, so the following script template can be used to transfer custom data.

```
UPDATE PMTimeActivity
SET UsrField1 = E.UsrField1,
UsrField2 = E.UsrField2
FROM PMTimeActivity T
INNER JOIN Obsolete_EPActivity E
ON E.CompanyID = T.CompanyID AND E.NoteID = T.RefNoteID
```

After the script is run, the `Obsolete_EPActivity` table can be deleted to save database space.

Upgrade Procedure: Screen-Based API Wrapper

Because of the connection of the screen-based SOAP API with Acumatica ERP forms, the applications that are developed based on this API are sensitive to the UI changes in the system. For interaction with Acumatica ERP, the client application uses the schema of an Acumatica ERP form, which defines the correspondence between the API elements and the internal data fields of Acumatica ERP. If something has been changed in the UI of Acumatica ERP, the schema is changed too. However, the application continues to use the schema for which it was compiled and fails. Thus, any changes made to the UI after the application is created require you to update and recompile the application.

As an alternate approach, you can change your client application so that it uses the screen-based API wrapper. This approach prevents UI changes from causing application failures after an upgrade to a newer version of Acumatica ERP and omits the step of updating and recompiling the application for each UI change.

The screen-based API wrapper is a special wrapper that was designed to prevent failure of your application that can happen because of the UI changes in the system. The screen-based API wrapper saves the schema of an Acumatica ERP form that was initially used by the application to a local XML file and restores this schema to the server each time the application is running, which prevents breaking of the application because of changes in the user interface of Acumatica ERP. For more information on the screen-based API wrapper, see *Screen-Based API Wrapper*.

Upgrade of a Client Application for Acumatica ERP 6

To prevent application failures with the update to Acumatica ERP 6, you can recompile your application so that it uses the screen-based API wrapper as follows:

1. Create a test copy of your production instance of Acumatica ERP.
2. Make the following changes to the project of your client application:
   a. Add a reference to `PX.Soap.dll` to your project. This file is installed automatically with Acumatica ERP 6 in the `ScreenBasedAPIWrapper` folder of your Acumatica ERP installation folder.
   b. Use the `Helper.GetSchema()` method from the `PX.Soap` library in the code of your application instead of the corresponding `Screen.GetSchema()` method to obtain the schema of each form.
3. Compile the application and test it with the test copy of your Acumatica ERP instance.
4. Update the test copy of your Acumatica ERP instance to Acumatica ERP 6.
5. Test the client application with the updated test copy of your Acumatica ERP instance.
6. Update the production instance of Acumatica ERP to Acumatica ERP 6.

For more information on the use of the screen-based API wrapper, see *To Use the Screen-Based API Wrapper*.

Organization: Improved Processing of Activities

Acumatica ERP 6 brings the following benefits to system performance:

- The processing of activities during various update operations has been streamlined and sped up.
- The database is upgraded more quickly due to a reduced number of indexes in the tables.

In previous versions, the `EPActivity` table stored all data related to activity records, which led to the multiple performance issues.

Changes to the Database

The `EPActivity` table has been split into the following four tables:

- `CRActivity`: This table stores all tasks, events, and activity records, such as information about phone calls.
- `SMEmail`: This table stores inbound and outbound emails and the email details required for the email processing and synchronization with Exchange.
- `PMTimeActivity`: This table stores the activity details required for reporting the time spent on activities. Time cards now use the data from only this table.
- `CRReminder`: This table stores reminders for activities.

Any type of activity (event, task, email, or activity) created in Acumatica ERP 6 is stored as a `CRActivity` record or a combination of two or three different types of records, which can be any of the following:

- `CRActivity` and `PMTimeActivity`
- `CRActivity` and `SMEmail`
- `CRActivity`, `SMEmail`, and `PMTimeActivity`

At the same time, `PMTimeActivity` and `SMEmail` records can represent self-sufficient entities: An `SMEmail` record can be an abridged version of an email (system email), and a `PMTimeActivity` record can be created on the *Employee Time Card (EP.30.50.00)* or *Employee Time Activities (EP.30.70.00)* form, with no related `CRActivity` record created.

The `EPActivity` table has been renamed to `Obsolete_EPActivity` to preserve any custom data.

Custom Data Transfer

The changes to the database may affect integrations, customizations, generic inquiries, and custom reports. An upgrade of the Acumatica Self-Service Portal to version 6 may cause its malfunctioning. Because integrations are not compatible across versions, any integration, if it uses the `EPActivity` data access class and if it has been implemented in Acumatica ERP 5.3 or any earlier version, has to be reconsidered and adjusted for use with Acumatica ERP 6. Also, you may need to review integrations and customizations that involve time cards and time activities.

During an upgrade to Acumatica ERP 6, the `EPActivity` table in the database is renamed to `Obsolete_EPActivity` to preserve all custom fields and database indexes used in the older version. After the update, you can run a script to transfer your custom data. Below is an example of such a script for the most probable case when each custom field is an add-up to the new `PMTimeActivity` table:

```
UPDATE PMTimeActivity
SET UsrField1 = E.UsrField1,
UsrField2 = E.UsrField2
FROM PMTimeActivity T
```

```
INNER JOIN Obsolete_EPActivity E
ON E.CompanyID = T.CompanyID AND E.NoteID = T.RefNoteID
```

After the script is run, the `Obsolete_EPActivity` table can be deleted to save database space.

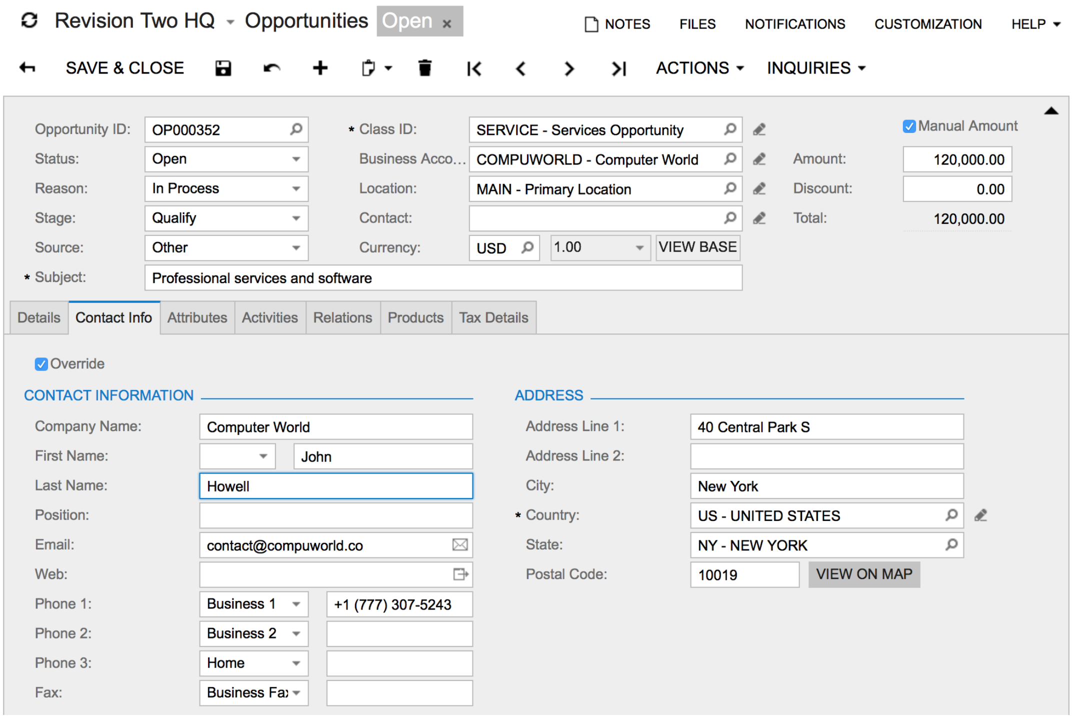
Changes to User Interface and Workflows

No user interface changes were introduced by these improvements.

If a detail line is added on the *Employee Time Card (EP.30.50.00)* form, the system records only the time reporting data and does not create an activity. That is why a user cannot open such a record by using the *Activities (EP.40.43.00)* form.

CRM: Contact Information for an Opportunity

In Acumatica ERP, an opportunity represents a potential sale to a new or existing customer. Previously, in order to view or modify contact information for an opportunity, you had to open the opportunity-related contact or business account on the *Contacts (CR.30.20.00)* or *Business Accounts (CR.30.30.00)* form. Now you can access contact information by using the new **Contact Info** tab of the *Opportunities (CR.30.40.00)* form. (See the following screenshot.)



Opportunity ID: OP000352
Status: Open
Reason: In Process
Stage: Qualify
Source: Other
* Subject: Professional services and software
* Class ID: SERVICE - Services Opportunity
Business Acco…: COMPUWORLD - Computer World
Location: MAIN - Primary Location
Contact:
Currency: USD 1.00
☑ Manual Amount
Amount: 120,000.00
Discount: 0.00
Total: 120,000.00

☑ Override

CONTACT INFORMATION

Company Name: Computer World
First Name: John
Last Name: Howell
Position:
Email: contact@compuworld.co
Web:
Phone 1: Business 1 +1 (777) 307-5243
Phone 2: Business 2
Phone 3: Home
Fax: Business Fa:

ADDRESS

Address Line 1: 40 Central Park S
Address Line 2:
City: New York
* Country: US - UNITED STATES
State: NY - NEW YORK
Postal Code: 10019

Figure: Contact Info tab

If you have a contact who is not registered in the system, but who is interested in your quote and has displayed an intention to buy, the contact can be regarded as a qualified opportunity. You can immediately create an opportunity for this contact by using the *Opportunities* form and specify basic contact information on the fly by using the **Contact Info** tab of the form, instead of first creating a new business account or contact on the *Business Accounts* or *Contacts* form, respectively, and then associating it with the opportunity.

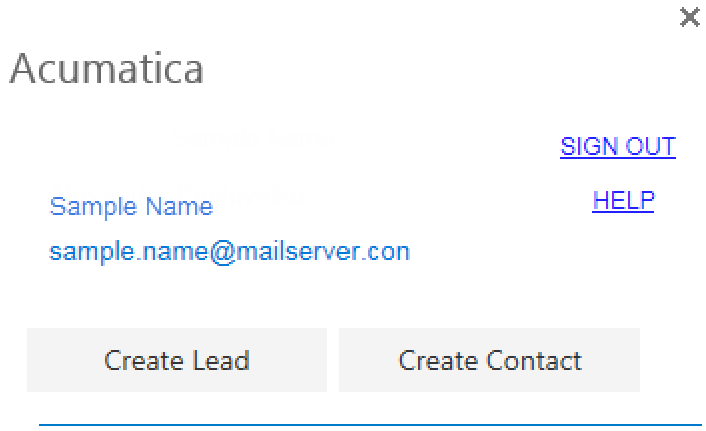
For details on creating opportunities for contacts that are not registered, see *Opportunity-Related Contact Information*.

CRM: Acumatica Add-In for Outlook

The Acumatica add-in for Outlook processes the incoming and outgoing mail in your Outlook mailbox and searches for leads, contacts, and employees in Acumatica ERP that correspond to the sender and recipient addresses in the emails. With this add-in, you can do any of the following:

- View details on lead and contact records that already exist in Acumatica ERP
- Create new leads and contacts right from your mailbox
- Create opportunities and cases associated with existing contacts
- Log your communication activities
- Customize the appearance of the Acumatica ribbon button in Outlook in accordance with your company's branding

If no matching records are found in the system, the **Create Lead** and **Create Contact** buttons appear on the add-in form so that you can create a lead or contact record based on the details of the selected email recipient or sender. (See the following screenshot.)

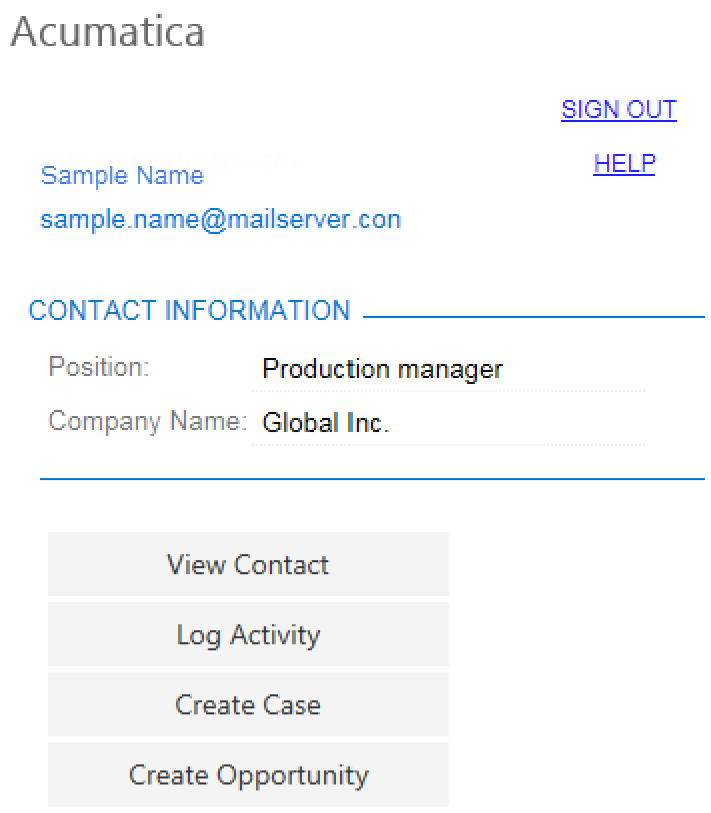


Sample Name

sample.name@mailserver.con

Figure: New contact

In the screenshot below, you can see an example of the information displayed for an existing contact and the action buttons available for this contact.



Sample Name
sample.name@mailserver.con

CONTACT INFORMATION

Position: Production manager
Company Name: Global Inc.

Figure: Existing contact

You must have an Office365 or Exchange account to be able to use the Acumatica add-in for Outlook. This add-in requires one of the following applications:

- Outlook 2013 or later
- Outlook on the Web

For the add-in to work properly, your Acumatica ERP instance must be hosted over HTTPS.

For details on the installation process and use of the add-in, see *Acumatica Add-In for Outlook*.

Finance: Approvals of Accounts Payable Documents

Acumatica ERP 6 provides functionality that you can use to set up the process of approving the Accounts Payable documents by the designated employees before these documents are processed in the accounting systems and paid. These approvals can be used, for example, for preventing double-payments of bills or payments for inaccurate or fraudulent bills.

Approvals can be performed by one person only or by multiple persons. When multiple persons are designated to approve a document, they can approve it either in parallel (the documents are reviewed by multiple approvers at the same time), or in multiple successive stages (that is, once one employee has approved the document, it becomes available for approval by the next employee). You can set up both of these processes in Acumatica ERP.

Setting Up the Approval of Accounts Payable Documents

The approval workflow functionality is set up on the *Accounts Payable Preferences (AP.10.10.00)* form. Before you set up the approval process on this form, you first configure the approval map or maps on the *Assignment and Approval Maps (EP.20.50.00)* form. A map defines the employee or employees designated for approving AP documents, and the conditions that AP documents must match to be assigned for approval. Once the maps are created, you specify the needed ones on the **Approval** tab of the *Accounts Payable Preferences* form. You can, if needed, specify the template of the notification email to be used by the system for notifying approvers about the documents pending their approval.

Processing AP Documents with Approvals Set Up

With approvals set up, when an AP document is created, saved, and taken off hold on the *Bills and Adjustments (AP.30.10.00)* form, it gets the *Pending Approval* status. The responsible approver receives an email notification (if an email notification template is specified for this approval process) about the document pending his or her approval, and can review the documents pending approval on the *Approvals (EP.50.30.10)* form. On this form, he or she can select one record or multiple records in the list and approve the documents, or approve or reject each particular document directly on the *Bills and Adjustments* form.

You can view the detailed approval information for each particular Accounts Payable document on the **Approval Details** tab of the *Bills and Adjustments* form. On this tab, you can view to whom the document was assigned, who actually approved the bill, and when approval was granted.

You can make changes in the approved document once you put it on hold. After you take the document off hold, the approval process starts again.

For details on setting up the approvals for Accounts Payable documents, see *Approvals of Accounts Payable Documents*.

Finance: Accounts Receivable Parent-Child Functionality Enhancements

If your company deals with companies with franchises or multiple branches, you can configure parent-child relationships in Acumatica ERP between the customer accounts that represent these companies.

In Acumatica ERP 6, the parent-child customer relationship functionality has been enhanced. In Acumatica ERP 6, the ability to manage the credit control on the parent account level has been implemented.

For details, see *Credit Policy Application*.

Credit Verification Rules

Now, the credit verification rules defined for the parent account can be shared with the child accounts. You can set up this capability on the *Customers (AR.30.30.00)* form by selecting the **Share Credit Policy** check box in the **Parent Info** section of the **Billing Settings** tab for child accounts. This check box becomes available when you select the **Consolidate Balance** check box in the same section. With these settings specified, the settings specified for the parent account in the **Credit Verification Rules** section on the **General Info** tab will be applied to the documents of all related child accounts.

Consolidated Dunning Letters

If your company policy requires that dunning letters be sent (and the *Dunning Letter Management* feature is enabled in your system), the system consolidates the overdue documents of a parent account and its child accounts that have the **Share Credit Policy** check box selected in the **Parent Info** section of the **Billing Settings** tab on the *Customers* form, and sends the dunning letters to the parent account contact. If you change the credit policy configuration for the parent account, the system will reflect these changed settings in its child accounts that share the parent account's credit policy.

AR Reports That Can Include Consolidated Data

In Acumatica ERP 6, you can generate reports that group child accounts (or their documents) under the parent account and show individual account balances along with the consolidated balance of the group. The child accounts that have the **Consolidate Balance** check box selected are grouped under their parent account, and their balances are consolidated with the parent balance. The following Acumatica ERP reports have a check box (either **Group by Parent Account** or **Consolidate Data by Parent Account**) that you can select to generate reports regarding parent-child relations:

- *AR Balance by GL Account (AR.63.20.00)*
- *AR Balance by Customer (AR.63.25.00)*
- *AR Balance by Customer MC (AR.63.30.00)*
- *AR Aged Past Due (AR.63.10.00)*
- *AR Aged Past Due MC (AR.63.11.00)*
- *AR Aged Outstanding (AR.63.15.00)*
- *AR Aged Outstanding MC (AR.63.16.00)*
- *AR Aged Period Sensitive (AR.63.05.00)*

: If any of these reports has been customized in your system, you need to customize these updated reports again because the new parameter has been added to the reports.

Platform: Dashboards and KPI Enhancements

The management and configuration of dashboards have been significantly enhanced in Acumatica ERP 6. It is now more convenient to add new dashboards, manage existing dashboards, and configure widgets on dashboards. For details on managing dashboards, see *Managing Dashboard Pages*. For information about the configuration of widgets, see *Dashboards*.

Implemented Enhancements

The following enhancements have been introduced in Acumatica ERP 6:

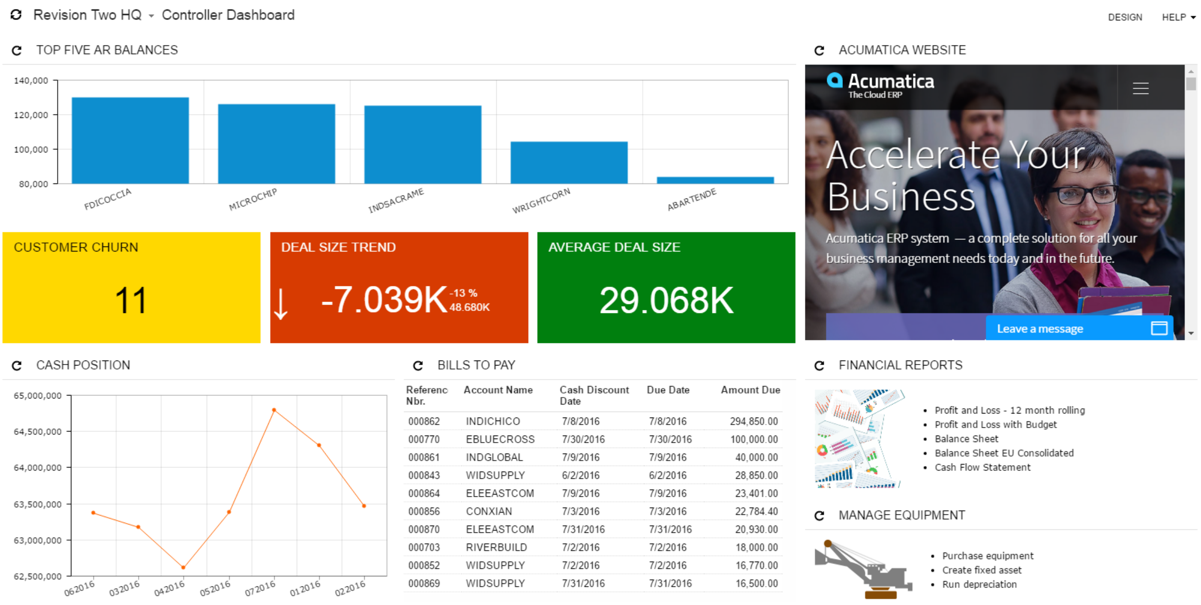
- A single form for managing all dashboards in the system: The new *Dashboards (SM.20.86.00)* form has been added, where you can add and remove dashboards and modify the properties of dashboards.
- Dashboard management by a dedicated user role: A user with a role that has full access to the *Dashboards* form can add and delete dashboards, set up access rights to dashboards for user roles, change the locations of dashboards in the site map, allow users to modify dashboards, and populate dashboards with widgets. The predefined *DashboardDesigner* role is no longer needed for managing dashboards.
- Ability to locate dashboards in any position of the site map: You can locate dashboards in the position of the Acumatica ERP site map where they are most convenient for the users who can use the dashboards in their daily work.
- Control of access rights to dashboards: You can control which user roles have the ability to view each dashboard. For users who can view the dashboard, you can also control whether the users can personalize the dashboard (that is, configure and modify the widgets of the dashboard).
- Personalization of dashboards without changing the original dashboard: A copy of a dashboard is created automatically when a user begins to modify any widget on the dashboard. Users work with their copy of the dashboard and do not change the original dashboard.
- Dashboard owner: Each dashboard now has an owner. You can assign one user to be the owner of all dashboards (for example, an administrator) or designate a dedicated user as the owner of each dashboard. The dashboard owner can configure the original dashboard. The predefined *DashboardDesigner* role is no longer needed for dashboard configuration.
- View and design modes of a dashboard: Two modes have been implemented for dashboards—view mode and design mode. In view mode, users can only view information on widgets; they cannot configure widgets. In design mode, users can add, delete, and modify widgets. Switching between these modes helps users to avoid occasional modifications of a dashboard.
- Support of new widgets: The list of supported widgets has been extended with the embedded page, the scorecard KPI, and the trend card KPI widgets. By using new widgets, you can extend the types of information to be placed on a dashboard.
- Easier chart widget configuration: The dialog box where you specify properties of a chart widget has been redesigned to unify settings and make it easier to configure the widget.
- Predefined dashboard layouts: You can select one of the predefined layouts of a dashboard to place widgets on, for convenient configuring of dashboards and viewing of information.

Supported Widgets

The following types of widgets can be added to dashboards:

- **Charts**: Graphical representations of data from Acumatica ERP forms. You can use charts of various types, such as bar charts and line charts.

- **Data tables**: Data from an Acumatica ERP form arranged into rows and columns.
- **Wiki pages**: Reference articles, procedures, business plans, and other content that is frequently referenced by users of Acumatica ERP.
- **Embedded pages**: Documents or images that are stored on external resources such as cloud storage and should be displayed on an Acumatica ERP dashboard.
- **Scorecard KPIs**: Parameters that are important for the business processes of your organization and that should be monitored on a daily basis.
- **Trend card KPIs**: Parameters whose dynamic change is important for the business processes of your organization.



Referenc Nbr.	Account Name	Cash Discount Date	Due Date	Amount Due
000862	INDICHICO	7/8/2016	7/8/2016	294,850.00
000770	EBLUECROSS	7/30/2016	7/30/2016	100,000.00
000861	INDGLOBAL	7/9/2016	7/9/2016	40,000.00
000843	WIDSUPPLY	6/2/2016	6/2/2016	28,850.00
000864	ELEEASTCOM	7/9/2016	7/9/2016	23,401.00
000856	CONXIAN	7/3/2016	7/3/2016	22,784.40
000870	ELEEASTCOM	7/31/2016	7/31/2016	20,930.00
000703	RIVERBUILD	7/2/2016	7/2/2016	18,000.00
000852	WIDSUPPLY	7/2/2016	7/2/2016	16,770.00
000869	WIDSUPPLY	7/31/2016	7/31/2016	16,500.00

- Profit and Loss - 12 month rolling
- Profit and Loss with Budget
- Balance Sheet
- Balance Sheet EU Consolidated
- Cash Flow Statement

- Purchase equipment
- Create fixed asset
- Run depreciation

Figure: Example of a dashboard with new widgets

The screenshot above illustrates all supported types of widgets.

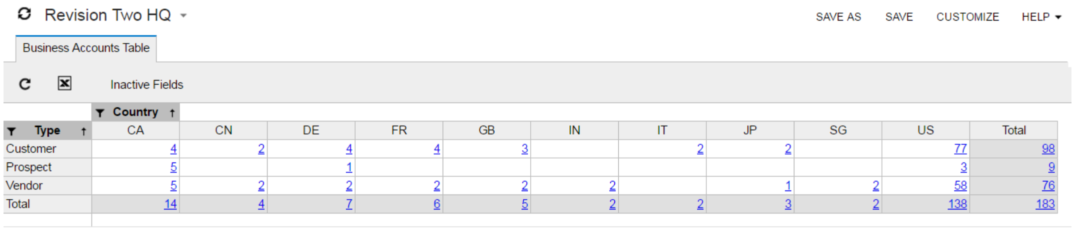
User Interface Changes

The following changes have been made in the user interface of Acumatica ERP to support new functionality of dashboards:

- The new *Dashboards (SM.20.86.00)* form has been added to the site map. You can use this form to maintain dashboards and access rights to them.
- The new **Add Widgets** dialog box has been implemented for adding and configuring widgets on dashboards. The dialog box can be opened from a dashboard page in design mode.
- The **Dashboards** title bar menu item has been removed from all forms. A dashboard can be designed on the dashboard page only.

Platform: Pivot Table Support on Inquiry Forms

New control has been implemented that provides pivot table functionality for the user and can accept generic inquiries as a data source in Acumatica ERP. A pivot table is a program tool that you can use to reorganize and summarize data in a generic inquiry, to view it from different perspectives. This capability is useful when you have large amounts of data in the generic inquiry and you need to build aggregate reports based on this data.



Type	CA	CN	DE	FR	GB	IN	IT	JP	SG	US	Total
Customer	4	2	4	4	3		2	2		77	98
Prospect	5		1							3	9
Vendor	5	2	2	2	2	2		1	2	58	76
Total	14	4	7	6	5	2	2	3	2	138	183

Figure: Number of Business Accounts by Country and Account Type

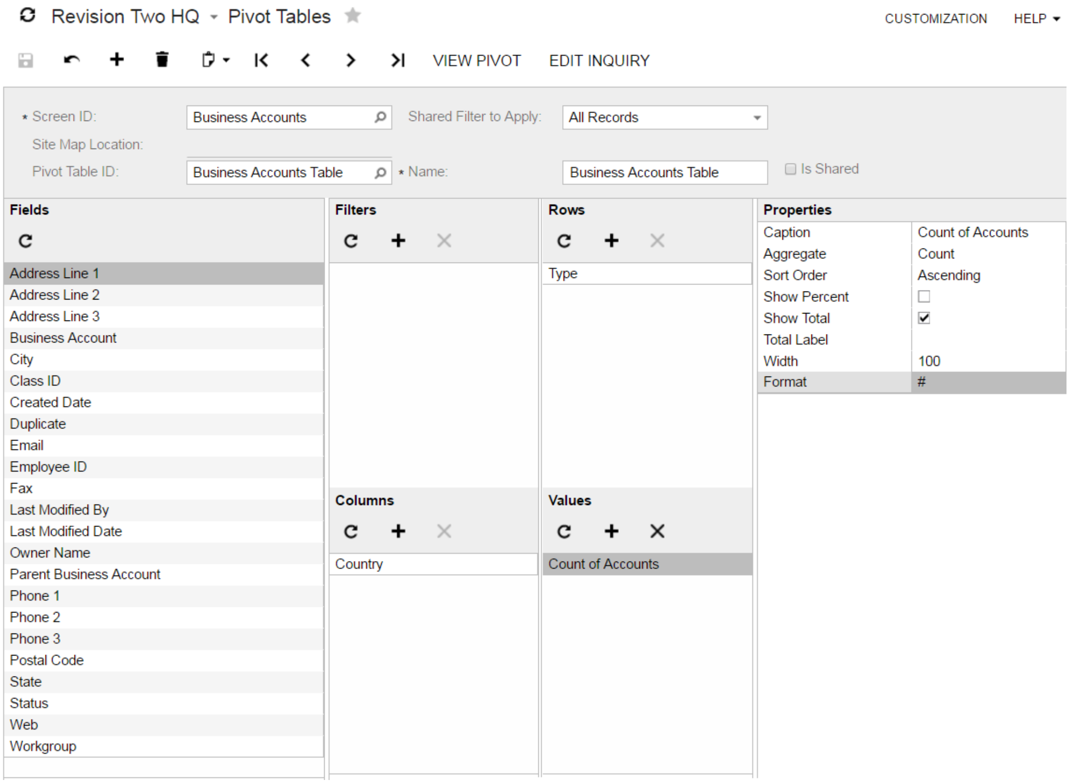
Among other functions, a pivot table can automatically sort, count, total, or give the average of the data stored in a generic inquiry, displaying the results in a separate table that shows the summarized data. You can configure and change the structure of pivot tables simply by dragging and dropping fields. For details, see *Pivot Tables in Acumatica ERP*.

The data for pivot tables is stored separately from the data for generic inquiries in the system database. This makes it possible for users to work with pivot tables even if the generic inquiry on which they are based is unavailable for some reason.

Users may open configured pivot tables as follows:

- Through the site map, if the pivot table was added there during configuration.
- From the generic inquiry form, by using the **Pivot Tables** menu item on the form title bar, which appears when at least one pivot table is created for the particular generic inquiry.

The new *Pivot Tables (SM.20.80.10)* form has been added to the site map where you can configure pivot tables for users. (See the following screenshot.)



* Screen ID:	Business Accounts	Shared Filter to Apply:	All Records
Site Map Location:			
Pivot Table ID:	Business Accounts Table	* Name:	Business Accounts Table

□ Is Shared

Fields

Address Line 1
Address Line 2
Address Line 3
Business Account
City
Class ID
Created Date
Duplicate
Email
Employee ID
Fax
Last Modified By
Last Modified Date
Owner Name
Parent Business Account
Phone 1
Phone 2
Phone 3
Postal Code
State
Status
Web
Workgroup

Filters

Rows

Type

Columns

Country

Values

Count of Accounts

Properties

Caption	Count of Accounts
Aggregate	Count
Sort Order	Ascending
Show Percent	□
Show Total	☑
Total Label	
Width	100
Format	#

Figure: The *Pivot Tables* Form

To configure a pivot table, you need to perform the following steps on the *Pivot Tables* form:

1. You open the form in one of the following ways:
   - From the generic inquiry form that will be used as a data source for the pivot table: On the form title bar, you click **Customization** > **Pivot Tables**.
   - From the site map: You navigate to **System** > **Customization** > **Manage** > **Pivot Tables**.
2. In the **Screen ID** box, you select a generic inquiry from the list of existing generic inquiries. (If you opened the *Pivot Tables* form from the generic inquiry form, the inquiry is selected automatically.)
3. By dragging the fields to the appropriate panes of the form, you select the generic inquiry fields that will be used in rows, in columns, and in the data area (the **Values** pane) of the pivot table; you also select the fields that should be used for filtering by dragging and dropping them to the **Filters** pane on the form.
4. Optional: You specify the display properties for the generic inquiry fields that are used in the pivot table, such as a custom caption, sorting type, aggregate functions, and cell width.
5. Optional: In the **Site Map Location** box in the Summary area of the form, you specify the position of the created pivot table in the Acumatica ERP site map if you want users of your Acumatica ERP instance to access it in the site map.
6. Optional: You share the pivot table within your Acumatica ERP instance by selecting the **Is Shared** check box if you want other users of Acumatica ERP to use it.
7. Optional: In the Summary area of the form, you select a shared filter that will be applied to the pivot table.

Platform: Localization Improvements

Acumatica ERP provides functionality that you can use to localize the system in multiple languages. You can maintain the user interface and wikis in multiple languages if you have multiple locales. Generally, localization includes the use of locale-specific settings and the translation of the strings used on the application interface.

In Acumatica ERP 6, this functionality has been expanded to cover the ability to translate user input to multiple languages, store translations in the database, and to apply locale-specific settings to reports and notifications. For details, see *Managing Locales and Languages*.

Multilingual User Input

You can translate user input to multiple languages and store translations in the database. For example, you can enter descriptions of General Ledger accounts in multiple languages and then print a localized version of the trial balance report with the descriptions of the accounts in the language of the locale you are currently signed with.

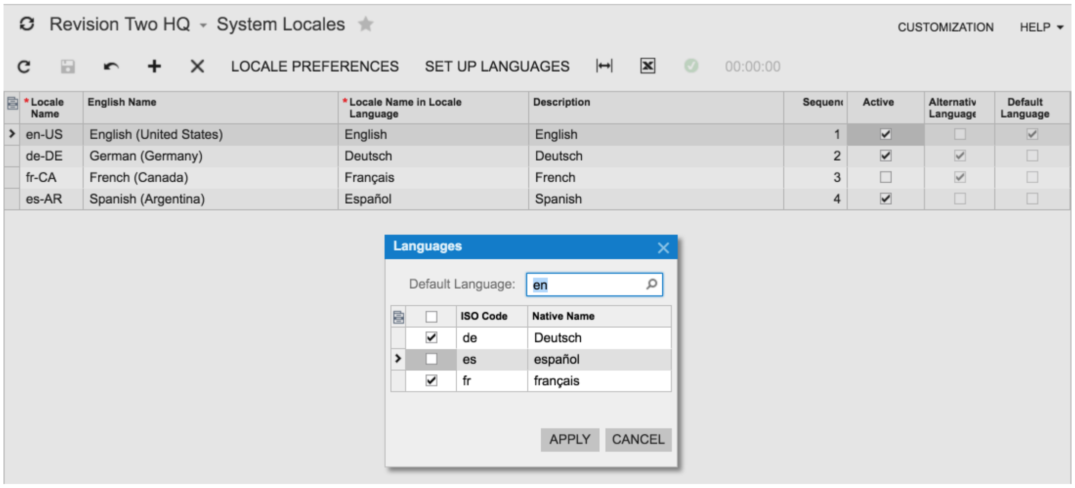
The system retrieves the ISO codes of languages from the locales (active and inactive) you have added on the *System Locales (SM.20.05.50)* form and creates new entities—languages that are each associated with a locale or multiple locales. This is the initial list of languages that you will use to define a set of languages to translate user input to.

: The system associates multiple locales with a language if the identifiers of these locales contain the same ISO code of a language. For example, the `en-US`, `en-GB`, `en-AU`, and similar locales are all associated with the language whose ISO code is `en`.

You define the list of languages to be used in user input translations by clicking the **Set Up Languages** button on the toolbar of the *System Locales* form. In the **Languages** dialog box that opens, you specify the languages in which you are going to localize values and define the default language. A language can be marked as the default language, an alternative language, or a language that is not used in the localization of user input. Specifying a default language enables the multi-language support for the text boxes in the system. The default and alternative languages form the list of languages for which you can provide translations.

: Once you set up languages, in the table on the *System Locales* form, you can view which locales are associated with languages that are the default, an alternative, or not used in localization by adding to the table the **Alternative Language** and **Default Language** columns, which are hidden by default.

In the screenshot below, you can see that the English language is marked as the default language, German and French are alternative languages, and Spanish is not involved in localizing user input. Note that in this case, each locale is associated with one language. The `fr-CA` locale is inactive, but you can mark French language as an alternative or the default. Once you activate the locale, the associated language can be used in localizing user input.



*Locale Name	English Name	*Locale Name in Locale Language	Description	Sequence	Active	Alternative Language	Default Language
en-US	English (United States)	English	English	1	☑	□	☑
de-DE	German (Germany)	Deutsch	Deutsch	2	☑	☑	□
fr-CA	Français (Canada)	Français	French	3	□	☑	□
es-AR	Spanish (Argentina)	Español	Spanish	4	☑	□	□

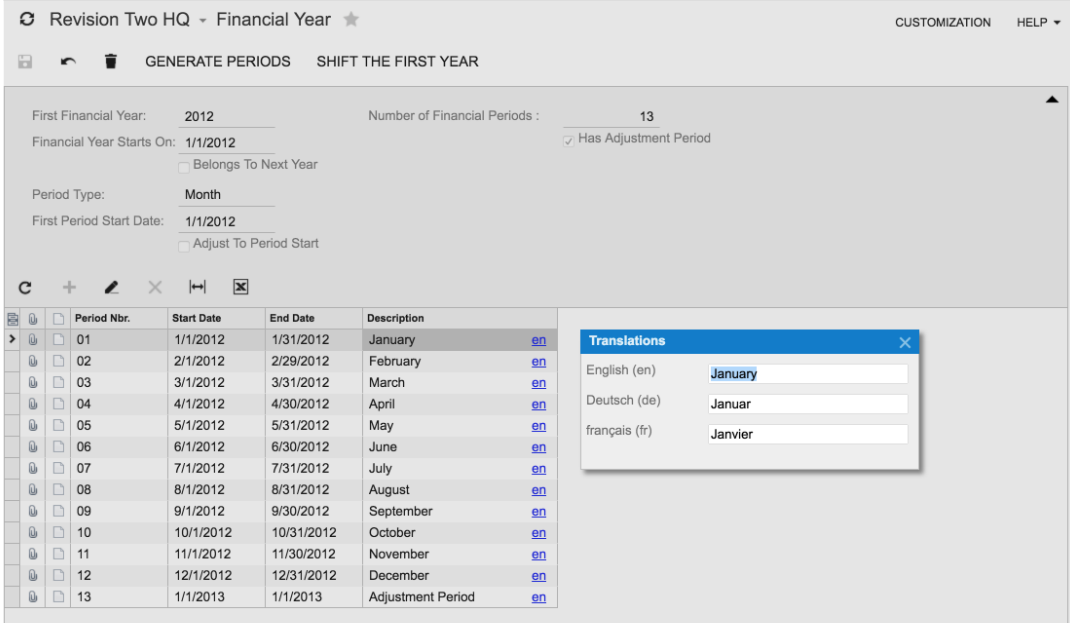
Languages

Default Language: en

□	ISO Code	Native Name
☑	de	Deutsch
□	es	español
☑	fr	français

Figure: Languages box

Once you have finished, you can add translations to the boxes with the multi-language support, regardless of the locale you are currently signed in with. That is, the link with the ISO language code is displayed next to multiple text boxes in the system. Clicking this link invokes the **Translations** dialog box, as shown in the following screenshot.



First Financial Year: 2012
Financial Year Starts On: 1/1/2012
□ Belongs To Next Year
Period Type: Month
First Period Start Date: 1/1/2012
□ Adjust To Period Start
Number of Financial Periods : 13
☑ Has Adjustment Period

Period Nbr.	Start Date	End Date	Description	
01	1/1/2012	1/31/2012	January	en
02	2/1/2012	2/29/2012	February	en
03	3/1/2012	3/31/2012	March	en
04	4/1/2012	4/30/2012	April	en
05	5/1/2012	5/31/2012	May	en
06	6/1/2012	6/30/2012	June	en
07	7/1/2012	7/31/2012	July	en
08	8/1/2012	8/31/2012	August	en
09	9/1/2012	9/30/2012	September	en
10	10/1/2012	10/31/2012	October	en
11	11/1/2012	11/30/2012	November	en
12	12/1/2012	12/31/2012	December	en
13	1/1/2013	1/1/2013	Adjustment Period	en

Translations

English (en): January
Deutsch (de): Januar
français (fr): Janvier

Figure: Translations dialog box

You can also upload translations from a file for those entities (such as account classes, ledgers, and financial periods) that support integration with Excel.

You can also import values to these text boxes and export values from them by using integration scenarios. For details, see *Multi-Language Fields in Import and Export Scenarios*.

For the list of boxes with multi-language support, see *Boxes that Have Multi-Language Support*.

You can expand this list by means of customization. Also, if the `PXDBString` attribute for a box from the list was customized in the earlier versions, after upgrade to Acumatica ERP 6 the box will not have multi-language support. For details, see *How to Provide Multi-Language Support for a Field*.

Localized Reports, Documents, and Notifications

The **Locale** box has been added to report forms. If you specify a locale in this box, the system prepares the report in the locale language if the report strings were translated to this language. Also, if you have set up multilingual user input and provided translations for fields that supply data to the report, the system will replace report data with translations in the language that is associated with the selected locale.

For example, suppose that you have translated the strings of the *Trial Balance Summary (GL.63.20.00)* report for some French locale (for example, `fr-CA` or `fr-FR`) and entered translations of the GL account descriptions in French. If you sign in with a locale associated with the English language (for example, `en-US` or `en-UK`) and you select a French locale on the report form, the system prints the fully localized report with the strings of the report (headers and footers) and with the GL account descriptions in French.

The **Locale** box has been added to notification templates. If you specify a locale in this box, the system applies the locale-specific settings to the notification template. Also, if you have set up multilingual user input and provided translations for fields that supply data to placeholders in a notification template, the system will replace notification placeholders with translations in the language that is associated with the selected locale.

Also, the **Locale** box has been added to the following forms:

- *Customer Classes (AR.20.10.00)*
- *Customers (AR.30.30.00)*
- *Vendor Classes (AP.20.10.00)*
- *Vendors (AP.30.30.00)*

You can associate a locale with a customer or vendor. If you do, when you create a document for the customer or vendor that includes data populated by the system from localized fields, the system will use the translations of these values in the language that is associated with the locale specified for the customer or vendor specified in the document. For example, suppose that you associated a customer with some French locale (for example, `fr-CA` or `fr-FR`) and entered translations of inventory item descriptions in French. When you create an invoice on the *Invoices and Memos (AR.30.10.00)* form, select the customer, and add an inventory item, the system populates the **Transaction Descr.** column with the item description in French. When you release the invoice, the same description in French is posted to the ledger as the GL transaction description.

For the configuration steps and implementation details, see *Usage of User Input Translations*.

Changes to the Database and API

The `PXDBLocalizableString` attribute have been introduced for a text field. You should use the attribute to declare a text field as a multi-language one in the data access class. For details, see *How to Provide Multi-Language Support for a Field*.

The following new attributes have been introduced in API with the localization improvements. For more information on these attributes, see *Working with Multi-Language Fields*.

Attribute	Description
`PXDBLocalizableString`	Indicates that a field can have values in multiple languages. You can use the methods of the corresponding `PXDBLocalizableStringAttribute` class to obtain the values of multilingual fields in specific languages.
`PXLocalizableDefault`	Specifies the default value of a multilingual field.

Changes to the User Interface

The following changes to the user interface have been introduced with the localization improvements.

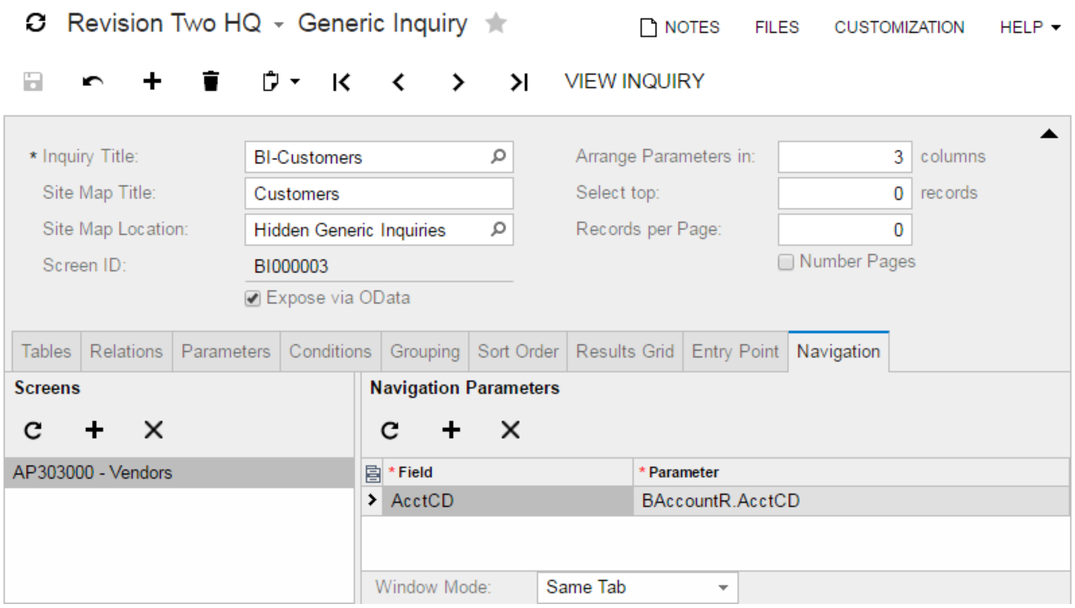
Area/Form	Element	Change	Description
System Locales (SM.20.05.50)	The **Set Up Languages** button The **Alternative Language** and **Default Language** columns (hidden by default)	Added to the form toolbar Added to the table	Clicking the button invokes the **Languages** dialog box, which allows you to form the list of languages to be used in localizing user input. The check boxes in the columns show whether a locale is associated with a language that is the default, an alternative, or not used in localization.
Report forms	The **Locale** box	Added to the Template area	A locale that you select here indicates to the system that the report should be prepared in the language associated with the selected locale. The value in the box overrides a locale specified for the report in the Report Designer (if any). The box is displayed if there are multiple active locales in the system.
Notification Templates (SM.20.40.03)	The **Locale** box	Added to the Summary area	A locale specified here indicates to the system that the region-specific settings of the locale should be applied to a notification and the placeholders in the notification body should be replaced with the data translated to the language associated with the selected locale.
Customer Classes (AR.20.10.00) *Customers (AR.30.30.00)* *Vendor Classes (AP.20.10.00)* *Vendors (AP.30.30.00)*	The **Locale** box	Added to the Summary area	A locale specified here defines the language in which vendors or customers prefer to receive documents. The box is displayed if there are multiple active locales in the system.
Multiple text boxes in the system	A link with an ISO code of a language	Displayed on the right of boxes, if multilingual user input is set up in the system.	If you click the link, the system brings up the **Translations** dialog box, where you can enter translations of the box value. For details, see *Translations Dialog Box*.

Platform: Improved Navigation from Generic Inquiry Screens

In Acumatica ERP 6, navigation from generic inquiry screens has been significantly improved. In previous versions of Acumatica ERP, if a user clicked the link on a generic inquiry form, the system could navigate to only the data entry form for the selected record. Now you can specify which form, inquiry, or report should be opened by the link.

Specifying the List of Screens to Use for Navigation

On the **Navigation** tab, which has been added to the *Generic Inquiries (SM.20.80.00)* form, you can specify the list of screens to be used for navigation from the inquiry. (See the following screenshot.)



* Inquiry Title: BI-Customers
Site Map Title: Customers
Site Map Location: Hidden Generic Inquiries
Screen ID: BI000003
☑ Expose via OData
Arrange Parameters in: 3 columns
Select top: 0 records
Records per Page: 0
□ Number Pages

Screens

AP303000 - Vendors

Navigation Parameters

* Field	* Parameter
AcctCD	BAccountR.AcctCD

Window Mode: Same Tab

Figure: The Navigation tab of the *Generic Inquiries* form

In the **Screens** pane (lower left), you can specify the list of screens to be used for navigation from the inquiry.

In the **Navigation Parameters** pane (lower right), for each screen, you can specify navigation parameters. You select a key field of the screen as the **Field** and select an inquiry parameter to be passed to this field as the **Parameter**. In the **Window Mode** box, you can select the way to open the screen: in the same browser tab instead of the inquiry (the default mode), in a new browser tab, or in a new pop-up window.

Selecting the Screen to Navigate To

On the **Result Grid** tab of the *Generic Inquiries* form, you can specify the form to navigate to for each column of the inquiry form. The following columns have been added to the tab instead of the **Suppress Navigation** column:

- **Default Navigation**: A check box that indicates (if selected, which is the default value) that the field value can be a link to the default entry form, if any. If the check box is cleared, the field value can be a link to the form selected in the **Navigate To** box.
- **Navigate To**: A form specified on the **Navigation** tab that the user can open by clicking the link in the column.

If the **Default Navigation** check box and the **Navigate To** box are cleared, the hyperlink is not displayed in the field.

Configuring an Inquiry to Replace an Entry Form

Now on the **Entry Point** tab, when you select an entry form as the **Entry Screen** to replace the selected entry screen with the inquiry, this form is automatically added to the **Navigation** tab. With the **Replace Entry Screen with this Inquiry in Menu** check box selected, the entry form always opens in the same browser tab when a user is adding a new record from this inquiry or viewing the details of an existing record from the inquiry.

On the **Navigation Parameters** section of the **Navigation** tab, you can modify the inquiry parameters that should be passed to the key fields of the entry form.

Reviewing User Interface Changes

The user interface changes that have been made to the *Generic Inquiry* form since previous versions are listed in the following table.

Area	Element	Change	Description
Summary area	The **Screen ID** box	Added	The system automatically generates the screen identifier of the inquiry, when you save the inquiry for the first time.
Results Grid tab	The **Default Navigation** column	Added	You select this check box to make the field a link to the default form, if any.
Results Grid tab	The **Navigate To** column	Added	You specify a screen on the **Navigation** tab to make the field a link to this screen.
Results Grid tab	The **Suppress Navigation**column	Deleted	Now to suppress navigation, you should clear the **Default Navigation** check box and the **Navigate To** box.
Entry Point tab	The **Open Entry Screen as Pop-Up by Default** check box	Deleted	Now the entry screen is always opened in the same browser tab.
Entry Point tab	The **Navigation Parameters** table	Moved	You can specify navigation parameters on the **Navigation** tab.
Navigation tab	The entire tab	Added	On this tab, you can specify the list of forms to be used for navigation from the inquiry.

For details on configuring navigation for generic inquiries, see *Generic Inquiry* and *To Configure a Generic Inquiry to Replace an Entry Form*.

API: New Version of System Contract in the Contract-Based API

Acumatica ERP provides a contract-based web services SOAP API that external systems can use to get data records from Acumatica ERP, process these records, and save new or updated records to Acumatica ERP. For each endpoint that the contract-based API provides, a contract of the endpoint defines the entities, with their actions and fields, that are available through the endpoint and the methods that you can use to work with these entities. In Acumatica ERP 5.3, one version of the system contract was available, which was used by the system endpoint *Default/5.30.001*.

In Acumatica ERP 6, Version 2 of the contract has been introduced, which provides the following new features, which are described in detail below:

- Access to all fields of an entity, including those that are not defined in the contract
- Selection of particular fields whose values are to be returned from Acumatica ERP

You can work with the new version of the contract through the system endpoint *Default/6.00.001*.

Access to All Fields of an Entity

In Acumatica ERP 5.3, through the contract-based web services API, you could work with only the fields that are defined in the contract of the endpoint and the custom fields that are defined in a customization project (through the special `CustomFields` property of the needed entity). If you needed to work with the fields that are not defined in the contract of an endpoint, you created a custom endpoint and add the needed entities, fields, and actions to it.

In Acumatica ERP 6, through the contract-based web services API, you can work with all fields, including those that are not defined in the contract, without creating a custom endpoint. To work with the fields that are not defined in the contract of an endpoint, you should use the `CustomFields` property of the needed entity. In Version 2 of the contract, this property provides access to both the predefined elements on an Acumatica ERP form that are not included in the entity definition and the elements that were added to the Acumatica ERP form in a customization project. However, this approach has a drawback: No compatibility guarantees are provided for the fields that are not defined in the contract. That is, if your code works with a field that is not defined in the contract and this field has been removed or modified in a new version of Acumatica ERP, your code will be broken and will require an update.

To get or set the value of an element that is not included in the entity definition, you should use the `CustomFields` property of the entity that contains this element. To work with the needed element in the `CustomFields` array, you specify the values of the `FieldName` and `ViewName` properties of the `CustomField` object of the needed type. In the `ViewName` property, you specify the name of the data view that contains the element, and in the `FieldName` property, you specify the name of the element. The code example below illustrates the use of the `CustomFields` property.

```
Customer newCustomer = new Customer
{
  CustomerID = new StringValue { Value = "TEDSMITH" },
  CustomerName = new StringValue { Value = "Ted Smith" },

  //Specify the values of the custom fields
  MainContact = new Contact
  {
    CustomFields = new[]
    {
      new CustomStringField
      {
        FieldName = "UsrPersonalIDType",
        ViewName = "DefContact",
        Value = new StringValue { Value = "Passport" }
      },
      new CustomStringField
      {
        FieldName = "UsrPersonalID",
        ViewName = "DefContact",
        Value = new StringValue { Value = customerPersonalID }
      },
      new CustomBooleanField
      {
        FieldName = "UsrCreditRecordVerified",
        ViewName = "DefContact",
        Value = new BooleanValue { Value = true }
      }
    }
  }
};
```

For more details on the `CustomFields` property, see *CustomFields Property (Contract Version 2)*.

Selection of Particular Fields to Be Returned

In Acumatica ERP 5.3, the only way to restrict the number of fields that are returned when you request records from Acumatica ERP was to specify the value of the `returnFullEntities` parameter of the `GetList()` method. By using this parameter, you could specify whether the detail lines of master-detail records should be returned.

In Acumatica ERP 6, you can specify which particular fields, linked objects, and details of entities you want to obtain when you request records from Acumatica ERP.

To specify the needed fields, linked objects, and details, you should use the `ReturnBehavior` property of the needed entity, which is available in Version 2 of the contract. By using the `ReturnBehavior` property, you can specify the following groups of fields whose values should be returned:

- No fields of the entity.
- Only the system fields—that is, the fields of the `Entity` class, such as `ID`, `RowNumber`, and `Note`.
- Only the specified fields of the entity and the system fields. You can specify the values to be returned or skipped by using the `*Return` and `*Skip` classes of the needed value type, such as `StringReturn` and `StringSkip`.
- All fields of the entity that are defined in the contract.

For more details on the `ReturnBehavior` property, see *ReturnBehavior Property (Contract Version 2)*.

: In endpoints with Contract Version 2, the `GetList()` method does not have the `returnFullEntities` parameter. If you want to restrict the number of returned fields, you should use the `ReturnBehavior` property of the needed entities.

API: Endpoint Extensions in the Contract-Based API

When you work with Acumatica ERP through the contract-based web services API, you use an endpoint to access the service. You can use either the needed version of the system endpoint that has the predefined API and is available in the product by default, or create a custom endpoint, which you can configure by yourself. In Acumatica ERP 5.3, the only way to create a custom endpoint was to create it from scratch. That is, you had to add all of the needed API elements and maintain this API by yourself.

In Acumatica ERP 6, you can create custom endpoints of contract-based web services API not only from scratch, but also as extensions of existing endpoints. That is, you can use all the API of an existing endpoint without modification and add the entities, fields, and actions that you need. If you create a custom endpoint as an extension of a system endpoint, you are responsible for the maintenance of the API that you add. The API that is inherited from the system endpoint is maintained automatically.

You may need to create an extension of an endpoint if you want to use the entities that are defined in the contract of the existing endpoint but also need some additional entities, fields, and actions in the contract. For example, suppose the contract of the system endpoint with the name *Default* and version *6.00.001* contains the `Address` entity, which includes the following fields: `AddressLine1`, `AddressLine2`, `City`, `Country`, `PostalCode`, and `State`. Suppose that you want to add the new `GPSCoordinates` field to the `Address` entity of the contract and use it with other API of the contract. You cannot edit the contract of the system endpoint; instead, you should create an endpoint that is based on this system endpoint, and add the new `GPSCoordinates` field to the `Address` entity of the contract of the new endpoint.

You create an extension for an endpoint on the *Web Service Endpoints (SM.20.70.60)* form by clicking **Extend Endpoint** on the form toolbar. For details, see *To Extend an Existing Endpoint*. For more information on endpoint extensions, see *Endpoint Extensions*.

API: New Contract-Based REST API

Acumatica ERP 6 introduces the new contract-based representational state transfer (REST) application programming interface (API). By using the contract-based REST API, a client application can create and update data records in Acumatica ERP, obtain records from Acumatica ERP, and process these records.

The REST API uses the same endpoints and contracts as the contract-based SOAP API, which was introduced in Acumatica ERP 5.3. That is, you can work with the same set of entities, fields, and actions that are defined by the contract of the endpoint through the REST API or the SOAP API. For the common information on the contract-based API, see *Configuring the Contract-Based Web Services API*.

One of the main benefits of the REST API is that it is very easy to create and test requests: You don't need any custom libraries or special wrappers as you do for the SOAP API; you can create and test requests directly from your web browser. If your client application works with Acumatica ERP by using the contract-based REST API, in the application, you can use the standard HTTP methods for working with records in Acumatica ERP as follows:

- The `GET` HTTP method with different parameters: To retrieve all available records, the first *N* records from the list, the needed record by key fields, the list of records that satisfy the specified conditions, or a file attached to a record
- The `PUT` HTTP method: To create a record, update a record, or attach a file to it
- The `POST` HTTP method: To perform an action, such as releasing an invoice
- The `DELETE` HTTP method: To delete a record from Acumatica ERP

The records are passed to and from Acumatica ERP in the JavaScript object notation (JSON) format in the HTTP request body.

For information on how to use the REST API to work with records, see *Working with the Contract-Based REST API*.

Other Improvements

Distribution

AC-69003: A landed cost amount that was allocated to partially or fully issued purchase receipts was recorded to the Landed Cost Variance account instead of the Cost of Goods Sold (COGS) account.

Fix Description: A landed cost amount that is allocated to partially or fully issued purchase receipts is always recorded to the COGS account.

Documentation

AC-57837: The User Security chapter in the User Guide has been restructured. All how-to topics have been reviewed and updated.

AC-63288: The Accounts Receivable chapter in the User Guide has been reviewed. The *Managing Invoices and Memos* and *Managing Payments* chapters have been replaced with the following chapters: *Processing AR Documents*, *Processing Invoices*, *Processing Payments and Applications*, and *Processing CC Payments*.

AC-64442: The Payroll Guide for the Payroll customization project has been developed from scratch and published in Help.

AC-67749: The overdue charges process description was refactored: The result of the calculation process is explained, and a table has been added with possible configurations of an overdue charge code.

AC-70296: The navigation of a wiki page has been enhanced with the addition of the **Back To Top** link.

AC-70572: The *Managing Self-Service Portal Wikis* chapter in the User Guide has been reviewed and updated to better reflect the tasks of the Self-Service Portal administrator.

AC-72576: The *System Requirements* topic in the Installation Guide has been updated and rearranged.

Finance

AC-43233: A customer refund applied to a credit memo was not displayed in the list of the credit memo applications on the **Application History** tab of the *Payments and Applications (AR302000)* form.

AC-51019: *Consolidate Posting to GL* was in the list of features on the *Enable/Disable Features (CS100000)* form, although it is a posting option.

Fix Description: The *Consolidate Posting to GL* feature has been removed from the *Enable/Disable Features* form and replaced by the **Generate Consolidated Batches** check box, which is now located on the *General Ledger Preferences (GL102000)* form in the **Posting and Retention Settings** section.

AC-51020: The **Keep Transactions for *n* Periods** settings were not utilized by the system but were available on the module preferences forms, which was confusing for users.

Fix Description: The **Keep Transactions for *n* Periods** box has been hidden from the following forms:

- *General Ledger Preferences (GL102000)*
- *Accounts Payable Preferences (AP101000)*
- *Accounts Receivable Preferences (AR101000)*
- *Inventory Preferences (IN101000)*
- *Time and Expenses Preferences (EP101000)*

- *Currency Management Preferences (CM101000)*

AC-58871: There was no ability to review the documents that were matched to the transactions imported from the bank statements.

Fix Description: The **View Matched Document** button has been added to the table toolbar on the *Import Bank Transactions (CA306500)* form. When you select a matched transaction and click the button, the system navigates to the form, where you can view details of the matched document or the batch for the selected transaction. The button is available for only the matched transactions.

AC-64156: The parameters on all financial report forms have been unified: The same captions are used for user interface elements (such as boxes and check boxes) with the same meaning on all report forms, and user interface elements have been rearranged on these forms.

AC-68817: If the **Validate Document Total on Entry** check box had been selected on the *Accounts Receivable Preferences (AR101000)* form, it was not possible to generate recurring invoices with taxes calculated by Avalara AvaTax.

AC-68820: On the *Generate Recurring Transactions (AP504000; AR504000)* forms in the Accounts Payable and Accounts Receivable modules, the processes copied taxes from the scheduled documents into the recurring documents, instead of recalculating taxes based on the tax settings that were effective on the generated document dates.

AC-68981: On the *Sales Orders (SO301000)* form, a user could enter and save a reference number of the authorization transaction returned by a processing center even if the currency of the order did not match the currency of the cash account associated with the payment method selected in the order. In these cases, the credit card transaction could not be neither captured nor voided, and the sales order itself could not be canceled.

Fix Description: Now the system ensures that the currency of the sales order matches the currency of the cash account.

AC-69067: On the *Run Contract Billing (CT501000)* form, when a user clicked the contract ID hyperlink just after the billing process had been completed, the system opened the *Customer Contracts (CT301000)* form in the new window, but displayed an incorrect contract status in the window.

AC-71949: The default size of the **Bank Transaction Rules** dialog box, which is invoked on the *Process Bank Transactions (CA204550)* form, was not big enough to fit all elements. On the dialog box, uppercase letters were not used to display the name of the rule entered by a user into the **Rule** box.

AC-72275: On the *Cash Sales (AR304000)* and *Quick Checks (AP304000)* forms, the **Transaction Descr.** box did not support localized item descriptions.

Organization

AC-71560: The email body of all emails was not displayed on the Acumatica Partner Portal.

AC-71995: After an upgrade to Version 6 from Version 5.3, the information on the **Summary** tab of the *Employee Time Card (EP305000)* form was empty for some time cards, and the time totals could be aggregated in a different line on the **Summary** tab.

AC-72230: The value of the **Billable Time** element was different for the same employee on the *Employee Time Card (EP305000)* data entry form and on the *Employee Time Card (EP406000)* generic inquiry form.

AC-72381: The Microsoft Outlook Add-in did not ignore inactive contact records when the user searched for an existing contact.

AC-72399: A user could not filter tasks by date on the *Tasks (EP404000)* form due to a performance issue.

AC-73303: When a user changed a class for a customer or a vendor, the attributes of the business account were lost.

System and Platform

AC-60868: In Acumatica ERP 6, Classic mode of application pools in Internet Information Services is no longer supported. Only Integrated mode of the application pool is supported. Support of only Integrated mode simplifies the testing of Acumatica ERP, the deployment of Acumatica ERP, and the upgrade to future versions of ASP.NET.

AC-67964: When a user added mapping for a field with the `PXIntListAttribute` attribute by using the **Populate Fields** dialog box, which is invoked on the *Web Service Endpoints (SM207060)* form, the system generated mapping of the `IntValue` type. As a result, a custom application that used this field generated the *Error in deserializing body of reply message for operation 'Get'* error message when it tried to retrieve any entity of this type.

AC-70122: A custom endpoint was not working if it was located not in the first (*System*) company.

AC-70335: The **DITA** button with its commands has been removed from the *Wiki (SM202005)* form.

AC-70806: The *Project Task cannot be found* error message occurred during the processing of an email that had a link to a project or a project task.

AC-70969: The *Violation of PRIMARY KEY constraint 'CSAnswers_PK'* error message occurred during the upgrade of a particular Acumatica ERP instance to Version 6 from Version 5.3.

AC-71330: A user account for an employee could not be created on the *Employee (EP203000)* form if this user account had been deleted earlier on the *Users (SM201010)* form.

AC-71436: Contacts could not be imported during synchronization with Microsoft Exchange server if a contact display name was empty.

AC-71438: During synchronization with Microsoft Exchange server, some fields could be empty when they come to Acumatica ERP (for example, the **CompleteName** of a contact), which could cause an error message. These empty fields are now processed correctly.

AC-71543: Filters did not work on generic inquiry forms.

AC-71660: When a user tried to export a help article to Microsoft Word, the *Object Reference not set* error message was displayed.

AC-71701: When a user executed a `PUT` request with the *LongSearch* type in the `RowNumber` parameter (that is, the user tried to access a generic inquiry with row limit) the system threw an exception.

Fix Description: The list of system-recognized types has been unified for all methods.

AC-71782: When a user tried to sort by using the **Business Account Name** element in the *Business Accounts (CR3030PL)* generic inquiry form, the *Cannot navigate to the record...* error message was displayed.

AC-71907: The `@ReportDescr` column set parameter, which allows the report description to be displayed in the column set header, was added to reports.

AC-72600: The `SearchTextMark` CSS class can be used to customize the highlighting style for search text.

AC-72624: If a column set of the ARM report used a percentage format like *P* for a calculated column, this column was displayed incorrectly after the report was exported to Microsoft Excel.

AC-72723: Sometimes a user could not clear the **Sync Account** check box on the *Exchange Server Configuration (SM204015)* form.

Improvements Implemented in 5.3 Updates

Improvements that have been introduced in multiple updates of Acumatica ERP 5.3 (listed below) are available in Acumatica ERP6.

AC-49473: The reclassification of GL transactions was implemented. With this functionality, users are able to:

- Reclassify the GL account, subaccount, and branch for posted journal entries.
- Exclude transactions that had been reclassified from the reports and inquiries.
- View the history of changes for each entry that had been reclassified.

The reclassification process generates a new transaction of the *Reclassification* type based on the original GL transaction. The reclassification transaction moves the amount from the wrong GL account, subaccount, or branch to the required one.

Important: Because of changes made to the *Journal Transactions (GL301000)* form as a part of the *Reclassification of GL Transactions* feature, custom applications that interact with this form through the screen-based API might require recompilation with the updated form definition. Applications that rely on the contract-based API will not require any changes.

AC-51423: Support of single sign-on in the mobile application has been implemented. If your Acumatica ERP instance supports single sign-on with identity management systems, such as Google or Microsoft Account, you can use your accounts in these systems to sign in to the Acumatica mobile app on a mobile device. Also, if your Acumatica ERP instance is integrated with Active Directory Federation Services or Azure Active Directory, you can use your domain credentials for signing in to the mobile app.

AC-63276: The ability to add and allocate landed costs to transfers has been added. Users can add landed costs directly to receipts of the *Transfer Receipt* type on the *Purchase Receipt (PO.30.20.00)* form or to bills on the *Bills and Adjustment (AP.30.10.00)* form and allocate them by using the same allocation methods as were developed for purchase receipts. Also, for stock items with the FIFO valuation method, a new cost layer with the date of the transfer receipt has been created.

Known Issues

: The updated list of the known issues is available on the Acumatica Partner Portal at https://portal.acumatica.com/release-notes/known-issues/.

System and Platform

AC-54601: When a user tries to update a detail for an entity by constructing an entity containing `*Search` in master (to find the entity) and `*Search` in detail (to find the detail), the system tries to insert a new detail instead of updating an existing detail.

AC-59992: When a search (through the `Get` or `GetList` command) that contains a restriction on the detail item is issued to the contract-based API, the system first filters the entities according to the restrictions on the entities and then filters the detail in every entity according to the detail restriction.

Workaround: Filter the received list of entities in your client application.

AC-67419: When an entity that is created with the contract-based API contains a space in either a property name or a type name, the system will successfully save it and produce WSDL, but any attempt to use the WSDL causes an error.

Workaround: Use only Latin characters and digits, and no spaces, in property names and type names.

AC-69335: An unclear exception occurs when contract-based API methods are called with a null `EntityImpl` object.

Workaround: If you receive an unclear exception, make sure that the `EntityImpl` object is not null.

AC-69353: The `NullReferenceException` occurs in the `CalculatePackages` function when a client application tries to add a shipment for a product on a date earlier than the requested date.

AC-70779: The *500 Internal Server Error: Object reference not set to an instance of an object* occurs when a user invokes an action without specifying parameters of an action through the REST API.

Workaround: Use the following form of the POST body for the `Invoke` command.

```
{
"entity": {some entity},
"parameters": null
}
```

AC-72946: The rows with the total values (summarized values of other rows) were not shown in the analytical reports if, in the calculation formulas, the references to the rows to be summarized had been used with the “@” sign.

Workaround: The formulas with the `sum` function should be used for calculating the summarized values.

AC-73005: A success status code is returned when a client application user tries to create a shipment with an incorrect payload by using the contract-based API.

Workaround: Double-check that you are using a correct payload when you create creating a shipment by using the contract-based API.

AC-73238: The *CREATE UNIQUE INDEX statement terminated* error message appears after upgrade of an Acumatica ERP database on Microsoft SQL Server to Version 6 if all of the following conditions are true:

- A full text index was not created on a system that has full text search functionality switched on.
- The `pp_DisableFullText` and `pp_EnableFullText` stored procedures are replaced with empty ones.

- A snapshot has been made while the previous two conditions were true.

Workaround: Before you upgrade to Acumatica ERP 6, run the following queries on your database:

- `UPDATE dbo.WikiRevision SET UID = NEWID()`
- `UPDATE dbo.SearchIndex SET IndexID = NEWID()`

AC-73318: The `$expand` parameter is required to receive a custom field of a linked entity in a `PUT` response (linked entity section), even if this entity is specified in the request payload.

AC-73575: After conversion of dashboards that have been created in earlier versions of Acumatica ERP, a table widget may display different data.

Workaround: Check settings of the table widget such as filter settings and adjust them according to your needs.

AC-73638: After conversion of dashboards that have been created in earlier versions of Acumatica ERP, the size of the doughnut chart can become smaller on the widget.

Workaround: Enlarge a widget size.

AC-73640: After conversion of dashboards that have been created in earlier versions of Acumatica ERP, the sorting of sections on a stacked column chart can be different due to new functionality of sorting values.

Workaround: Specify sorting settings according to your needs in the widget properties.

AC-73920: In the analytical reports (ARM), the total values were rounded twice if the rounding had been applied to calculated values in the column cells.