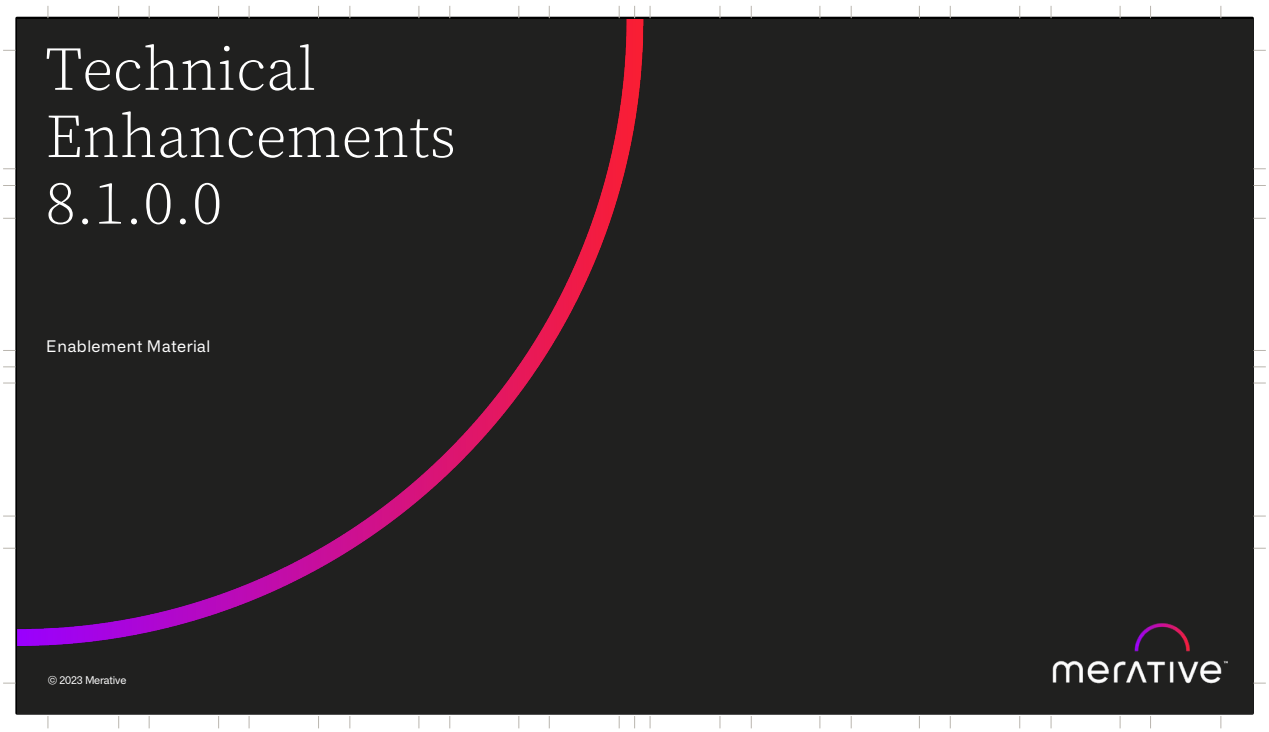


Hello and welcome to this deep dive presentation that introduces the technical enhancements that are available in Merative Social Program Management (SPM) V8.1.0.0.

Agenda

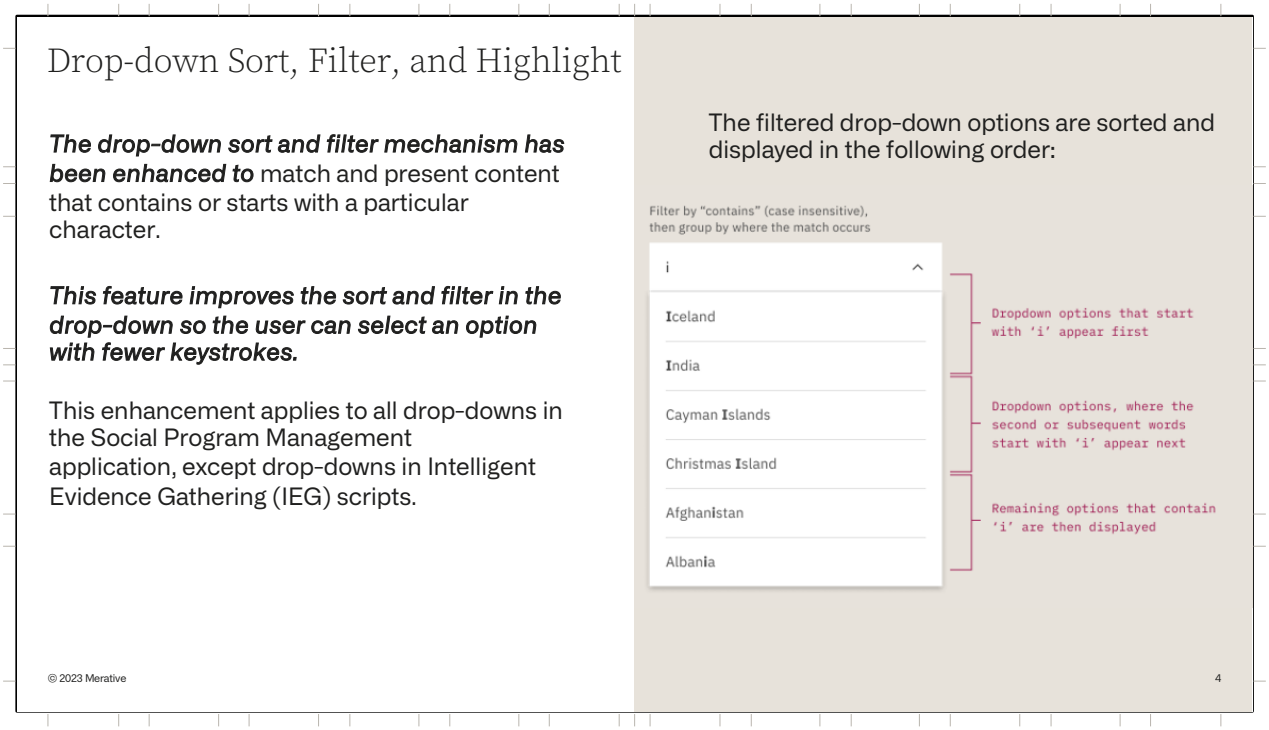
1. Drop-down sort, filter, and highlight
2. Indication of sortable columns in a list
3. Carbon 11 Update
4. Multi-locale support for Work Queues, Quick Links, and Organization Structure
5. New financial hook points

© 2023 Merative

MERATIVE

Drop-Down Sort, Filter, and Highlight

© 2023 Merative



Drop-downs in the caseworker application use a combo-box pattern, which enables caseworkers to filter the list of options based on the characters that they type into the field.

*Before V8.1.0.0, when the user entered one or more characters, the content that **starts** with the characters was shown.*

*In 8.1.0.0, the sorting and filtering logic has been enhanced for non-IEG drop-downs. When the user enters one or more characters, the content that **contains** or starts with the characters is shown, with the most likely option at the top, which allows the user to select options with fewer keystrokes.*

In the 'Country' drop-down sample, the filtered drop-down options are sorted and displayed in the following order:

- First, drop-down options that start with the characters that are entered are displayed, for example, entering 'i' returns Iceland.
- Then, for options that contain more than one word, those that contain a word that starts with the entered characters are displayed, for

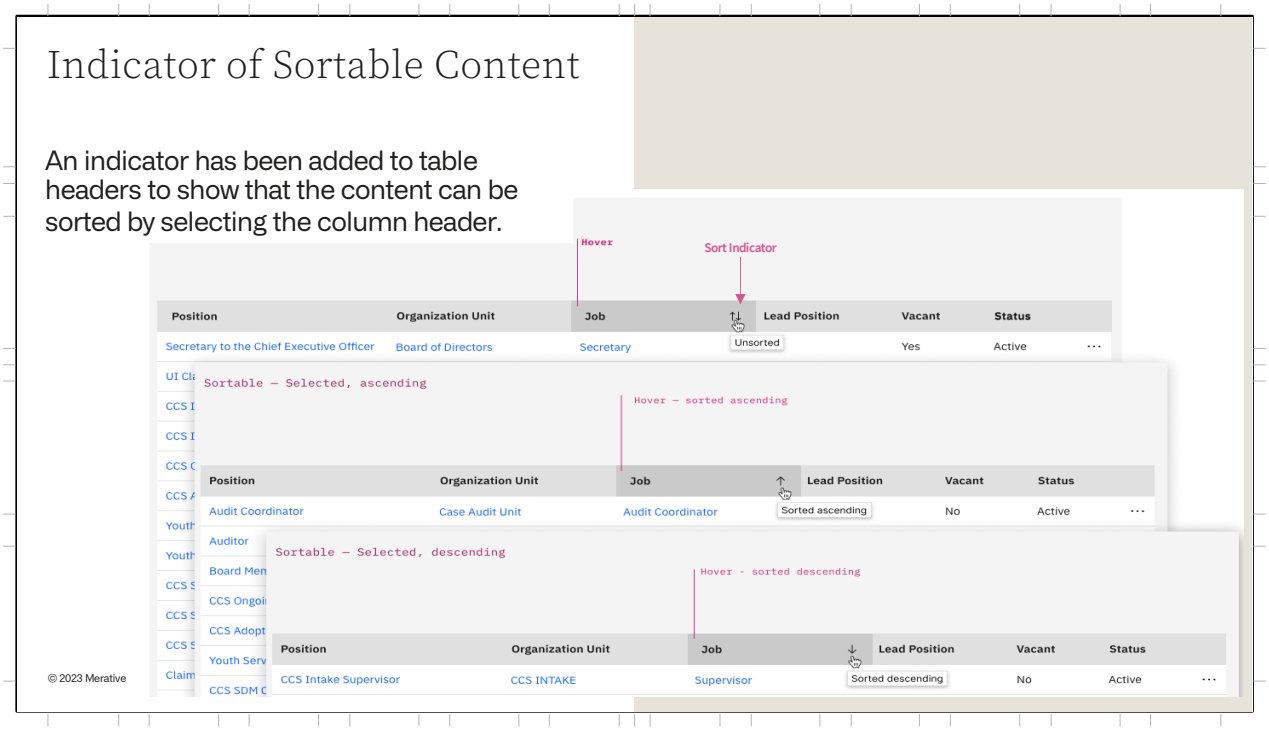
example, entering ‘I’ returns Cayman Island.
- Then, the remaining options that contain the entered characters are displayed, for example, entering 'i' returns Albania.

The characters used to order the content are highlighted to help caseworkers understand why an option was listed in a particular order and why it was included by the filter.

MERATIVE

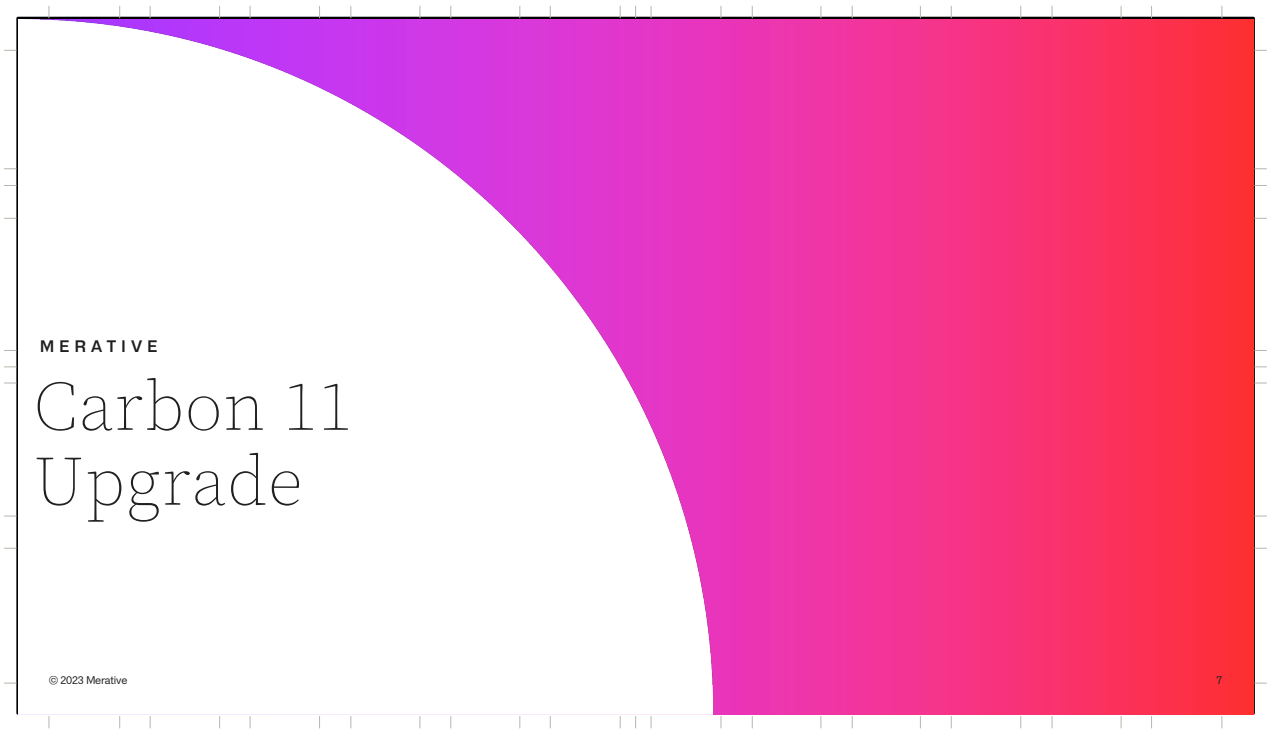
Indication of Sortable Columns on a List

© 2023 Merative



There is now an indicator to a user that the columns on a list page can be sorted by selecting column headers. The indicator is an arrow on the right of the column header. Two arrows (one up and one down) mean that the content is unsorted. Up arrow means the content is sorted in ascending order. Down arrow means the content is sorted in descending order. No arrow means that you cannot sort the content.

There is no change to the actual sorting functionality.


MERATIVE
Carbon 11
Upgrade
© 2023 Merative
7

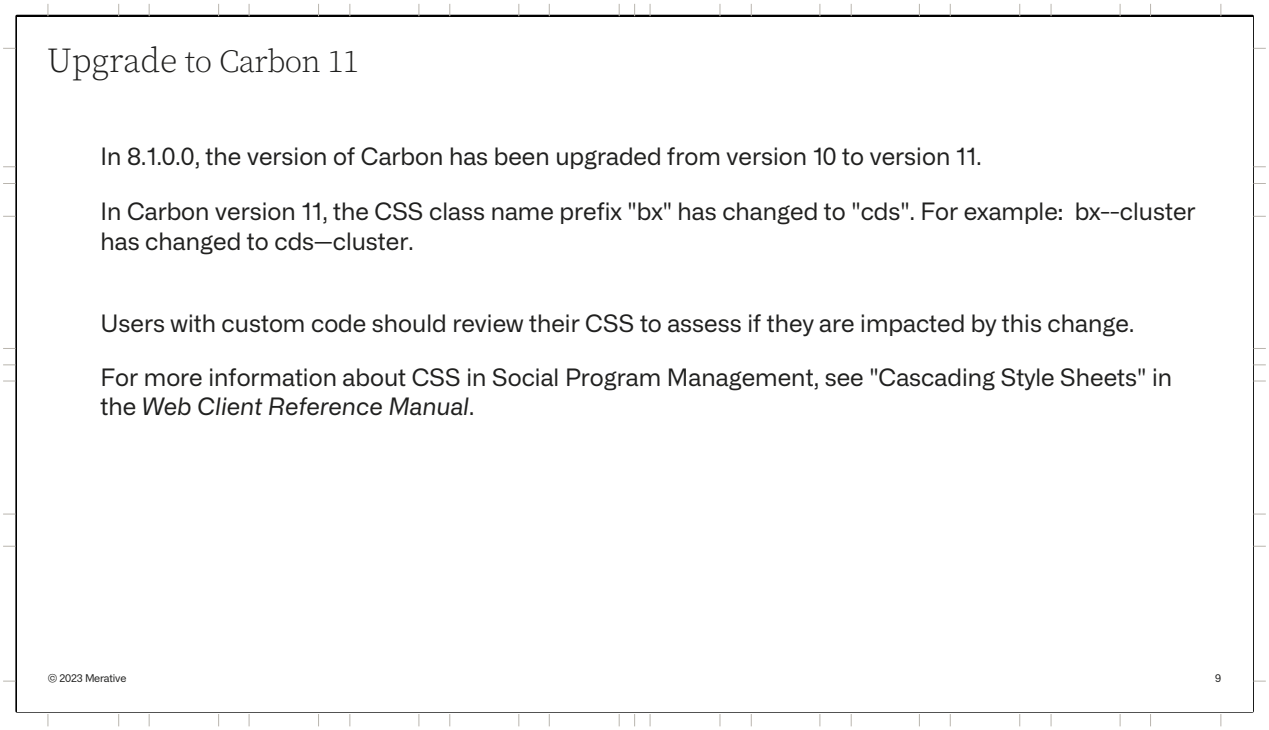
Upgrade to Carbon 11

Carbon v11 focuses on updates for designers and developers including token and prop renaming for ease of use, component API updates for a more predictable developer experience, and introducing CSS grid for accurate layout building in line with the design language.

What's new

- Changes in tokens names to make them easier to use
- Improved theming (enables theming and dark mode support in the future)
- The introduction of CSS Grid
- A 90% decrease in compilation for Styles from Carbon (Moving to Dart Sass , migrating to Sass Modules)
- Updates to the accessibility of components

© 2023 Merative



In 8.1 the version of carbon has been upgraded from version 10 to version 11 .

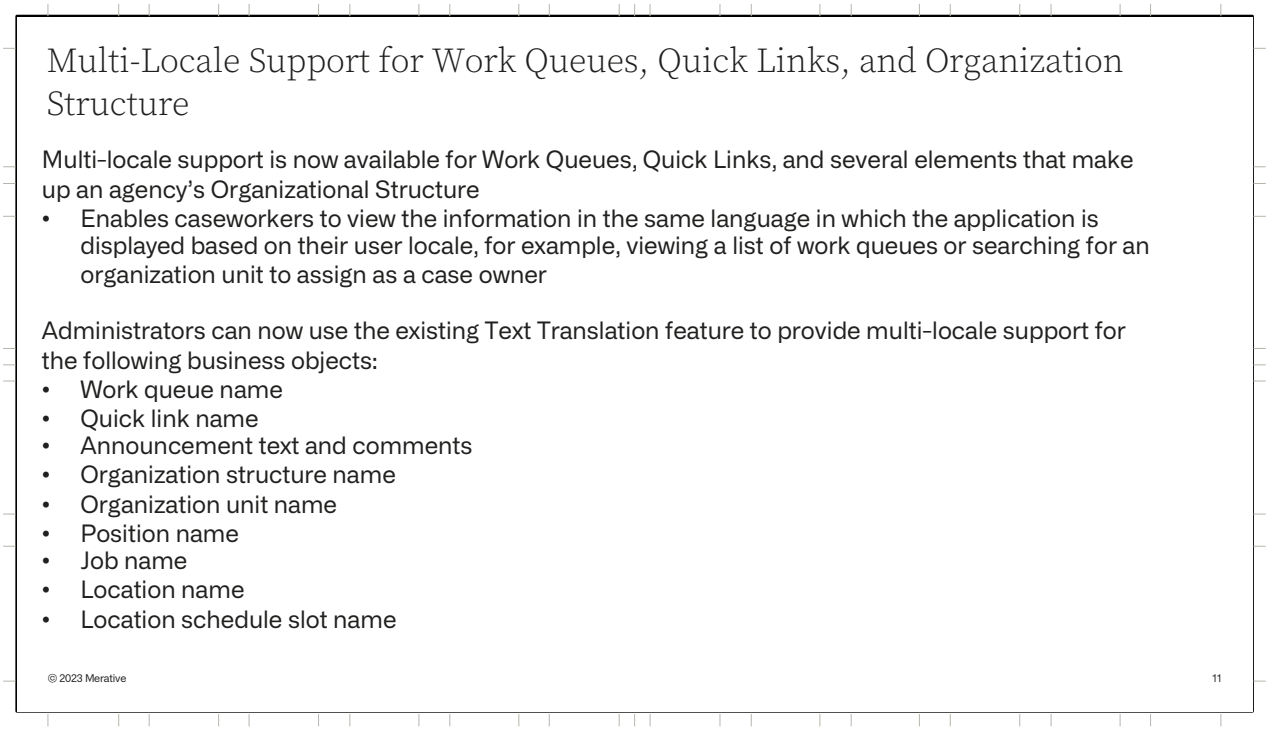
In Carbon version 11, the CSS class name prefix "bx" has changed to "cds". For example: bx--cluster has changed to cds--cluster

Users of custom code (https://www.ibm.com/docs/en/spm/8.0.2?topic=customization-cascading-stylesheets) should review their CSS to assess if they are impacted by this change.

MERATIVE

Multi-Locale Support for Work Queues, Quick Links, and Organization Structure

© 2023 Merative



Multi-locale support is now available for Work Queues, Quick Links, and several elements that make up an agency's Organizational Structure.

- This enables caseworkers to view the information, such as when viewing a list of work queues or searching for an organization unit to assign as a case owner, in the same language in which the application is displayed.

Administrators can now use the existing Text Translation feature to provide multi-locale support for the following business objects:

- Work queue name
- Quick link name
- Announcement text and comments
- Organization structure name
- Organization unit name
- Position name
- Job name
- Location name

- Location schedule slot name

When caseworkers use the language toggle to change their application language, the information is now displayed in the language based on the user's locale.

Technical and Upgrade Considerations

To support multiple locales, the text fields for the business objects listed previously have been re-modeled and coded using the Localizable Text entity.

- Text fields stored on the business entity, such as ***WorkQueue.name***, are now represented as an identifier, such as ***WorkQueue.nameTextID,*** instead of a piece of text in a single locale.
- Identifier relates to the localizable text, which can have multiple associated text translations of the specific text

Text displays in a user's locale where possible; however, if translations are not available a locale fallback mechanism is used:

- The first available translation identified in the following order of priority is displayed.
- If no translations are available, then no text is displayed.

Priority	Translation locale	Example
1	User locale	fr_CA
2	User locale language	fr
3	Default server locale	en_US
4	Default server locale language	en
5	Legacy text if the field is a migrated legacy field	-
6	Blank	-

Note: It is an administrator's responsibility to configure a locale-specific version of each localizable text record and to prioritize the configuration of the final fallback translation.

© 2023 Merative

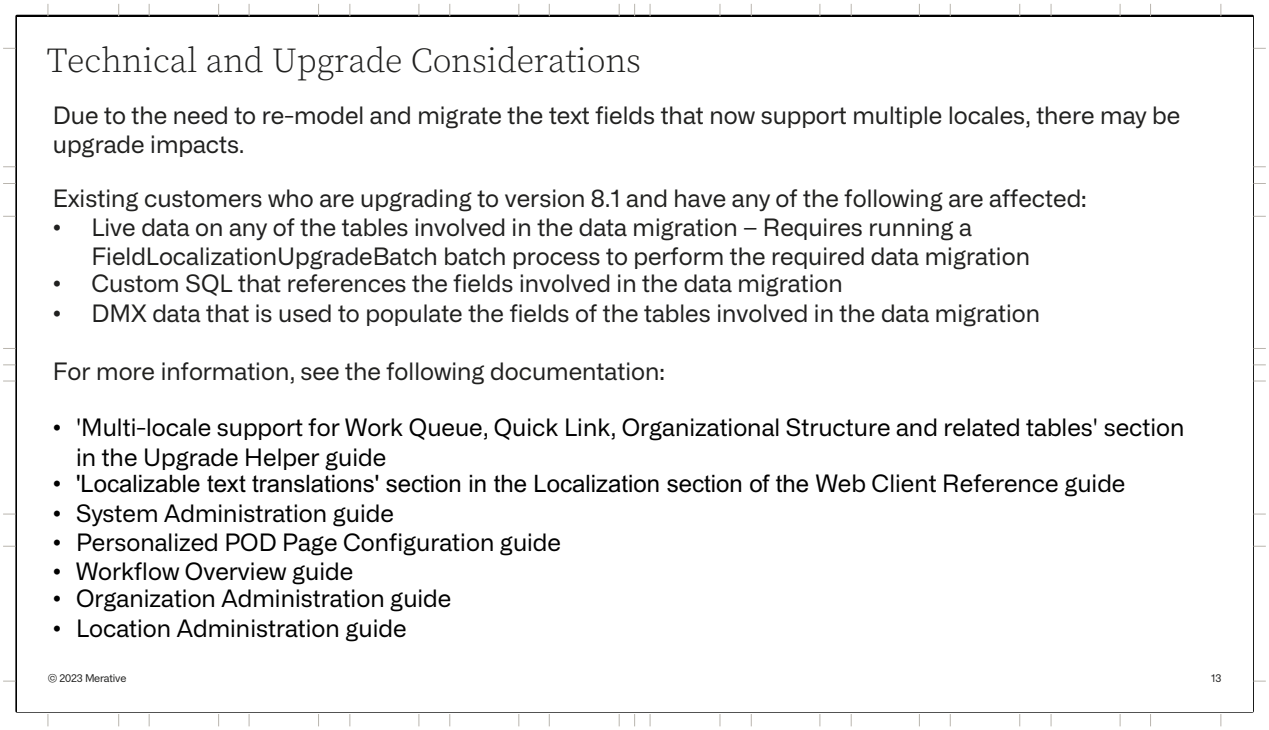
To support multiple locales, the text fields for the business objects listed previously have been re-modeled and coded using the Localizable Text entity.

The text field data that is stored on the business entity, such as WorkQueue.name, is now represented as an identifier, such as WorkQueue.nameText ID, instead of a piece of text in a single locale. The identifier relates to the localizable text, which can have multiple associated text translations of the specific text.

Text displays in a user’s preferred locale where possible. However, if translations are not available in the user's locale, a locale fallback mechanism is used to predictably identify and display the best available alternative translation. For localizable text, the fallback mechanism is based on the JDK class java.util.ResourceBundle.

The first available translation that the fallback mechanism identifies in the order of priority as illustrated on the slide is displayed when it retrieves the display text. If no translations are available in this set, no text is displayed.

Note: In multi-locale environments, it is an administrator's responsibility to configure a locale-specific version of each localizable text record and to prioritize the configuration of the final fallback translation.



Due to the need to re-model and migrate the text fields that now support multiple locales, there may be upgrade impacts.

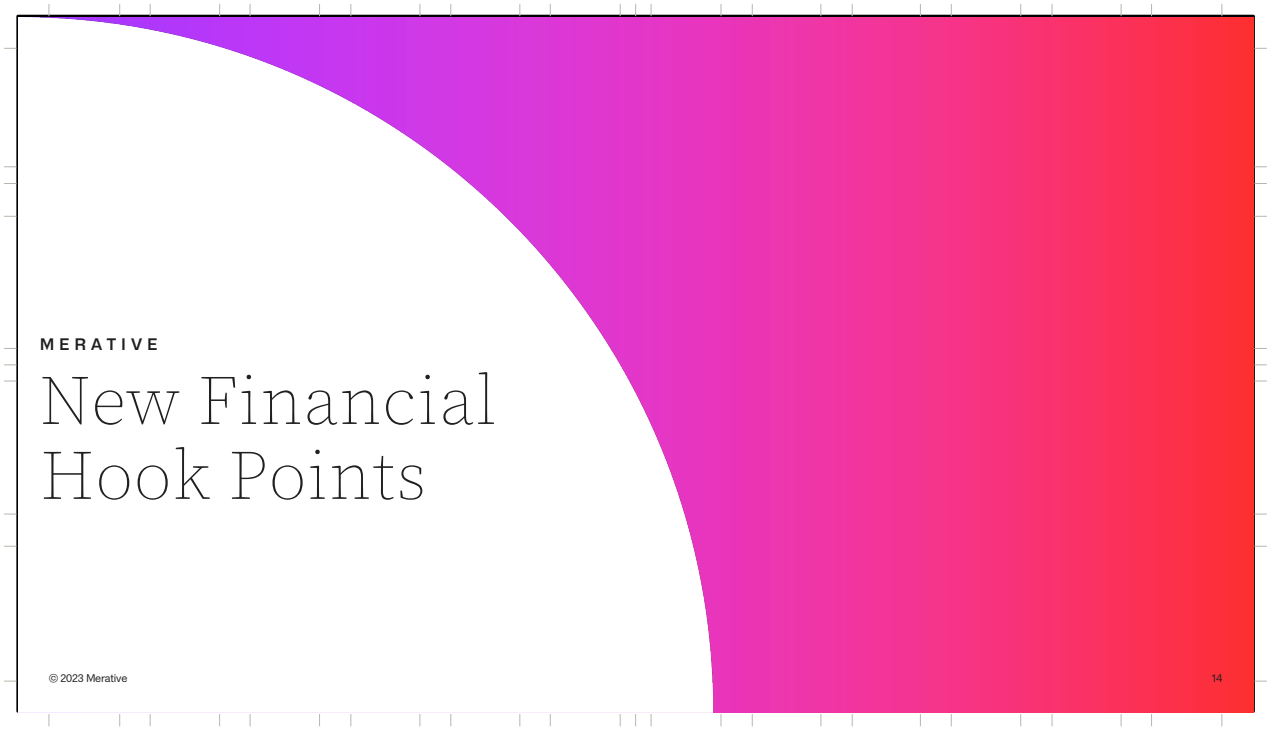
Existing customers who are upgrading to version 8.1 and have any of the following are affected:

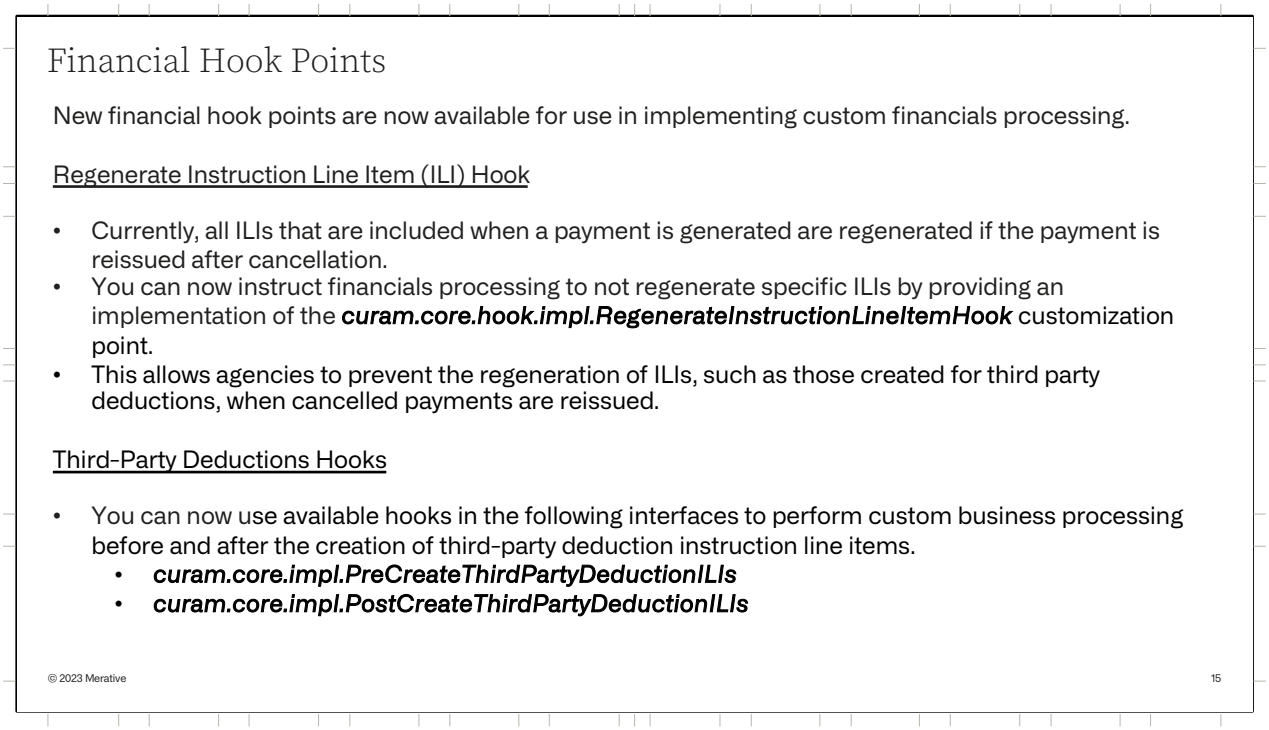
- Live data on any of the tables involved in the data migration. You must run the FieldLocalizationUpgradeBatch batch process to perform the required data migration.
- Custom SQL that references the fields involved in the data migration.
- DMX data that is used to populate the fields of the tables involved in the data migration.

For more information see the following documentation:

- 'Multi-locale support for Work Queue, Quick Link, Organizational Structure and related tables' section in the 'Upgrade Helper' guide
- 'Localizable text translations' section in the Localization section of the Web Client Reference guide
- System Administration guide

- Personalized POD Page Configuration guide
- Workflow Overview guide
- Organization Administration guide
- Location Administration guide


MERATIVE
New Financial
Hook Points
© 2023 Merative
14



New financial hook points are now available for use in implementing custom financials processing in a compliant manner.

Regenerate Instruction Line Item Hook

- Currently, all ILIs that are included when a payment is generated are regenerated if the payment is reissued after cancellation.
- You can now instruct financials processing to not regenerate specific ILIs by providing an implementation of the ***curam.core.hook.impl.RegenerateInstructionLineItemHook*** customization point.
- This allows agencies to prevent the regeneration of ILIs, such as those created for third party deductions, when cancelled payments are reissued.

Third Party Deductions Hooks

- You can now use available hooks in the following interfaces to perform custom business processing before and after the creation of third-party deduction instruction line items .
  - **curam.core.impl.PreCreateThirdPartyDeductionILIs**

- **curam.core.impl.PostCreateThirdPartyDeductionILIs**

Developers can now use the following new financial hook points to implement custom financials processing:

- Third Party Deductions hooks to perform custom business processing before and after the creation of third part deduction Instruction Line Items
- Instruction Line Item post insert hook to implement custom business processing, such as updating custom entities.
- Payment Instrument Post Insert and Post Modify hooks to implement custom business processing, such as updating custom entities
- Regenerate Instruction Line Item hook to skip the regeneration of an ILI when cancelled payments are reissued.

Use the available hooks in the interfaces curam.core.impl.PreCreateThirdPartyDeductionILIs and curam.core.impl.PostCreateThirdPartyDeductionILIs to perform custom business processing before and after the creation of third-party deduction instruction line items. For more information, see the Javadoc for the associated class.

Use the available hook in curam.core.hook.impl.RegenerateInstructionLineItemHook to prevent the regeneration of an instruction line item (ILI) when cancelled payments are reissued.

About This Task

By default, all ILIs that are included when a payment is generated are regenerated if the payment is reissued after cancellation. You can specify that a particular ILI such as an ILI for a third-party deduction, is not regenerated by providing an implementation of the curam.core.hook.impl.RegenerateInstructionLineItemHook customization point. For more information, see the Javadoc for the associated class.

Use the available post insert hook on the curam.core.sl.event.impl.InstructionLineItemEvents abstract class to implement custom business processing, such as updating custom entities.

Use the available post insert and post modify hooks on the curam.core.sl.event.impl.PaymentInstrumentInsertModifyEvents abstract class to implement custom business processing, such as updating custom entities.

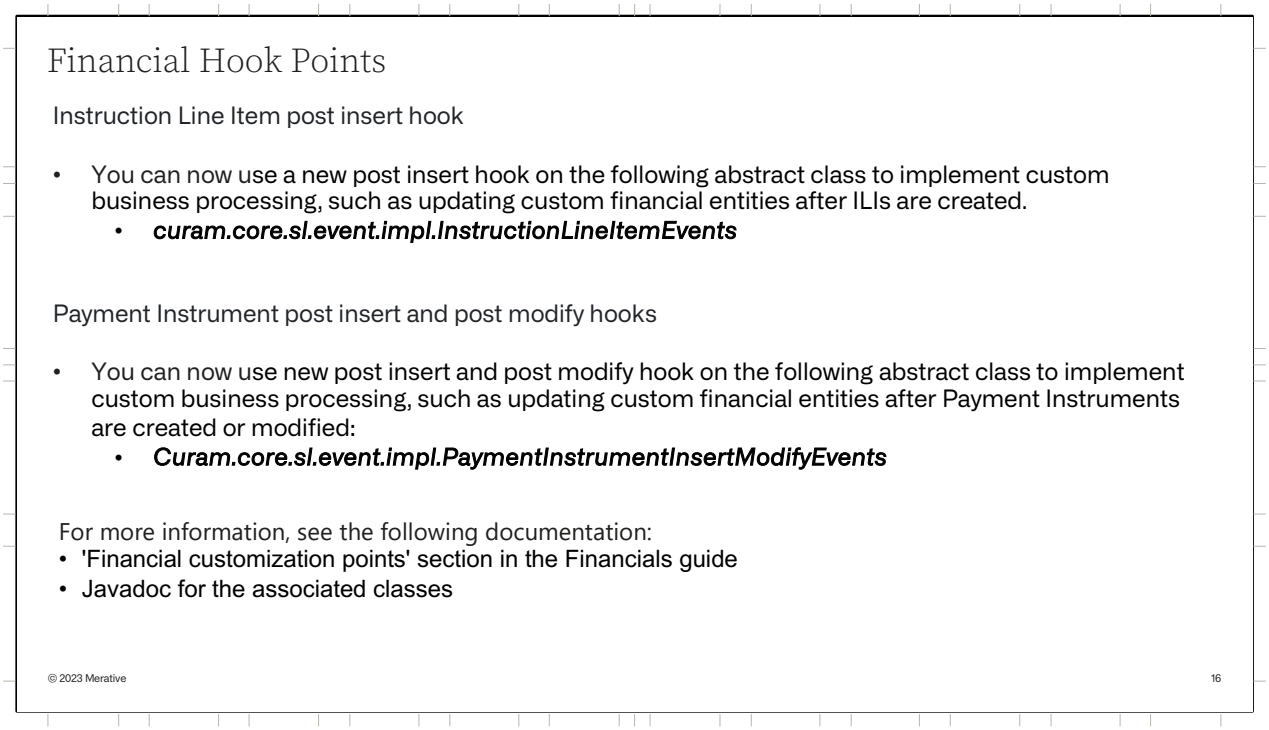
Two new customization points are now available on the curam.core.impl.FinancialHooks abstract class to offset deductions against standalone underpayments.

curam.core.impl.FinancialHooks.isComponentDeductible()
By default, an objective that is associated with an underpayment financial component on the original benefit case is not deductible. You can specify that a standalone underpayment objective is deductible by providing an implementation of the curam.core.impl.FinancialHooks.isComponentDeductible() customization point.

curam.core.impl.FinancialHooks.deductFromStandaloneUnderpayment()
You can also instruct financials processing to apply a specified case deduction to a standalone underpayment by providing an implementation of the curam.core.impl.FinancialHooks.deductFromStandaloneUnderpayment() customization point.

Provide an implementation of the core.impl.FinancialHooks.isComponentDeductible() customization point to specify that a standalone underpayment objective is deductible.

Provide an implementation of the curam.core.impl.FinancialHooks.deductFromStandaloneUnderpayment() customization point to instruct financials processing to apply a specified case deduction to a standalone underpayment.



Instruction Line Item post insert hook

- You can now use a new post insert hook on the following abstract class to implement custom business processing, such as updating custom financial entities after ILIs are created.
  - **Curam.core.sl.event.impl.InstructionLineItemEvents**

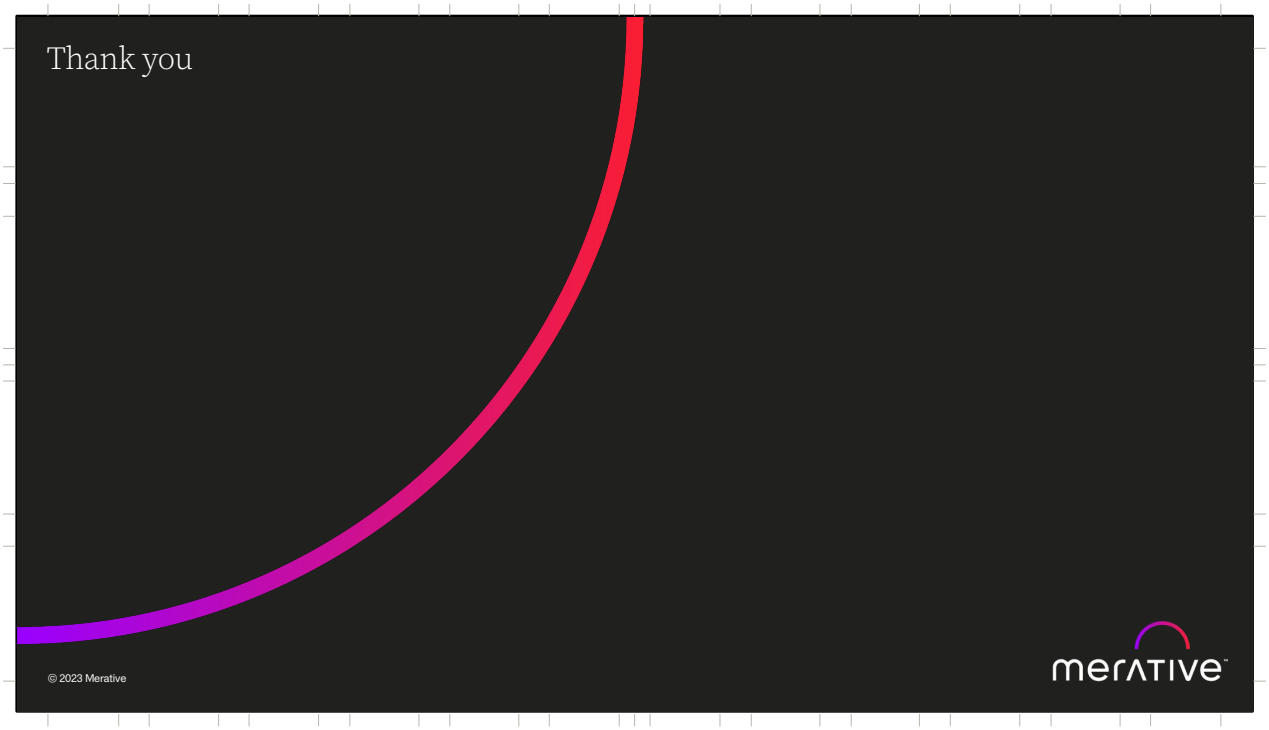
Payment Instrument post insert and post modify hooks

- You can now use new post insert and post modify hook on the following abstract class to implement custom business processing, such as updating custom financial entities after Payment Instruments are created or modified:
  - **curam.core.sl.event.impl.PaymentInstrumentInsertModifyEvents**

For more information, see the following docementation:

- Financial customization points' section in the Financials guide

- Java doc for the associated classes


Thank you
© 2023 Merative
merative

Forward Looking Statement

Merative's statements regarding its plans, directions and intent are subject to change or withdrawal without notice at Merative's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

© 2023 Merative

© Merative 2023. All Rights Reserved.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on Merative's current product plans and strategy, which are subject to change by Merative without notice. Merative shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from Merative or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of Merative products.

Product release dates, availability and/or capabilities referenced in this presentation may change at any time at Merative's sole discretion, and are not intended to be a commitment to future product or feature availability in any way. Not all Merative products are available in all jurisdictions in which Merative operates. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific performance results, and an individual user may achieve results different than any stated here.

Any customer examples described are presented as illustrations of how those customers have used Merative products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Merative, the Merative logo, and merative.com are trademarks of Merative, registered in many jurisdictions worldwide. Other product and service names might be trademarks of Merative or other companies.

© 2023 Merative

merATIVE