QAD Enterprise Applications
Enterprise Edition

Training Guide

Customization

70-3243-2012EE
QAD 2012 Enterprise Edition
August 2012

This document contains proprietary information that is protected by copyright and other intellectual property laws. No part of this document may be reproduced, translated, or modified without the prior written consent of QAD Inc. The information contained in this document is subject to change without notice.

QAD Inc. provides this material as is and makes no warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. QAD Inc. shall not be liable for errors contained herein or for incidental or consequential damages (including lost profits) in connection with the furnishing, performance, or use of this material whether based on warranty, contract, or other legal theory.

QAD and MFG/PRO are registered trademarks of QAD Inc. The QAD logo is a trademark of QAD Inc.

Designations used by other companies to distinguish their products are often claimed as trademarks. In this document, the product names appear in initial capital or all capital letters. Contact the appropriate companies for more information regarding trademarks and registration.

Copyright ©2012 by QAD Inc.

Customization_TG_v2012EE.pdf/biw

QAD Inc.
100 Innovation Place
Santa Barbara, California 93108
Phone (805) 566-6000
http://www.qad.com

Contents

Change Summary xiii

About This Course 1

Course Description 2

Virtual Environment Information 2

QAD Web Resources 3

Chapter 1 Course Overview 5

Course Overview 6

Course Description 7

Objectives 8

Daily Schedule 9

Agenda 10

Your Instructor 11

Facilities 12

Questions 13

Chapter 2 Browse Customizations 15

Browse Customizations 16

Browse Maintenance 17

Browse Maintenance 2 18

Creating a Browse 19

Selecting Fields 20

Using the Query Tab 21

Field Properties 22

Header Data Tab 23

Other Browse Options 24

Performance Considerations 25

Table Joins 26

Exporting Browses 27

Importing Browses 28

Deleting Multiple Browses 29

Browse Link Maintenance 30

Creating a Browse Link 31

Browse URL Maintenance 32
Creating a URL Link to a Program 33
Creating a Browse Link to a Web Page 34
Creating a Browse Link to a Web Page 2 35
Creating a Browse Link to Financials 36
Enterprise Financials Database Schema 37
Database - New Tables 38
Terminology Mapping 39
Database - Fields in New Tables 40
Database - Fields in New Tables 2 41
Database - Fields in New Tables 3 42
Database - Fields in New Tables 4 43
Data Replication/Data Ownership 44
Data Replication/Data Ownership 2 45
Data Replication/Data Ownership 3 46
Data Replication/Data Ownership 4 47
Data Replication/Data Ownership 5 48
Data Replication/Data Ownership 6 49
Replicated Tables 50
Replicated Tables 2 51
Replicated Tables 3 52
Business Data Model 53
Levels of Data 54
Shared Set Data 55
Shared Sets 56
Shared Sets - Assignment to Domain 57
Shared Sets - Assignment to Entity 58
Browses on Entity Data 59
Browses on Shared Data 60

Chapter 3 Customization Using UI Design Mode. 61

Customization Using QAD Application Features and Functions UI Design Mode 62
Configurable Screens 63
Types of Configuration 64
User Groups and Templates 65
Error Handling and Template Conflicts 66
Configurable Screens Setup 67
Creating a Template 68
Template Considerations 69
Performance Considerations 70
Configure Screen 71
Available Fields 72
Field Properties 73

Adding Lookups to Fields 74
Using Added Fields in Character Code 75
Adding Frames 76
Configurable Screens Report 77
Hands-on Exercise (5) 78
UI Design Mode 79
UI Design Mode 2 80
UI Design Mode 3 81
UI Design Mode 4 82
UI Design Mode 5 83
UI Design Mode 6 84
UI Design Mode 7 85
User-Defined Fields 86
User-Defined Fields 2 87
User-Defined Fields 3 88
Hands-on Exercise (6) 89

Chapter 4 Non-intrusive Customization of Enterprise Financials .91

Non-intrusive Customizations of Enterprise Financials 92
Non-Intrusive Customization: UI Design 93
UI Design Mode 94
UI Design Mode 2 95
UI Design Mode 3 96
UI Design Mode 4 97
UI Design Mode 5 98
UI Design Mode 6 99
Hands-on Exercise (1) 100
User-Defined Fields 101
User-Defined Fields 2 102
User-Defined Fields 3 103
Hands-on Exercise (2) 104
Non-intrusive Customization: Concepts 105
Concepts 106
Layered Application 107
Business Components 108
Proxy Implementations 109
Predefined Application Patterns 110
Business Fields 111
Business Fields 2 112
Business Activities 113
Object Datasets 114
(BL) Component Instantiations 115
Logical Transactions 116

Session Component 117
User-Defined Fields 118
User-Defined Tables + Components 119
Related Information / Objects 120
Non-instrusive Customization: Architecture 121
Introduction to QAD EE Architecture 122
Introduction to QAD EE Architecture 2 123
Introduction to QAD EE Architecture 3 124
Application Layers 125
Application Layers 2 126
Application Layers 3 127
Application Layers 4 128
Business Layer Component Structure 129
Inheritance Structure 130
Inheritance Structure 2 131
Inheritance Structure 3 132
Business Layer Component Structure - Terminology 133
Business Layer Component Structure - Terminology 2 134
Application Patterns Overview 135
Application Patterns Client 136
Application Patterns Client 2 137
Application Patterns Client 3 138
Application Patterns Client 4 139
Application Patterns Client 5 140
Application Patterns Business Logic Flow 141
Application Patterns Business Logic Flow 2 142
Application Patterns Business Logic Flow 3 143
Application Patterns Business Logic Flow 4 144
Application Patterns Business Logic Flow 5 145
Application Patterns Business Logic Flow 6 146
Application Patterns Business Logic Flow 7 147
Application Patterns Report Flow 148
Application Patterns Report Flow 2 149
Data Handling - Datasets 150
Data Handling - Datasets 2 151
Data Handling - Datasets 3 152
Data Handling - Datasets 4 153
Data Handling - Datasets 5 154
Data Handling - Datasets 6 155
Data Handling - Datasets 7 156
Data Handling - Datasets 8 157
Data Handling - Data Flow In Common Patterns 158
Data Handling - Data Flow in Patterns 159

Data Handling - Data Handling in Object Datasets - New Object . . . 160
Data Handling - Data Flow in Patterns . . . 161
Data Handling - Data Handling in Object Datasets - Modify Object . 162
Data Handling - Data Flow in Patterns . . . 163
Data Handling - Data Handling in Object Datasets - Delete Objects . 164
Error Handling . . . 165
Error Handling 2 . . . 166
Error Handling 3 . . . 167
Non-intrusive Customization: Development . . . 168
Requirements for Customization . . . 169
Requirements for Customization 2 . . . 170
BL Customizations . . . 171
BL Customizations 2 . . . 172
BL Customizations 3 . . . 173
BL Customizations 4 . . . 174
Customization Diagram . . . 175
Customization Diagram 2 . . . 176
Customization - Development Steps . . . 177
Customization - Development Steps 2 . . . 178
Customization - Development Steps 3 . . . 179
Customization - Development Steps 4 . . . 180
BL Customizations . . . 181
BL Customizations 2 . . . 182
Customization Exercise - Set Up . . . 183
BL Customizations . . . 184
BL Customizations 2 . . . 185
BL Customizations 3 . . . 186
BL Customizations 4 . . . 187
BL Customizations 5 . . . 188
BL Customizations 6 . . . 189
BL Customizations 7 . . . 190
BL Customizations 8 . . . 191
BL Customizations - InitialValues . . . 192
BL Customizations - InitialValues 2 . . . 193
Hands-on Exercise (3) . . . 194
BL Customizations - Calculate . . . 195
BL Customizations - Calculate 2 . . . 196
Hands-on Exercise (4) . . . 197
BL Customizations - ValidateComponent . . . 198
BL Customizations - ValidateComponent 2 . . . 199
Hands-on Exercise (5) . . . 200
BL Customizations - AdditionalUpdates . . . 201
BL Customizations - Additional Updates 2 . . . 202

BL Customizations - PreSave and PostSave 203
BL Customizations - PostSave 204
Hands-on Exercise (6) 205
BL Customizations - GetBusinessFields 206
BL Customizations - Business Fields Mechanism 207
BL Customizations - GetBusinessFields 208
Hands-on Exercise (7) 209
BL Customizations - GetTranslation 210
BL Customizations - GetTranslation 2 211
Translations 212
Translations 2 213
Translations 3 214
Translations 4 215
Hands-on Exercise (8) 216

Chapter 5 Non-intrusive Customization: Supporting Tools 217

Non-intrusive Customization: Supporting Tools 218
BL Customization - Tools 219
Launching HTML Documentation 220
Launching HTML Documentation 2 221
Navigating HTML Documentation 3 222
Navigating HTML Documentation 4 223
Navigating HTML Documentation 5 224
Navigating HTML Documentation 6 225
Navigating HTML Documentation 7 226
Navigating HTML Documentation 8 227
Navigating HTML Documentation 9 228
Navigating HTML Documentation 10 229
Navigating HTML Documentation 11 230
Navigating HTML Documentation 12 231
Navigating HTML Documentation 13 232
Navigating HTML Documentation 14 233
Navigating HTML Documentation 15 234
Hands-on Exercise (9) 235
BL Customizations - Tools 236
BL Customizations - BL Code Tracing 237
Hands-on Exercise (10) 238
BL Customizations 239
BL Customizations 2 240
BL Customizations 3 241
BL Customizations 4 242
BL Customizations 5 243
BL Customizations 6 244

BL Customizations 7 245
BL Customizations 8 246
BL Customizations 9 247
BL Customizations 10 248
BL Customizations 11 249
BL Customizations 12 250
Typical Customization: Case 1 251
Demo Customization: Case 1 252
Demo Customization: Case 1 (2) 253
Demo Customization: Case 1 (3) 254
Demo Customization: Case 1 (4) 255
Demo Customization: Case 1 (5) 256
Demo Customization: Case 1 (6) 257
Demo Customization: Case 1 (7) 258
Demo Customization: Case 1A 259
Hands-on Exercise (11) 260
Typical Customization: Case 2 261
Demo Customization: Case 2 262
Demo Customization: Case 2 (2) 263
Demo Customization: Case 2 (3) 264
Demo Customization: Case 2 (4) 265
Demo Customization: Case 2 (5) 266
Custom Tables 267
Custom Tables 2 268
Custom Tables 3 269
Demo Customization: Case 2 - DefineCustomRelations 270
Demo Customization: Case 2 - DataLoad 271
Demo Customization: Case 2 - PostSave 272
Demo Customization: Case 2 - Compile and Deploy 273
Demo Customization: Case 2 - Add Grid 274
Hands-on Exercise (12) 275
Typical Customization: Case 3 276
Demo Customization: Case 3 - Database Model 277
Demo Customization: Case 3 - Database Model 2 278
Demo Customization: Case 3 - Field Mapping 279
Custom Components 280
Custom Components 2 281
Demo Customization: Case 3 - Define User-Defined Component . . . 282
Demo Customization: Case 3 - Define User-Defined Fields 283
Demo Customization: Case 3 - Define Menu Items 284
Demo Customization: Case 3 - Define Role Permissions 285
Demo Customization: Case 3 - UI Design Mode 286
Demo Customization: Case 3 - UI Design Mode 2 287

Demo Customization: Case 3 - UI Design Mode 3 288
Hands-on Exercise (13) 289
Report Customization 290
Report Framework for Financials Reports 291
Report Framework for Financials Reports 2 292
Report Framework for Financials Reports 3 293
Report Framework for Financials Reports 4 294
Report Framework for Financials Reports 5 295
Report Framework for Financials Reports 6 296
Report Framework for Financials Reports 7 297
Report Framework for Financials Reports 8 298
Report Framework for Financials Reports 9 299
Report Framework for Financials Reports 10 300
Report Framework for Financials Reports 11 301
Report Framework for Financials Reports 12 302
Report Framework for Financials Reports 13 303
Report Framework for Financials Reports 14 304
Report Framework for Financials Reports 15 305
Report Framework for Financials Reports 16 306
Report Framework for Financials Reports 17 307
Report Framework for Financials Reports 18 308
Report Framework for Financials Reports 19 309
Report Framework for Financials Reports 20 310
Report Framework for Financials Reports 21 311
Report Framework for Financials Reports 22 312
Report Customizations 313
Report Customization - User-Defined Fields 314
Report Customization - Debugging 315
Hands-on Exercise (14 + 15 + 16) 316

Chapter 6 Integration with Enterprise Financials: Backend 317

Integration with Enterprise Financials: Backend 318
Integrating with Financials Backend 319
Calling the Application API 1 320
Calling the Application API 2 321
Calling the Application API 3 322
Calling the Application API 4 323
Calling the Application API 5 324
Calling the Application API 6 325
Calling the Application API 7 326
Calling the Application API 8 327
Calling the Application API 9 328
Calling the Application API 10 329

Calling the Application API 11 330
Calling the Application API 12 331
Calling the Application API 13 332
Calling the Application API 14 333
Calling the Application API 15 334

Change Summary

The following table summarizes significant differences between this document and the last published version.

Date/Version	Description	Reference
August 2012/2012EE	Numerous editorial changes	---
November 2012/2010EE	Initial version. No changes.	---

About This Course

Course Description

This course is designed to teach Enterprise Edition customization concepts and practice. It consists of the following modules:

- Browse Customizations
- Customization using QAD Application Features and Functions UI Design Mode
- Non-intrusive Customization of Enterprise Financials
- Non-intrusive Customization: Supporting Tools
- Integration with Enterprise Financials: Backend

Course Objectives

By the end of this class, students will:

- Understand non-intrusive development
- Understand the impact on development
- Know how to perform customizations using QAD application features and functions
- Know how to customize component-based QAD applications

Audience

This training is for developers/designers with Progress 4GL experience who want to learn non-intrusive customization techniques for component-based programs and functions (typically the new Financials functions).

Prerequisites

Students are expected to have an intermediate understanding of Progress (OpenEdge 10). Completion of the following online classes (available on the Learning Portal) is required prior to attending this class:

- Introduction to Java and Tomcat for QAD Enterprise Edition
- Introduction to Georgia SoftWorks Telnet Server for QAD Enterprise Edition
- Introduction to Supported Operating Systems – QAD Enterprise Edition
- Introduction to the QAD .NET UI – QAD Enterprise Edition
- Introduction to the QAD Interoperability (QXtend) – QAD Enterprise Edition
- Introduction to the QAD Interoperability (EDI eCommerce) – QAD Enterprise Edition
- QAD Application Architecture – QAD Enterprise Edition

Virtual Environment Information

The hands-on exercises in this book should be used with the QAD Enterprise Edition training environment specified by your instructor.

QAD Web Resources

From QAD's main site, you can access QAD's Learning or Support sites.

http://www.qad.com/

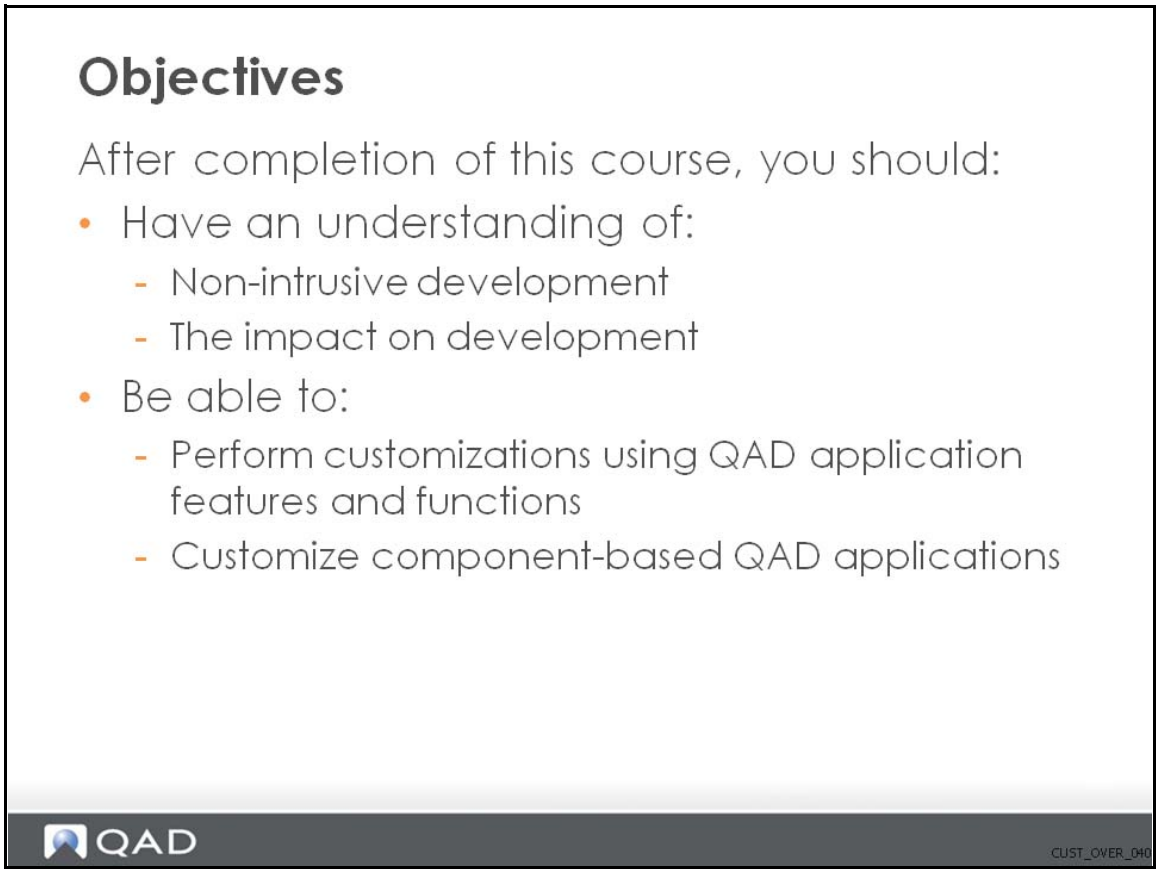
Chapter 1

Course Overview

Course Overview

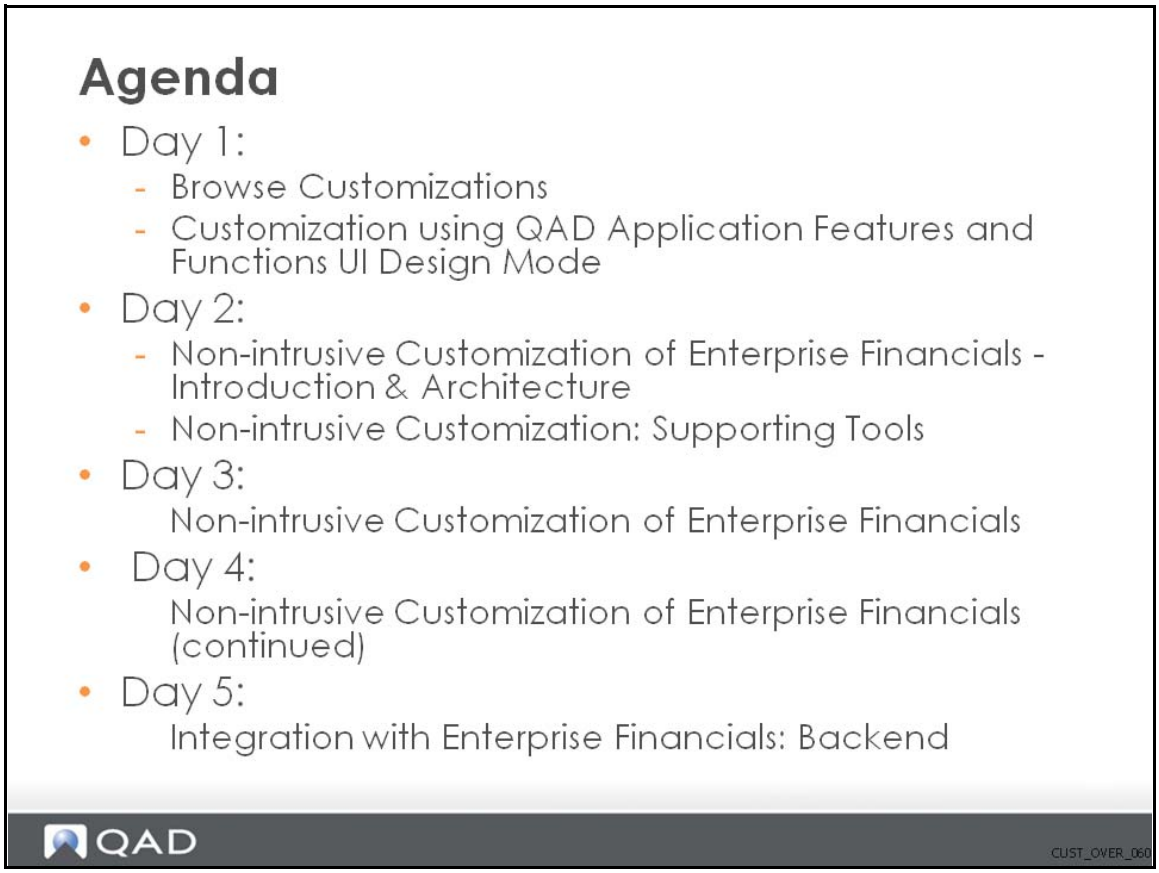
Course Description

Objectives

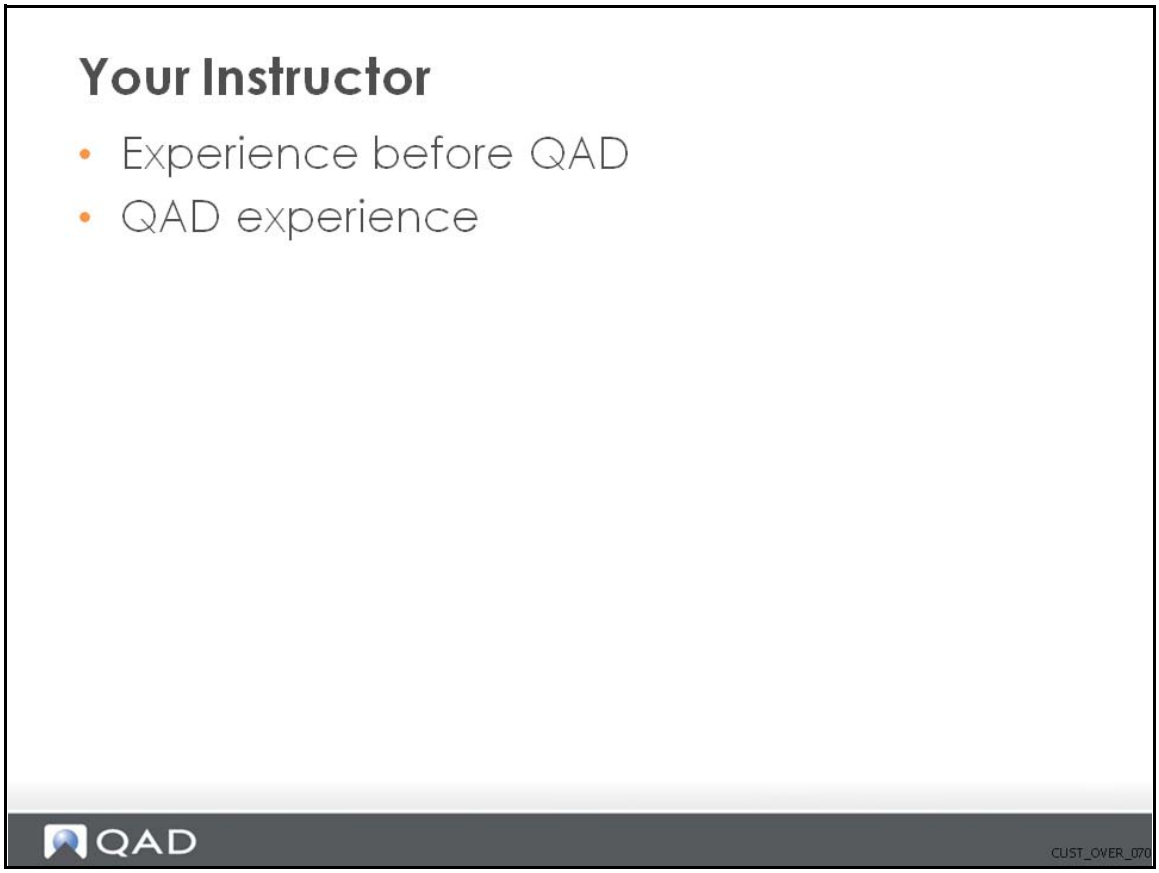

Objectives
After completion of this course, you should:
• Have an understanding of:
- Non-intrusive development
- The impact on development
• Be able to:
- Perform customizations using QAD application features and functions
- Customize component-based QAD applications
QAD
CUST_OVER_040

Daily Schedule

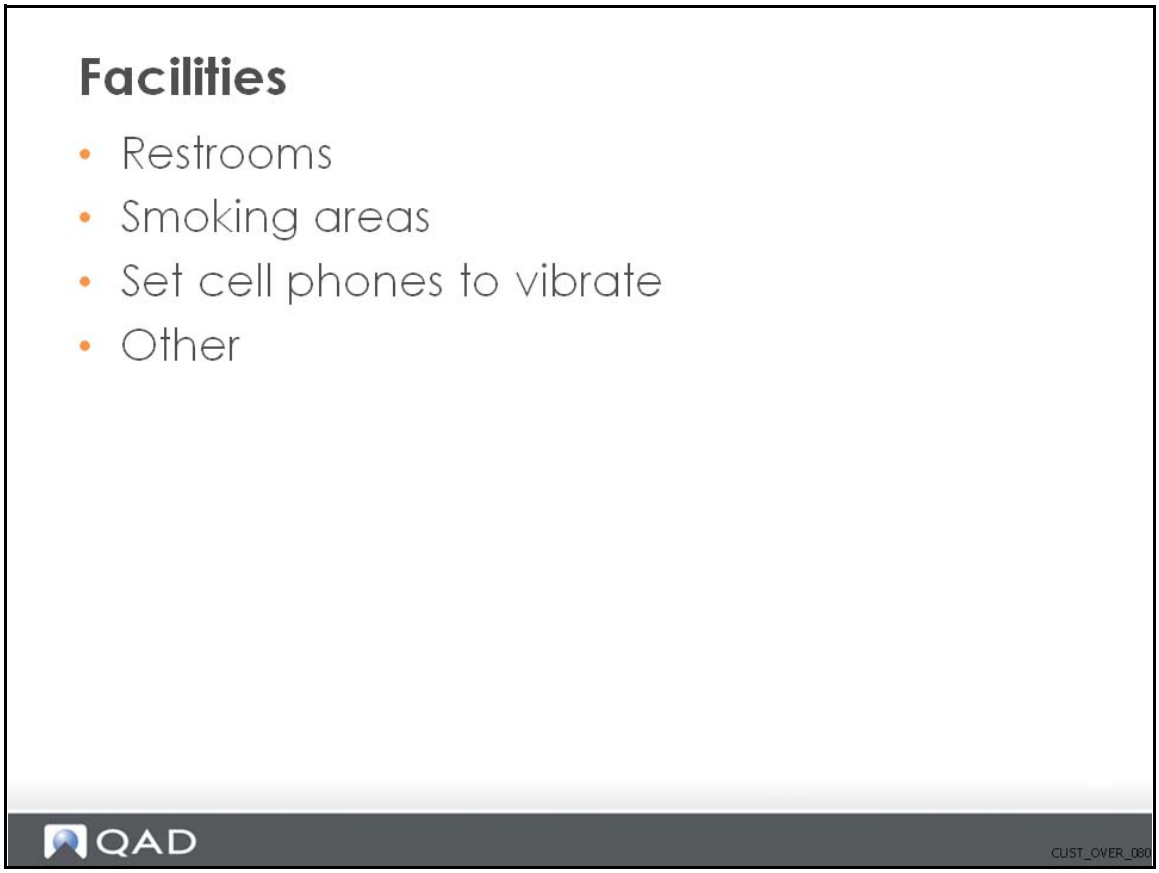
Agenda


Agenda
Day 1:
Browse Customizations
Customization using QAD Application Features and Functions UI Design Mode
Day 2:
Non-intrusive Customization of Enterprise Financials - Introduction & Architecture
Non-intrusive Customization: Supporting Tools
Day 3:
Non-intrusive Customization of Enterprise Financials
Day 4:
Non-intrusive Customization of Enterprise Financials (continued)
Day 5:
Integration with Enterprise Financials: Backend
QAD
CUST_OVER_060

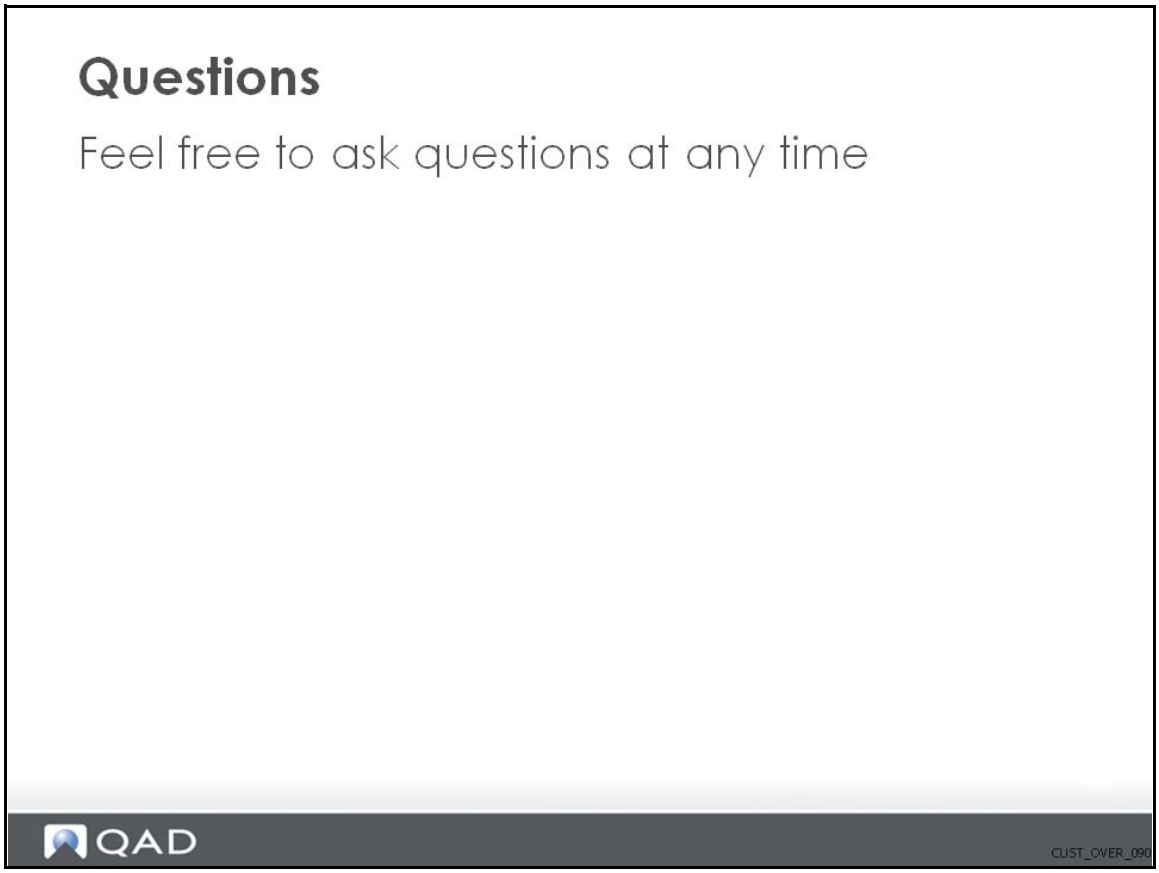
Your Instructor



Facilities


Facilities
Restrooms
Smoking areas
Set cell phones to vibrate
Other
QAD
CUST_OVER_080

Questions

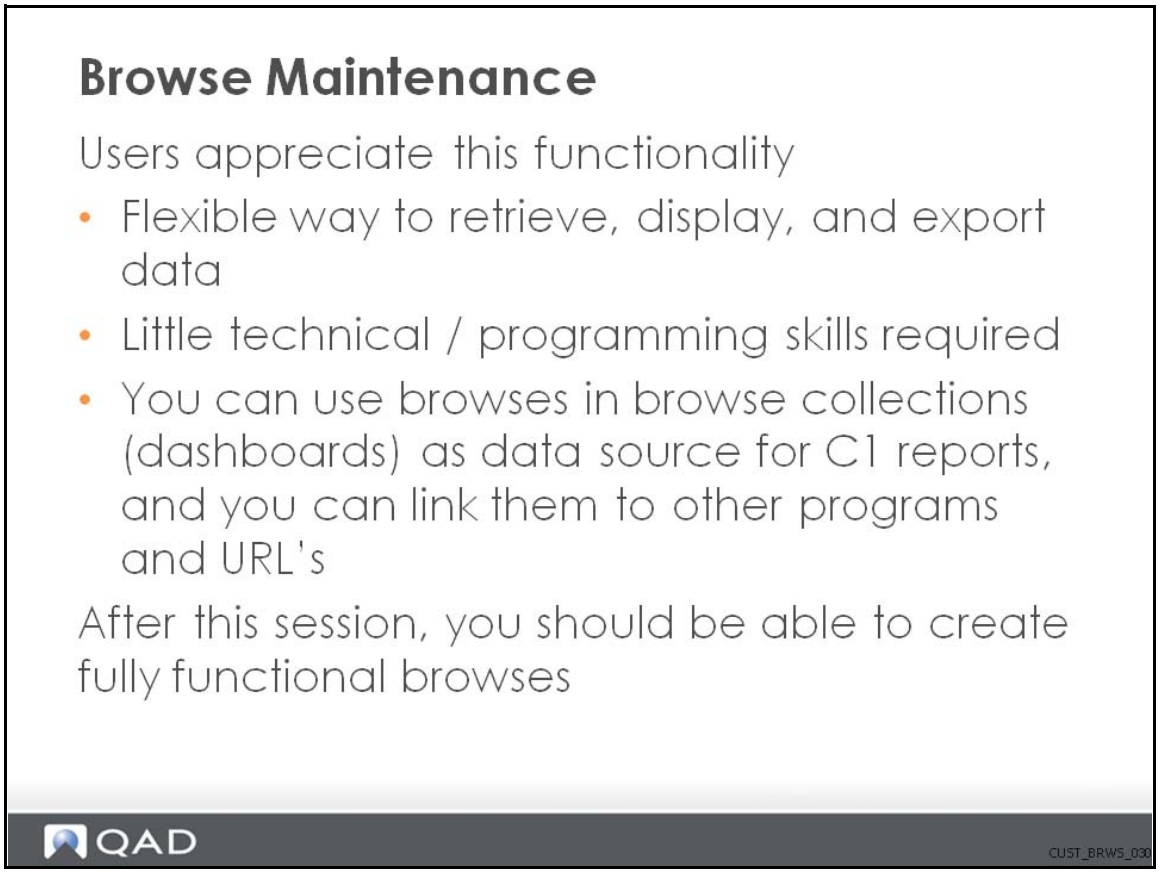

Questions
Feel free to ask questions at any time
QAD
CUST_OVER_090

Chapter 2

Browse Customizations

Browse Customizations

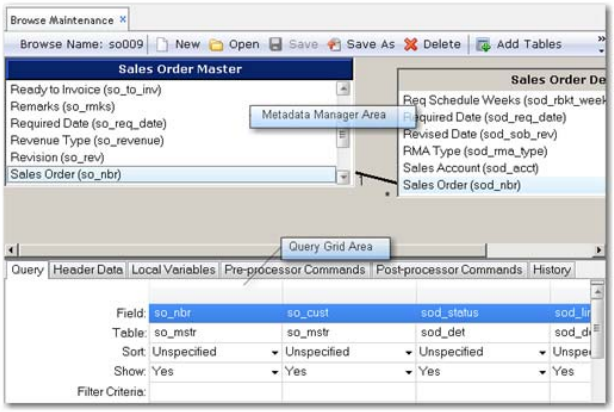
Browse Maintenance



Browse Maintenance 2

Browse Maintenance

- Design tool for creating, editing, duplicating and deleting browses
- .NET UI program
- Import and export browses
- Run stand-alone
- Run as a lookup on custom fields



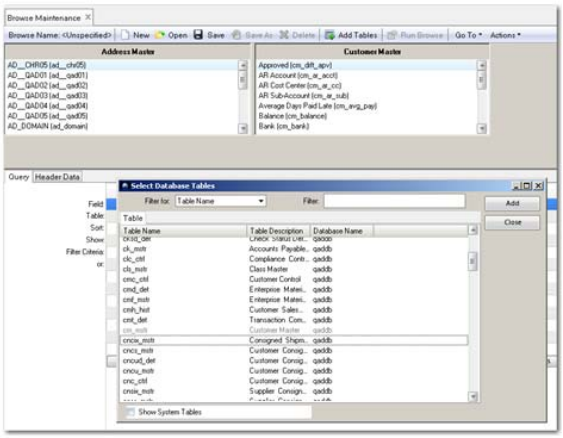
QAD

CUST_BRWS_040

Creating a Browse

Creating a Browse

- Add left and right tables
- Include or exclude system tables
- Can include up to 30 tables in a new browse
- Tables defined in `client-session.xml`
- Restricted tables are not displayed



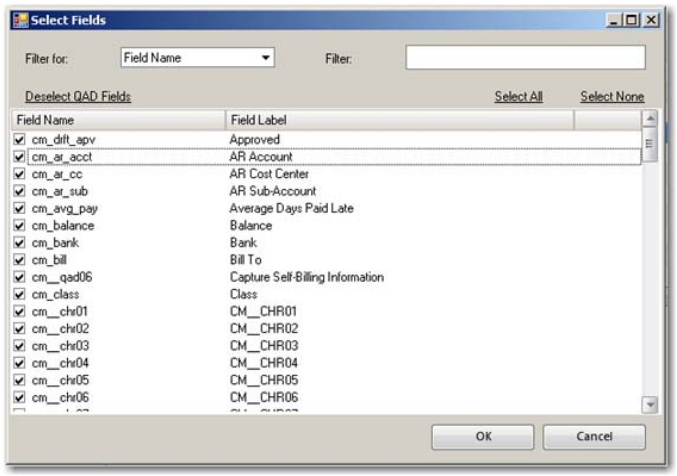
QAD

CUST_BRWS_050

Selecting Fields

Selecting Fields

- Right-click Table Area
- Select or deselect custom QAD fields



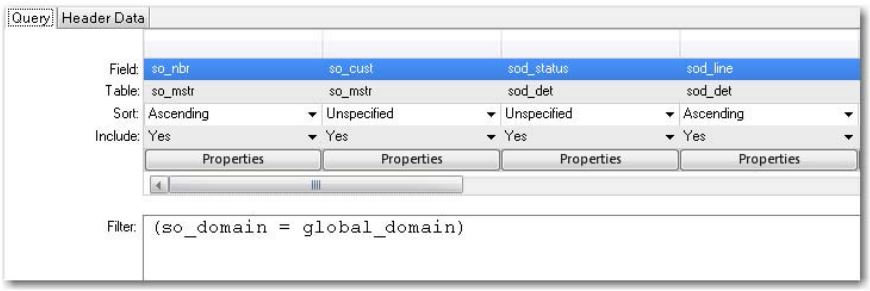
QAD

CUST_BRWS_060

Using the Query Tab

Using the Query Tab

- Add fields to the browse using the Query tab
- Set field Sort and Include properties
- Set field filters
- View and modify field properties



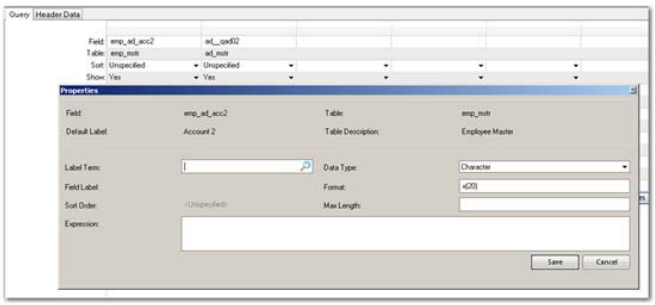
QAD

CUST_BRWS_070

Field Properties

Field Properties

- Specify a label term from the label master table
- Sort order if column is set to Ascending
- Expressions for local variables
- Data Types for local variables
- Format depends on data type
- Display length



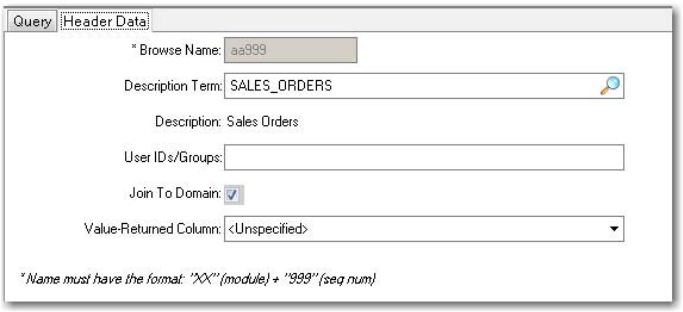
QAD

CUST_BRWS_080

Header Data Tab

Header Data Tab

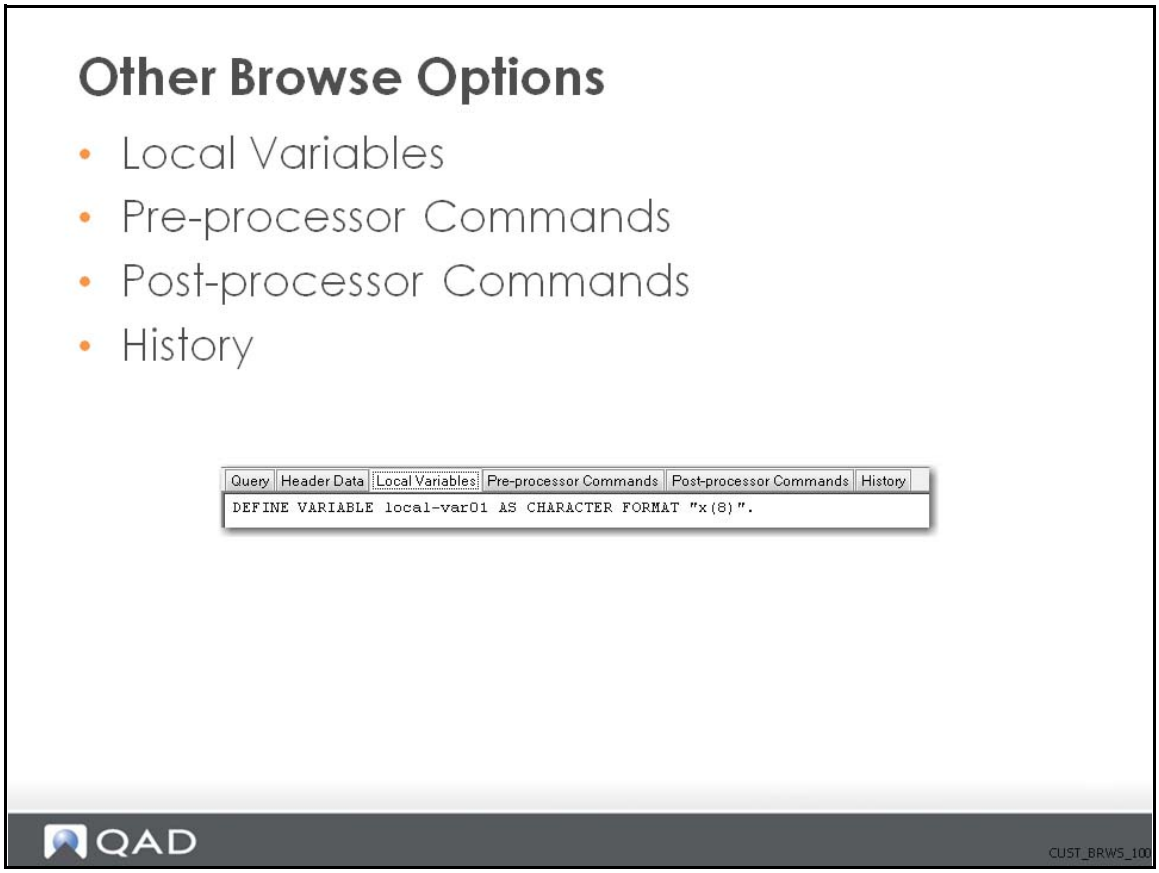
- Name and description for the browse
- Optionally restrict access to the browse
- Join the browse tables to the global domain
- Specify the value returned by double-clicking on a browse row



QAD

CUST_BRWS_090

Other Browse Options



Performance Considerations

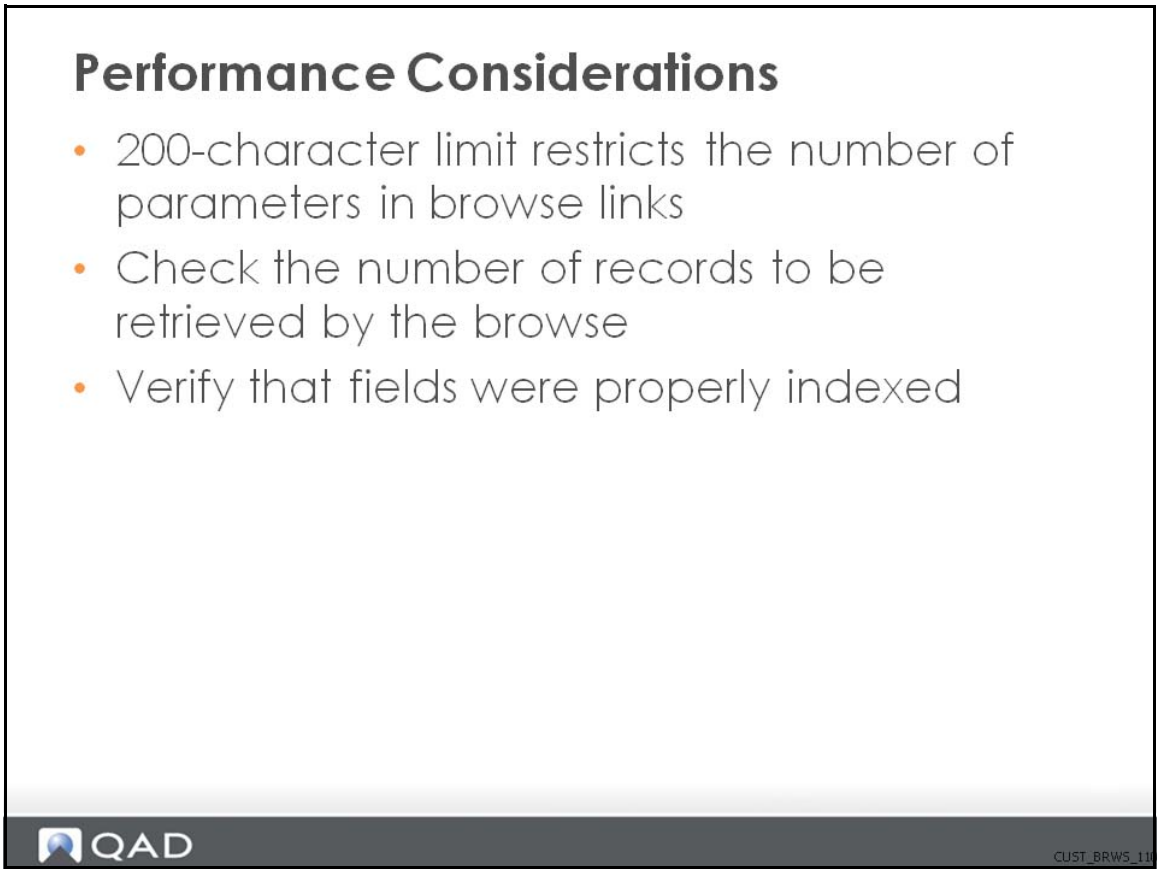
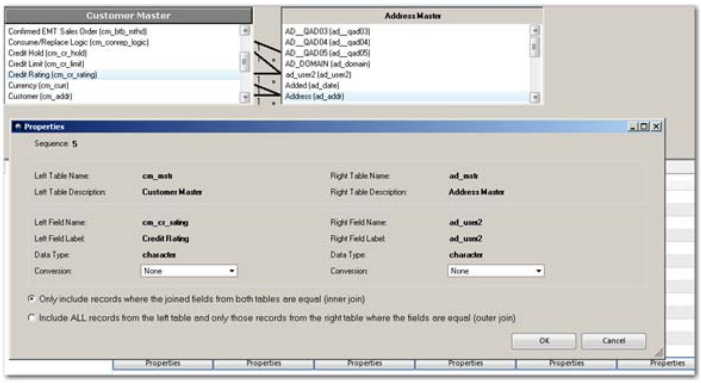


Table Joins

Table Joins

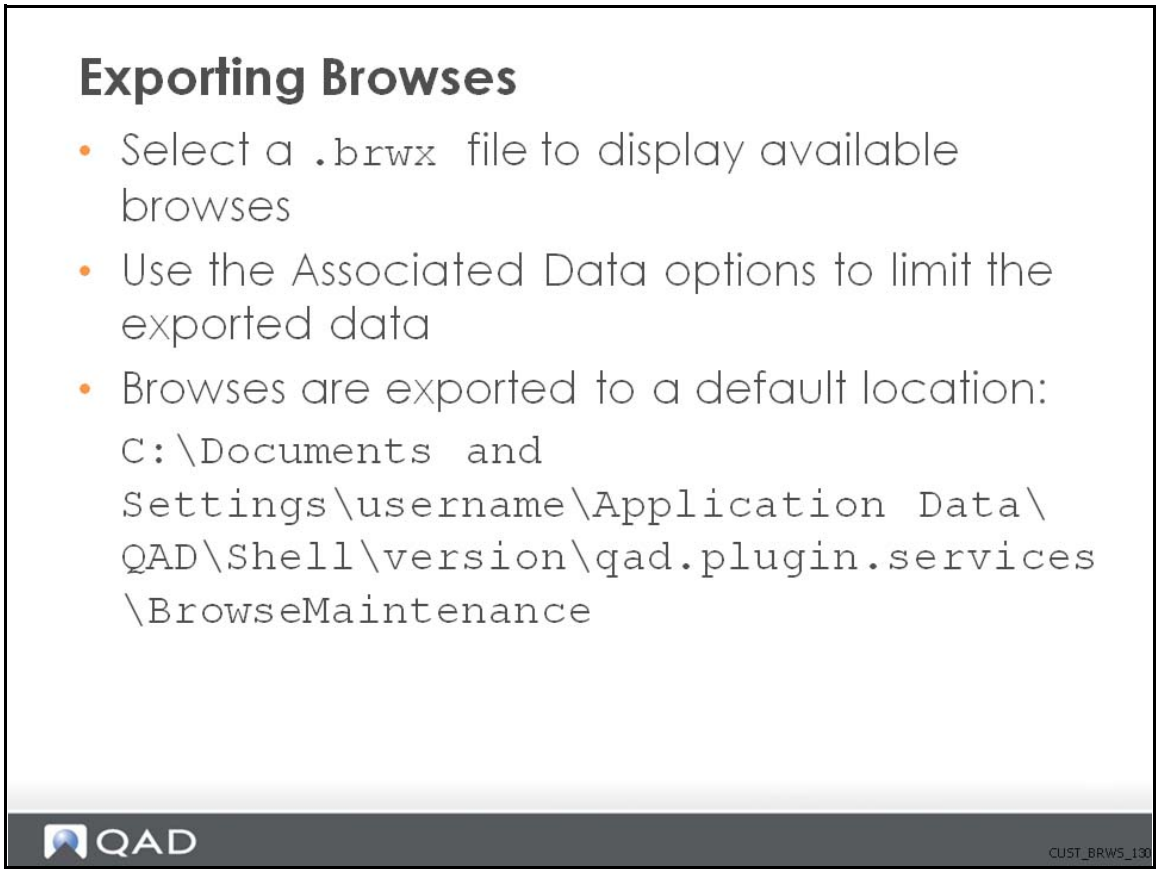
- Drag and drop to create a join
- View and modify join properties
- Data type conversion
- Inner and Outer Joins



QAD

CUST_BRWS_120

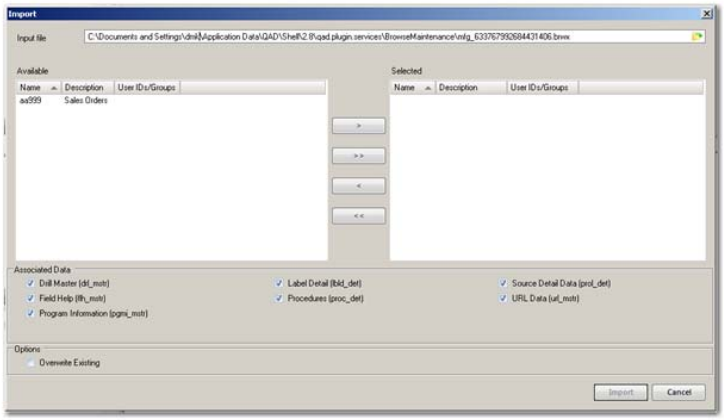
Exporting Browses



Importing Browses

Importing Browses

- Select a browse definition file (`.brwx`) using the Open lookup



- Specify the data to import using the Associated Data options

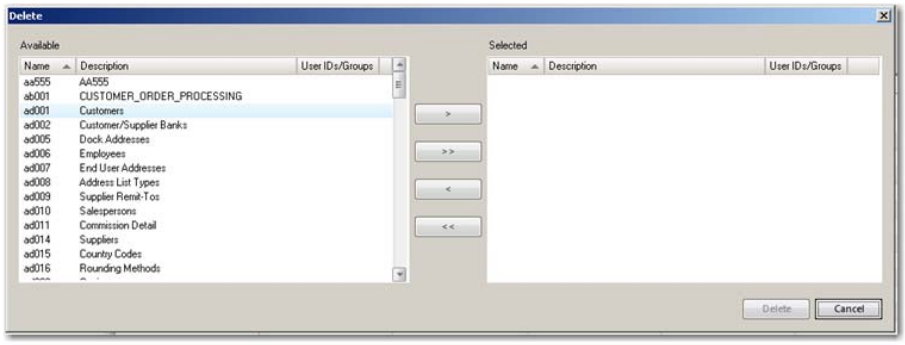
QAD

CUST_BRWS_140

Deleting Multiple Browses

Deleting Multiple Browses

Security considerations: Restricting the use of Browse Import, Export and Multiple Delete functions by modifying the `client-session.xml` file



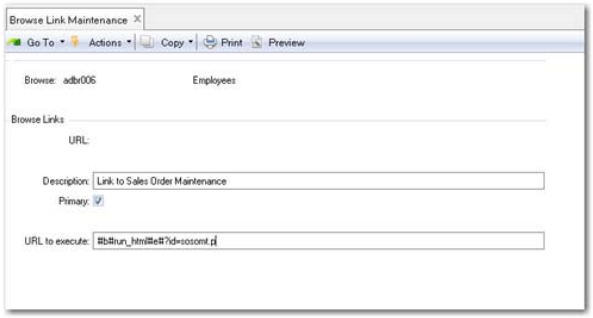
QAD

CUST_BRWS_150

Browse Link Maintenance

Browse Link Maintenance

- Creating New and Edit buttons on browses that link to Create and Modify programs
- Programs linked as URLs
- Data in the current browse is passed to the linked Modify program



QAD

CUST_BRWS_160

Creating a Browse Link

Creating a Browse Link

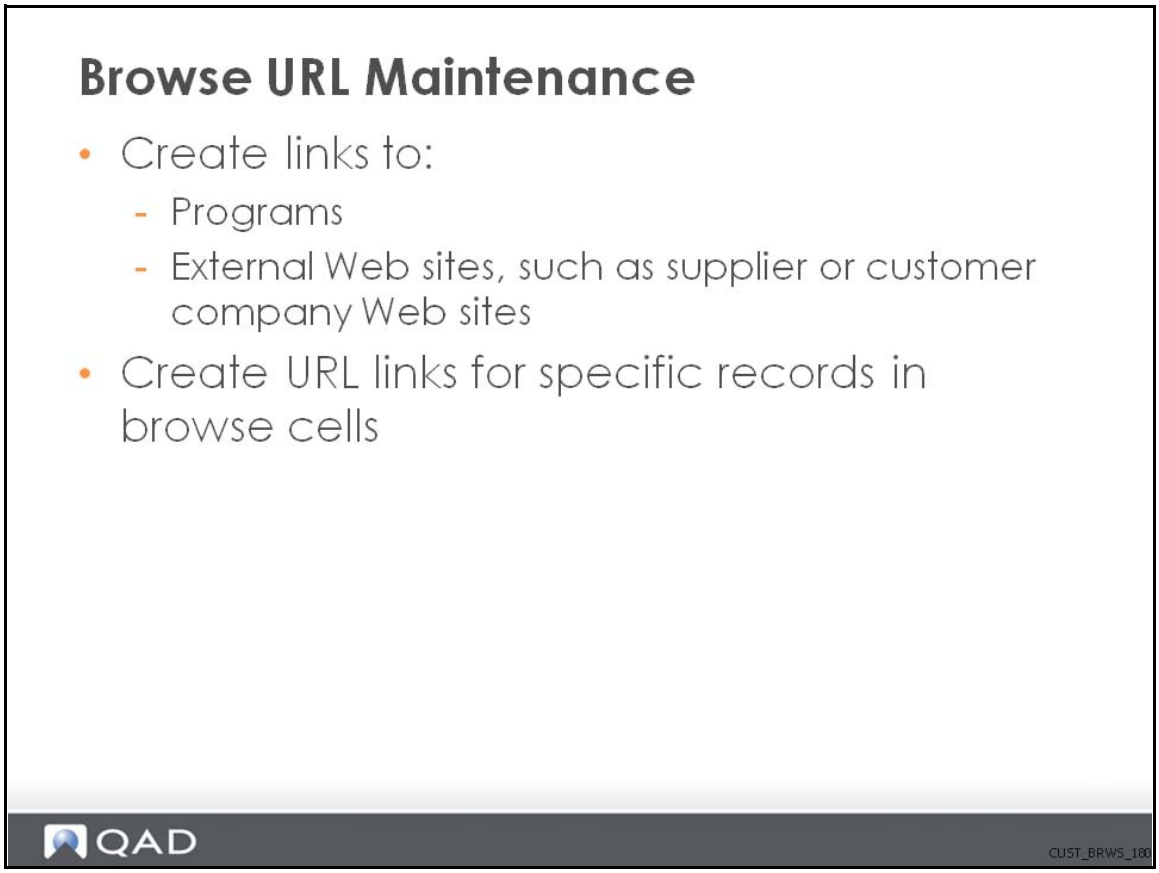
- Select a browse
- Specify the program link as a URL to Execute using the syntax:

  `#b#run_html#e#?id=program_name`
- Set the HTTP parameters for the program
- Index is the number of submits the program executes
- Field and Value are the field and value you want to load in the linked program

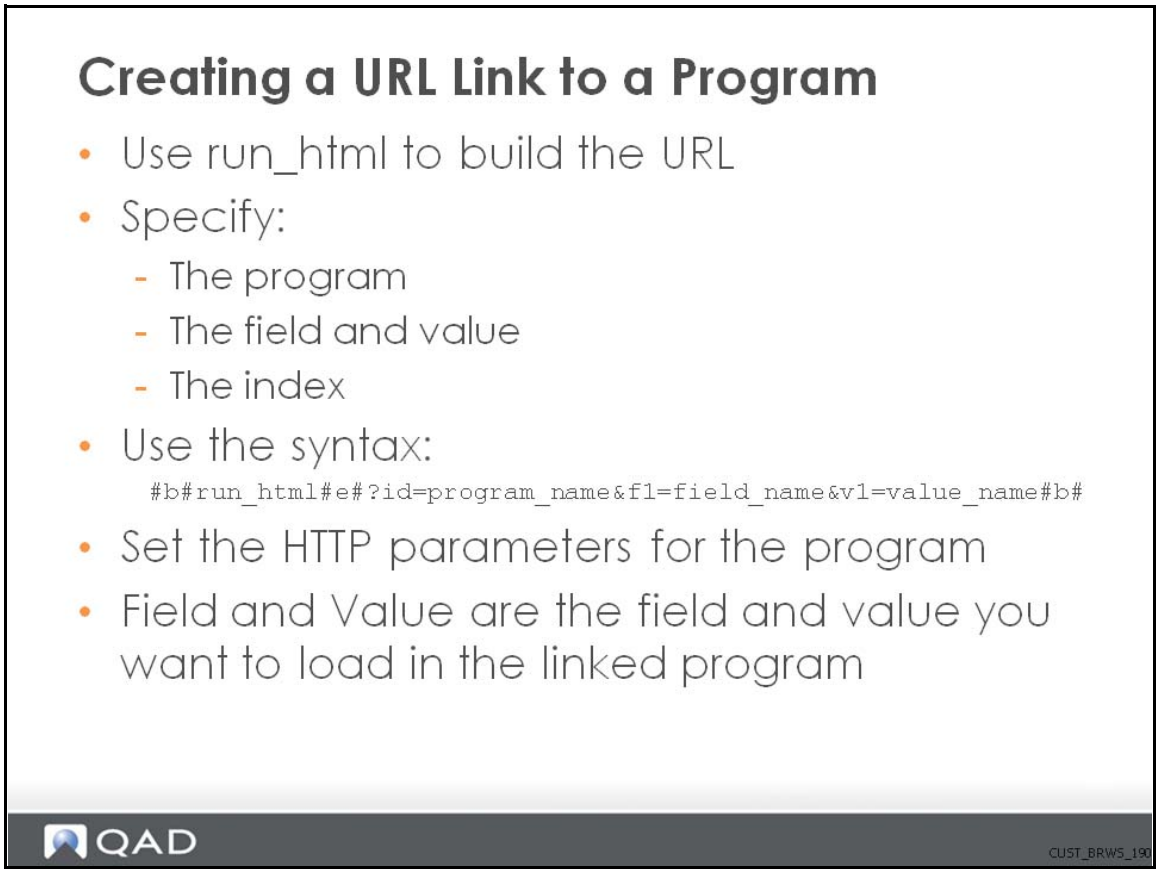
QAD

CUST_BRWS_170

Browse URL Maintenance



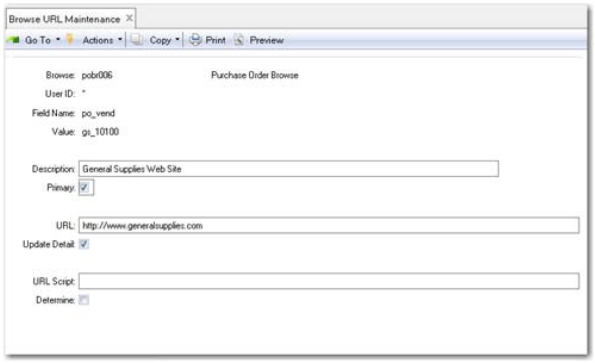
Creating a URL Link to a Program



Creating a Browse Link to a Web Page

Creating a Browse Link to a Web Page

- Specify:
  - The browse
  - Field name and value
  - The URL
- Use the syntax: `http://www.website.com`



QAD

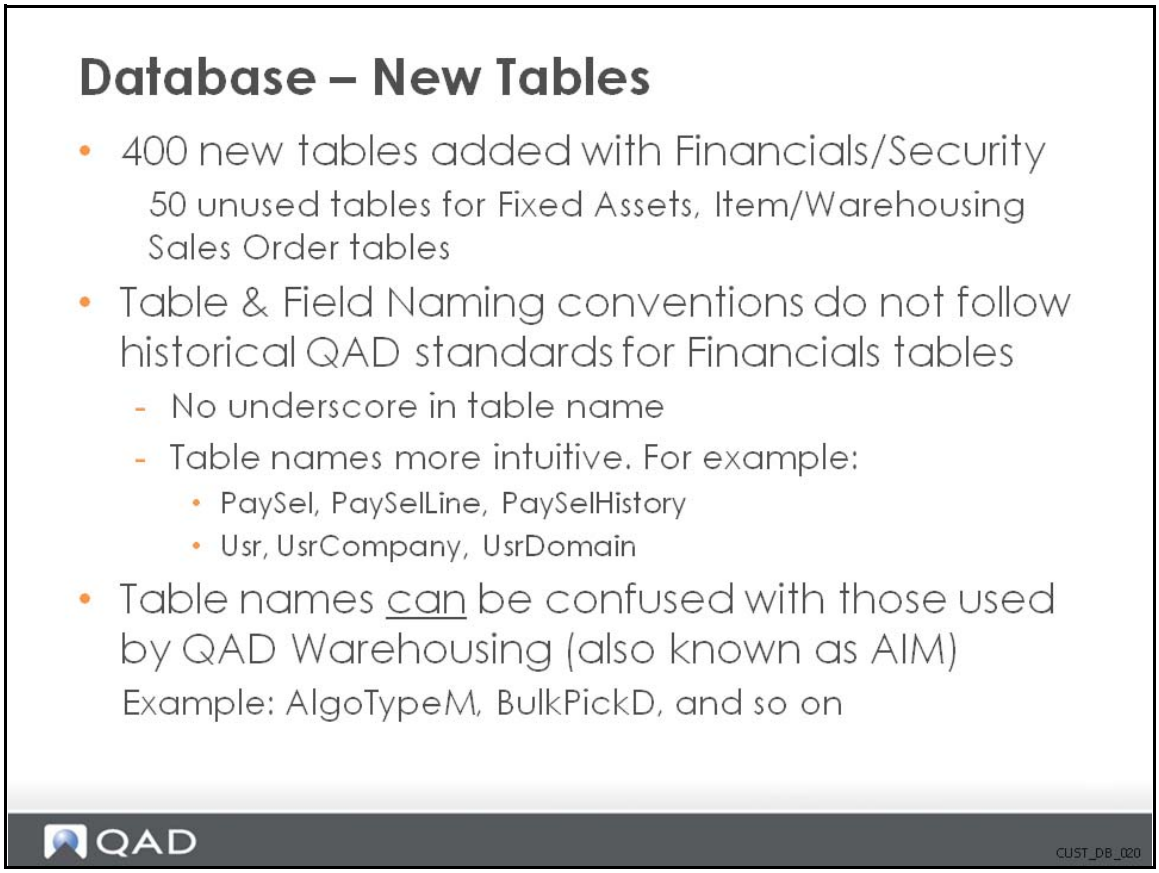
CUST_BRWS_20

Creating a Browse Link to a Web Page 2

Creating a Browse Link to Financials

Enterprise Financials Database Schema

Database - New Tables



Terminology Mapping

Terminology Mapping

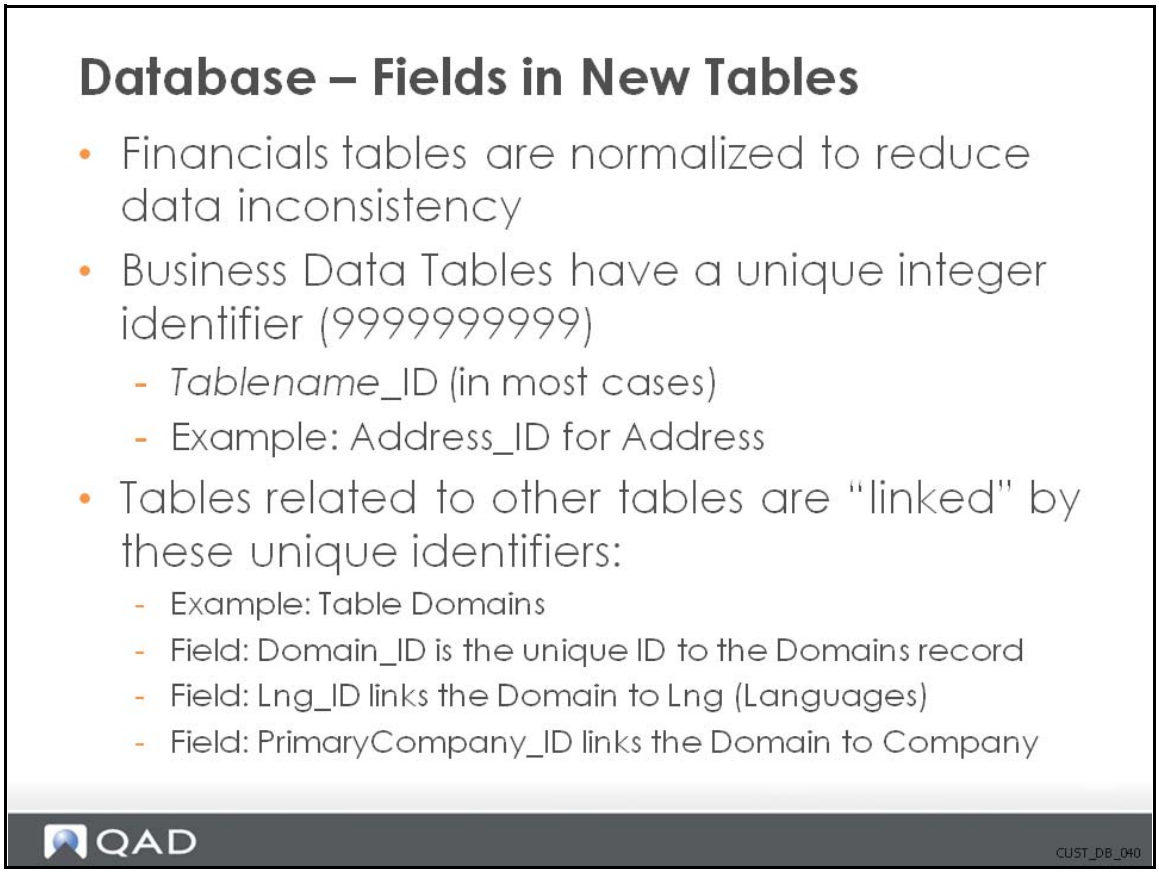
Terminology – mapping between logical application term and database table in Financials (special cases only)

	A	B
1	**Data Element**	**Financials Table**
2	Address Data	Address
3	AP Matching Data	APMatching
4	Cost Center Data	CostCentre
5	Country Data	Country
6	Credit Terms Data	PaymentCondition
7	Credit Terms Staged Data	PaymentConditionStaged
8	Customer Data	Debtor
9	Daybook Data	Journal
10	Domain Data	Domains
11	Employee Data	Employee
12	End User Contact Data	Contact
13	End User Data	DebtorEndUser
14	Entity Data	Company
15	Exchange Rate Data	ExchangeRate
16	GL Account Data	GL
17	GL Calendar Data	GLCalendar
18	GL Mask Data	GLMask
19	GL Period Data	Period
20	Invoice Data	DInvoice
21	Language Data	Lng
22	Posting Data	Posting
23	Role Data	Role
24	Rounding Method Data	RoundingMethod
25	Sub-Account Data	Division
26	Supplier Data	Creditor
27	Tax Data	Vat

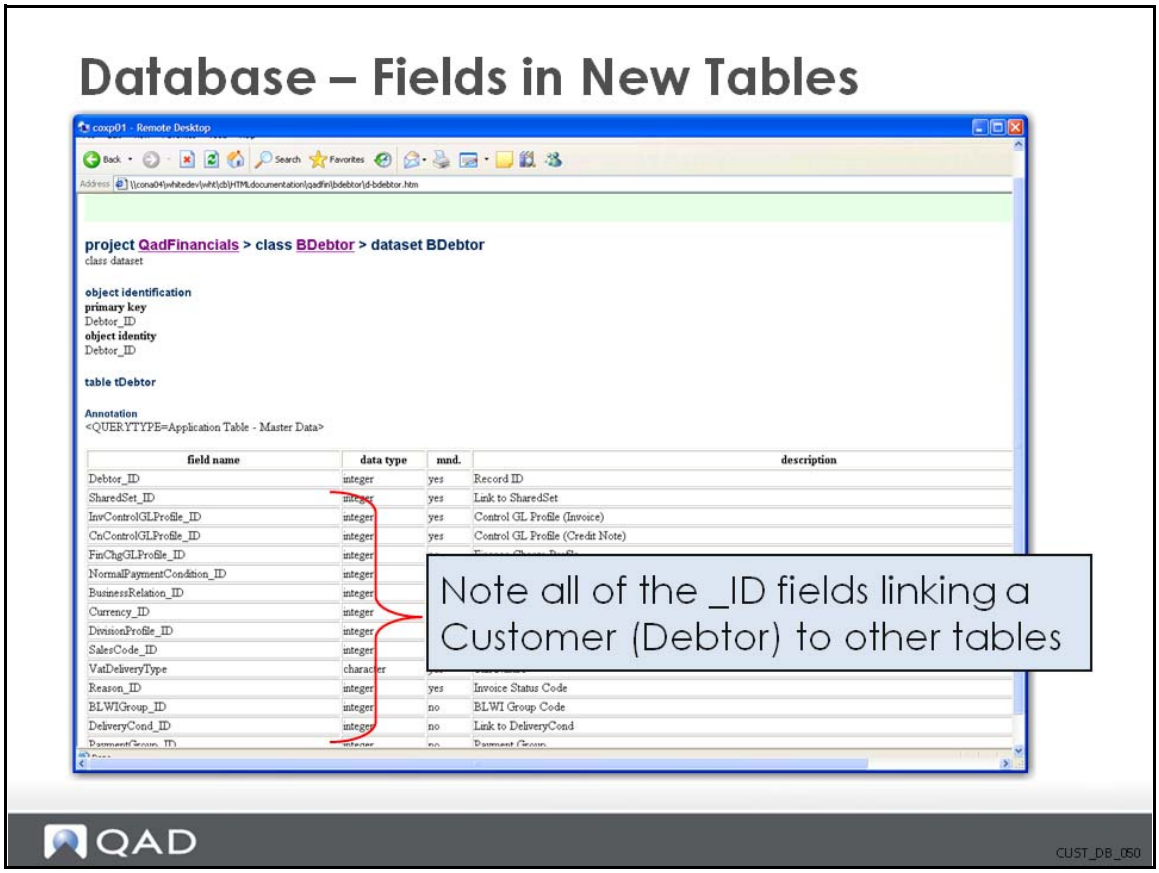
QAD

CUST_DB_030

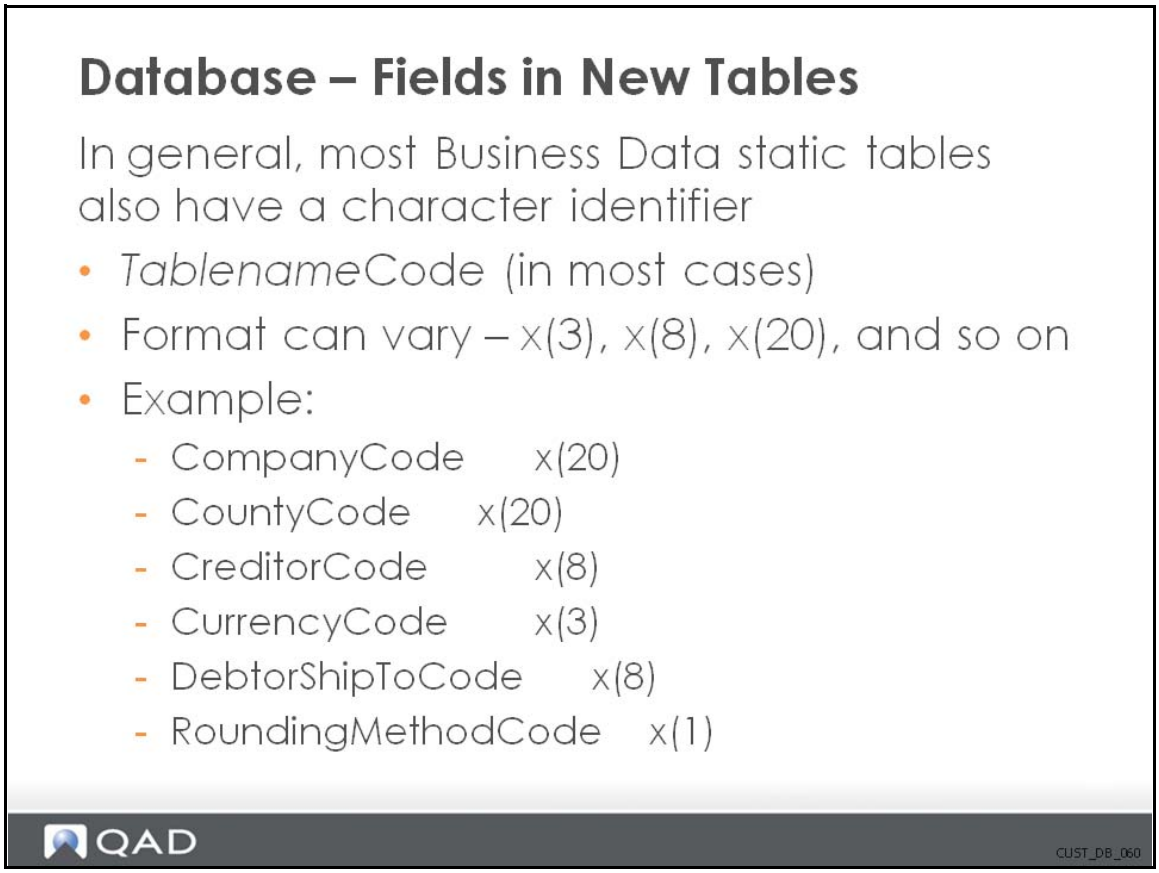
Database - Fields in New Tables



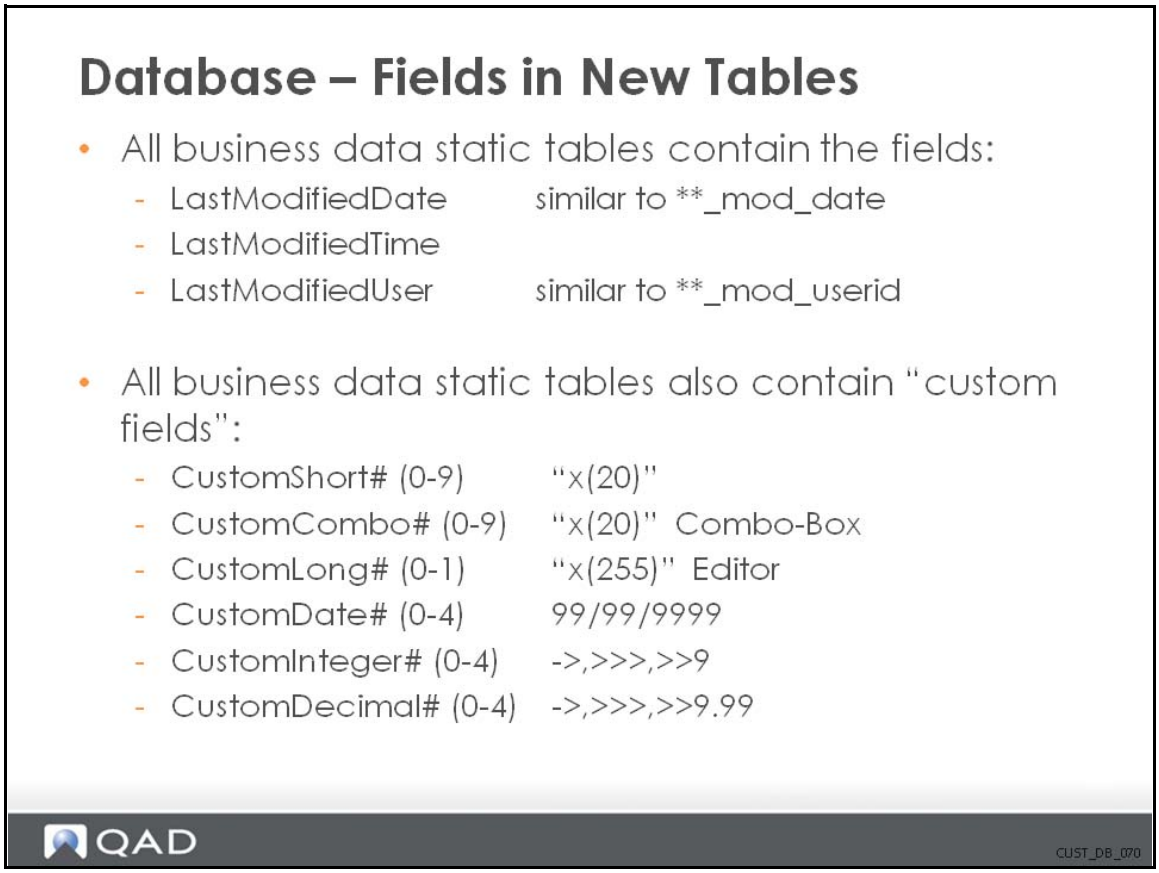
Database - Fields in New Tables 2



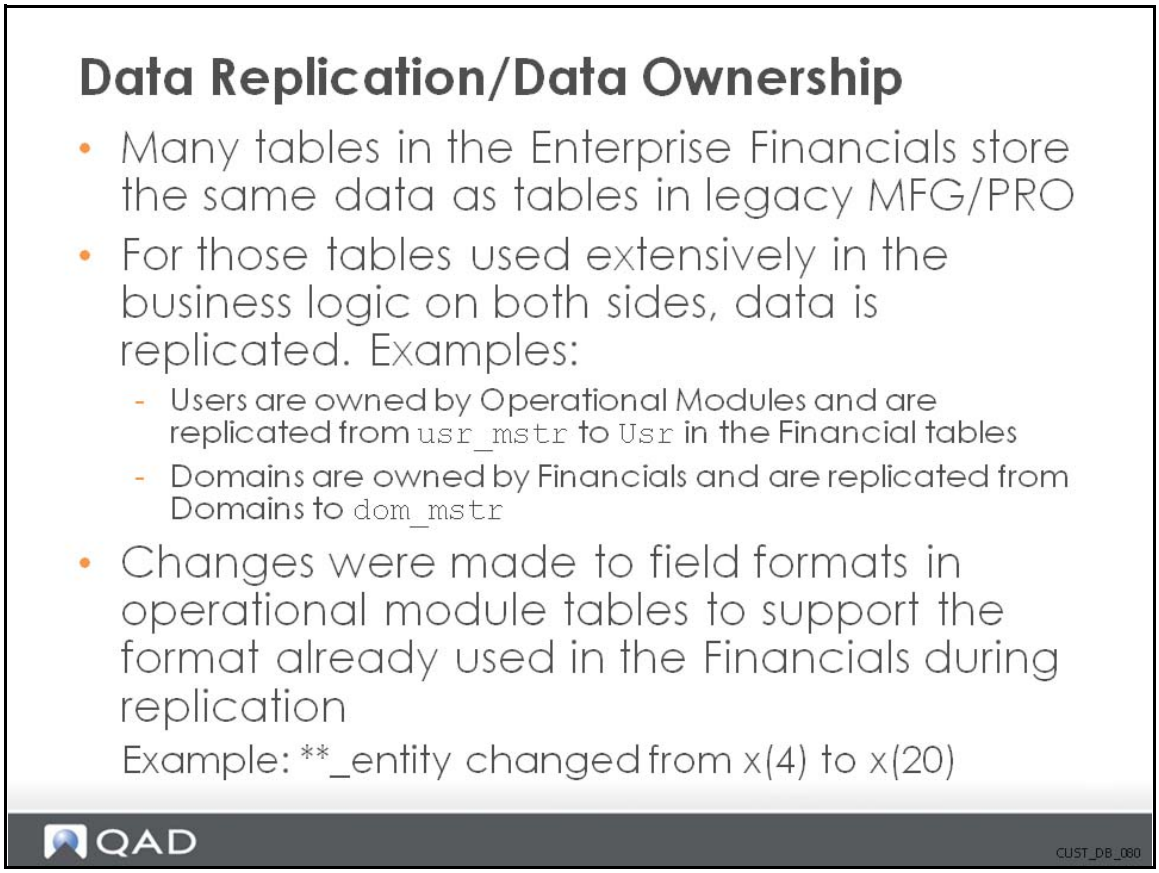
Database - Fields in New Tables 3



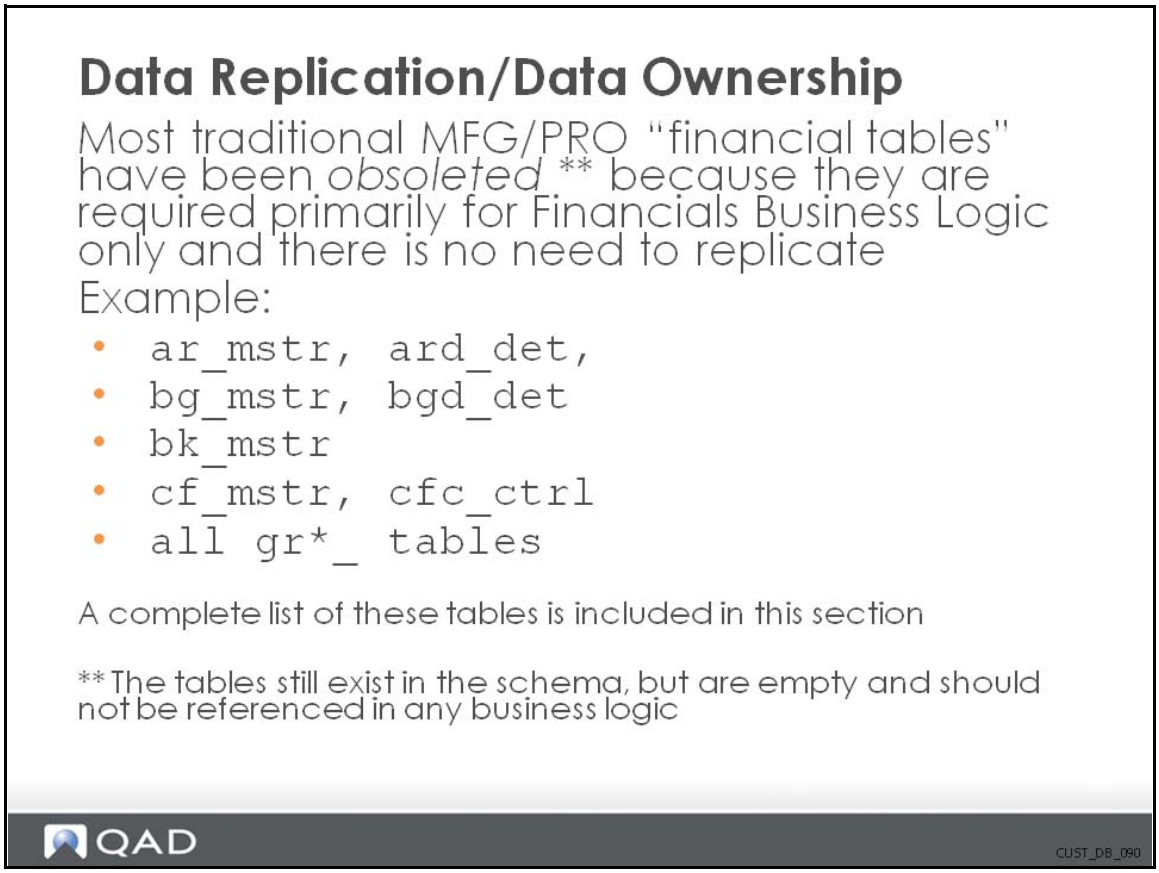
Database - Fields in New Tables 4



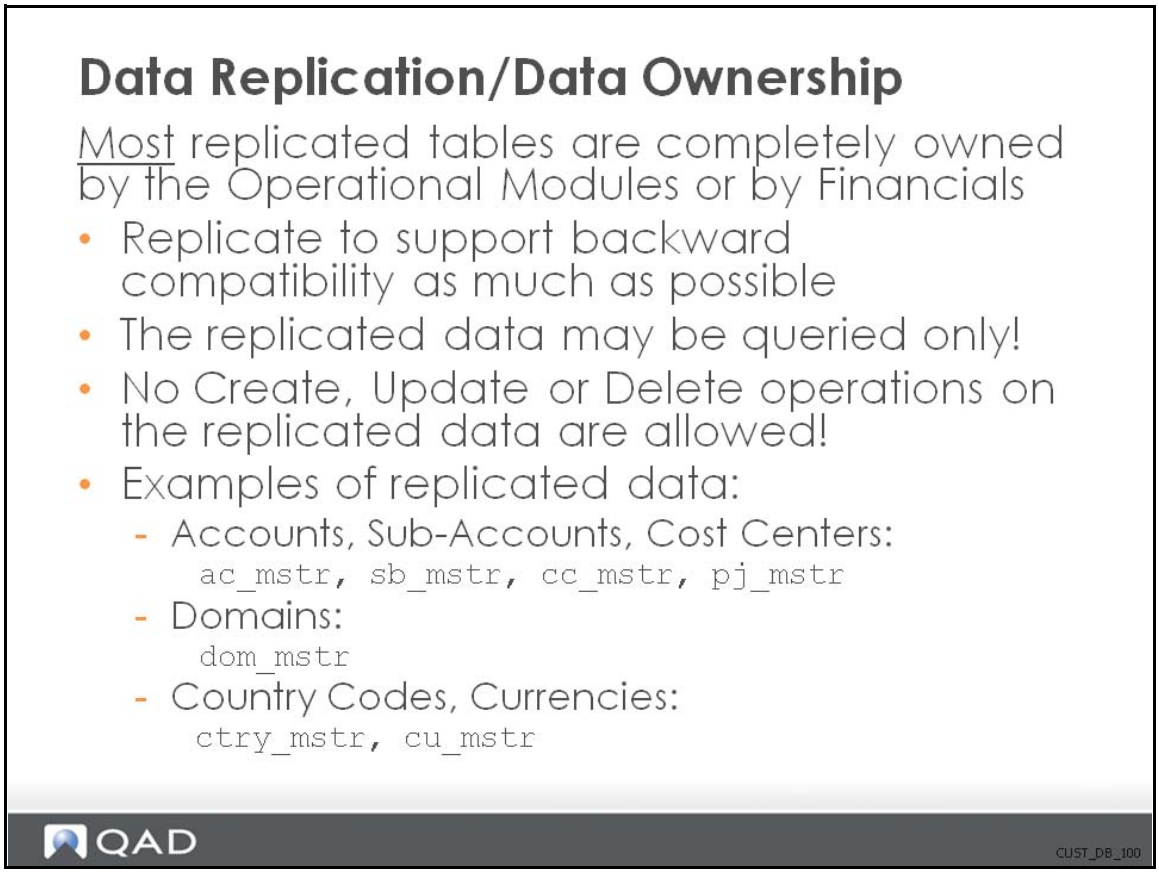
Data Replication/Data Ownership



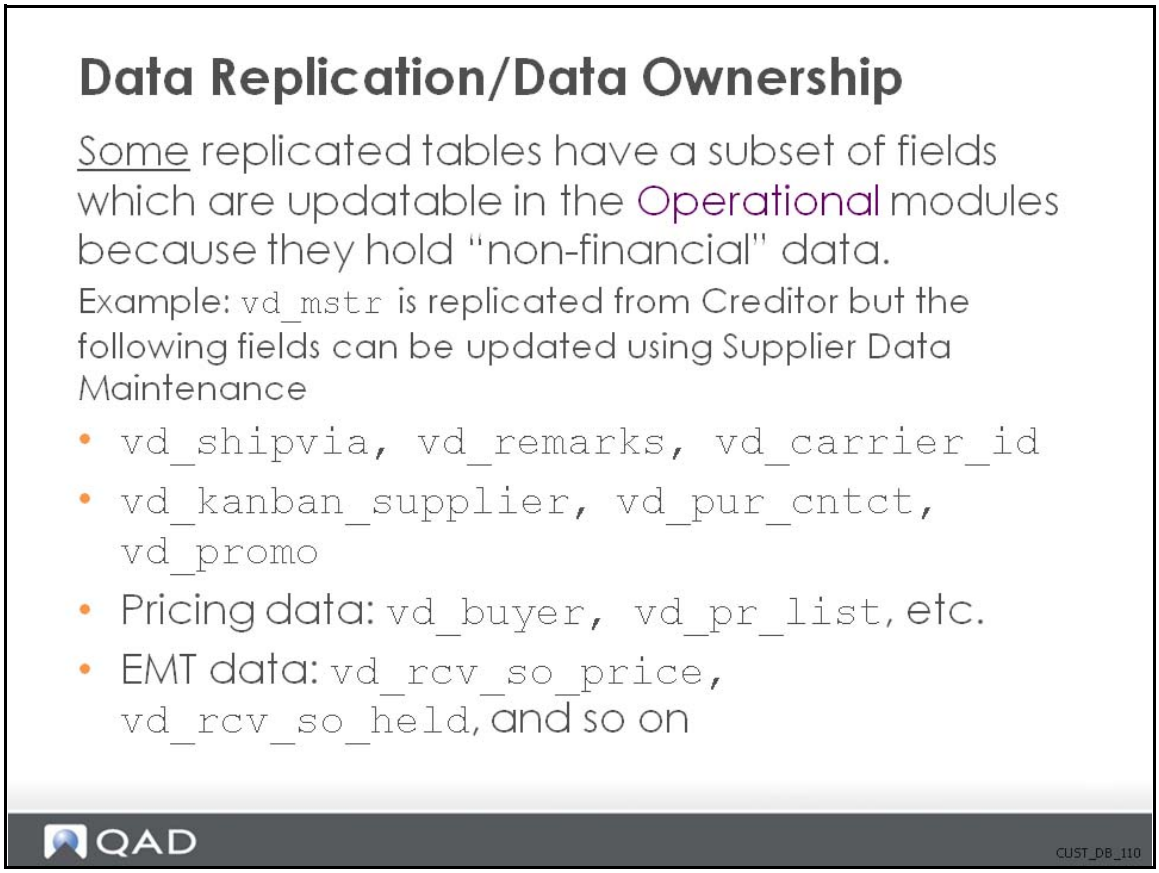
Data Replication/Data Ownership 2



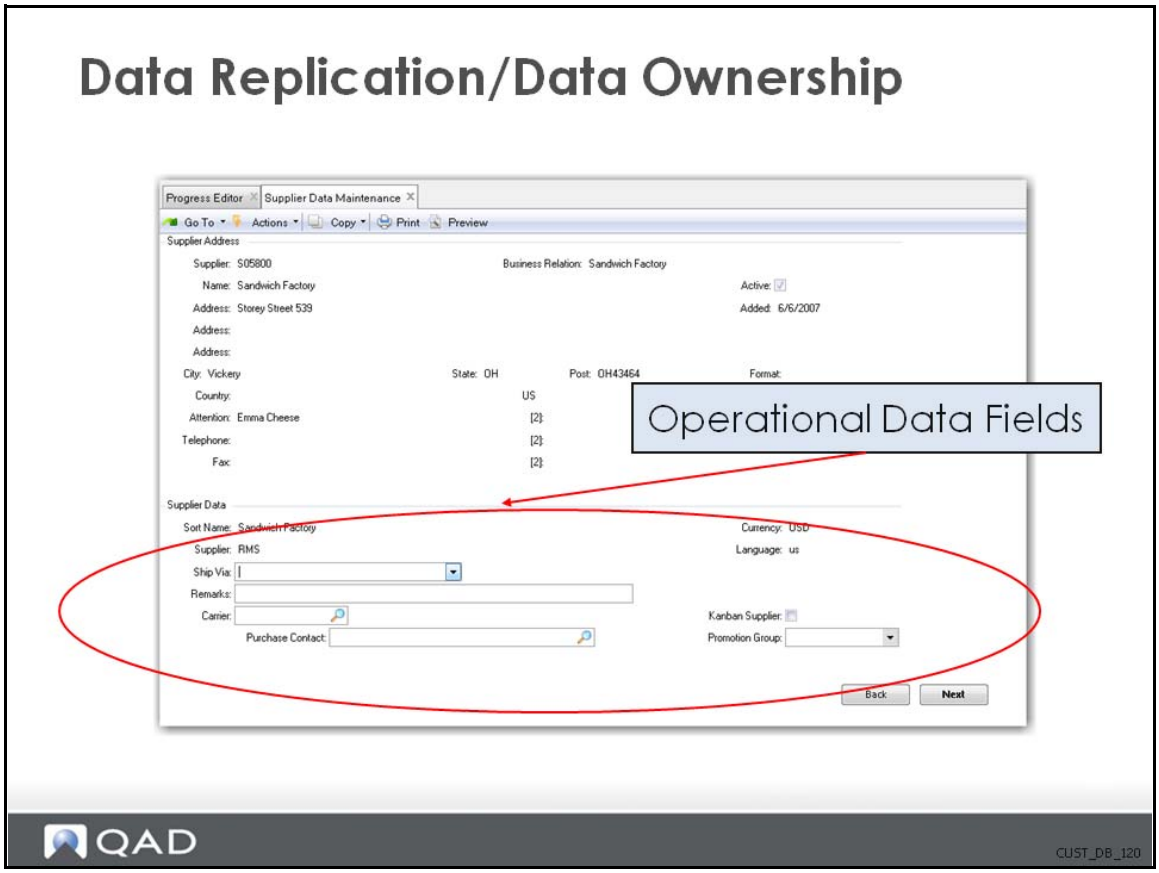
Data Replication/Data Ownership 3



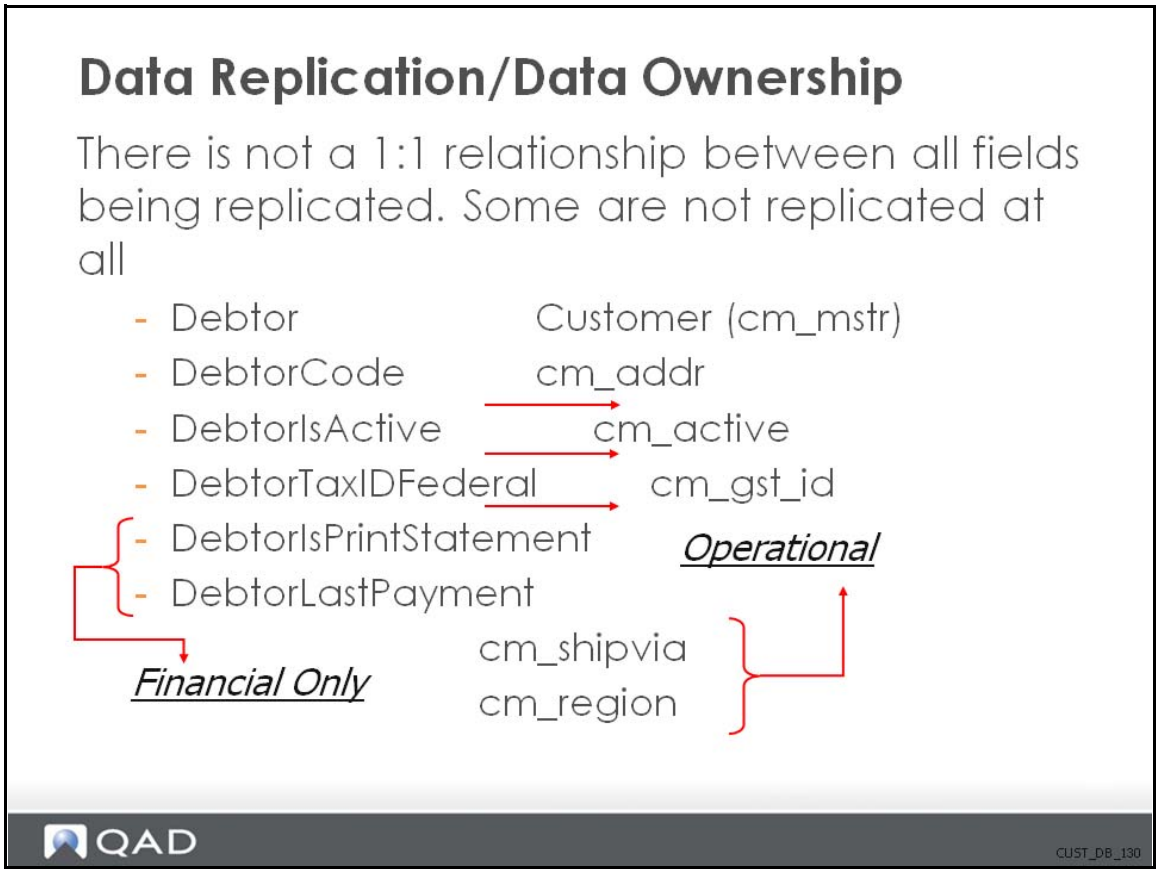
Data Replication/Data Ownership 4



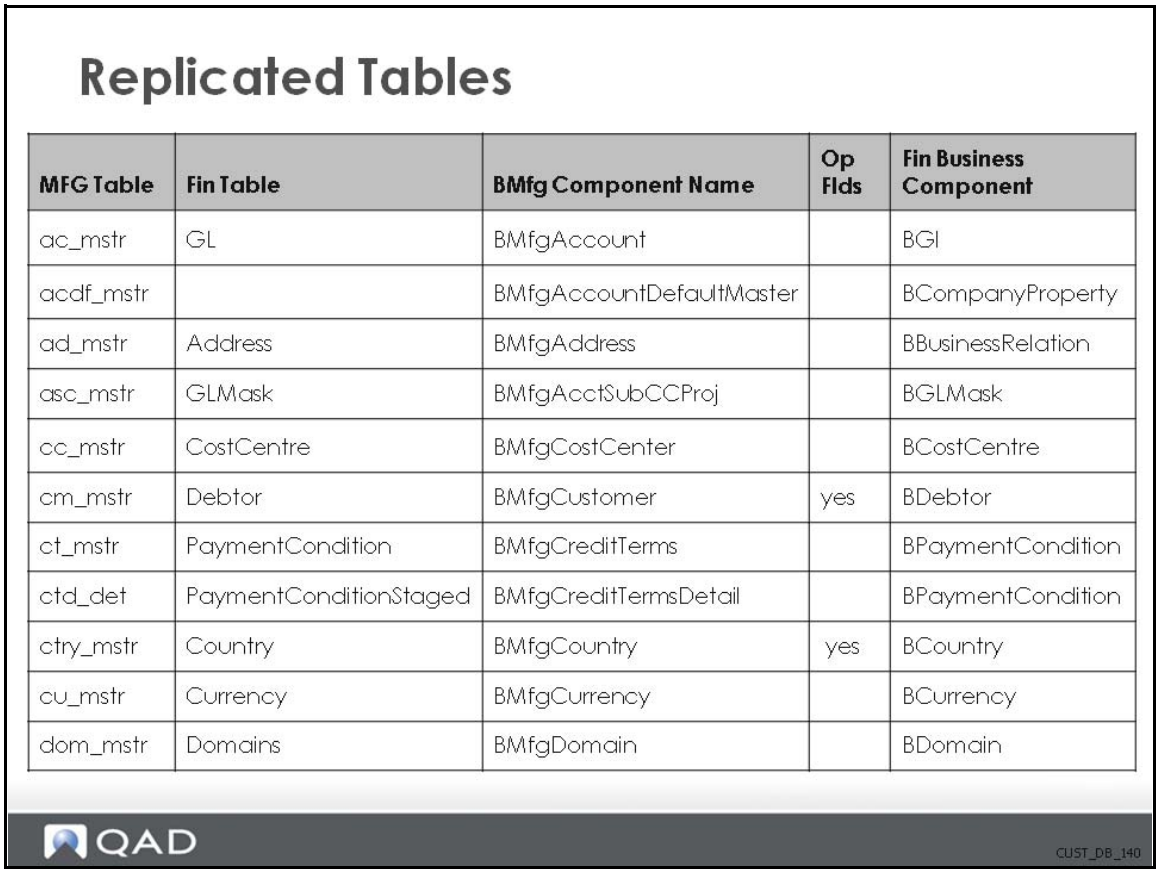
Data Replication/Data Ownership 5



Data Replication/Data Ownership 6



Replicated Tables



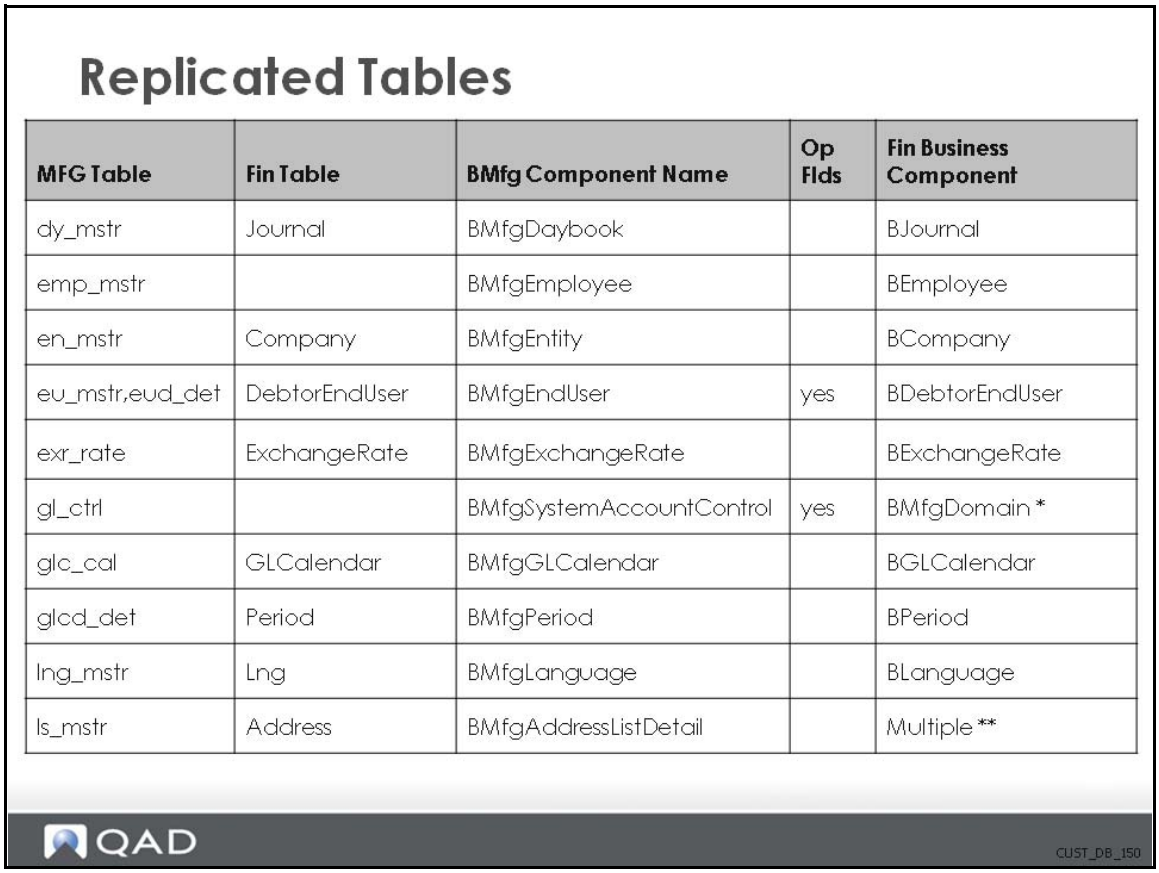
Replicated Tables

MFG Table	Fin Table	BMfg Component Name	Op Flds	Fin Business Component
ac_mstr	GL	BMfgAccount		BGl
acdf_mstr		BMfgAccountDefaultMaster		BCompanyProperty
ad_mstr	Address	BMfgAddress		BBusinessRelation
asc_mstr	GLMask	BMfgAcctSubCCProj		BGLMask
cc_mstr	CostCentre	BMfgCostCenter		BCostCentre
cm_mstr	Debtor	BMfgCustomer	yes	BDebtor
ct_mstr	PaymentCondition	BMfgCreditTerms		BPaymentCondition
ctd_det	PaymentConditionStaged	BMfgCreditTermsDetail		BPaymentCondition
ctry_mstr	Country	BMfgCountry	yes	BCountry
cu_mstr	Currency	BMfgCurrency		BCurrency
dom_mstr	Domains	BMfgDomain		BDomain

QAD

CUST_DB_140

Replicated Tables 2



Replicated Tables

MFG Table	Fin Table	BMfg Component Name	Op Flds	Fin Business Component
dy_mstr	Journal	BMfgDaybook		BJournal
emp_mstr		BMfgEmployee		BEmployee
en_mstr	Company	BMfgEntity		BCompany
eu_mstr,eud_det	DebtorEndUser	BMfgEndUser	yes	BDebtorEndUser
exr_rate	ExchangeRate	BMfgExchangeRate		BExchangeRate
gl_ctrl		BMfgSystemAccountControl	yes	BMfgDomain *
glc_cal	GLCalendar	BMfgGLCalendar		BGLCalendar
glcd_det	Period	BMfgPeriod		BPeriod
lng_mstr	Lng	BMfgLanguage		BLanguage
ls_mstr	Address	BMfgAddressListDetail		Multiple **

QAD

CUST_DB_150

Replicated Tables 3

Replicated Tables

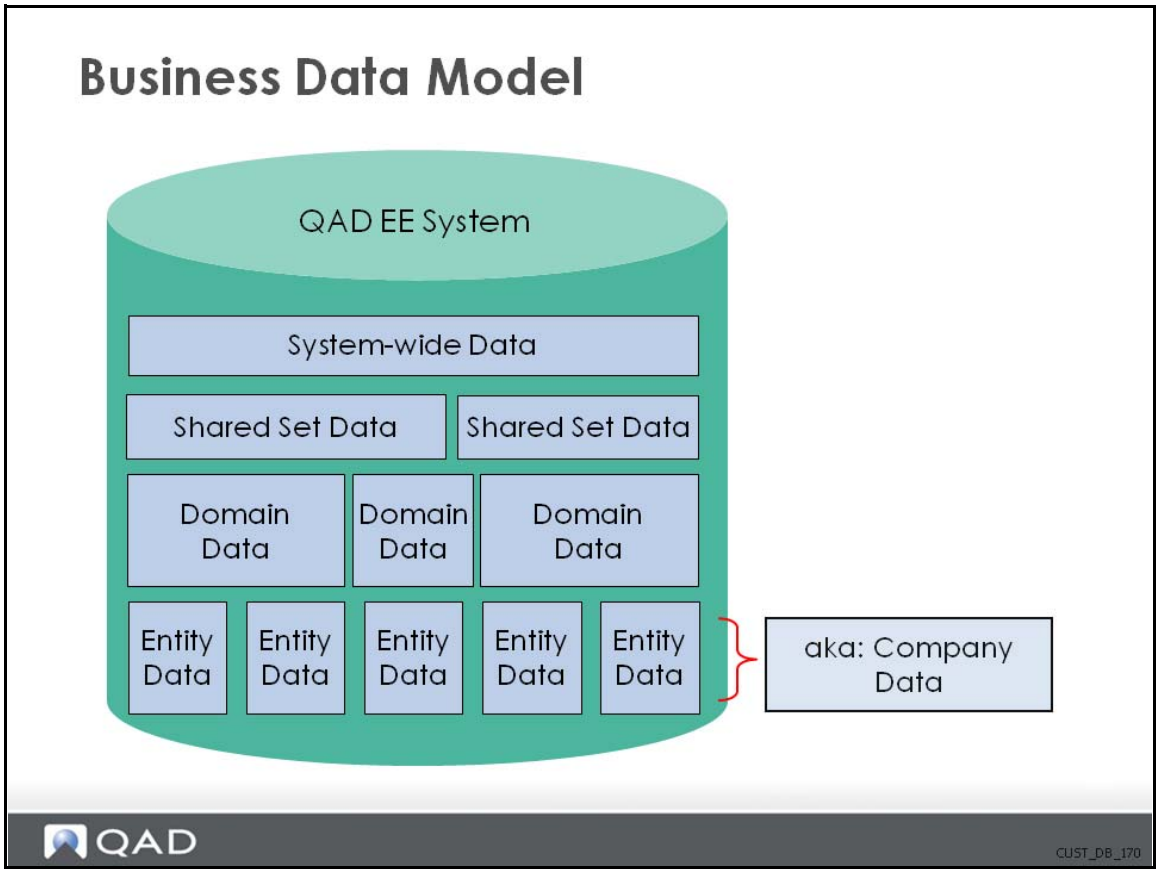
- The only tables currently replicated from ERP to Financials are usr_mstr & tx2_mstr
- There is also some "replication" of Resource data from Menu Maintenance

MFG Table	Fin Table	BMfg Component Name	Op Flds	Fin Business Component
nr_mstr	JournalBook	BMfgNumberRangeManagement	**	BJournal
pj_mstr	Project	BMfgProject		BProject
rnd_mstr	RoundingMethod	BMfgRoundingMethod		BRoundingMethod
sb_mstr	Division	BMfgSubAccount		BDivision
udd_det	UsrDomain	BMfgUserDomainForUser		BUser
usrg_mstr	Role	BMfgRole		BRole
usrgd_det	UserRoleDomain	BMfgUserRole		BUserRole
vd_mstr	Creditor	BMfgSupplier	yes	BCreditor

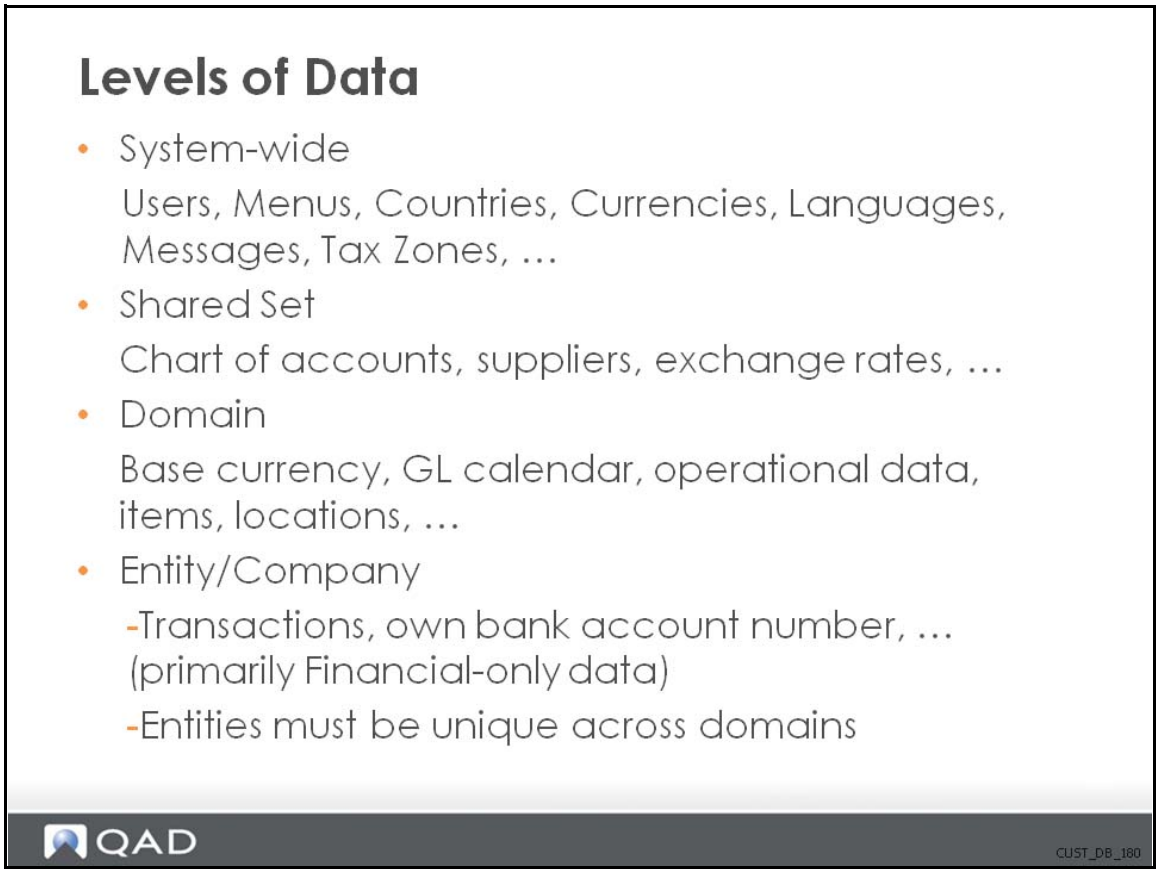
QAD

CUST_DB_160

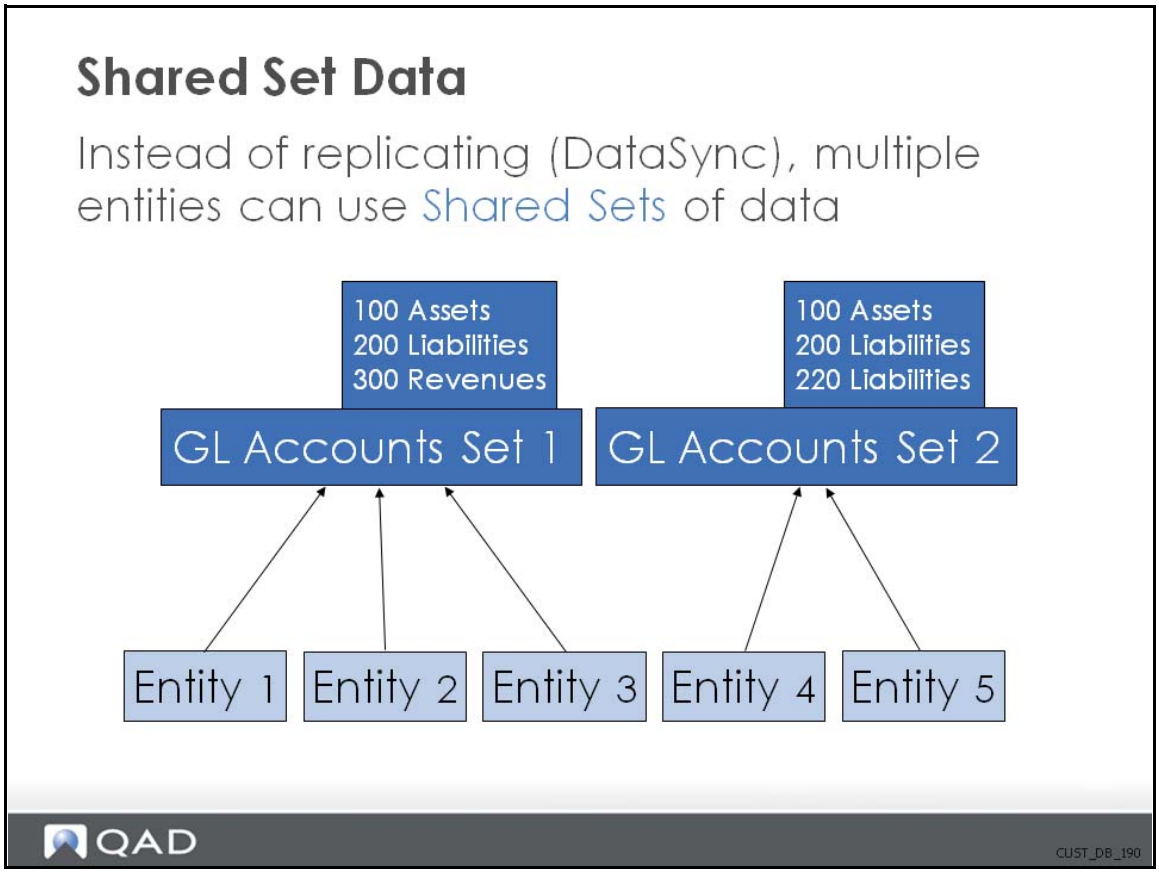
Business Data Model



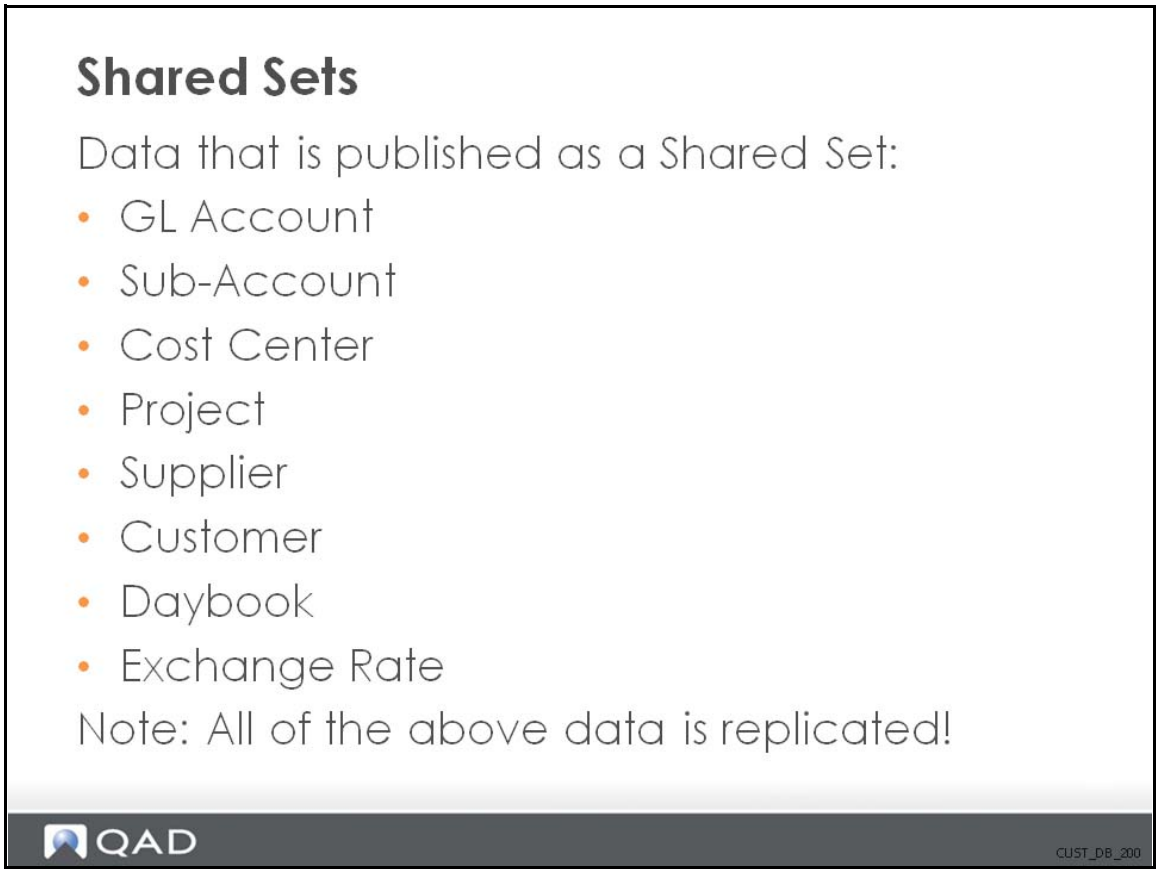
Levels of Data



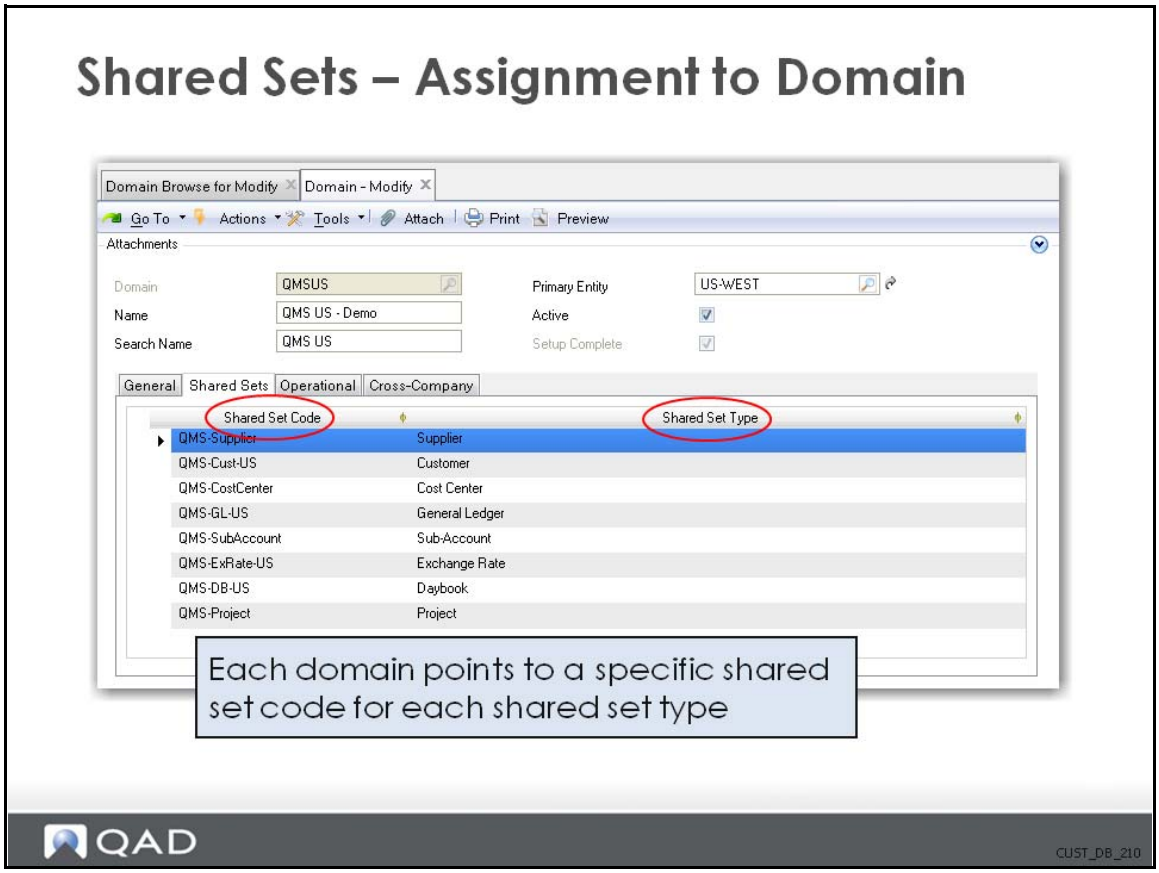
Shared Set Data



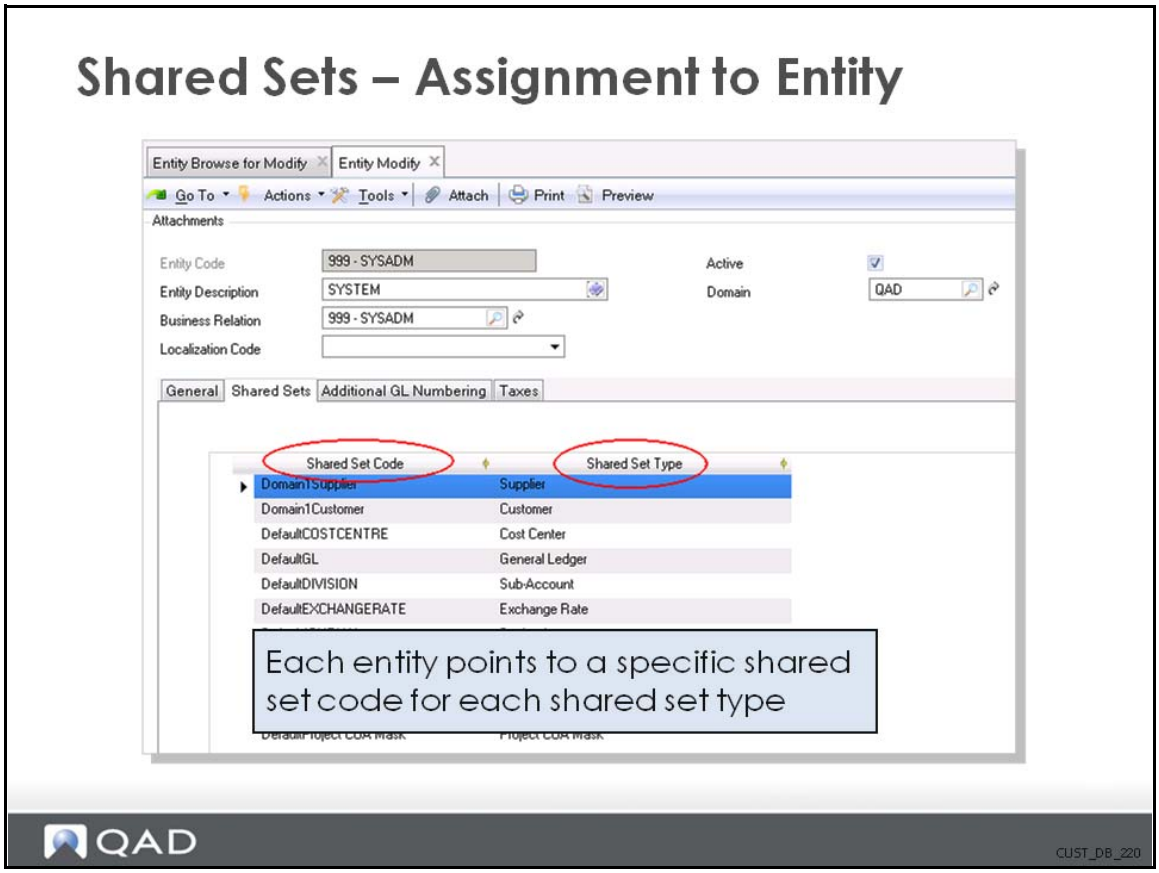
Shared Sets


Shared Sets
Data that is published as a Shared Set:
• GL Account
• Sub-Account
• Cost Center
• Project
• Supplier
• Customer
• Daybook
• Exchange Rate
Note: All of the above data is replicated!
QAD
CUST_DB_200

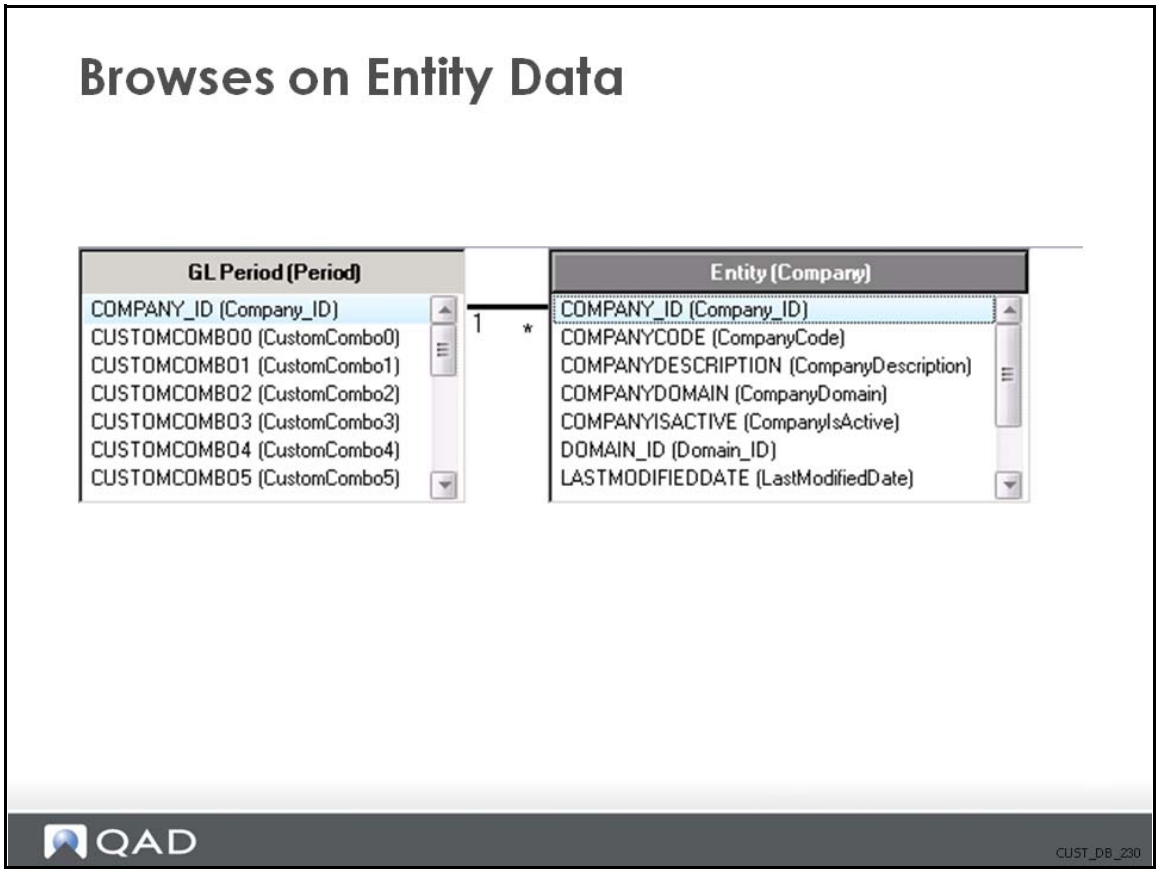
Shared Sets - Assignment to Domain



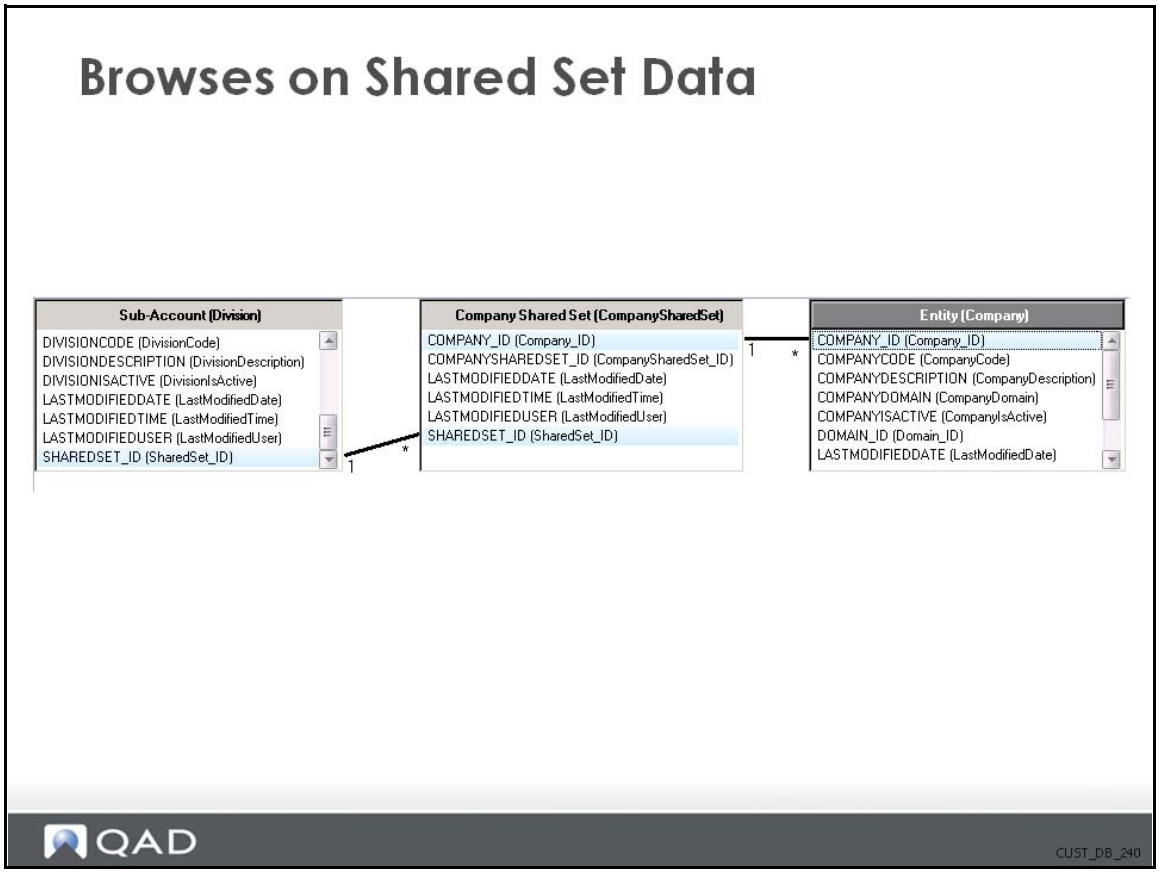
Shared Sets - Assignment to Entity



Browses on Entity Data



Browses on Shared Data

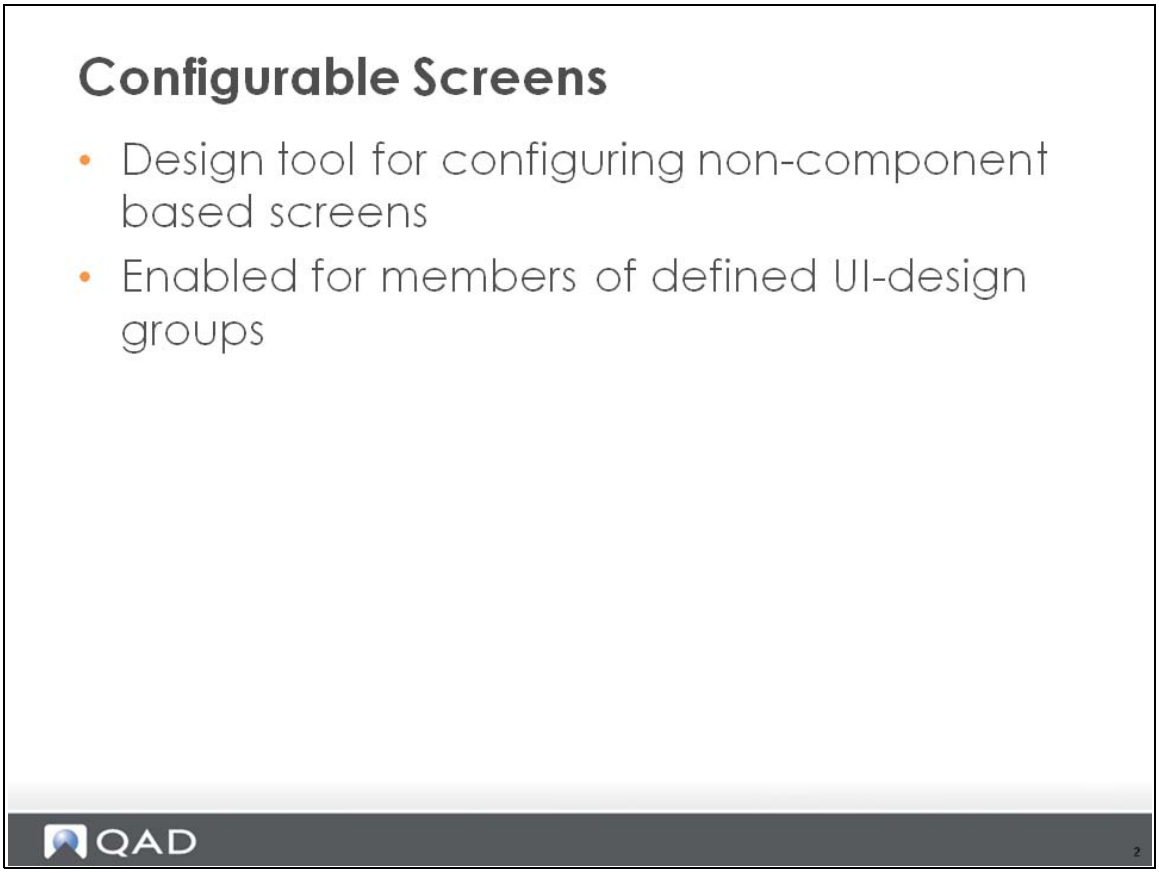


Chapter 3

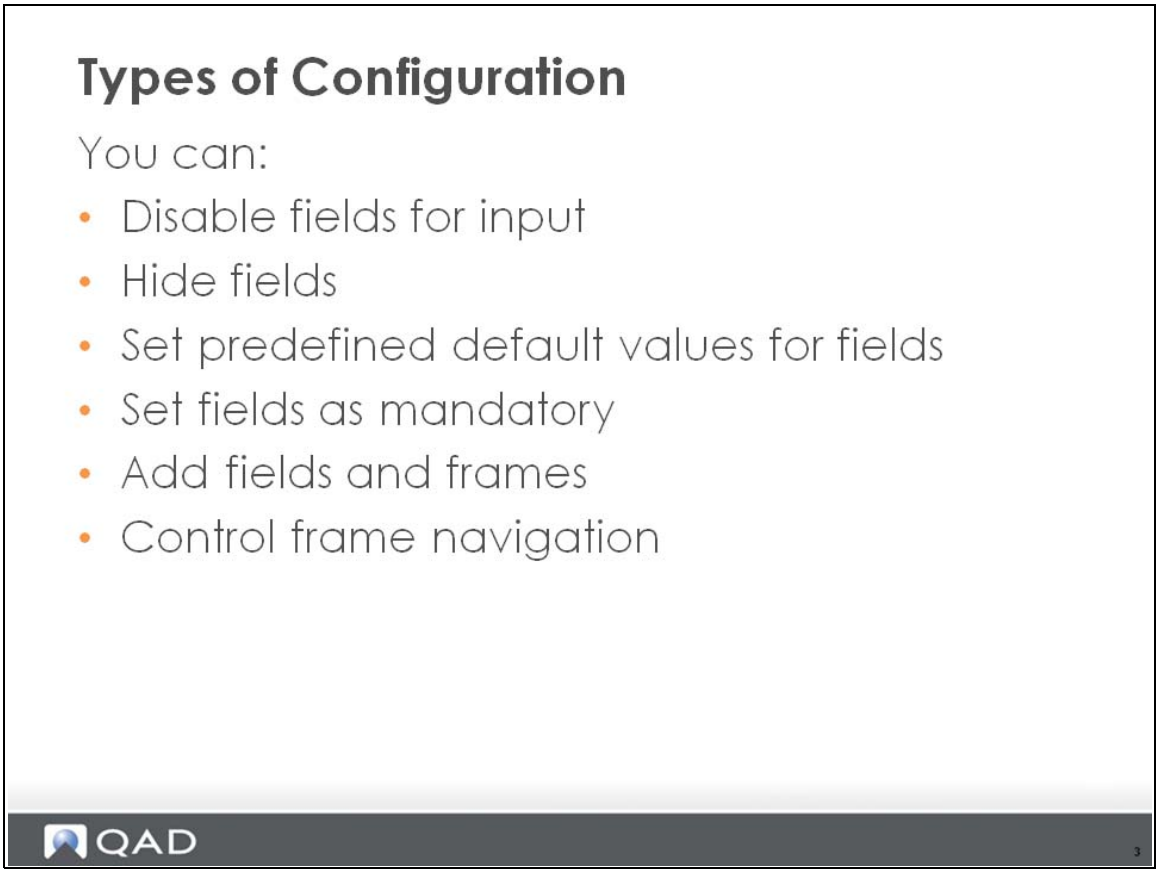
Customization Using UI Design Mode

Customization Using QAD Application Features and Functions UI Design Mode

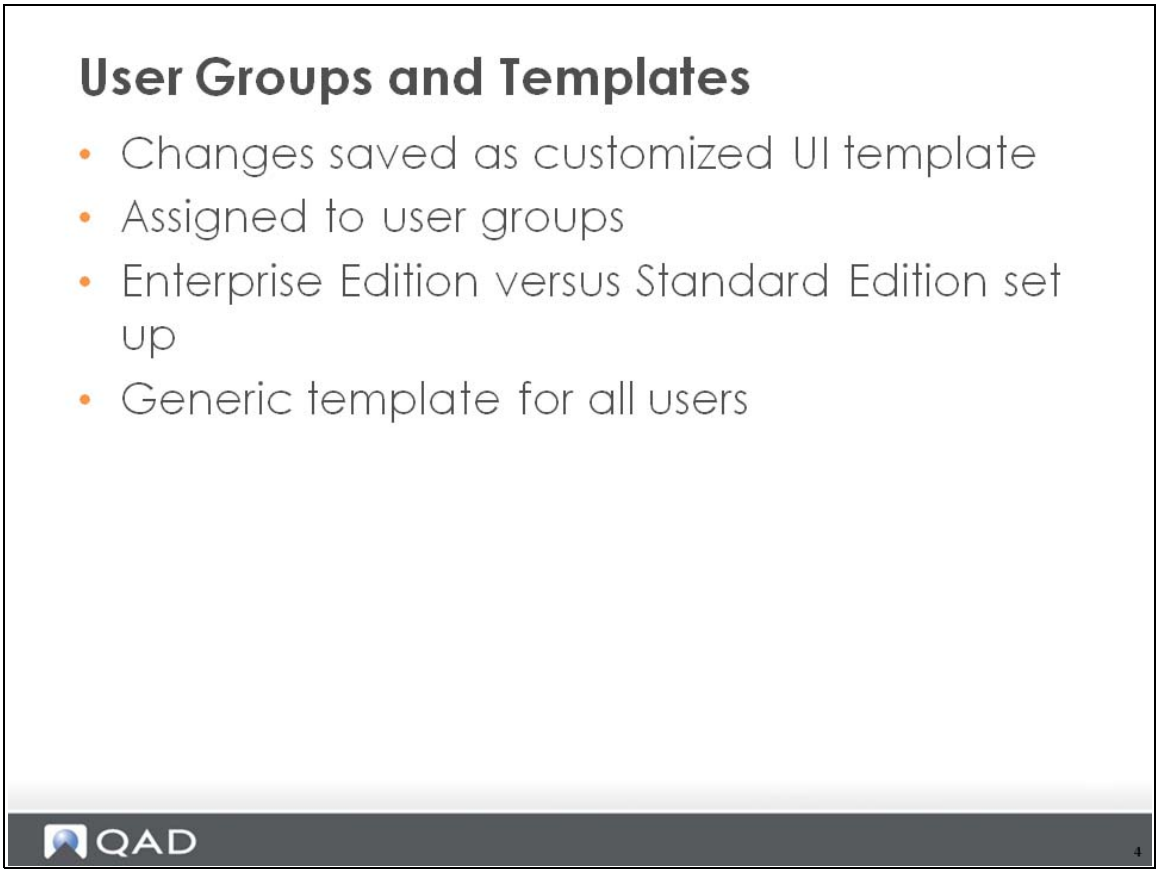
Configurable Screens



Types of Configuration



User Groups and Templates



Error Handling and Template Conflicts

Error Handling and Template Conflicts

- Error messages displayed when mandatory fields hidden or removed
- Possible conflicts when users belong to more than one group
- Conflict management tool to list users, programs, and conflicts

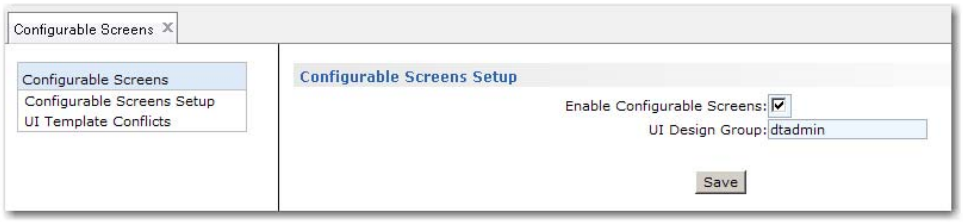
QAD

5

Configurable Screens Setup

Configurable Screens Setup

- Enable the function
- Specify the role that can use this design group
- Save to enable



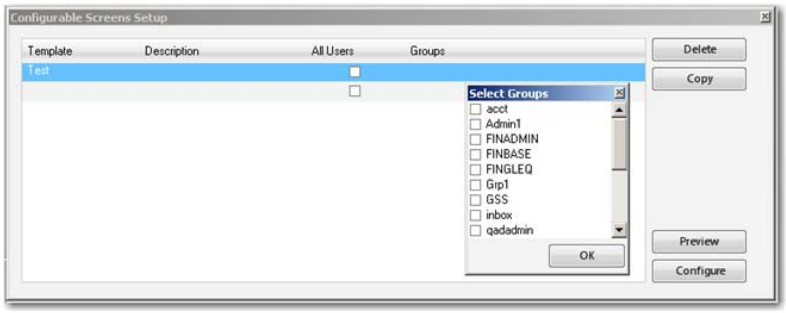
QAD

6

Creating a Template

Creating a Template

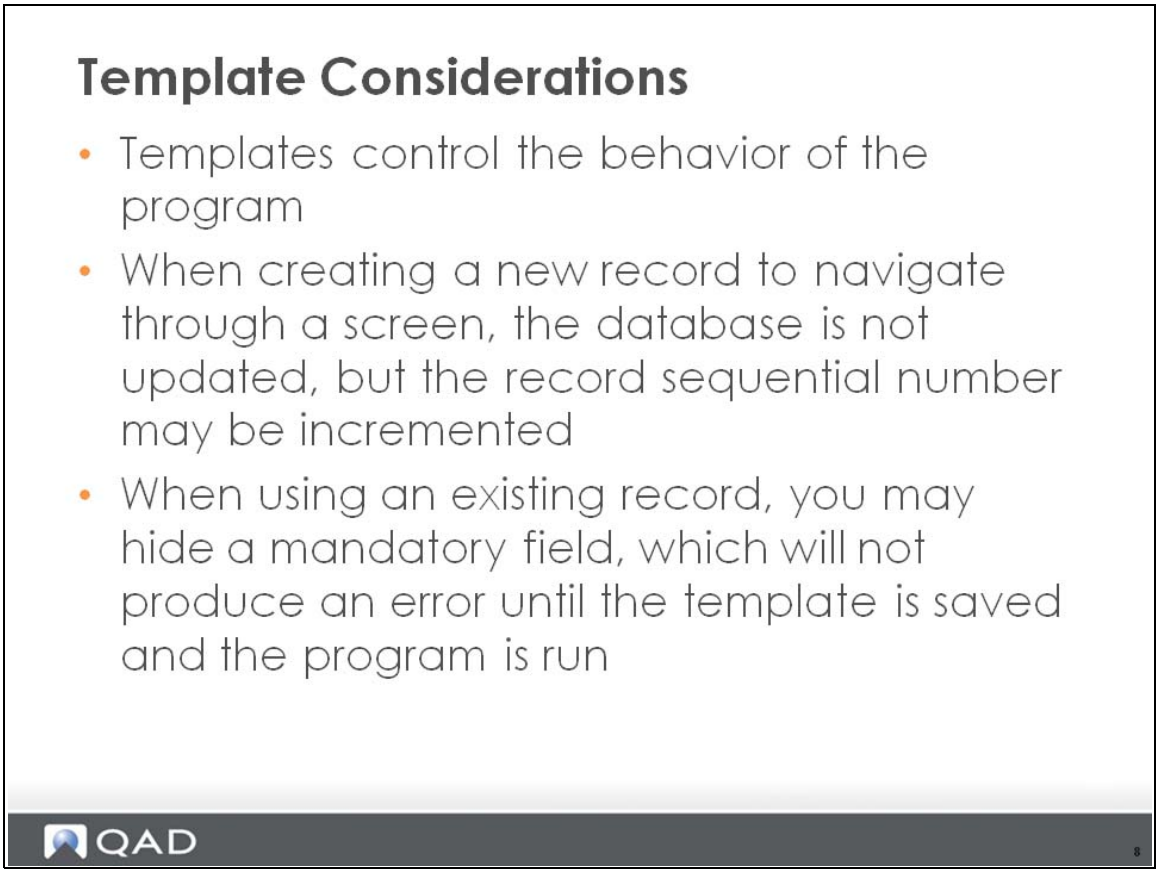
- Select the program to configure
- Right-click to select Design
- Enter a name, description, and roles for the template
- Click Configure to enter Design mode



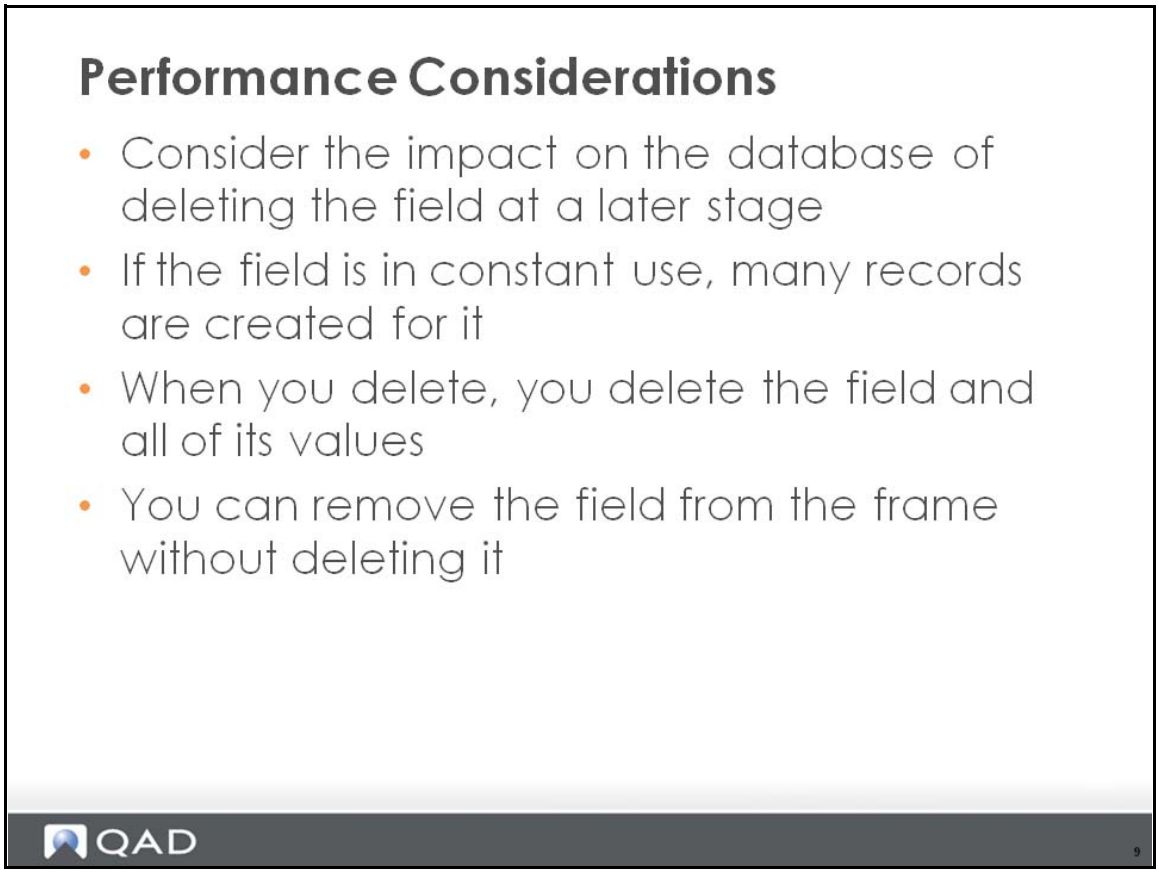
QAD

7

Template Considerations



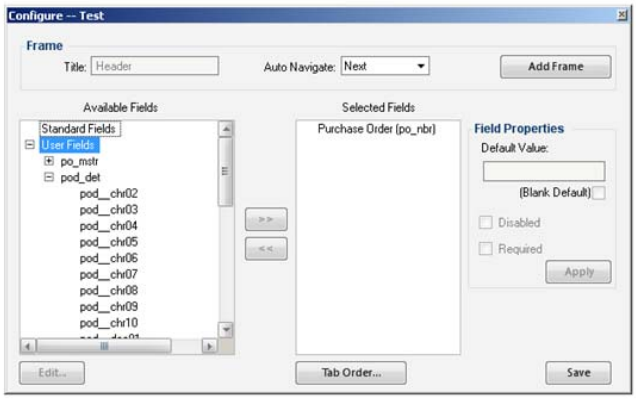
Performance Considerations



Configure Screen

Configure Screen

- Auto Navigate
- Available and Selected Fields
- Field Properties
- Tab Order



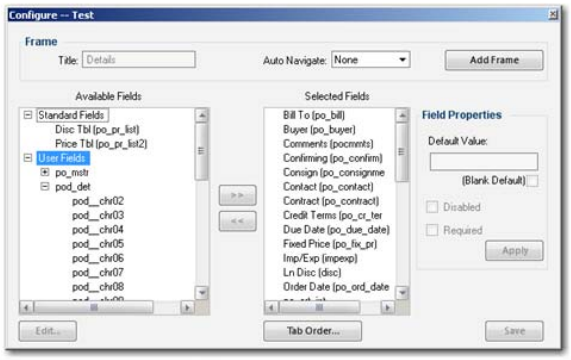
QAD

10

Available Fields

Available Fields

- Standard Fields and User Fields
- Tables and fields defined in `configscreens.xml`
- Add new user fields to `configscreens.xml`
- Default values for fields

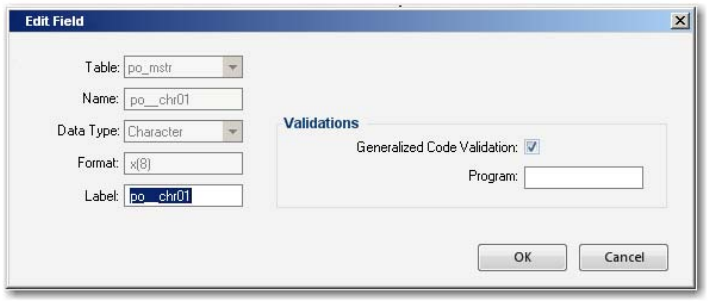


QAD 11

Field Properties

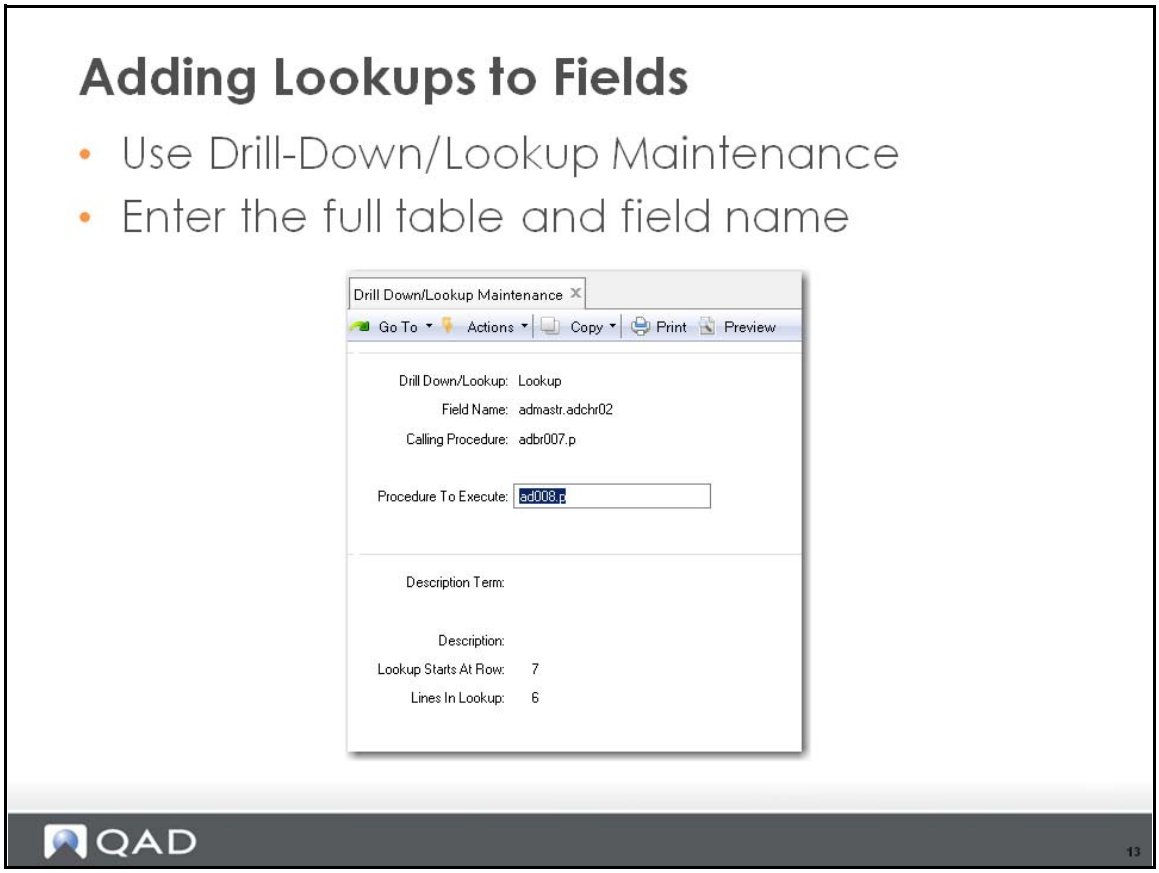
Field Properties

- Editing Field Properties
- Field Validation
- Generalized Codes
- Program Validation

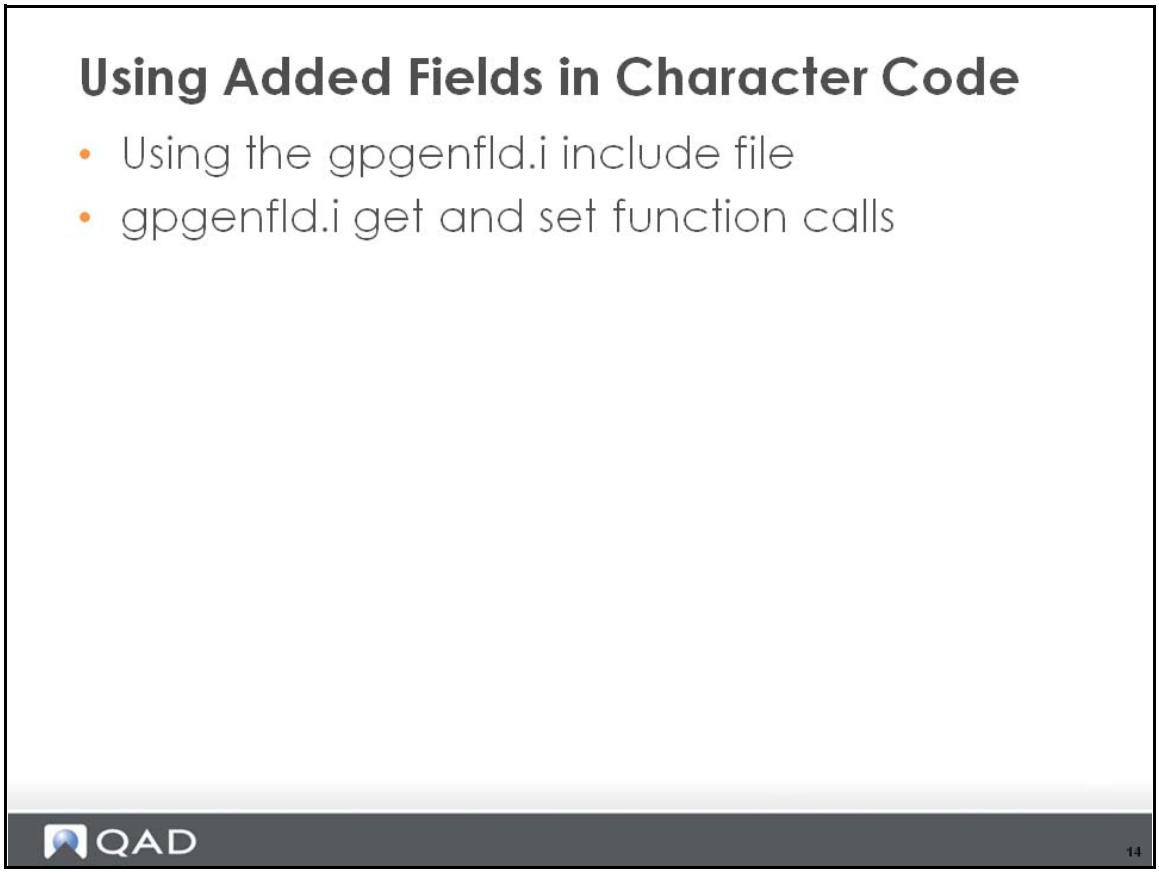


QAD 12

Adding Lookups to Fields



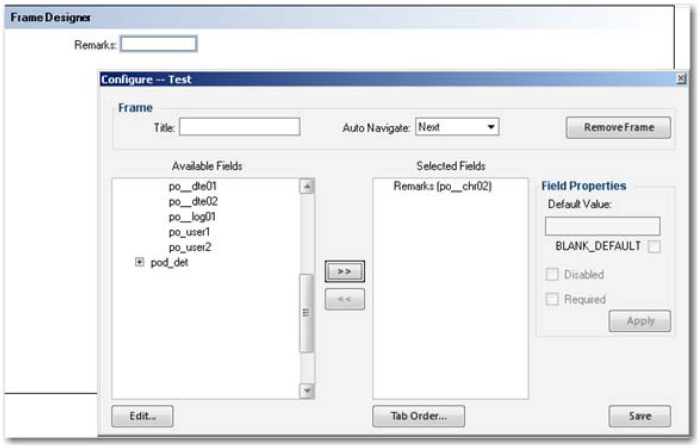
Using Added Fields in Character Code



Adding Frames

Adding Frames

- Frames option is only available when fields are available
- Use Tab Order to set the tabbing order



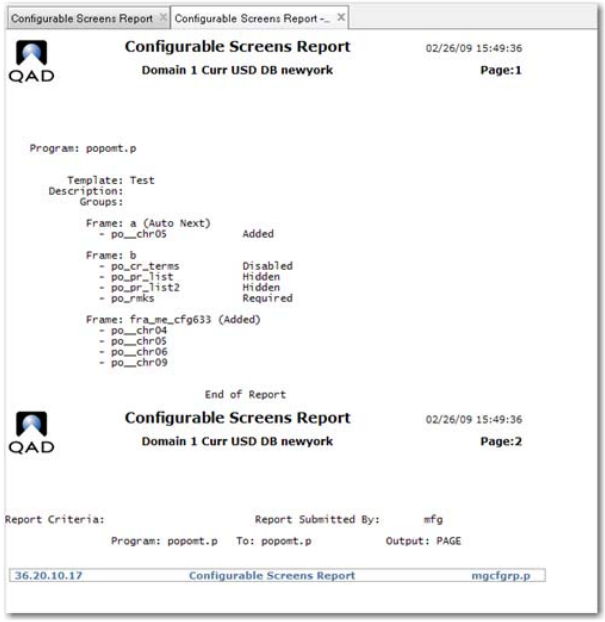
QAD

15

Configurable Screens Report

Configurable Screens Report

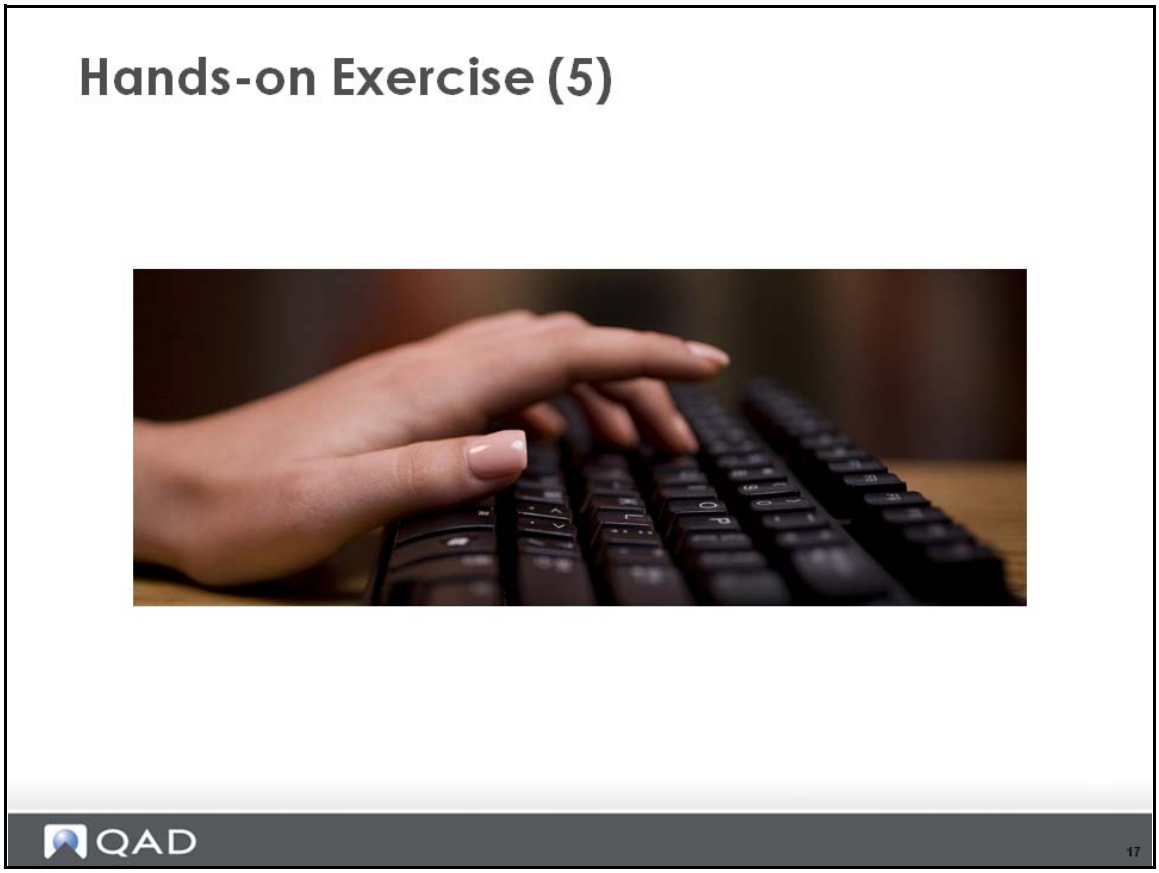
- All configured screens
- Details on individual fields



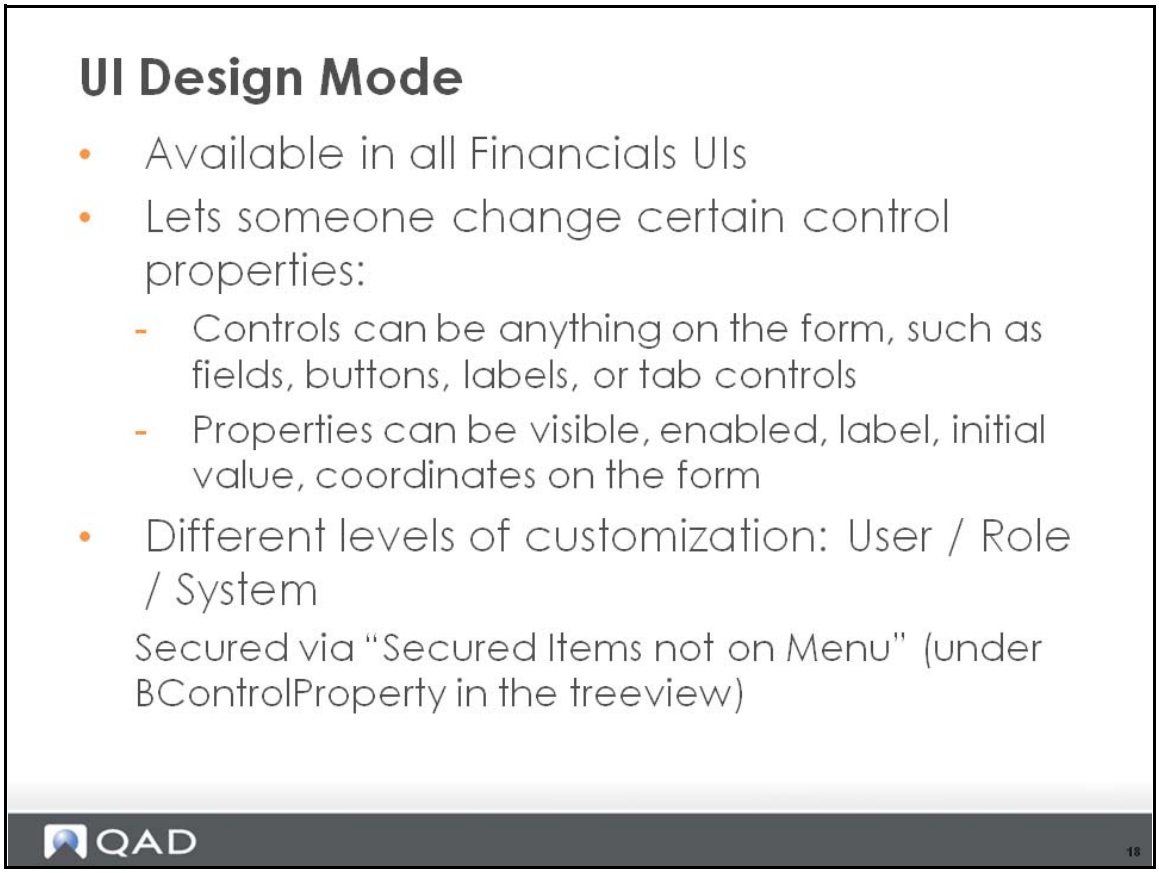
QAD

16

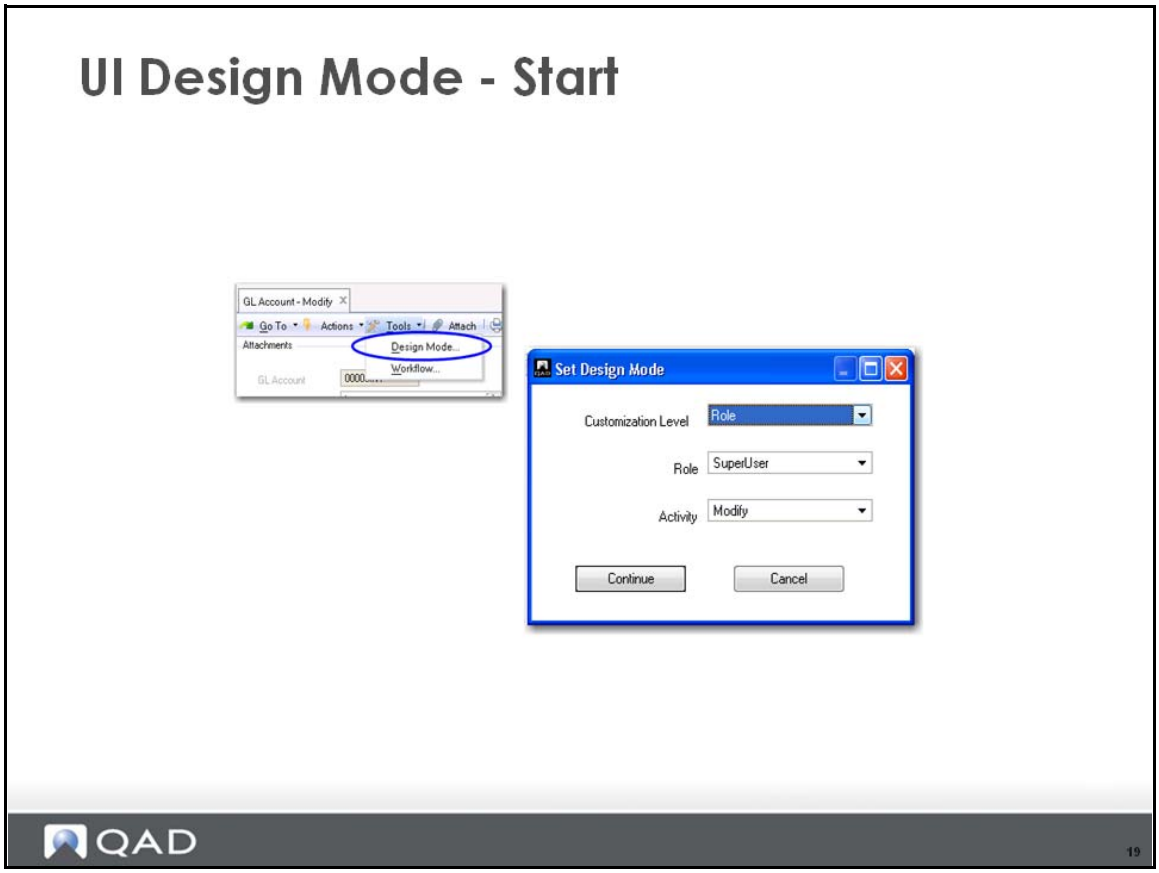
Hands-on Exercise (5)



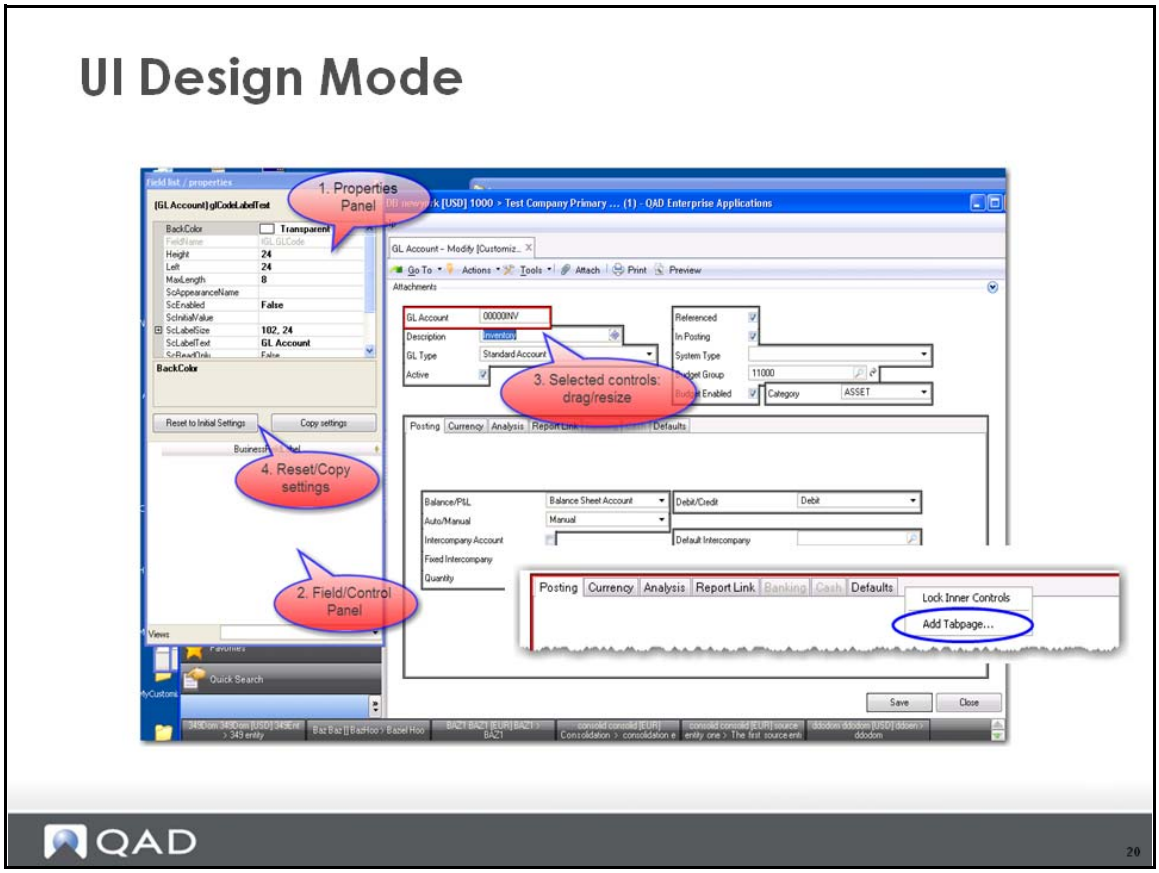
UI Design Mode



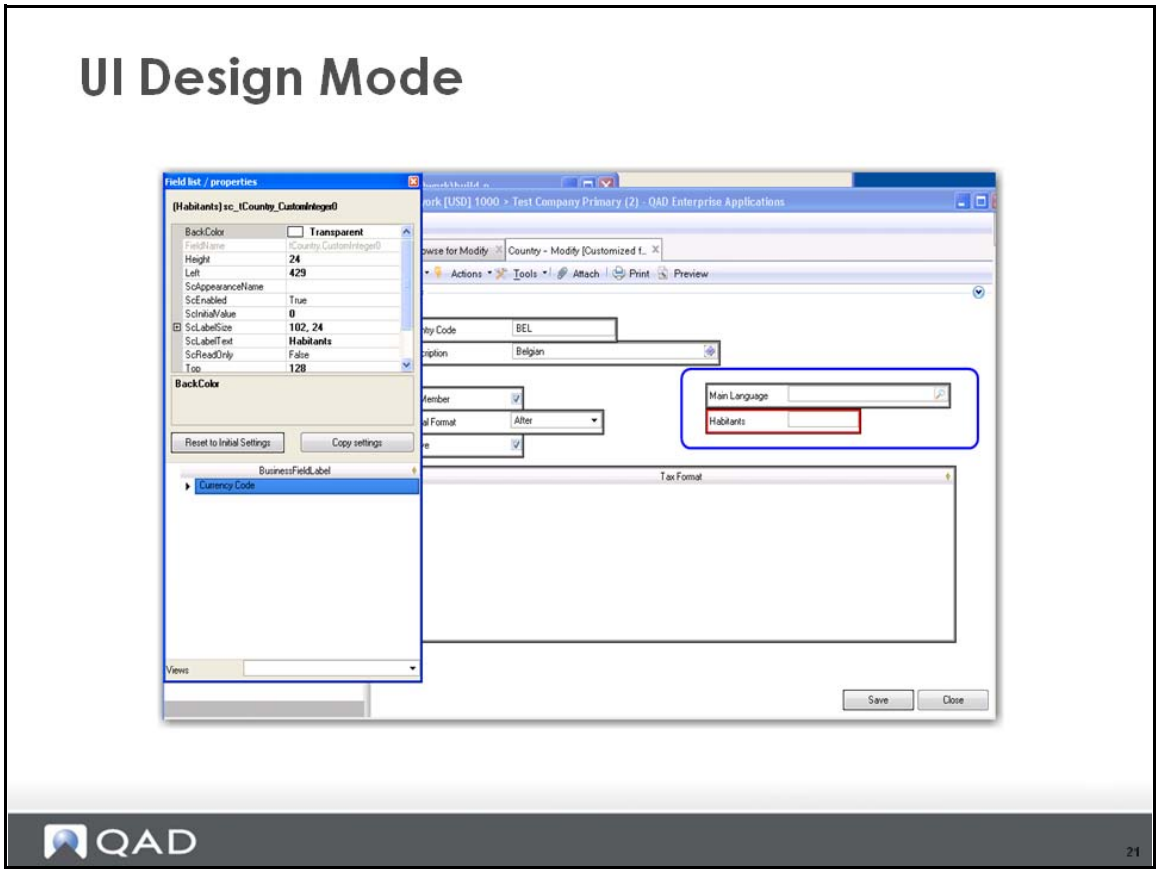
UI Design Mode 2



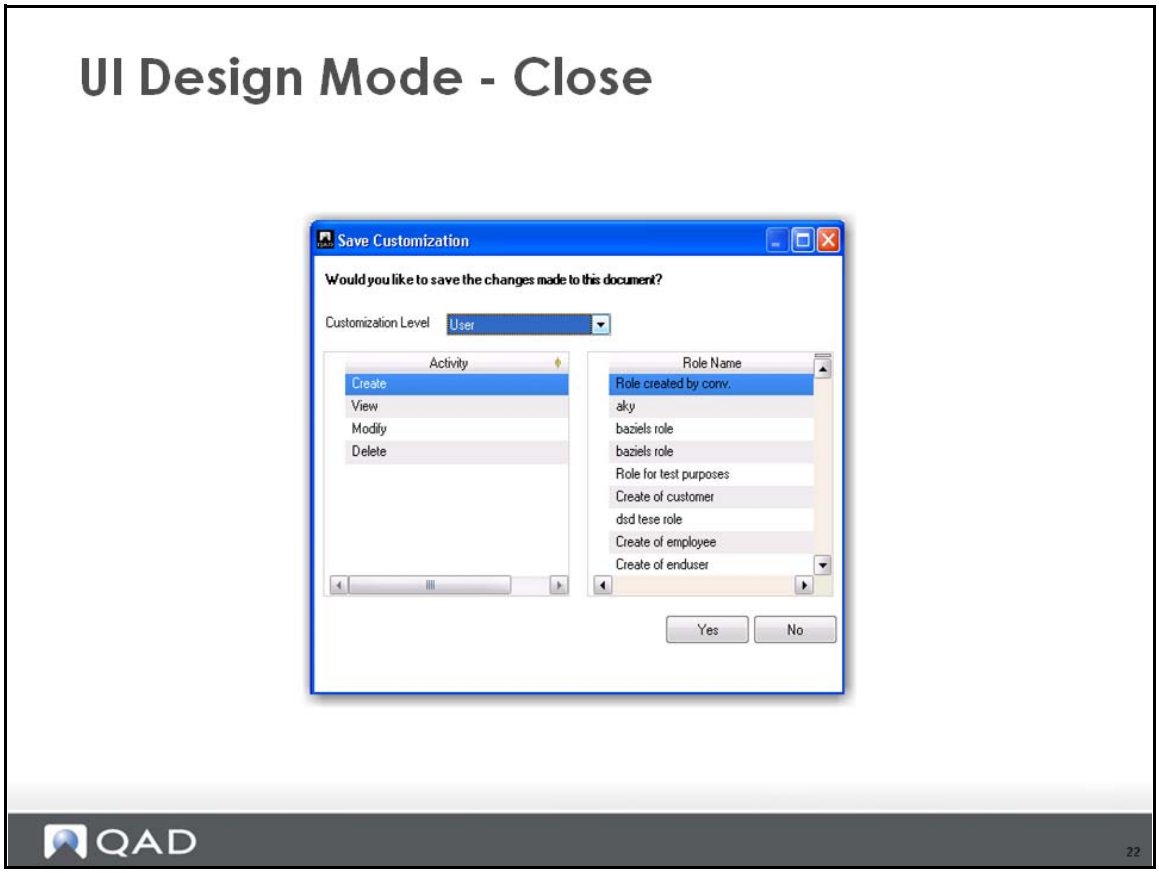
UI Design Mode 3



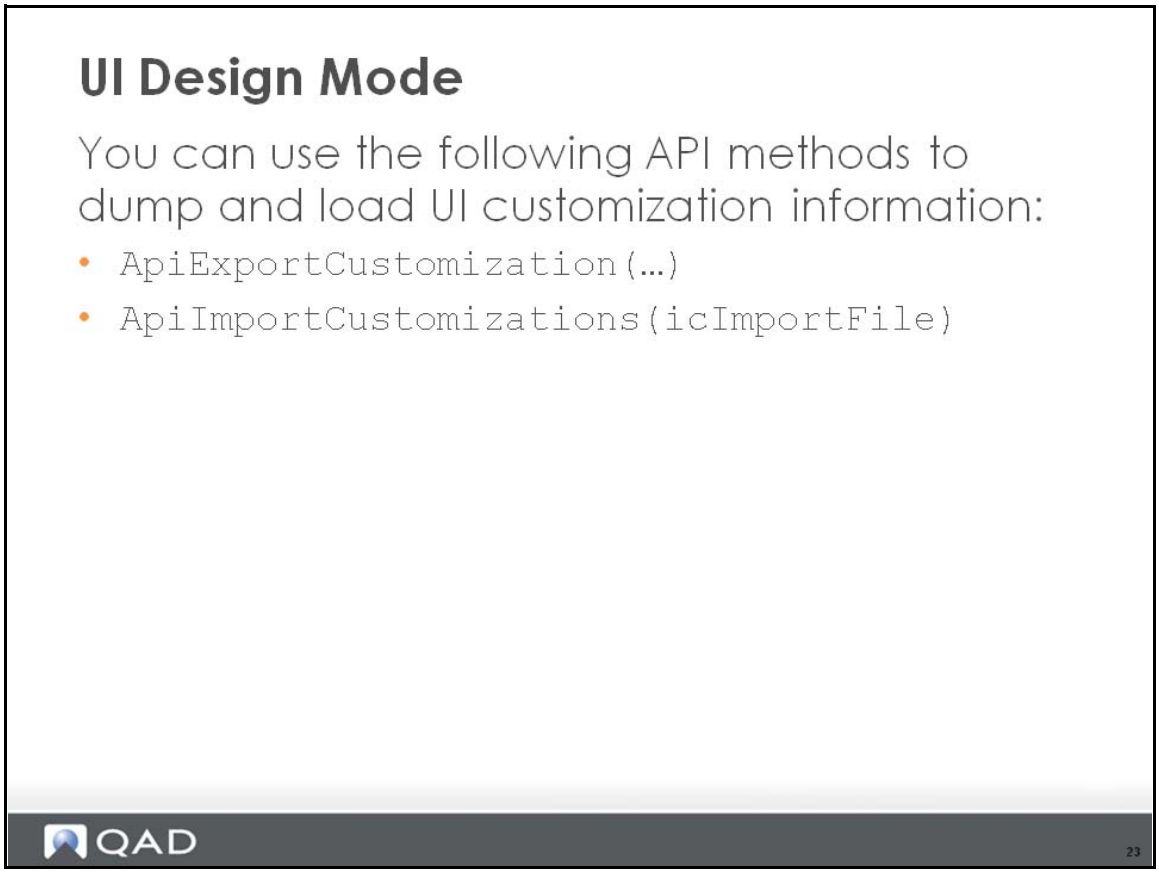
UI Design Mode 4



UI Design Mode 5



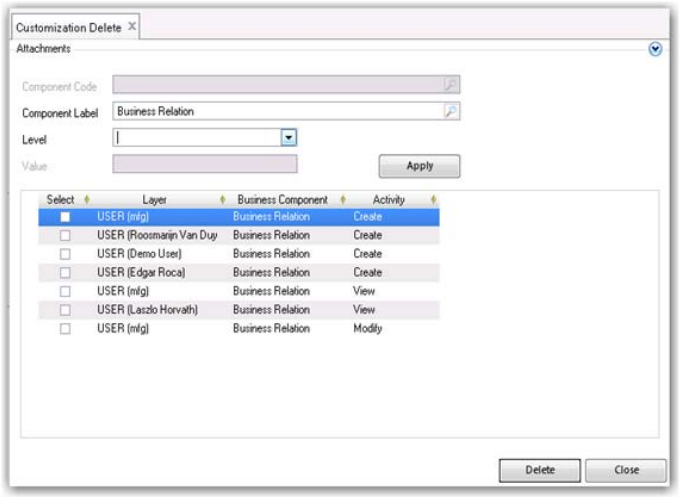
UI Design Mode 6



UI Design Mode 7

UI Design Mode - Maintenance

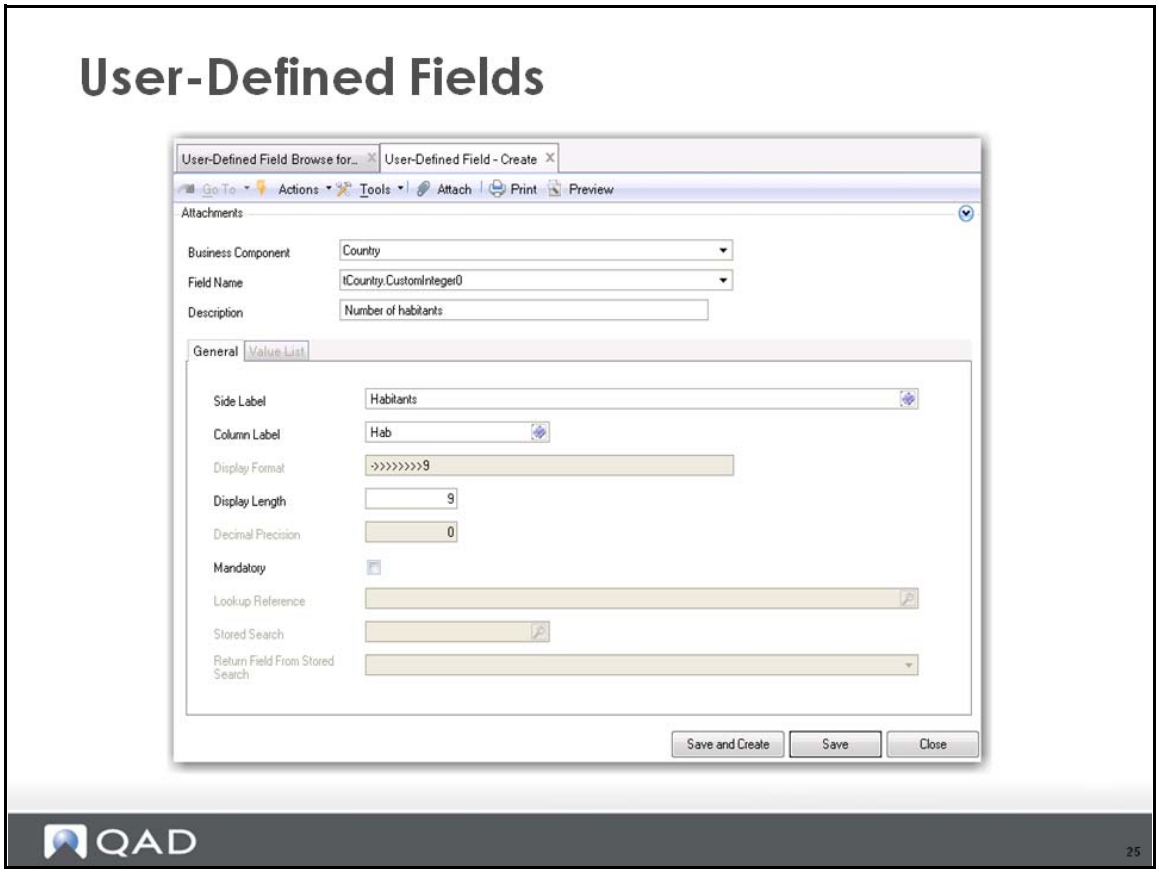
- Definition and update via UI Design Mode
- Deletion via Customization Delete



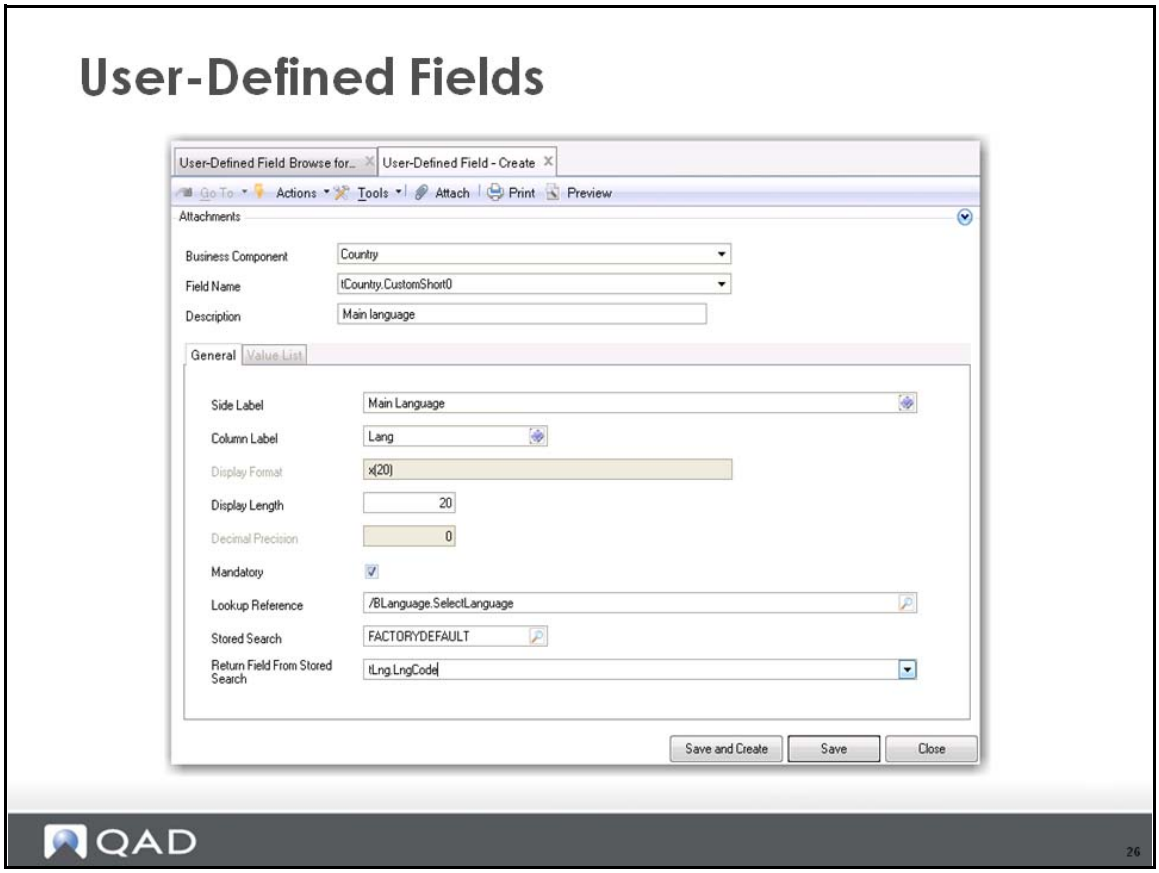
QAD

24

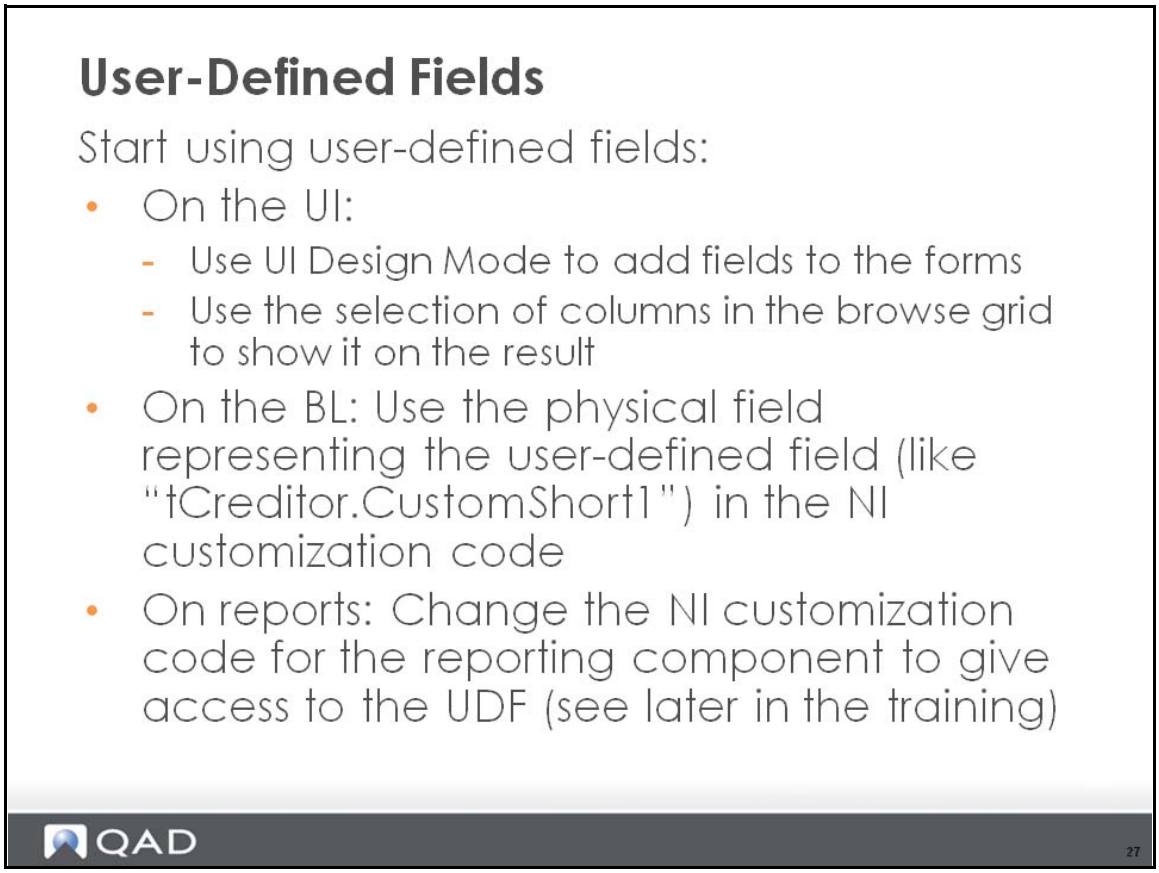
User-Defined Fields



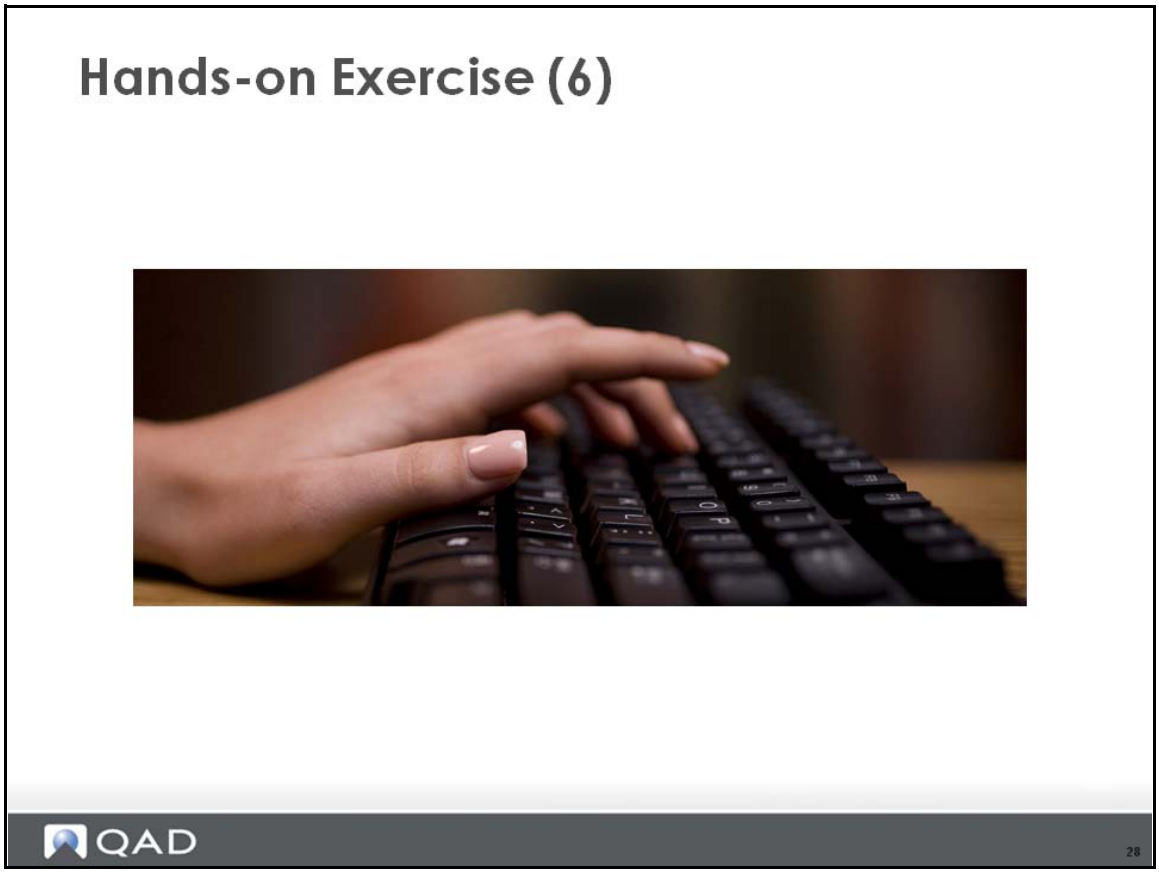
User-Defined Fields 2



User-Defined Fields 3



Hands-on Exercise (6)



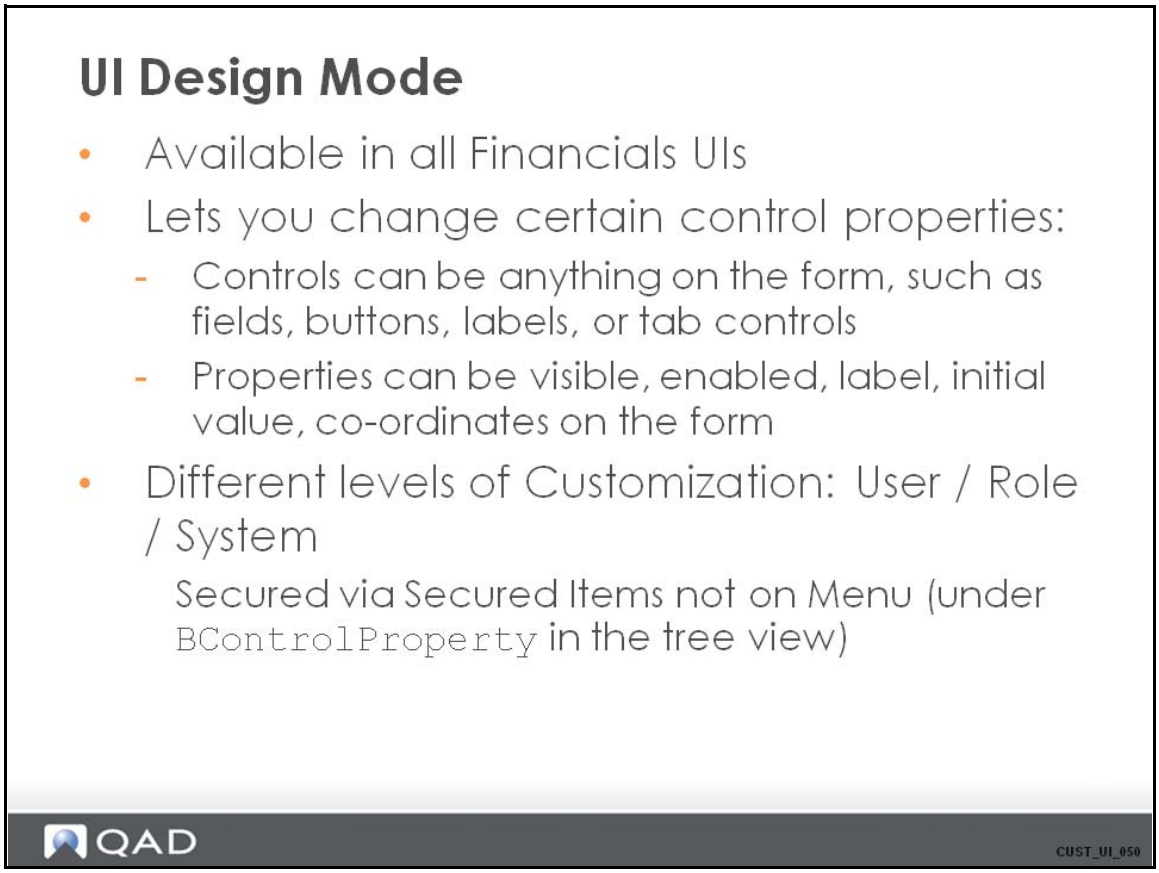
Chapter 4

Non-intrusive Customization of Enterprise Financials

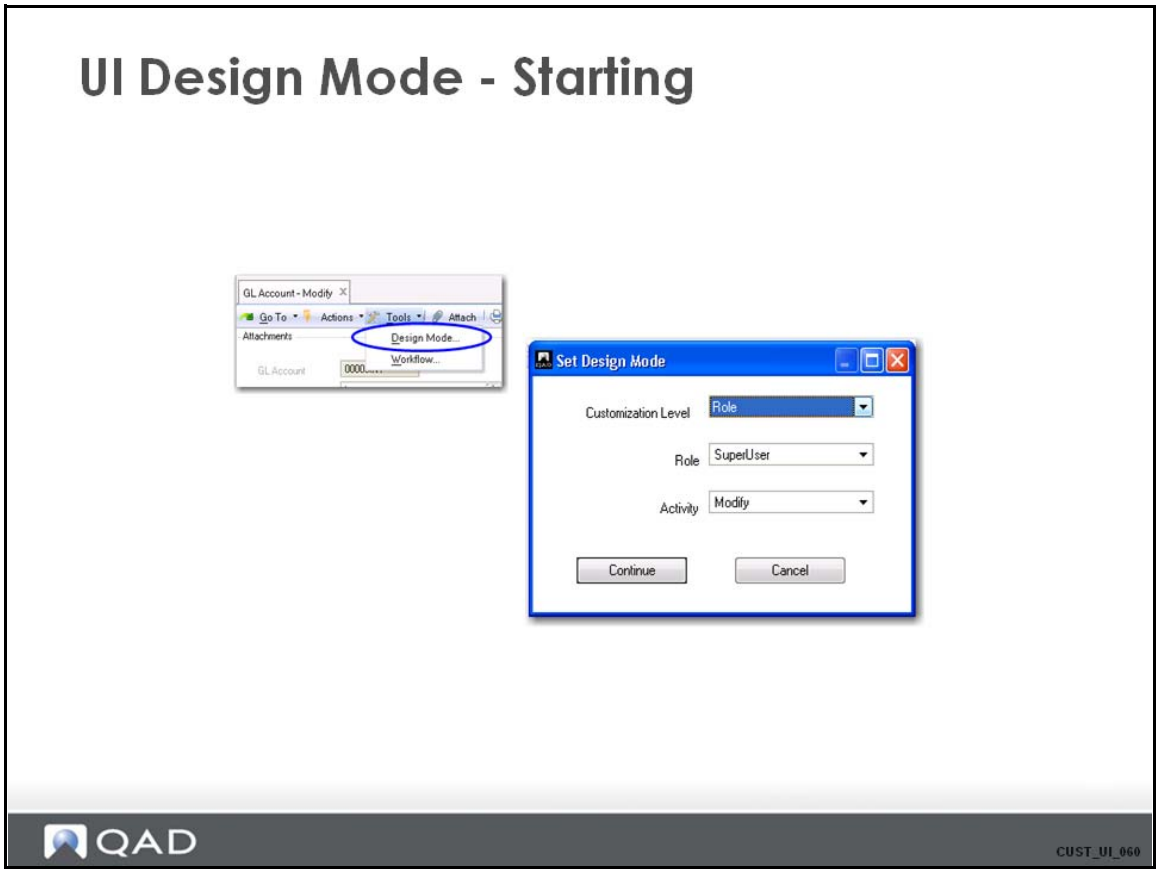
Non-intrusive Customizations of Enterprise Financials

Non-Intrusive Customization: UI Design

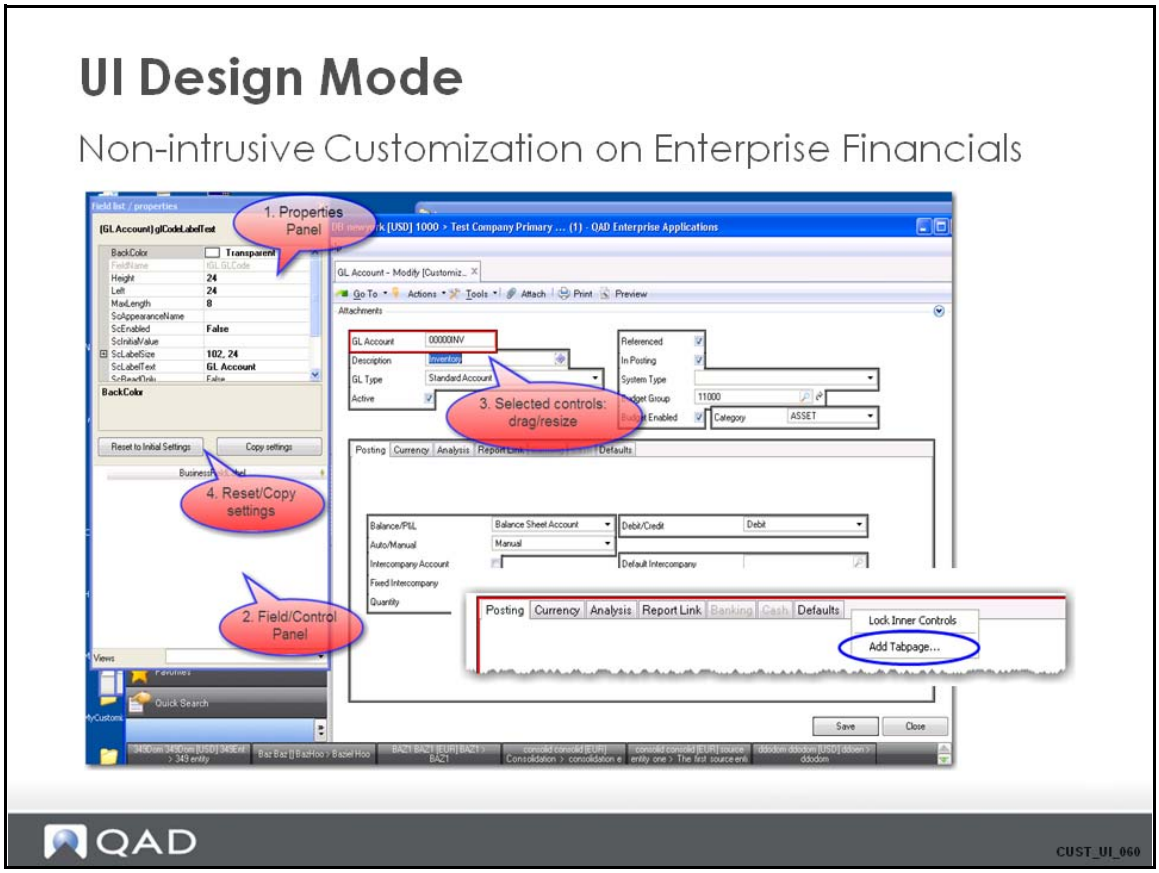
UI Design Mode



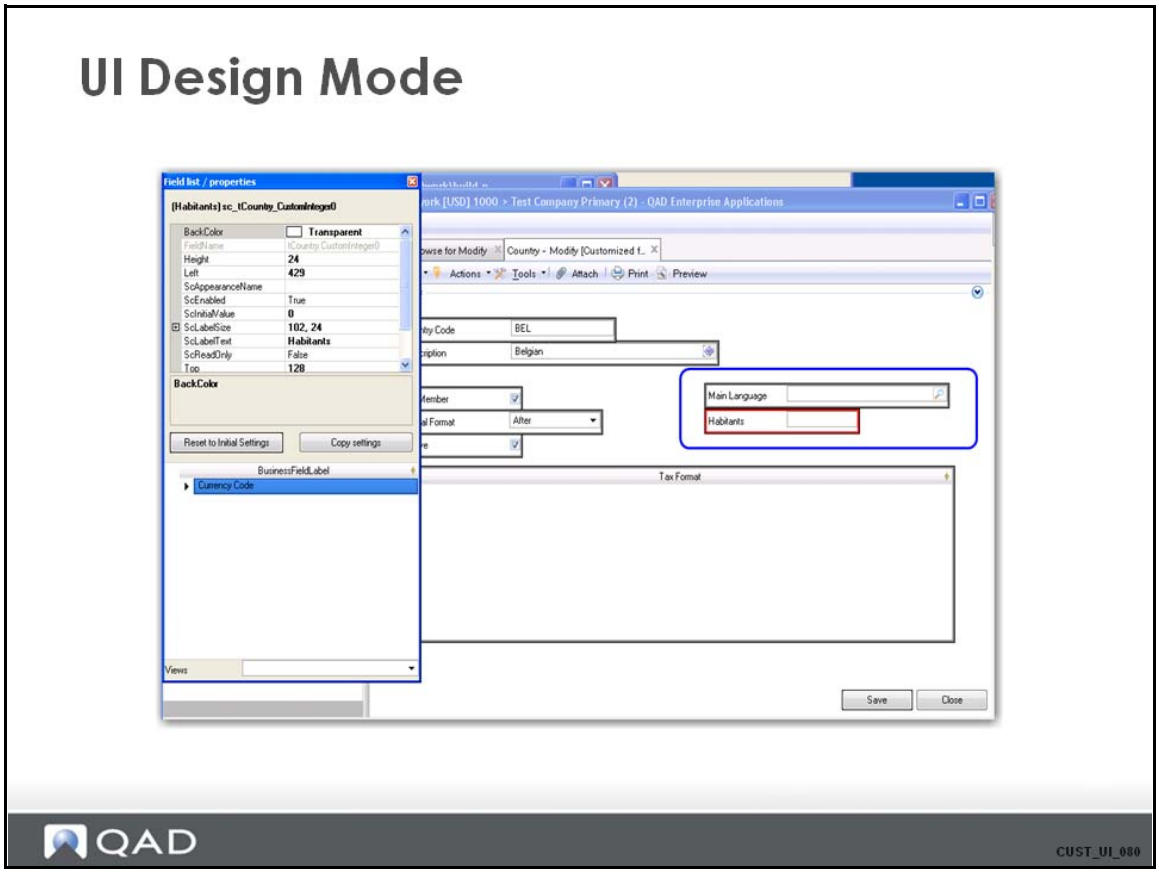
UI Design Mode 2



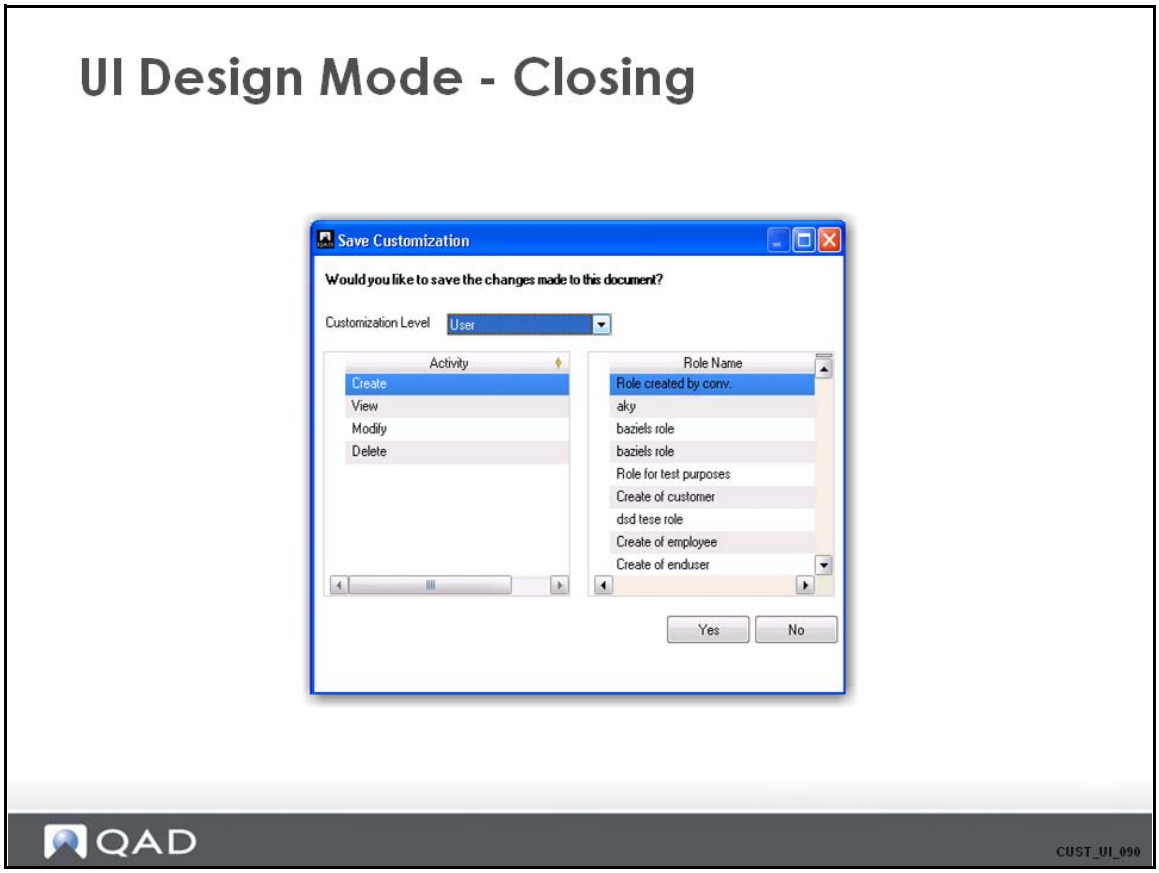
UI Design Mode 3



UI Design Mode 4



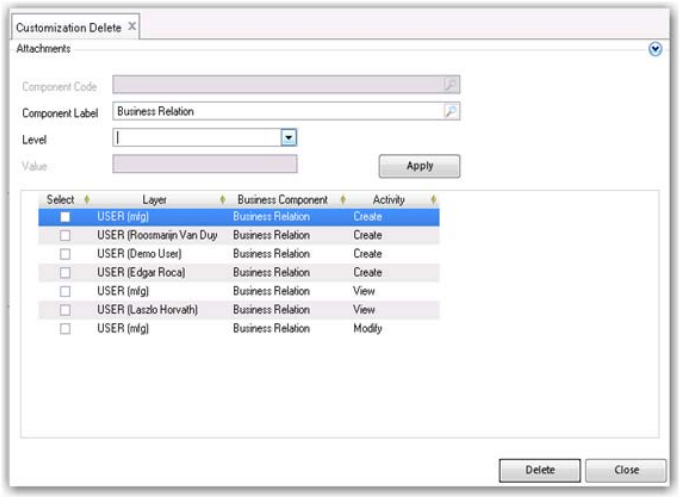
UI Design Mode 5



UI Design Mode 6

UI Design Mode - Maintenance

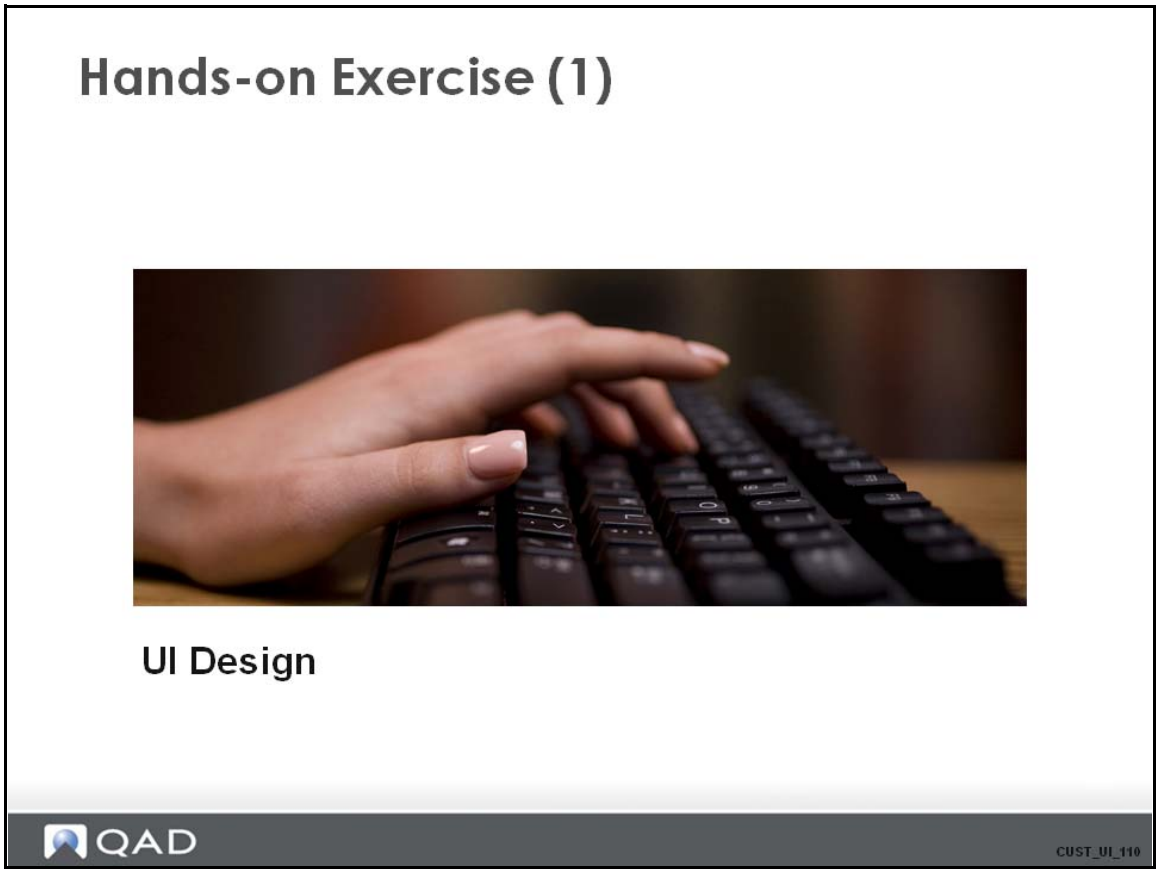
- Definition and update via UI Design Mode
- Deletion via Customization Delete



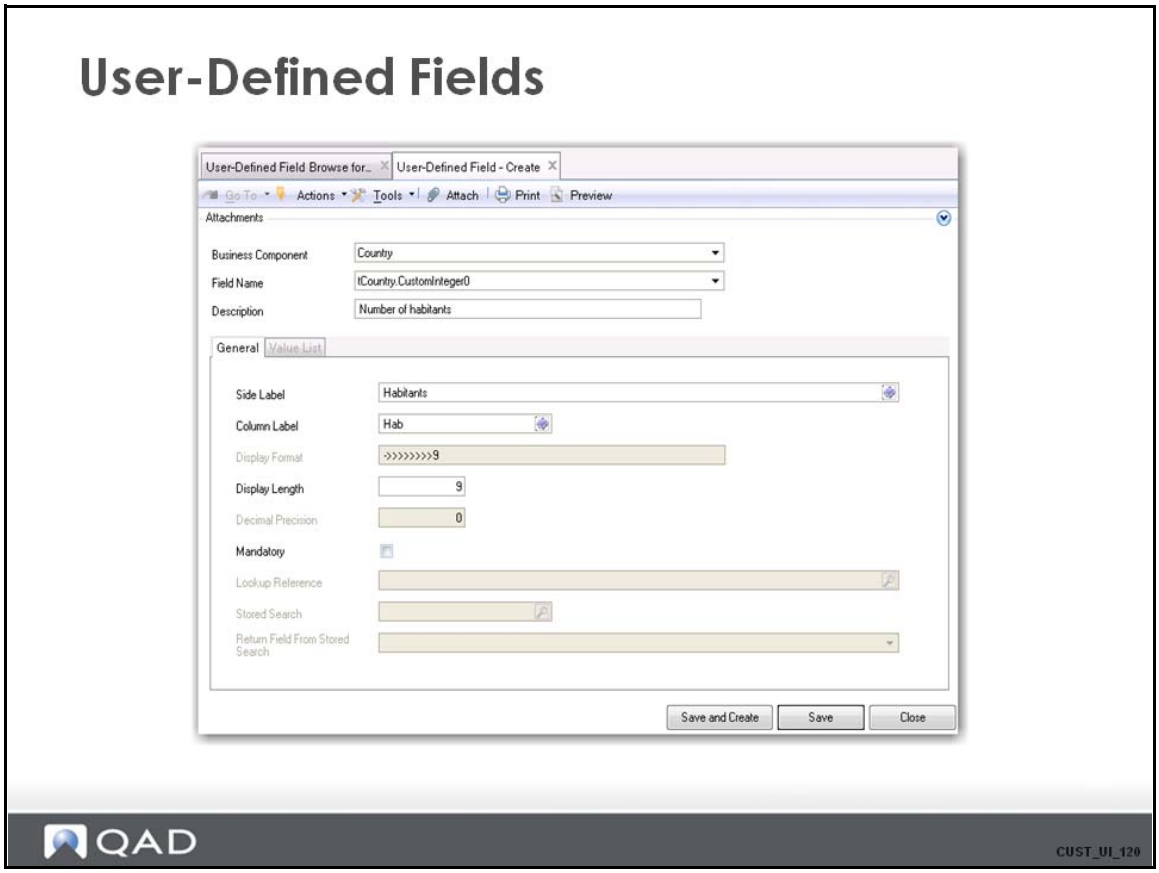
QAD

CUST_UI_100

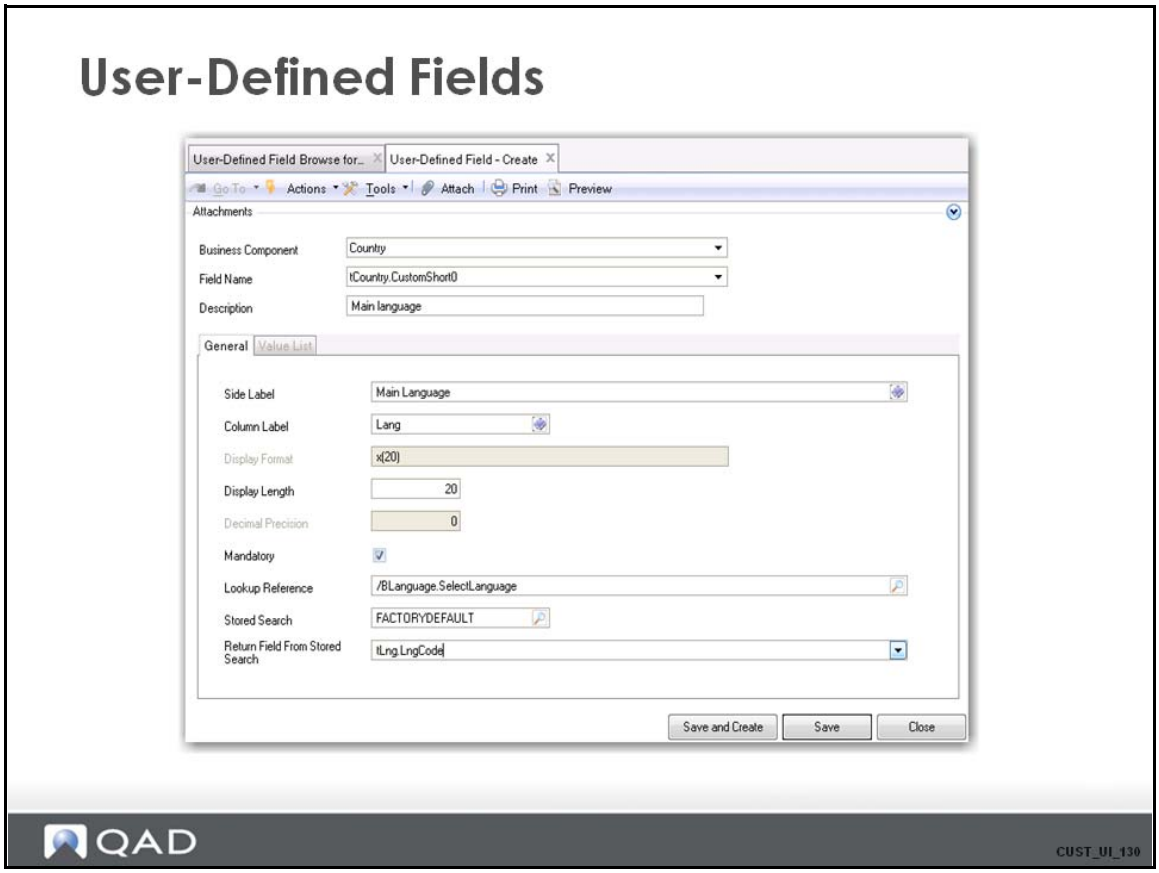
Hands-on Exercise (1)



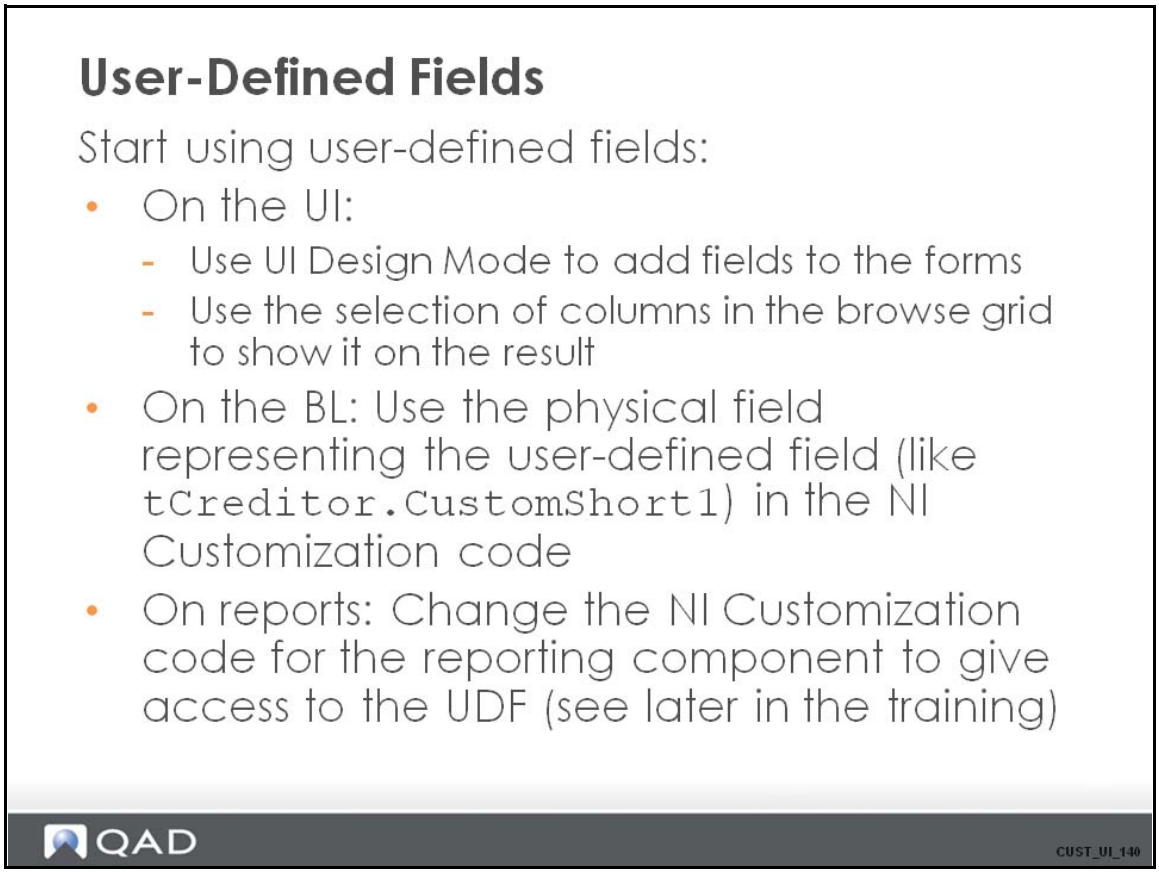
User-Defined Fields



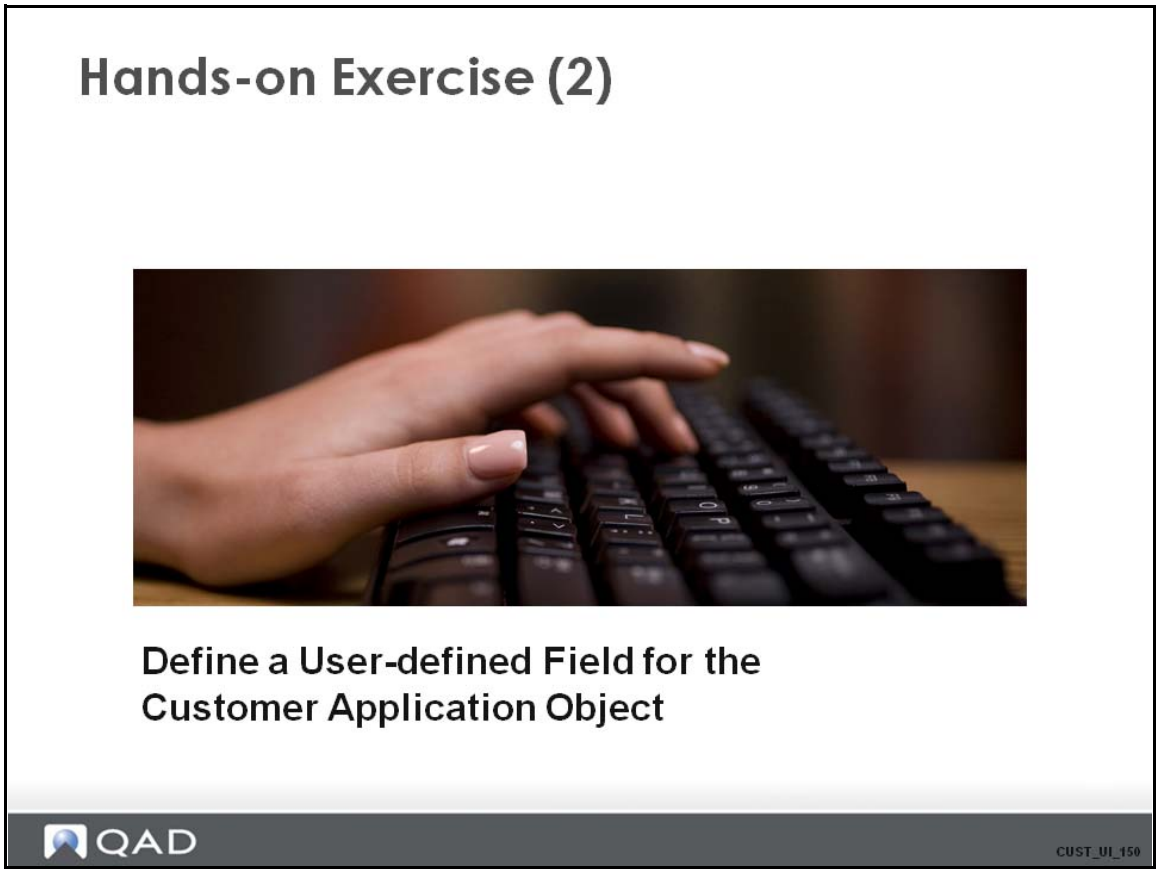
User-Defined Fields 2



User-Defined Fields 3

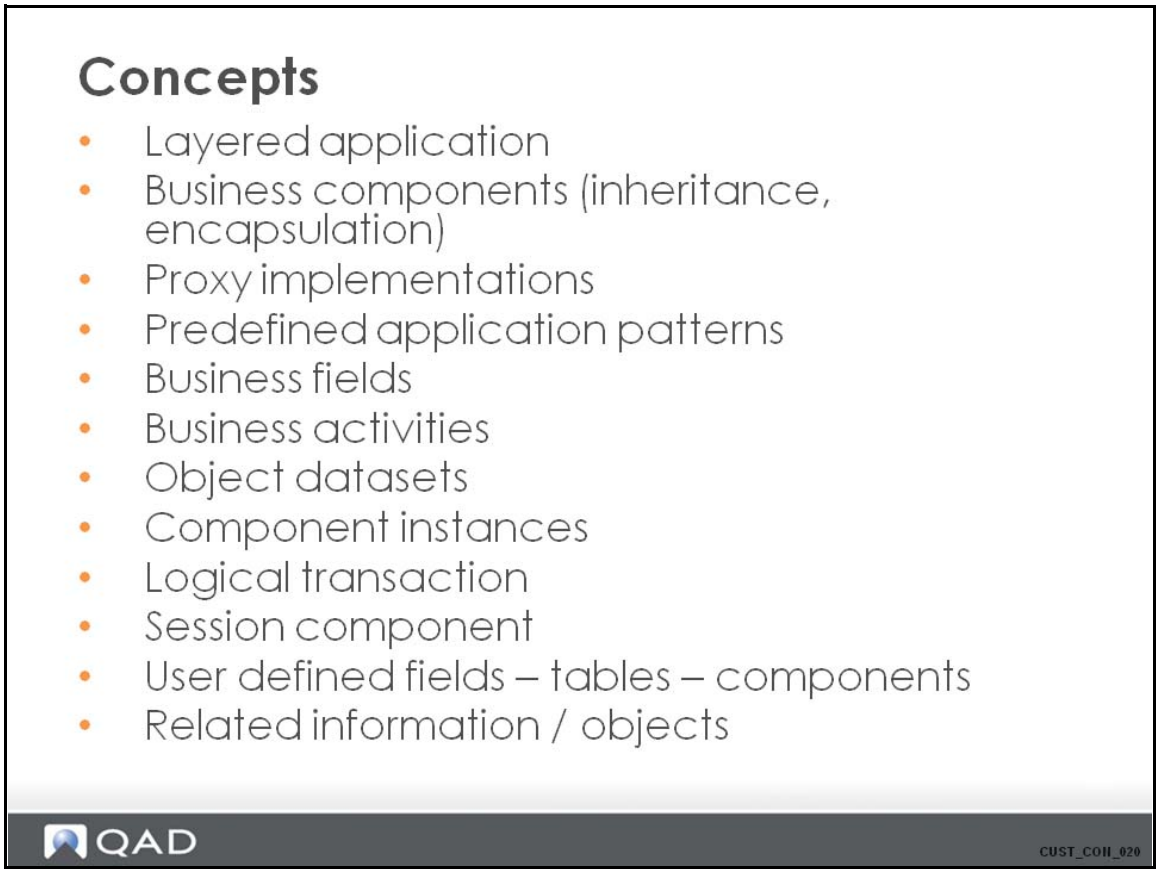


Hands-on Exercise (2)

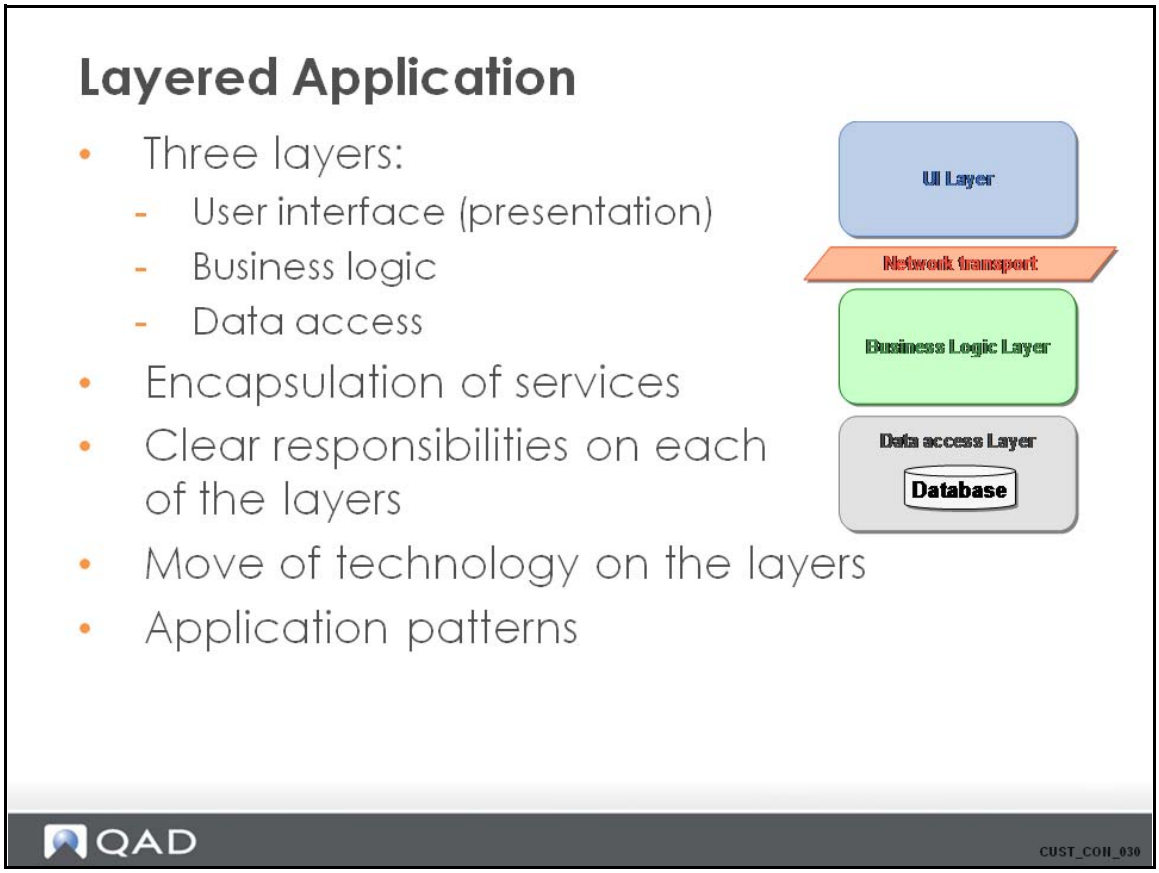


Non-intrusive Customization: Concepts

Concepts


Concepts
Layered application
Business components (inheritance, encapsulation)
Proxy implementations
Predefined application patterns
Business fields
Business activities
Object datasets
Component instances
Logical transaction
Session component
User defined fields – tables – components
Related information / objects
QAD
CUST_CON_020

Layered Application

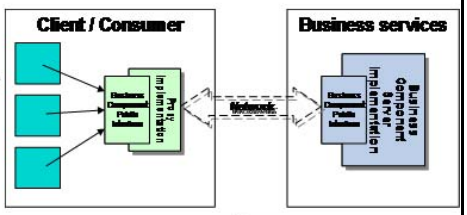


Business Components

Proxy Implementations

Proxy Implementations

- Surrogates for business components that provide access to it on the client (consumer) side.
  Client deployment of stubs representing business functions



- Proxies hide the complexity of connection infrastructure, session handling, and state handling for the client
- Proxies are generated and can be used in different types of technologies

QAD

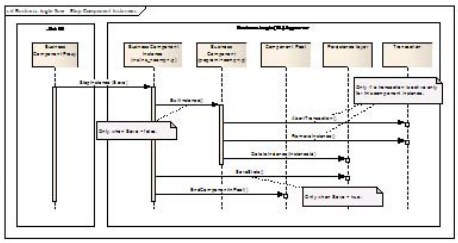
CUST_CON_050

Predefined Application Patterns

Predefined Application Patterns

Provide predefined program flows on different levels:

- Examples high-level patterns:
  - Creation of an object
  - Modification of an object
  - Deletion of an object
  - Maintenance of object set
  - Selection of an object

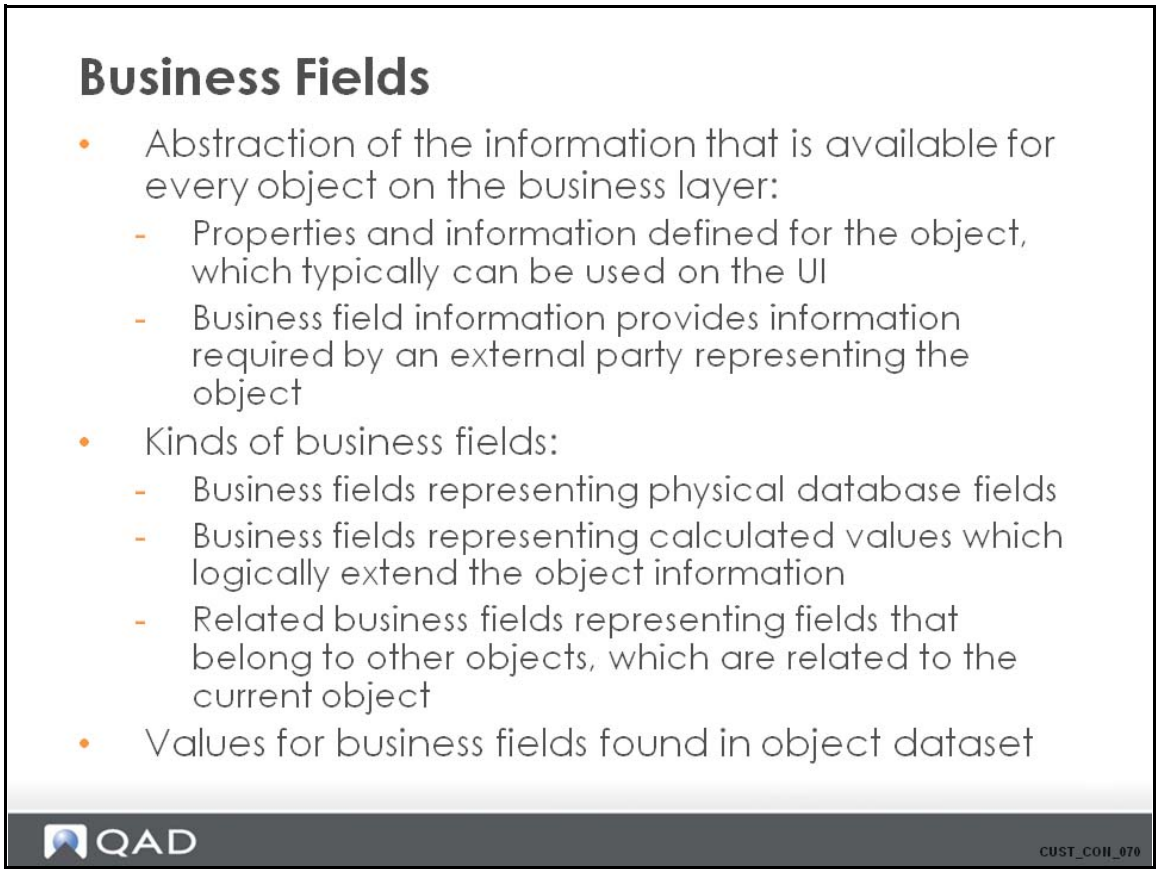


- Examples of lower level patterns:
  - Instantiation of components
  - Error processing

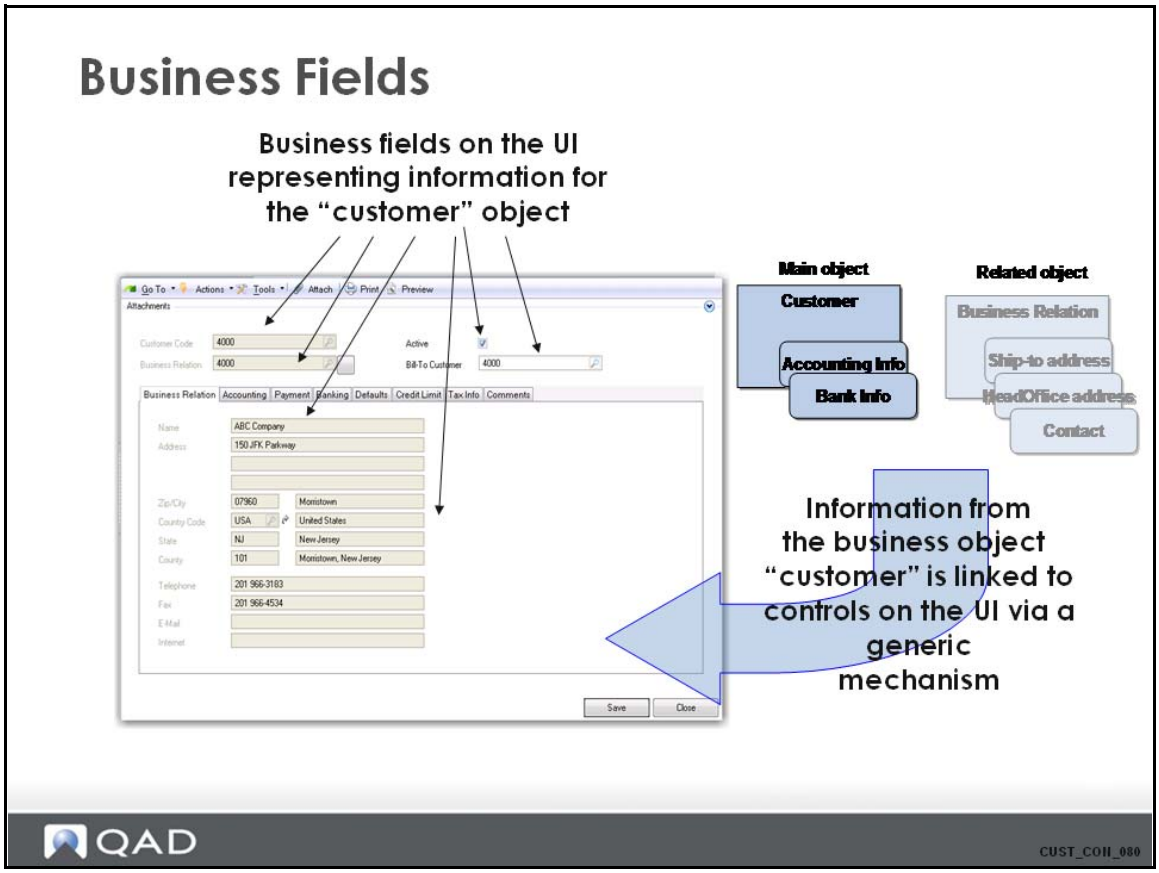
QAD

CUST_CON_060

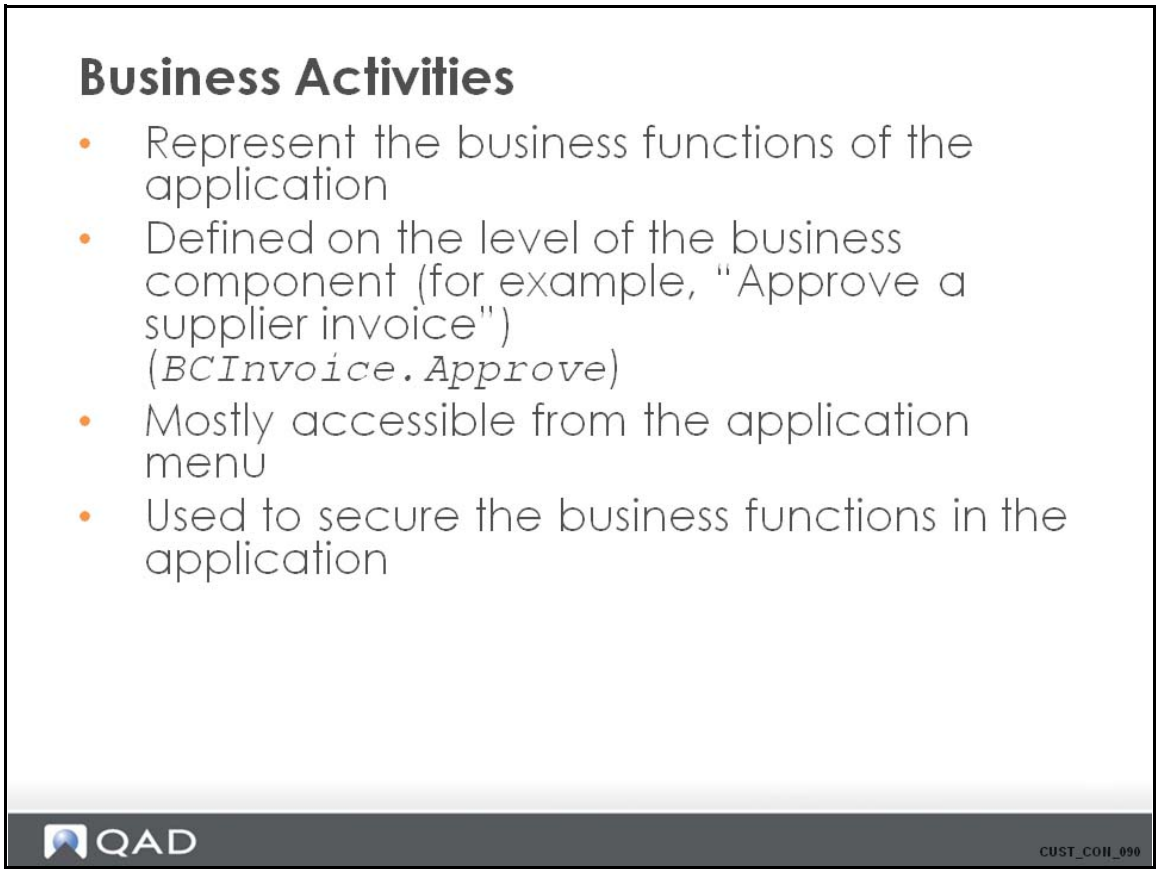
Business Fields



Business Fields 2



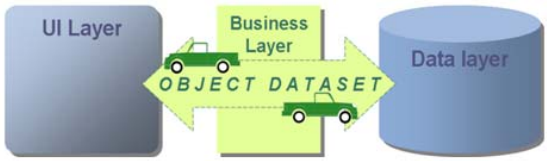
Business Activities



Object Datasets

Object Datasets

- The data associated with an object is passed between the different layers using object datasets

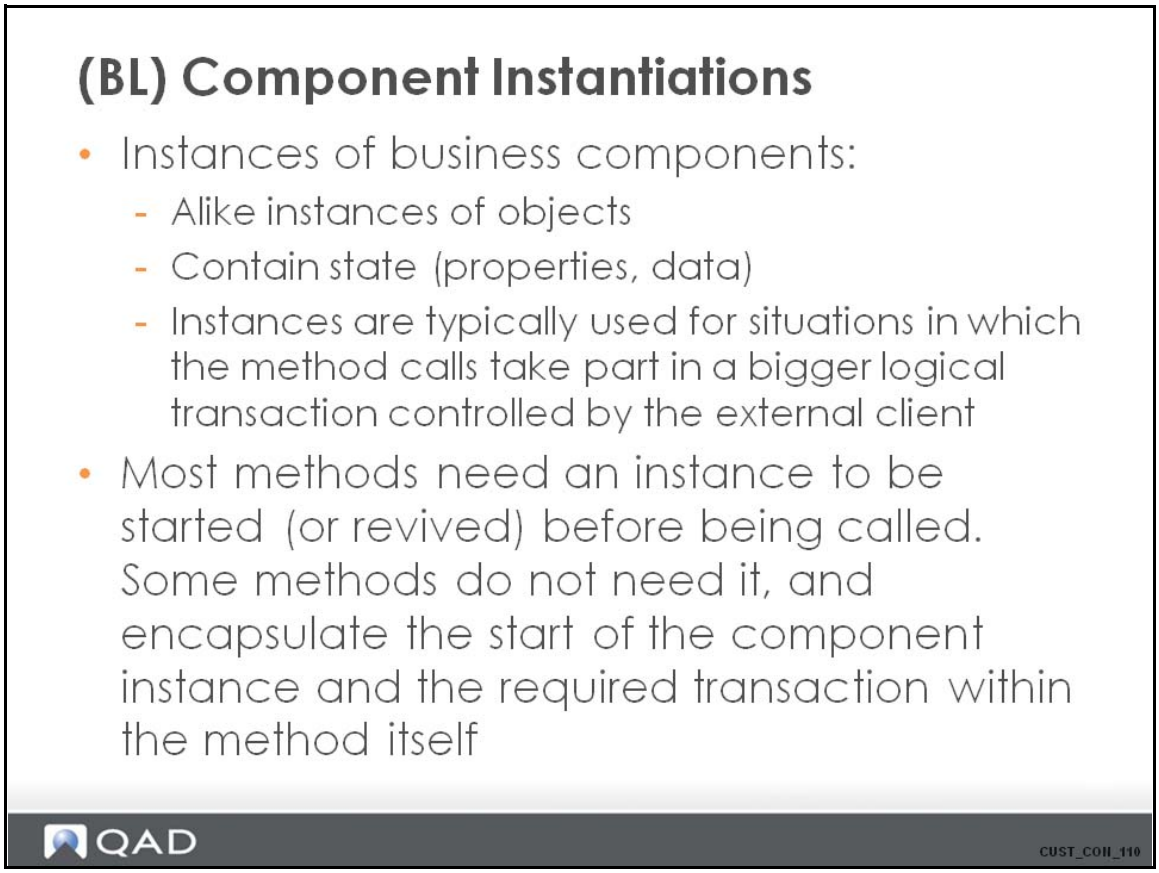


- Object datasets represented by:
  - OpenEdge ProDatasets on the BL side
  - .NET datasets on the UI side
- Object datasets are not only used by the UI client, but all types of consumers of the business services (integration, load functionality, event publishing...)

QAD

CUST_CON_100

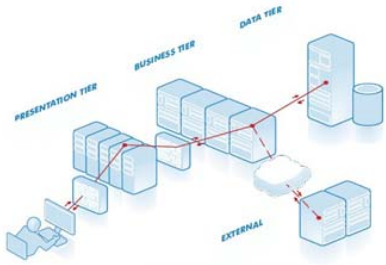
(BL) Component Instantiations



Logical Transactions

Logical Transactions

- Physical transactions do not work well with stateless applications. We must try to eliminate the risk of record locking by optimistic locking

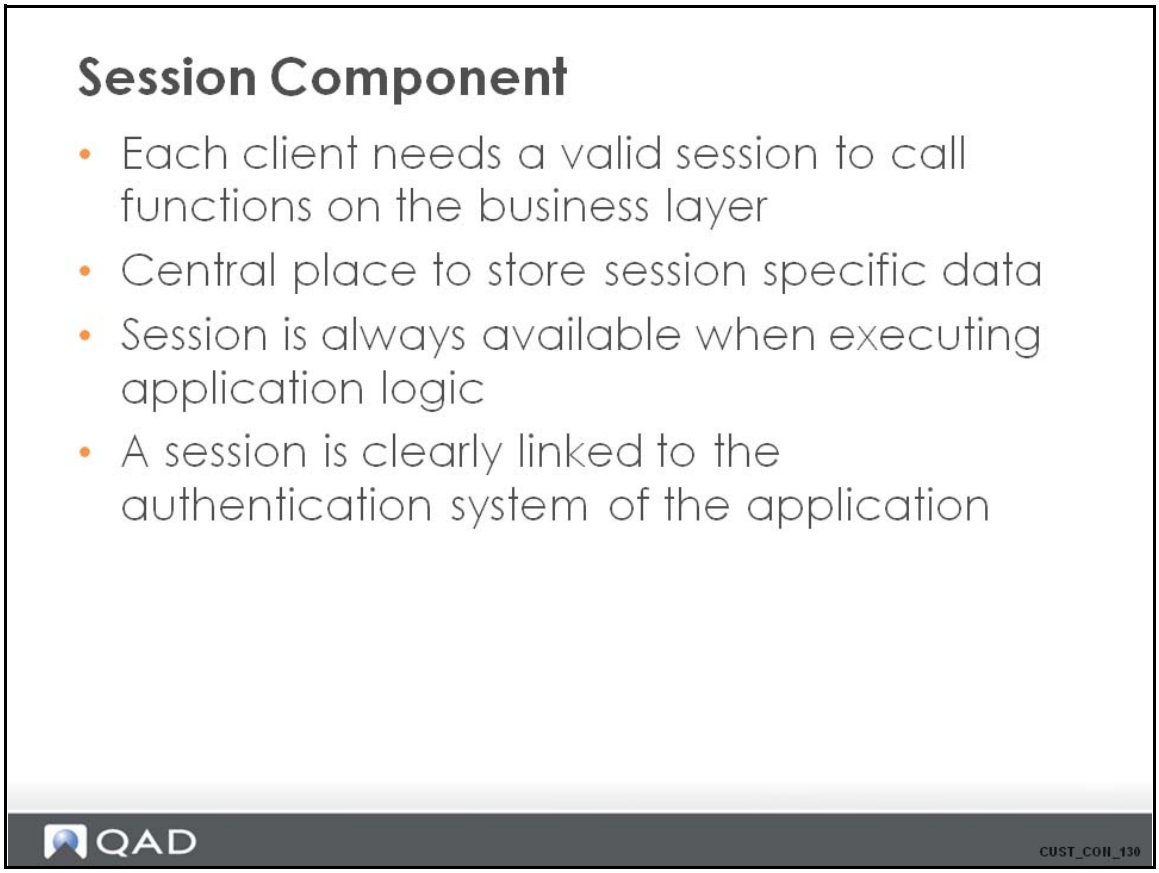


- The transaction control can be encapsulated in the business logic level, but can also be on the level of the UI or an external consumer
- Component instances must be aware of logical transactions

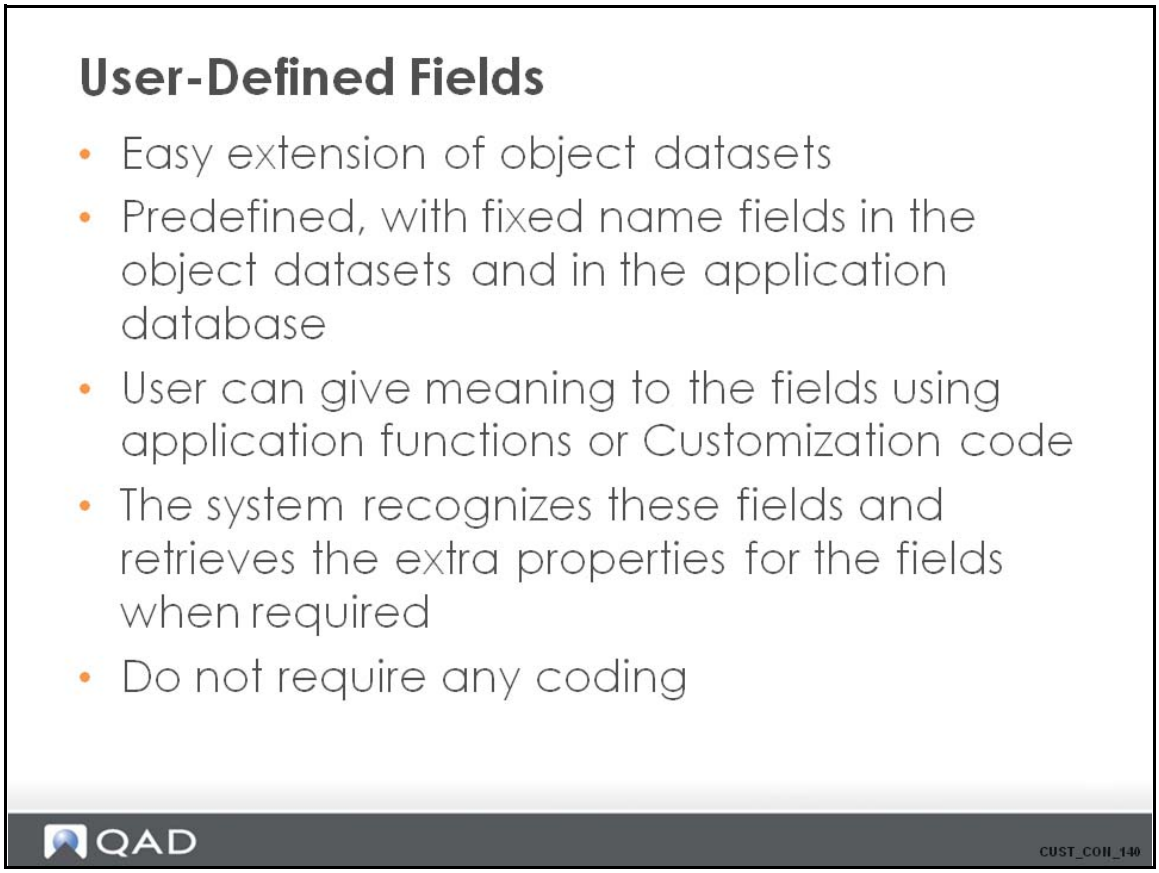
QAD

CUST_CON_120

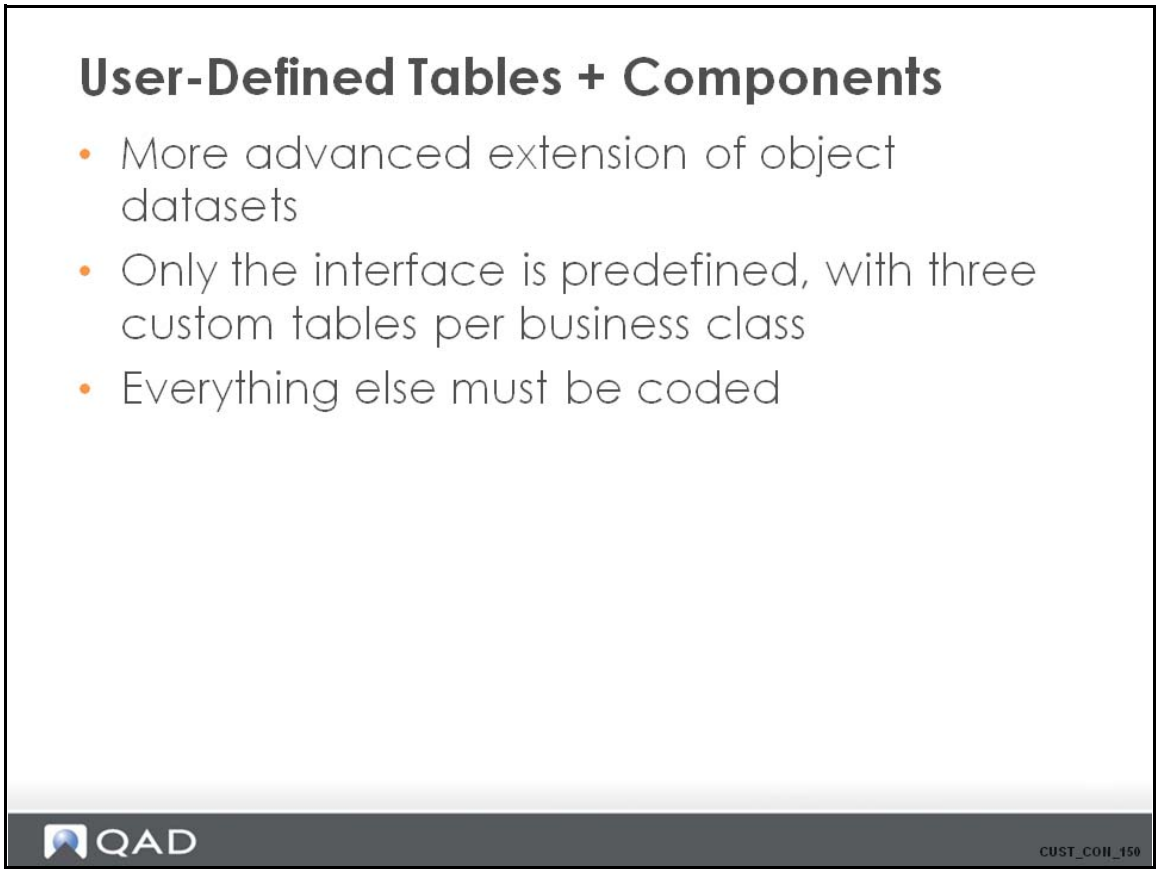
Session Component



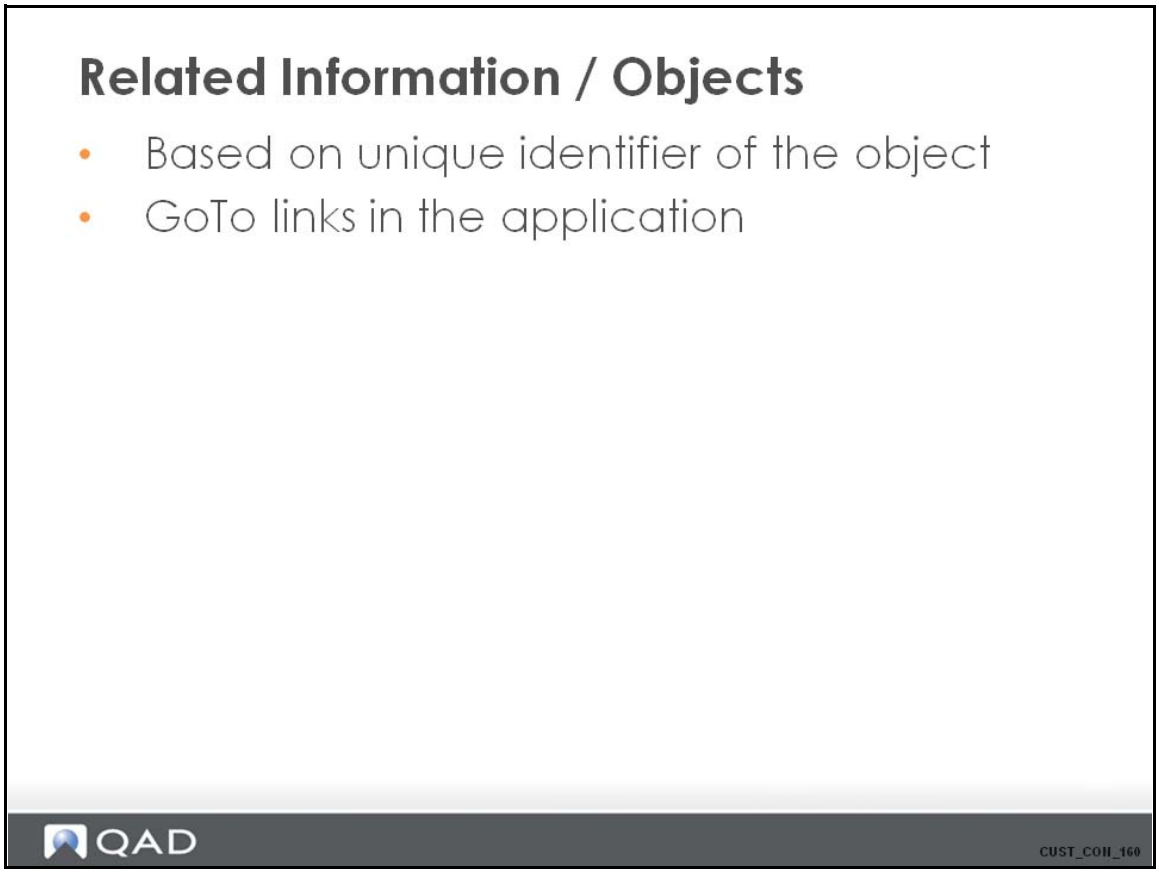
User-Defined Fields



User-Defined Tables + Components

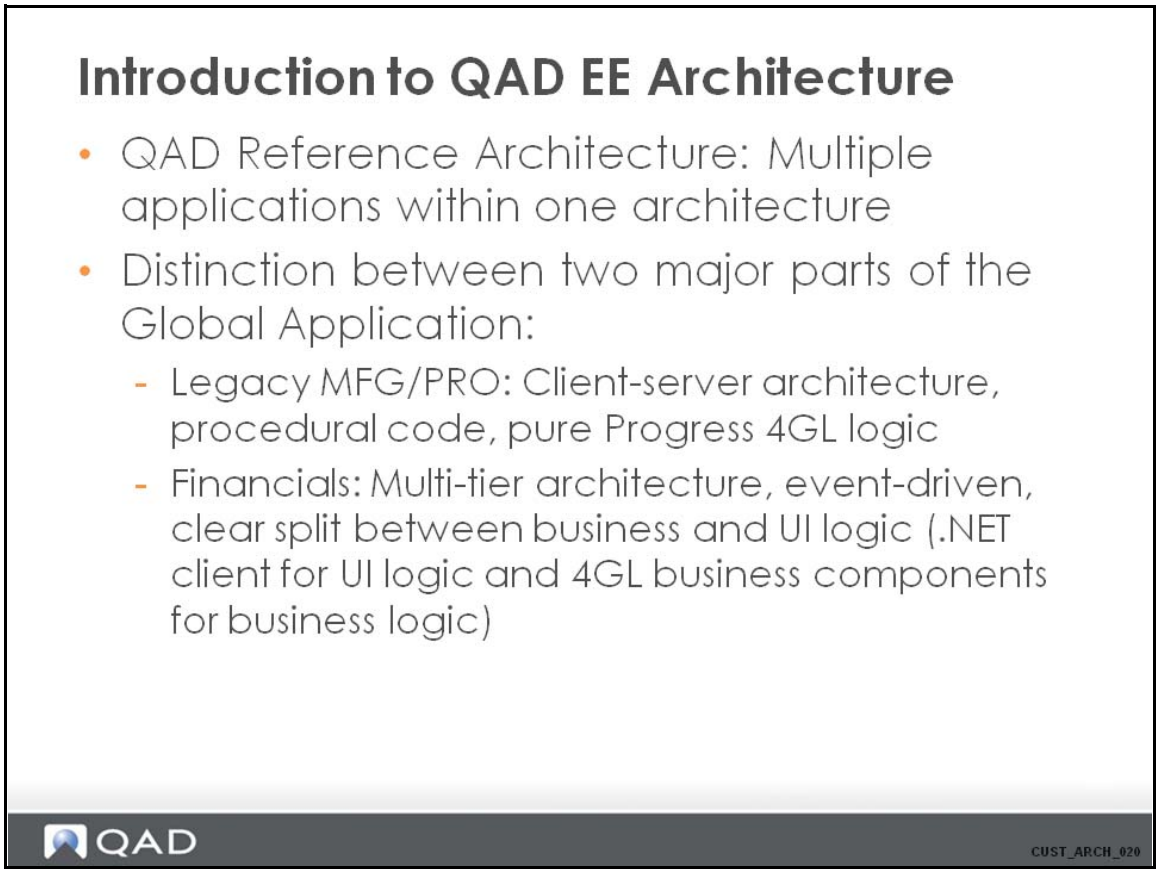


Related Information / Objects

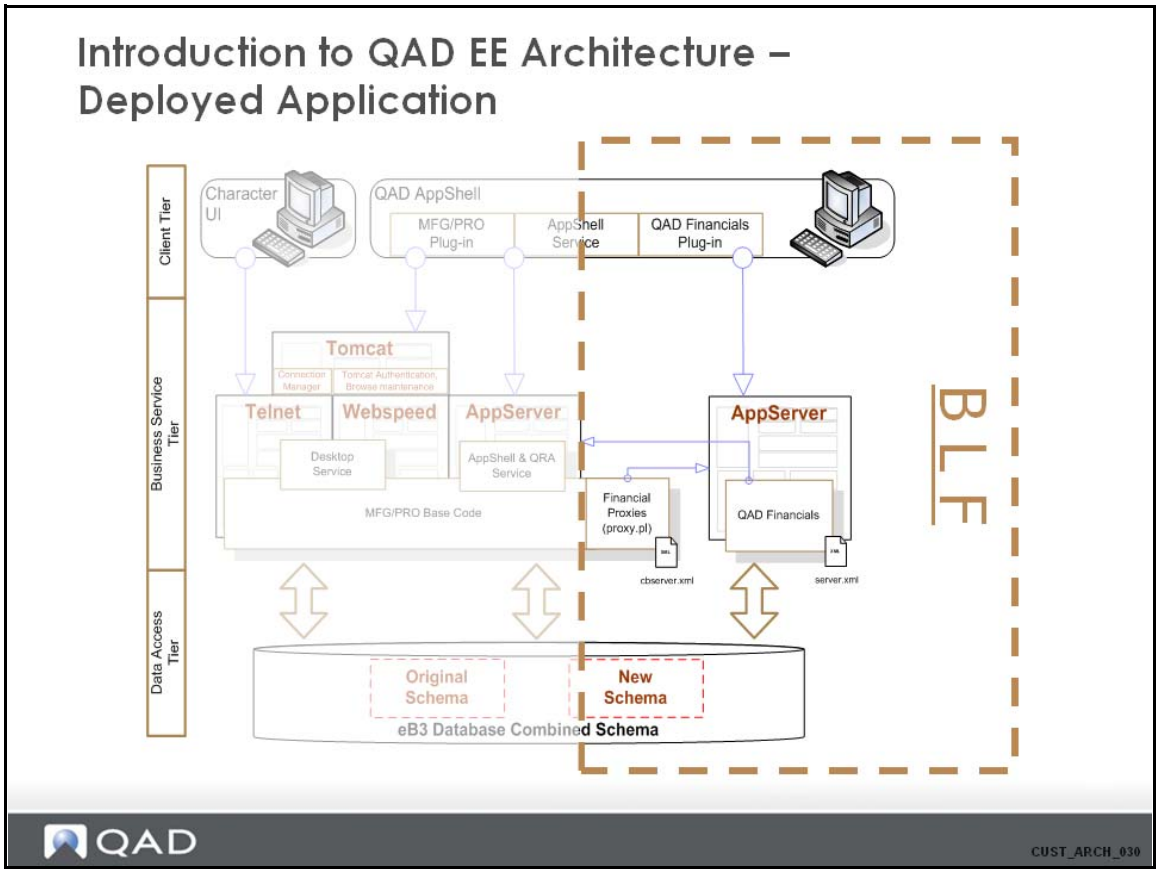


Non-instrusive Customization: Architecture

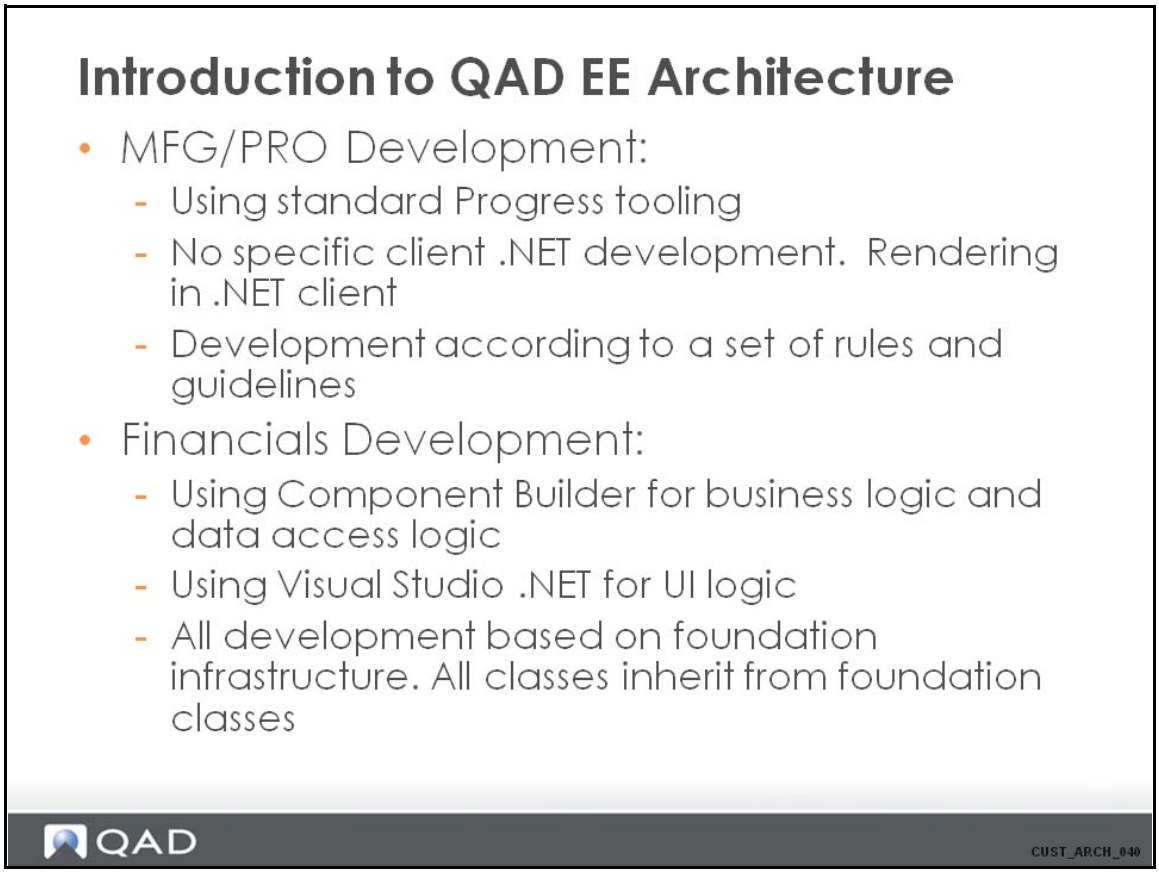
Introduction to QAD EE Architecture



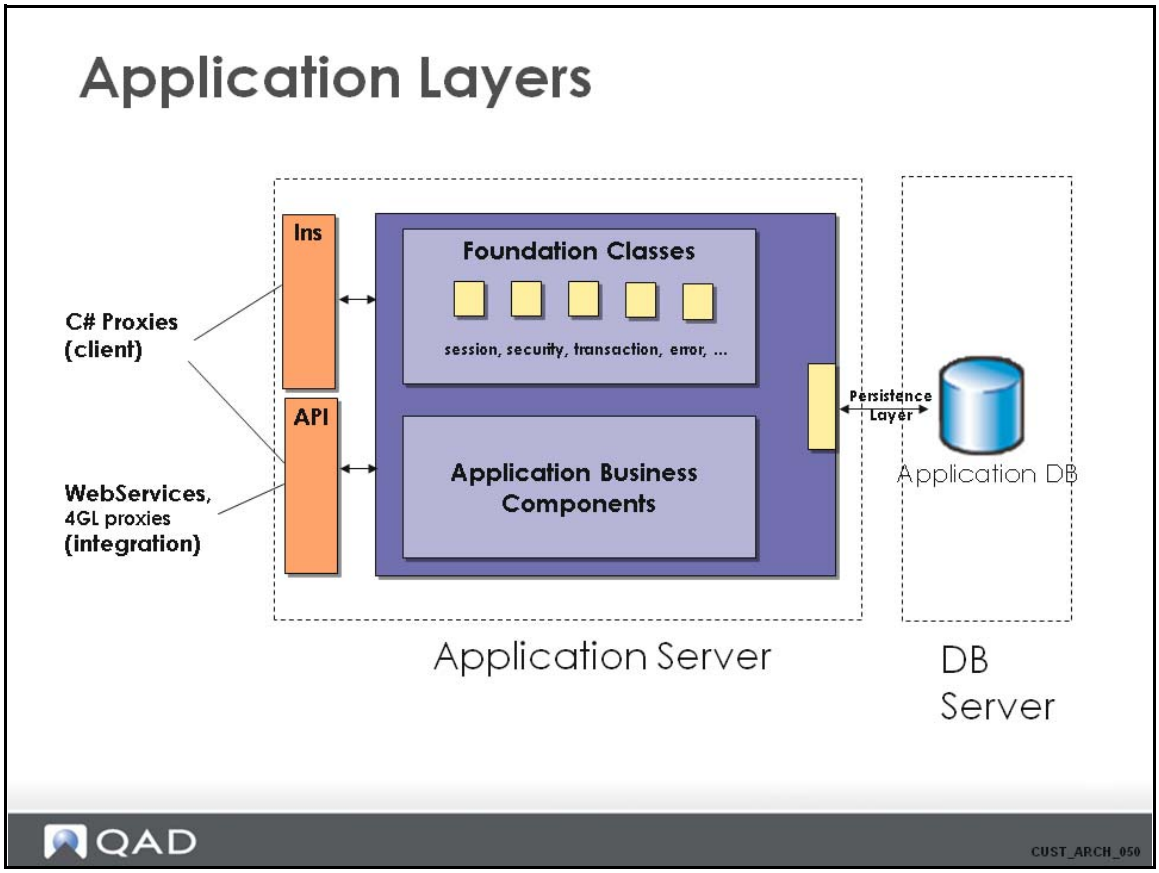
Introduction to QAD EE Architecture 2



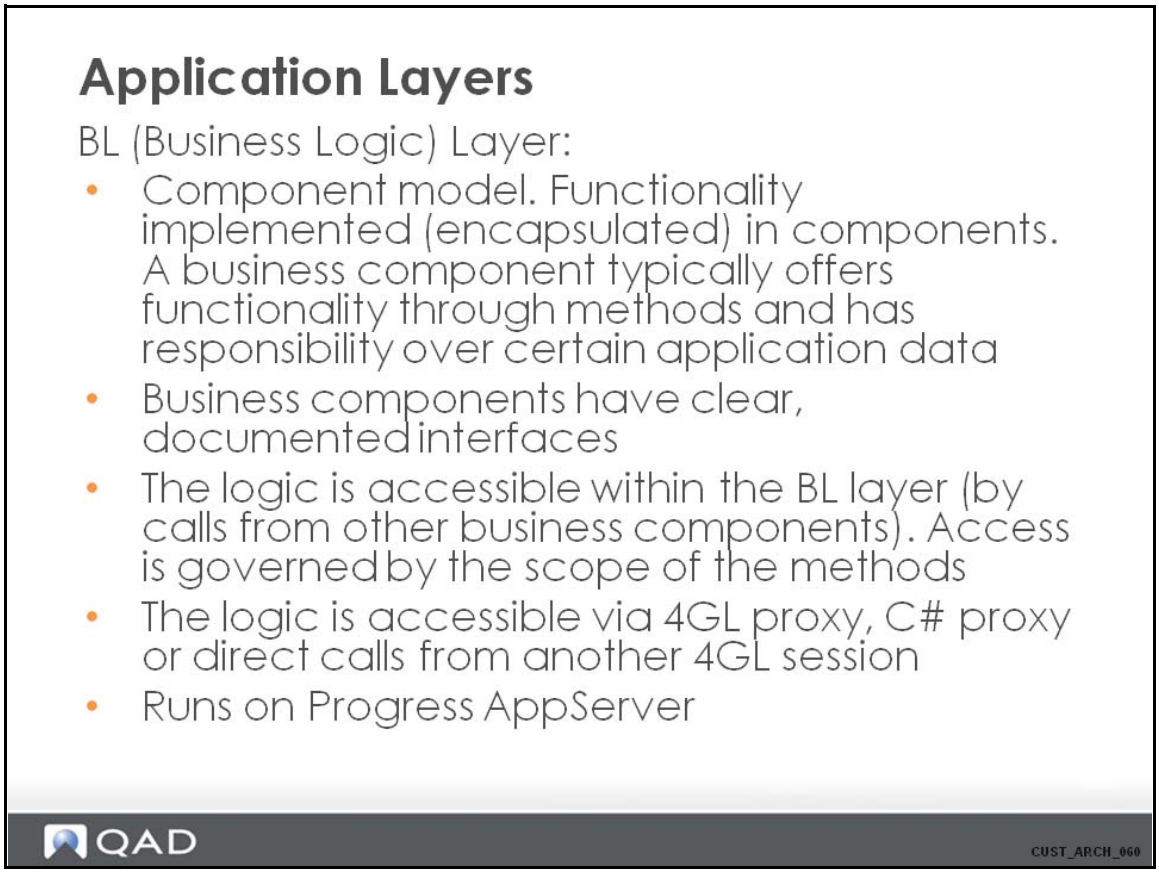
Introduction to QAD EE Architecture 3



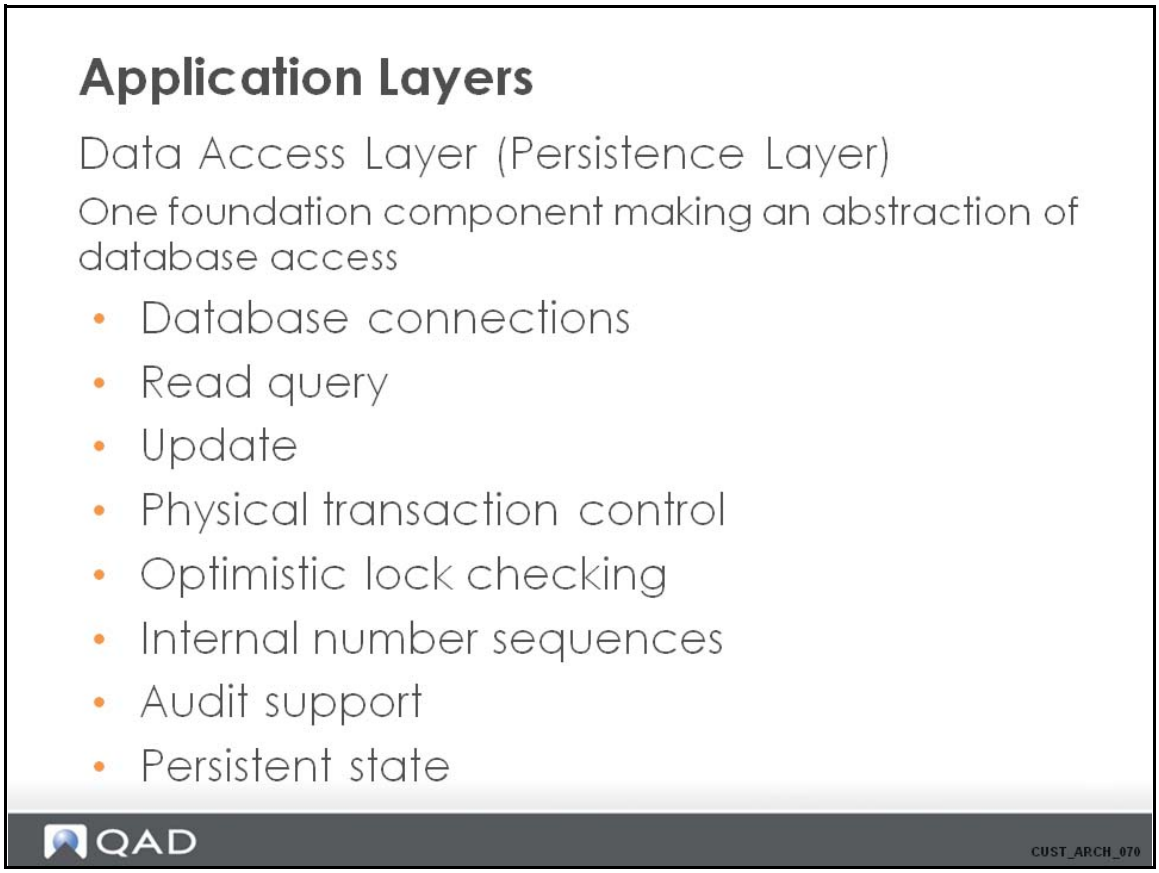
Application Layers



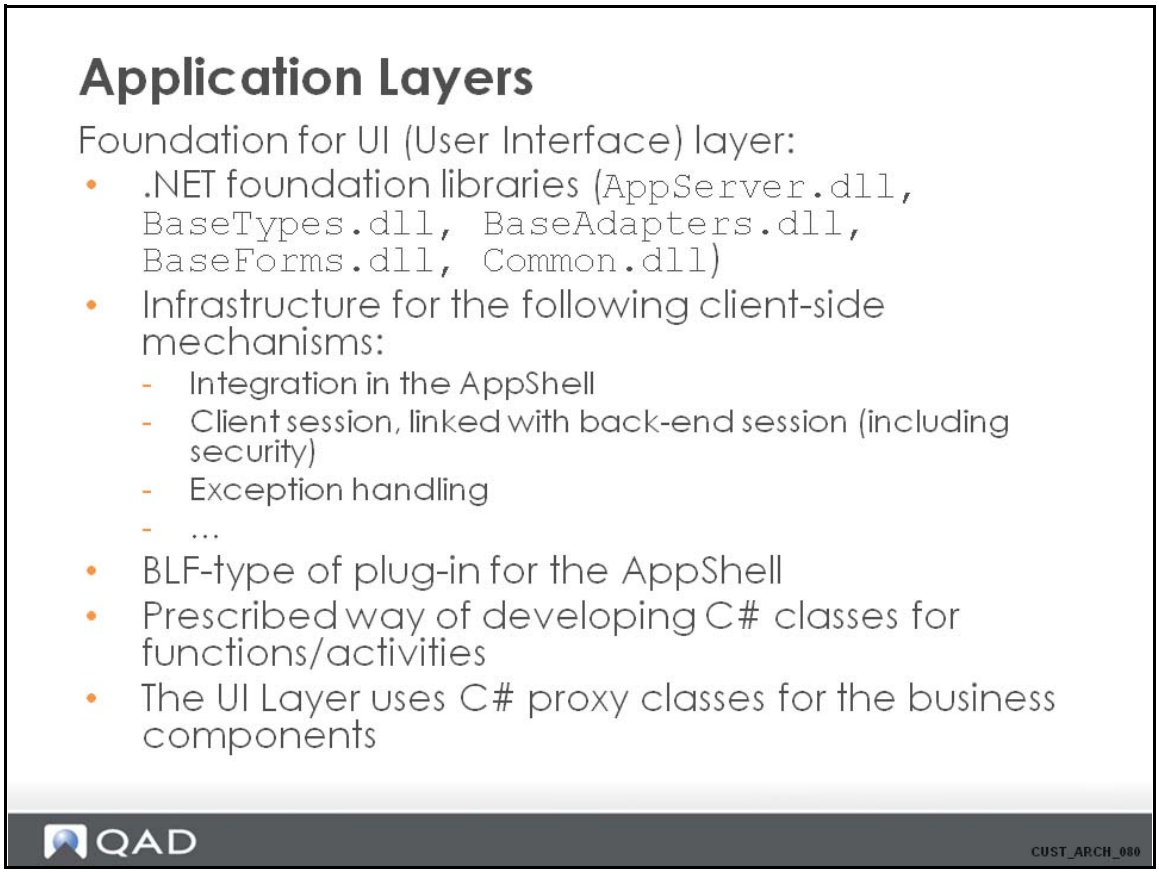
Application Layers 2



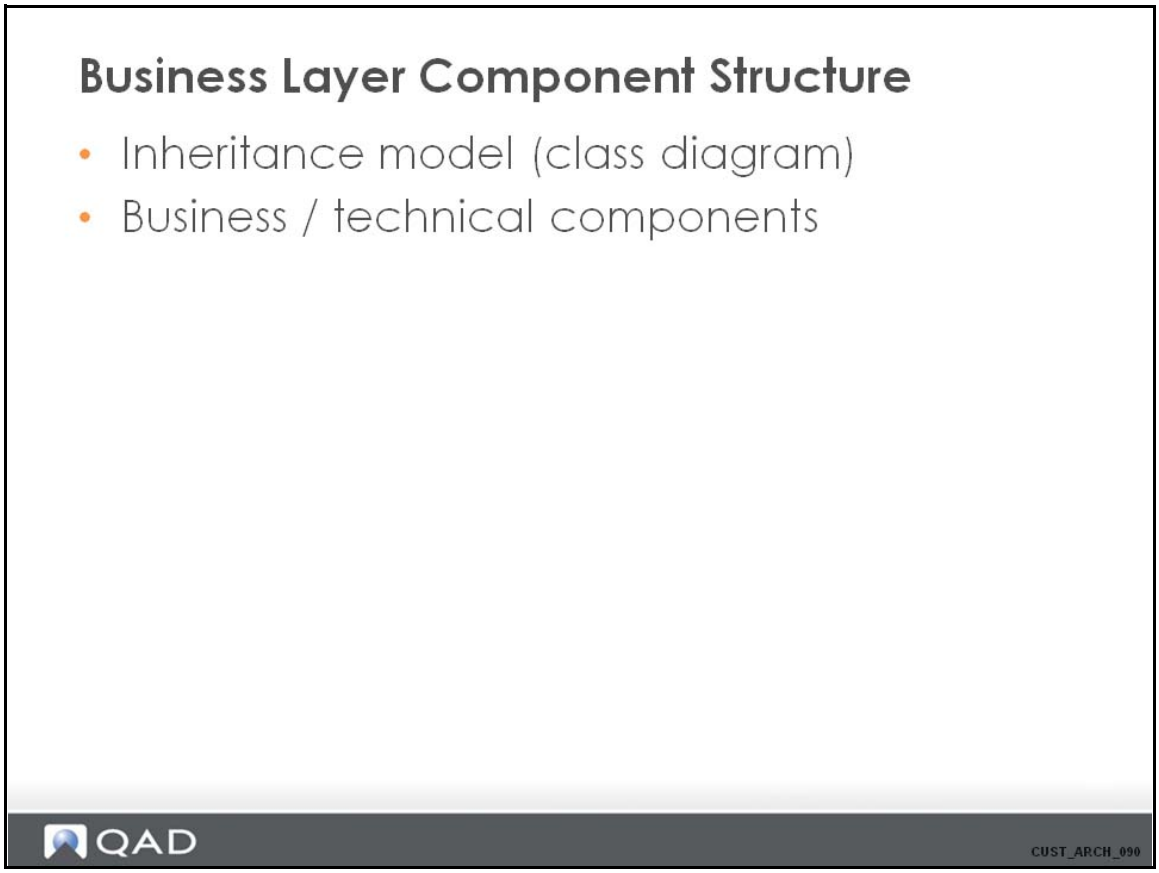
Application Layers 3



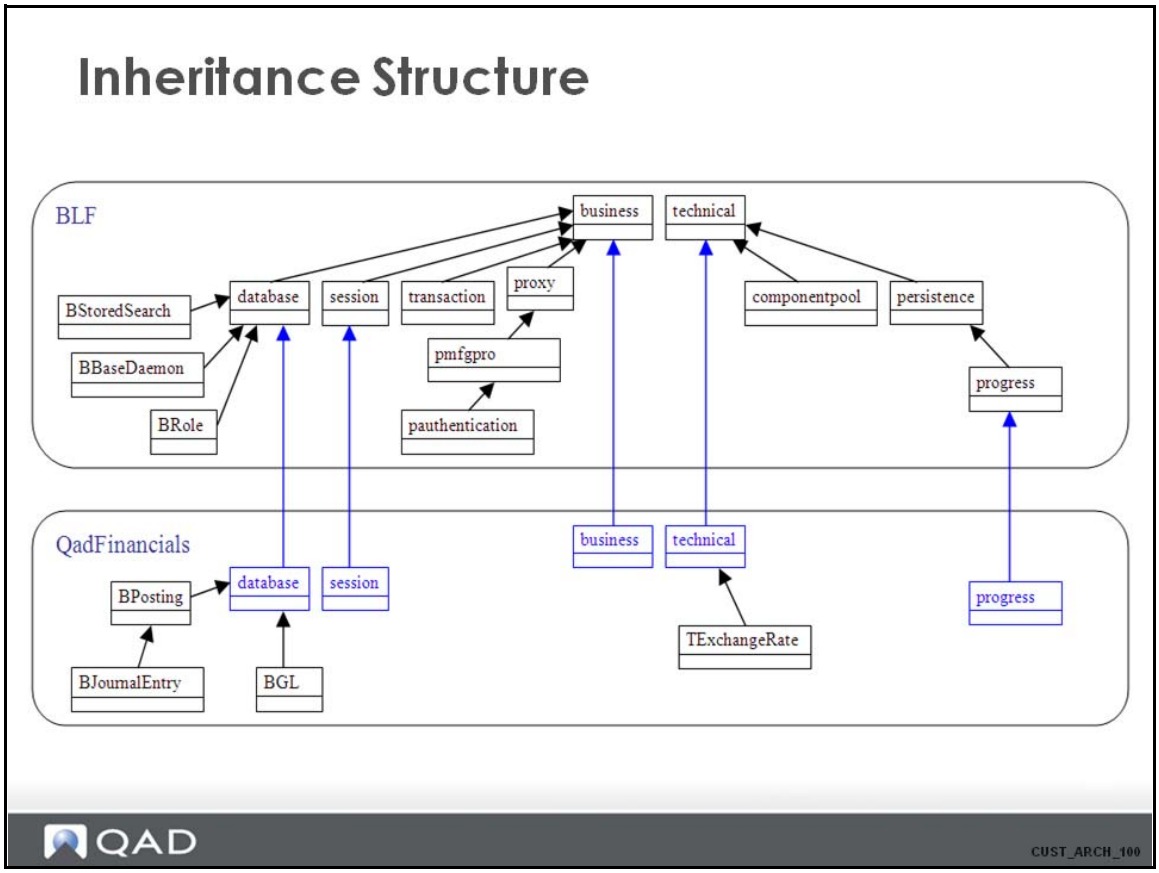
Application Layers 4



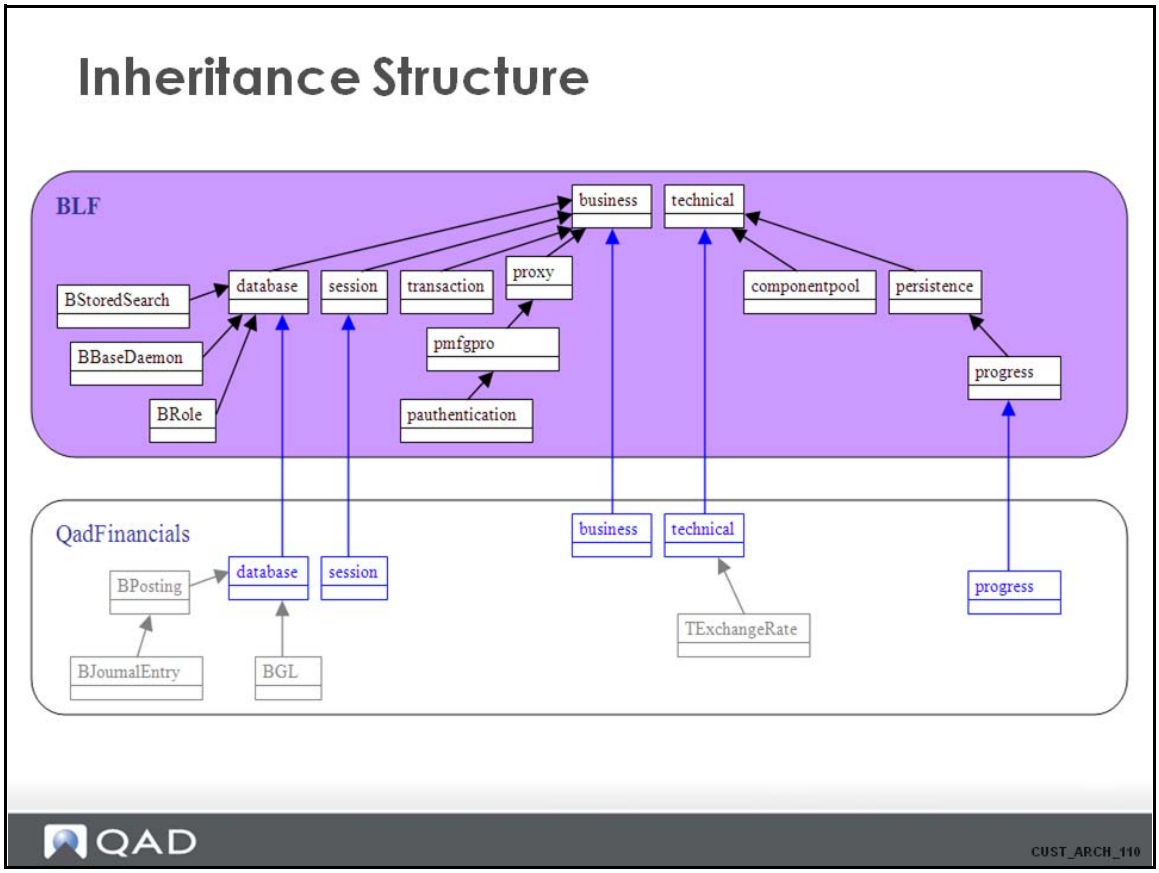
Business Layer Component Structure



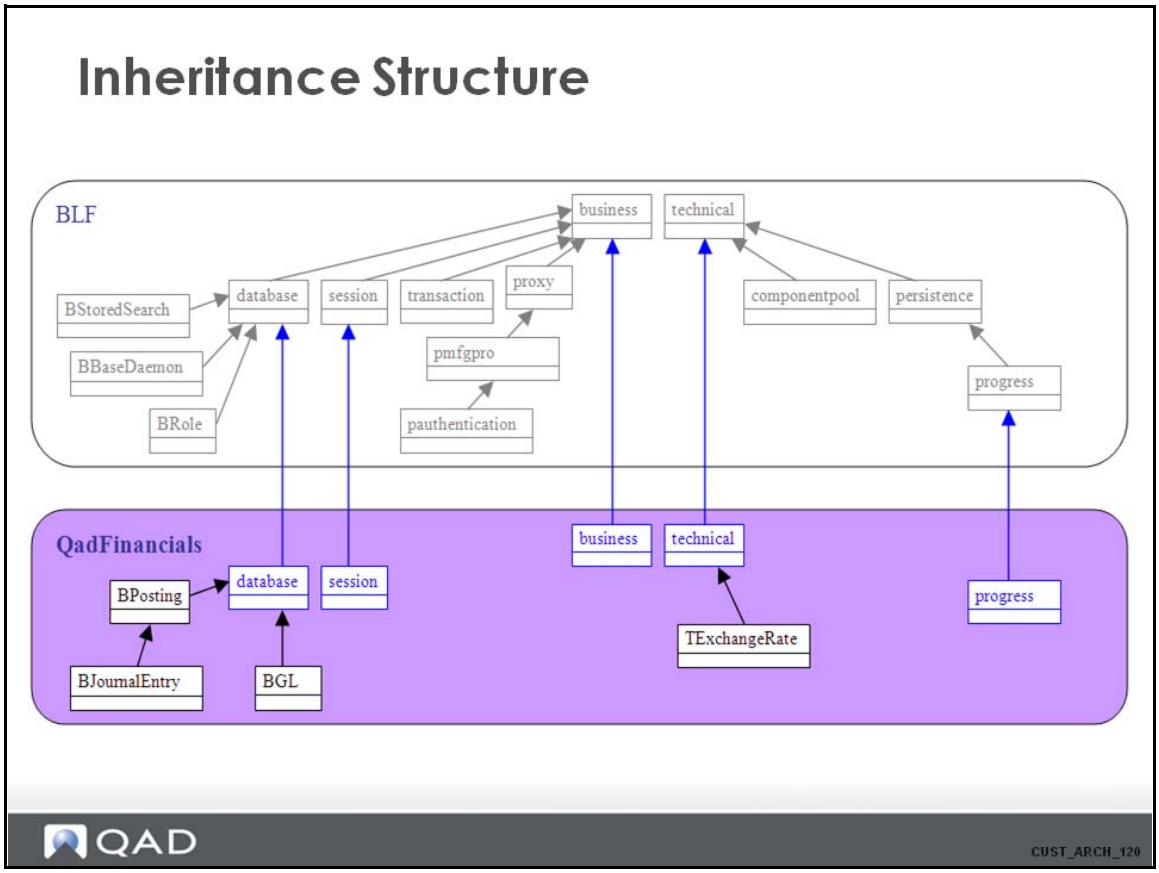
Inheritance Structure



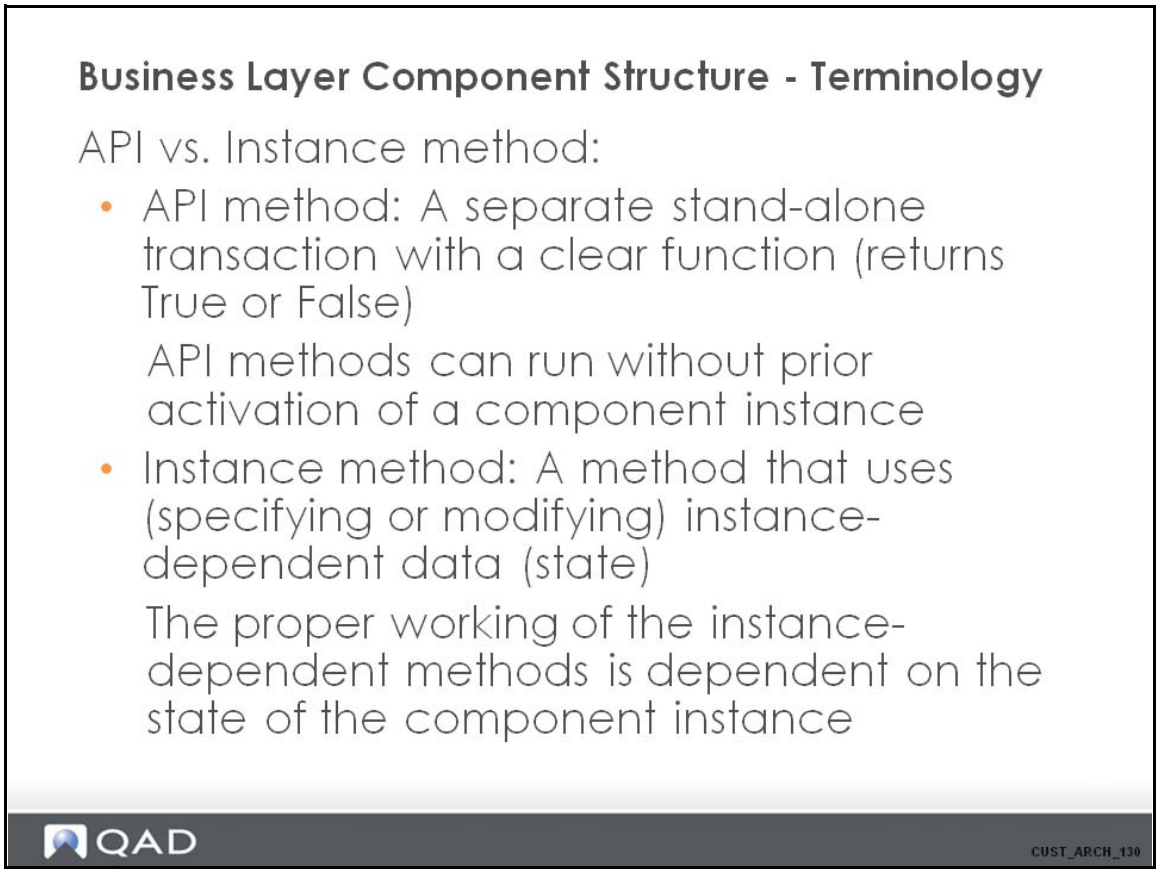
Inheritance Structure 2



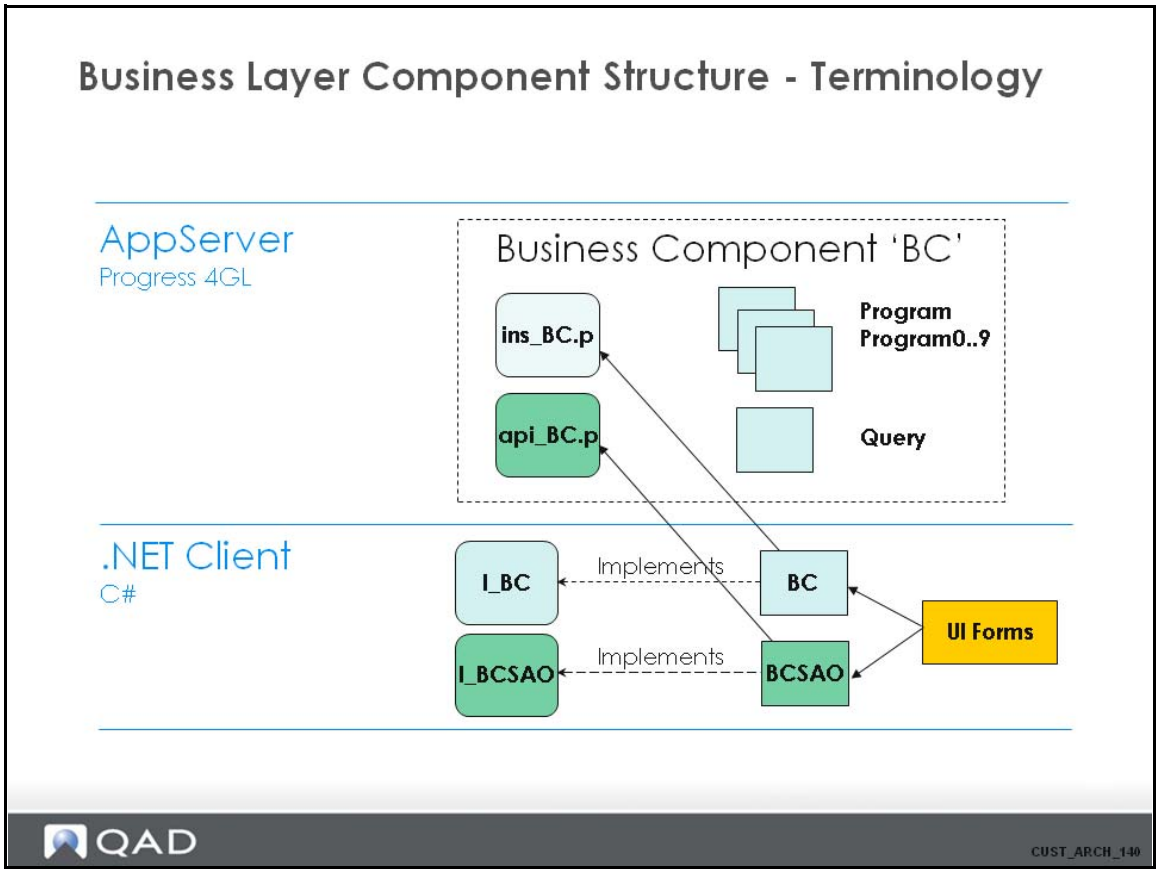
Inheritance Structure 3



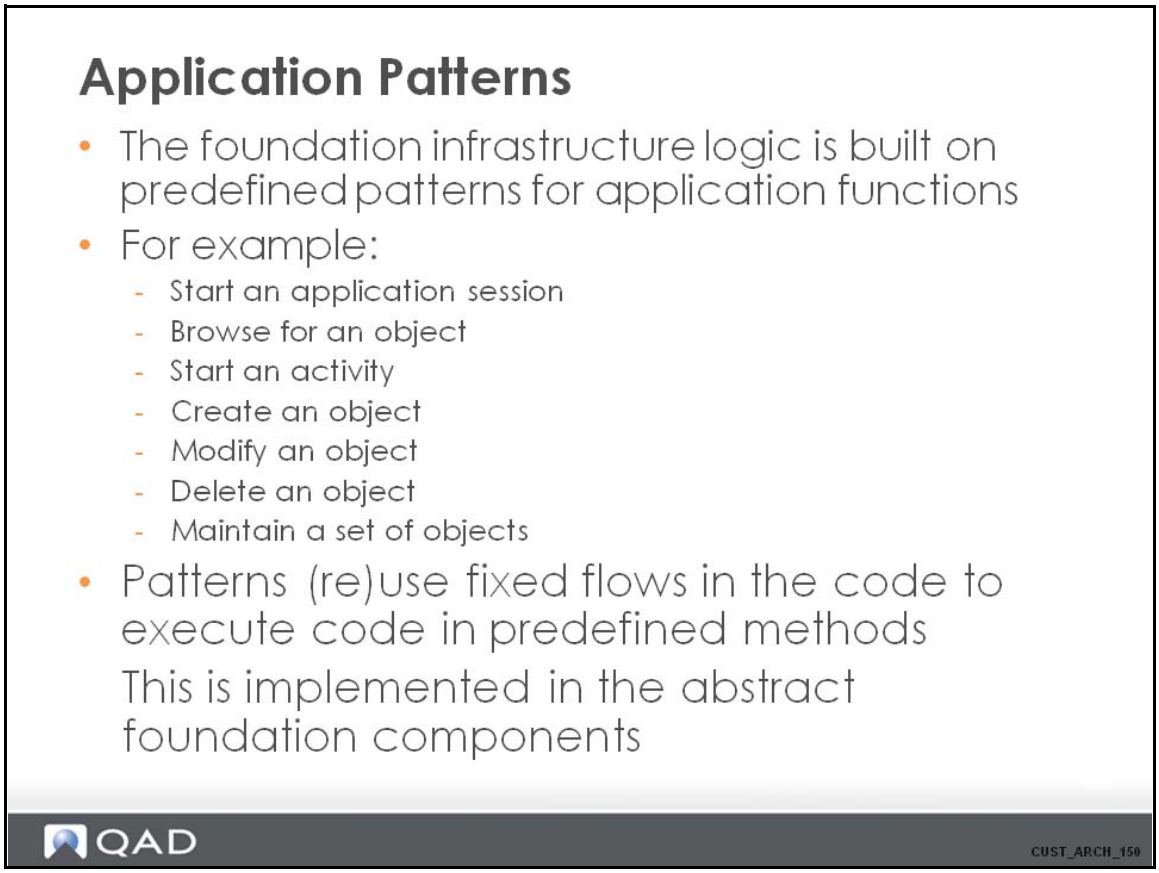
Business Layer Component Structure - Terminology



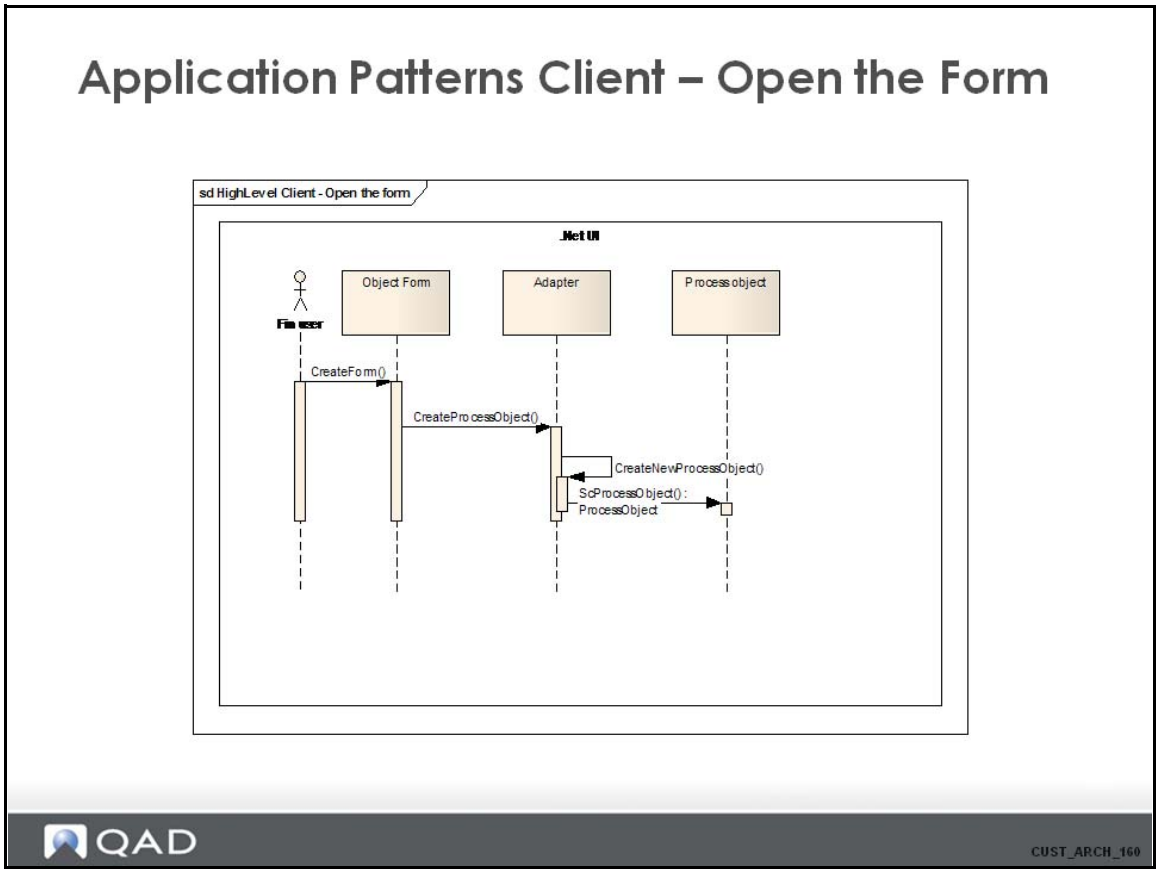
Business Layer Component Structure - Terminology 2



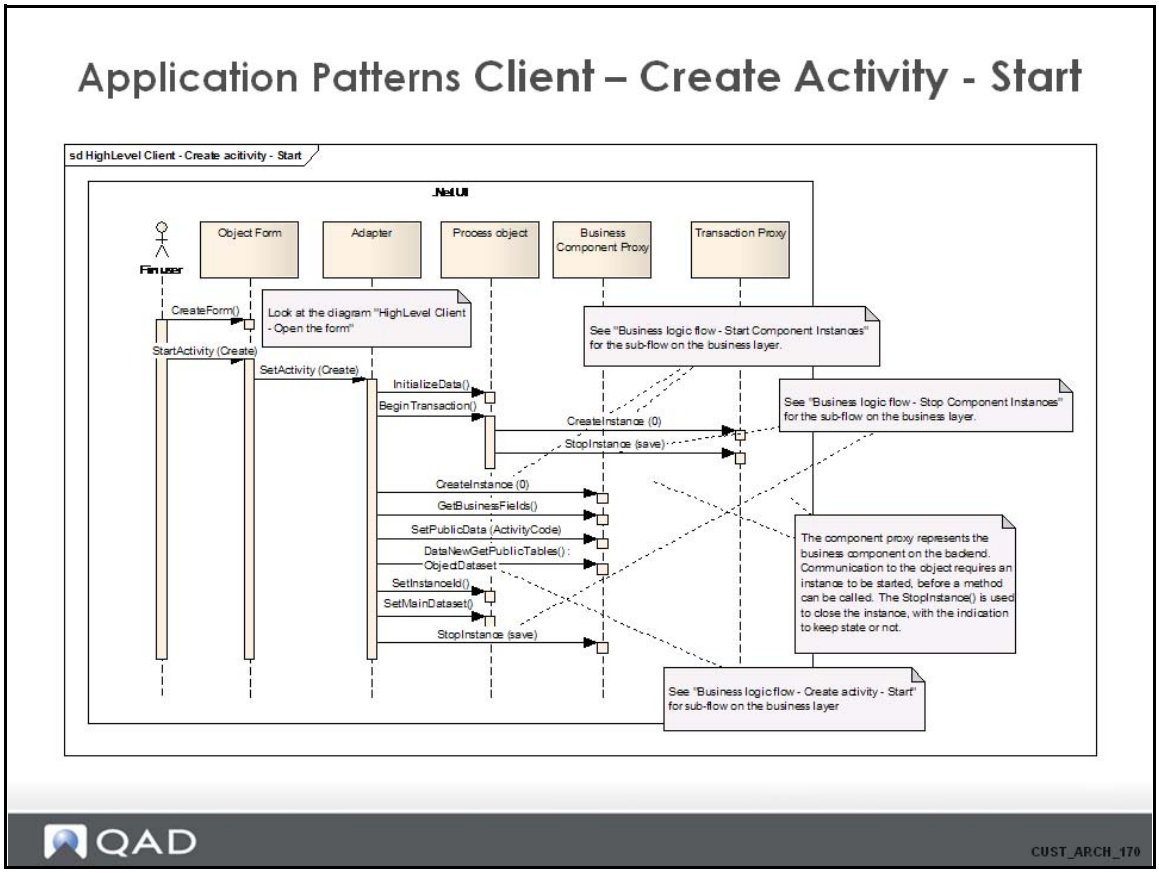
Application Patterns Overview



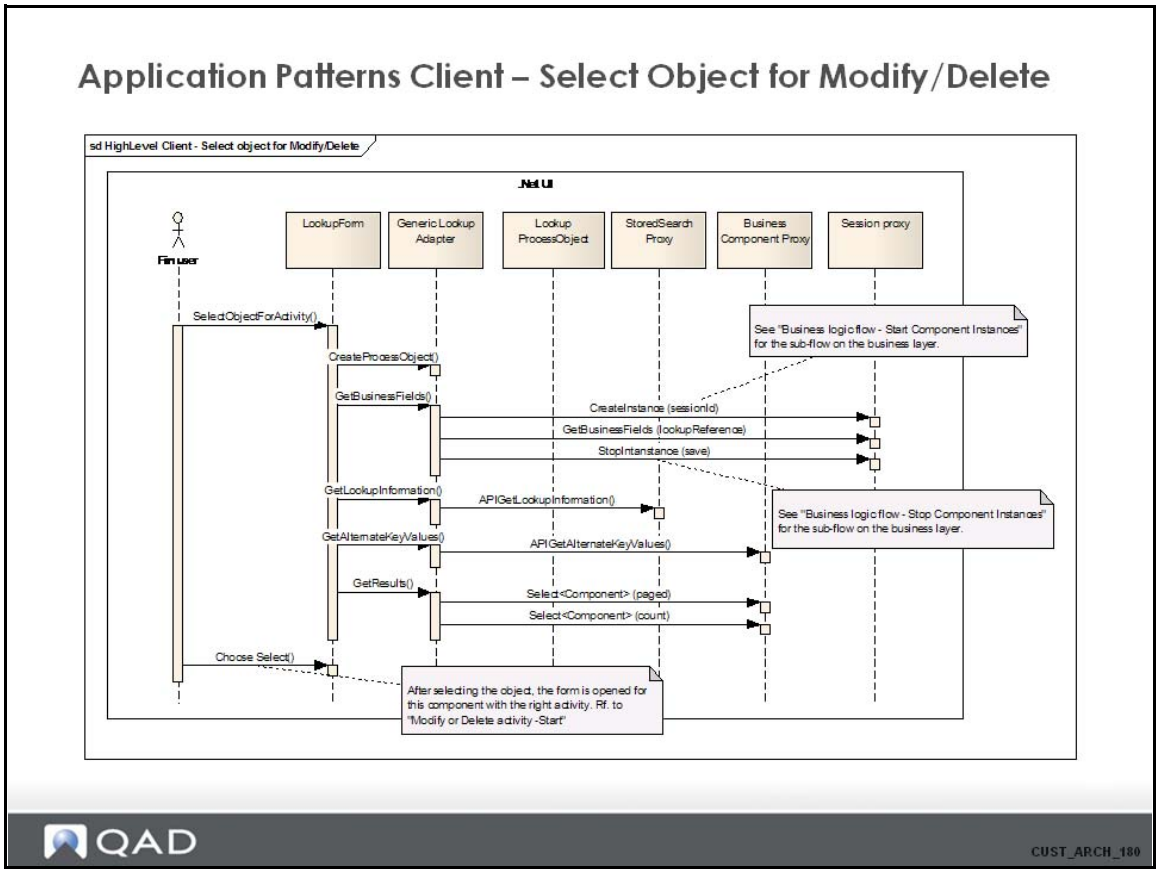
Application Patterns Client



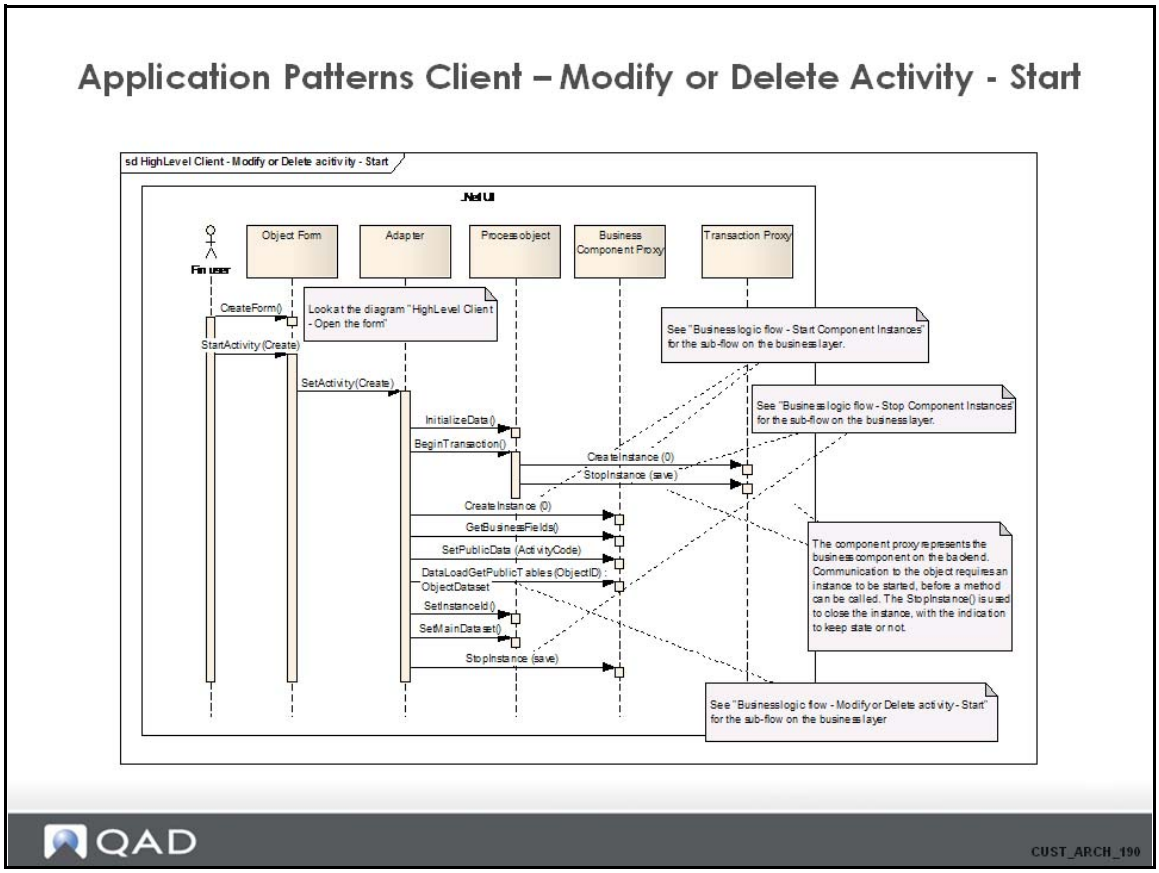
Application Patterns Client 2



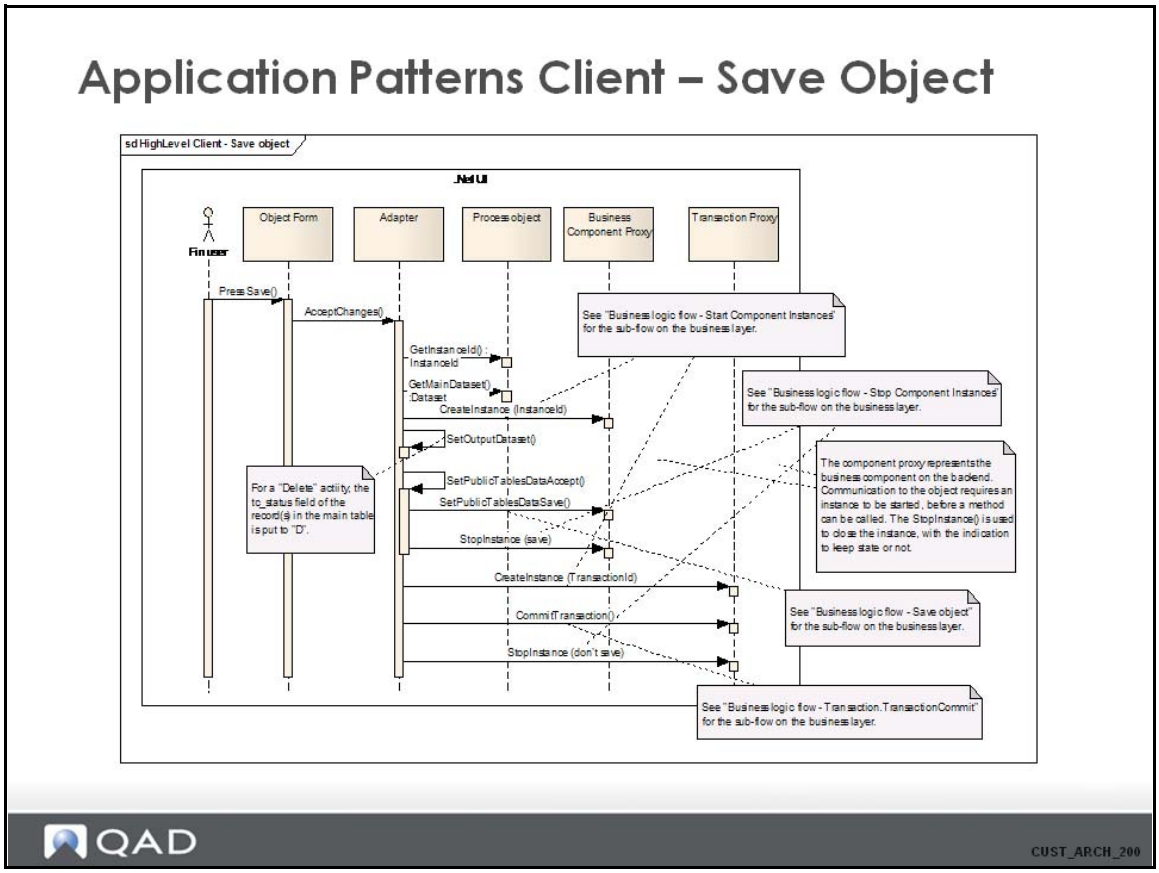
Application Patterns Client 3



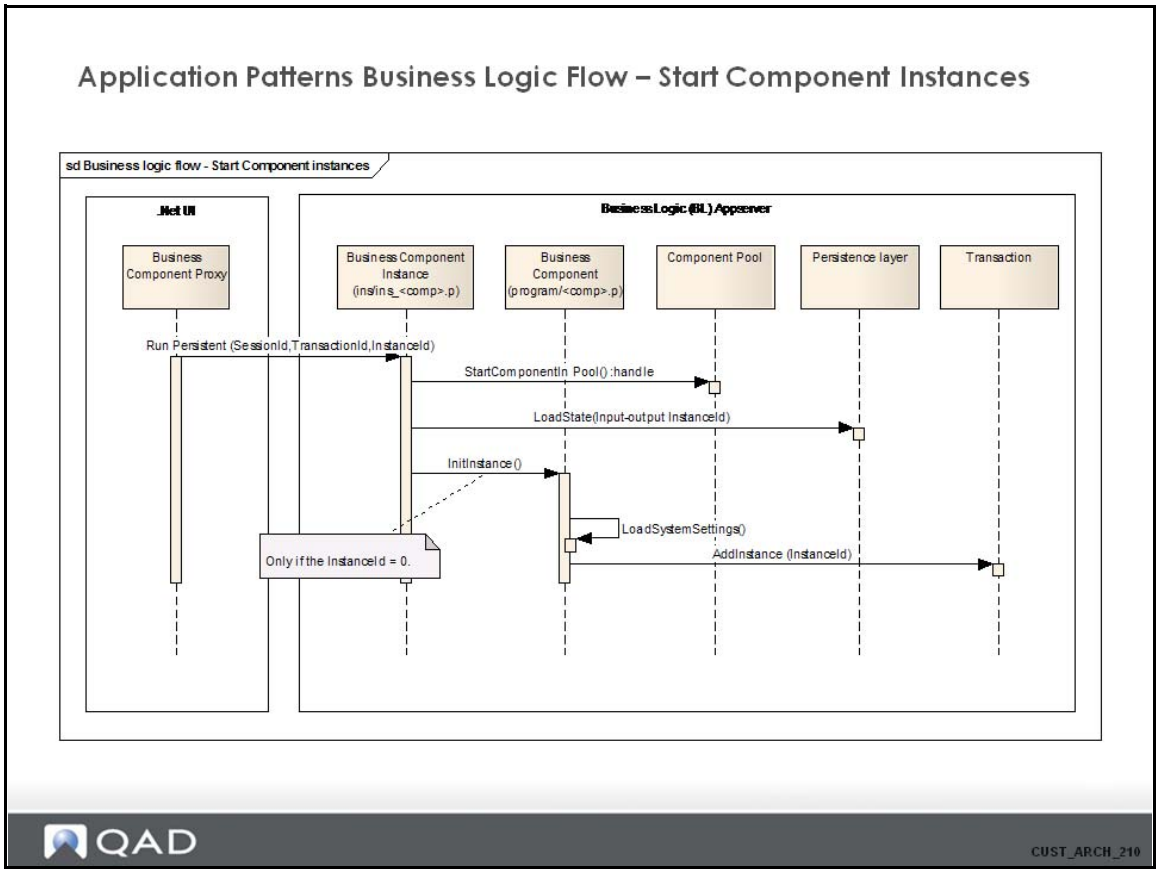
Application Patterns Client 4



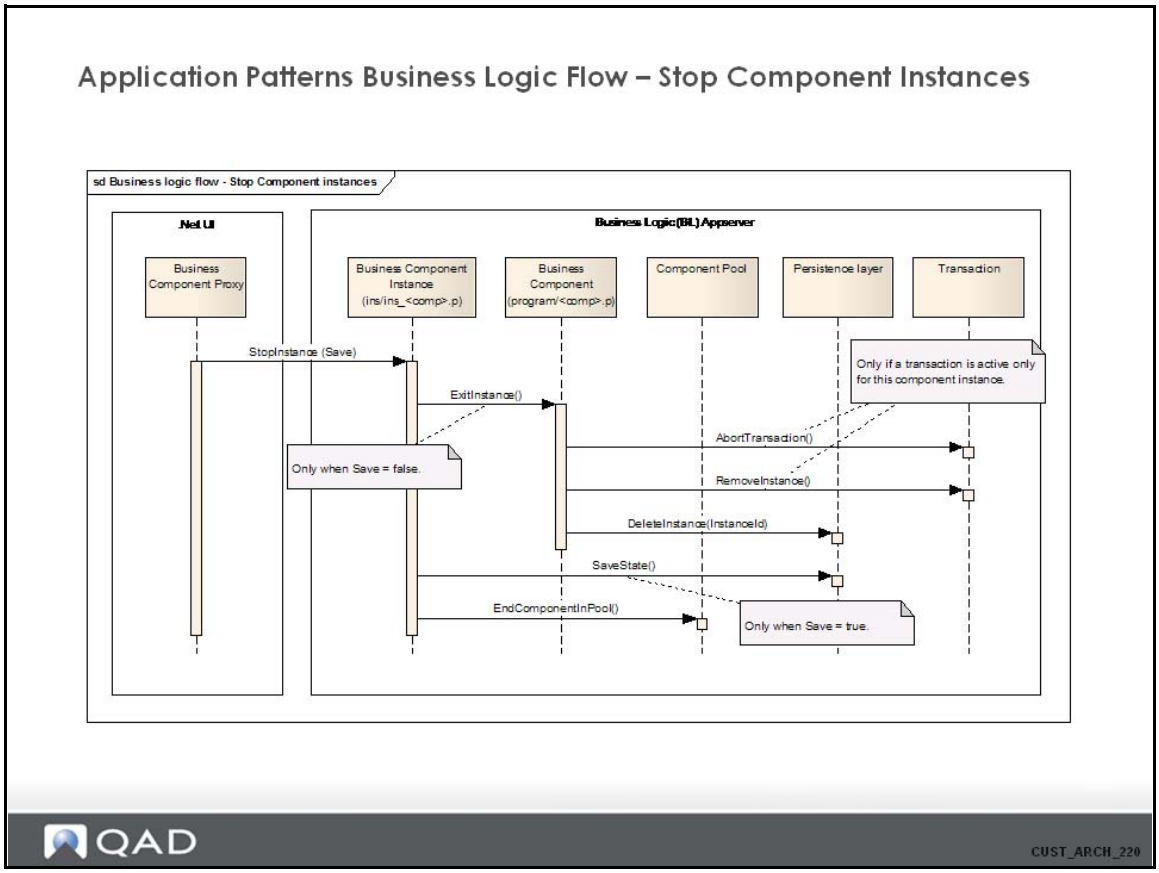
Application Patterns Client 5



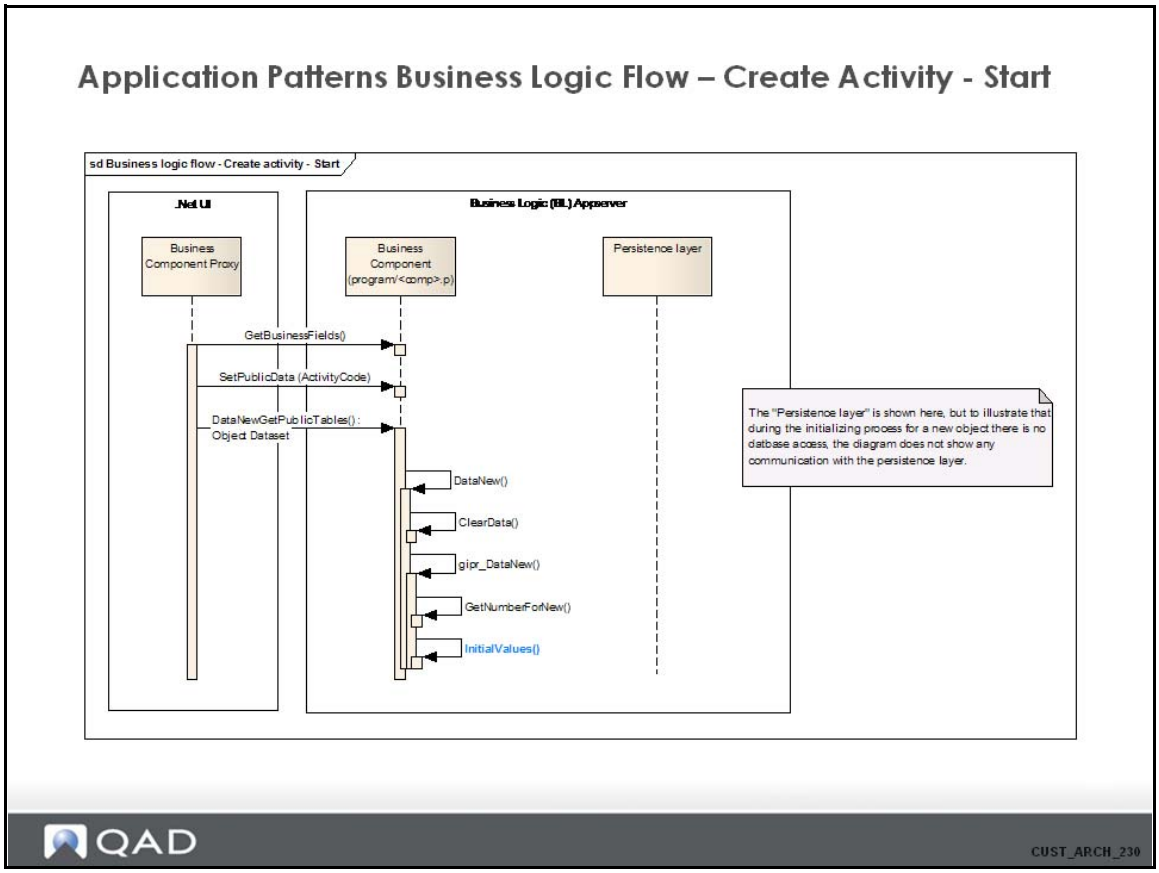
Application Patterns Business Logic Flow



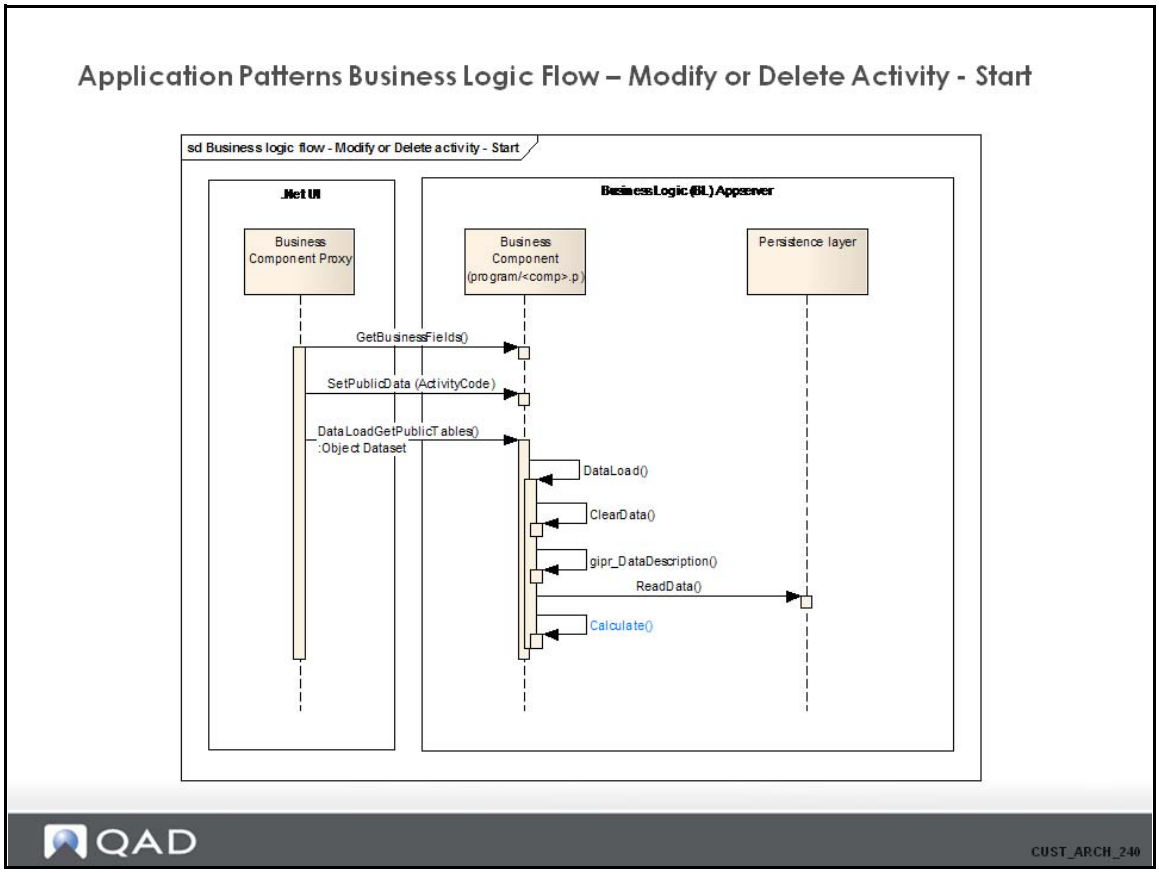
Application Patterns Business Logic Flow 2



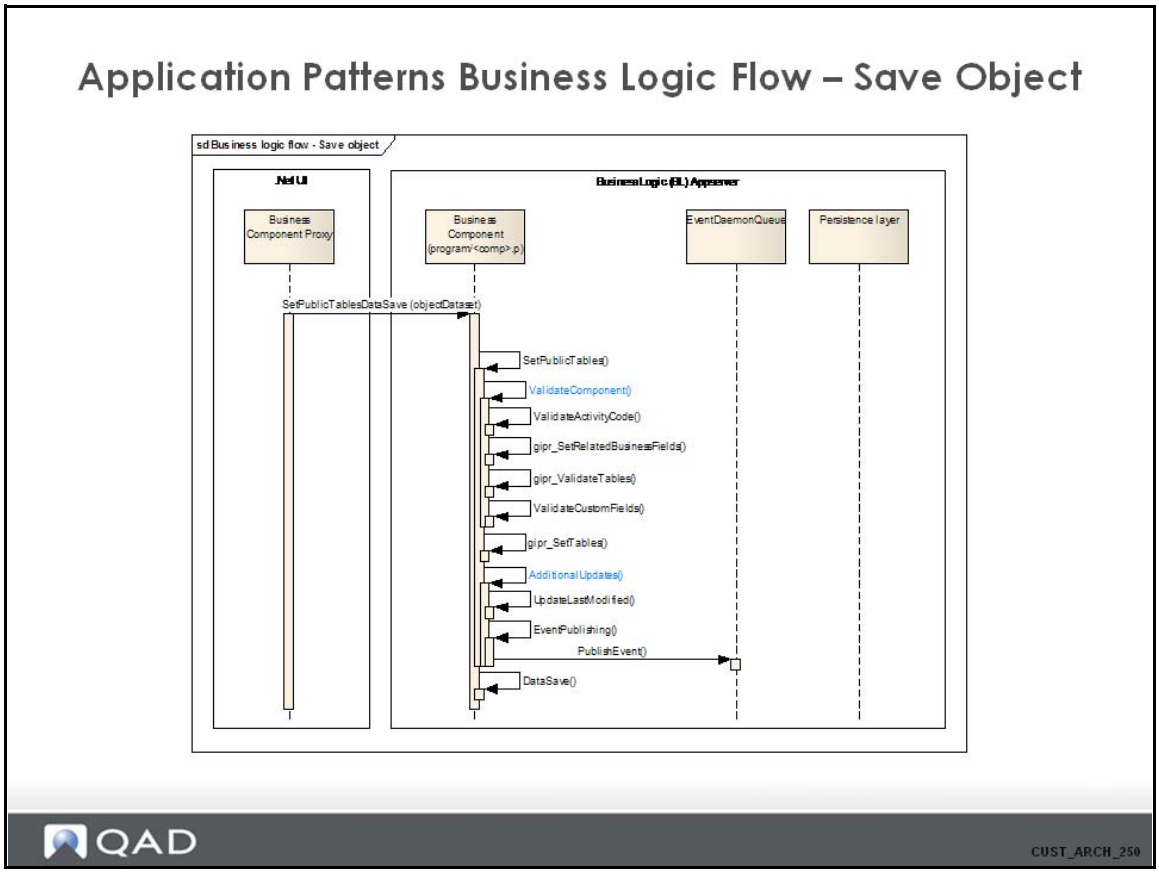
Application Patterns Business Logic Flow 3



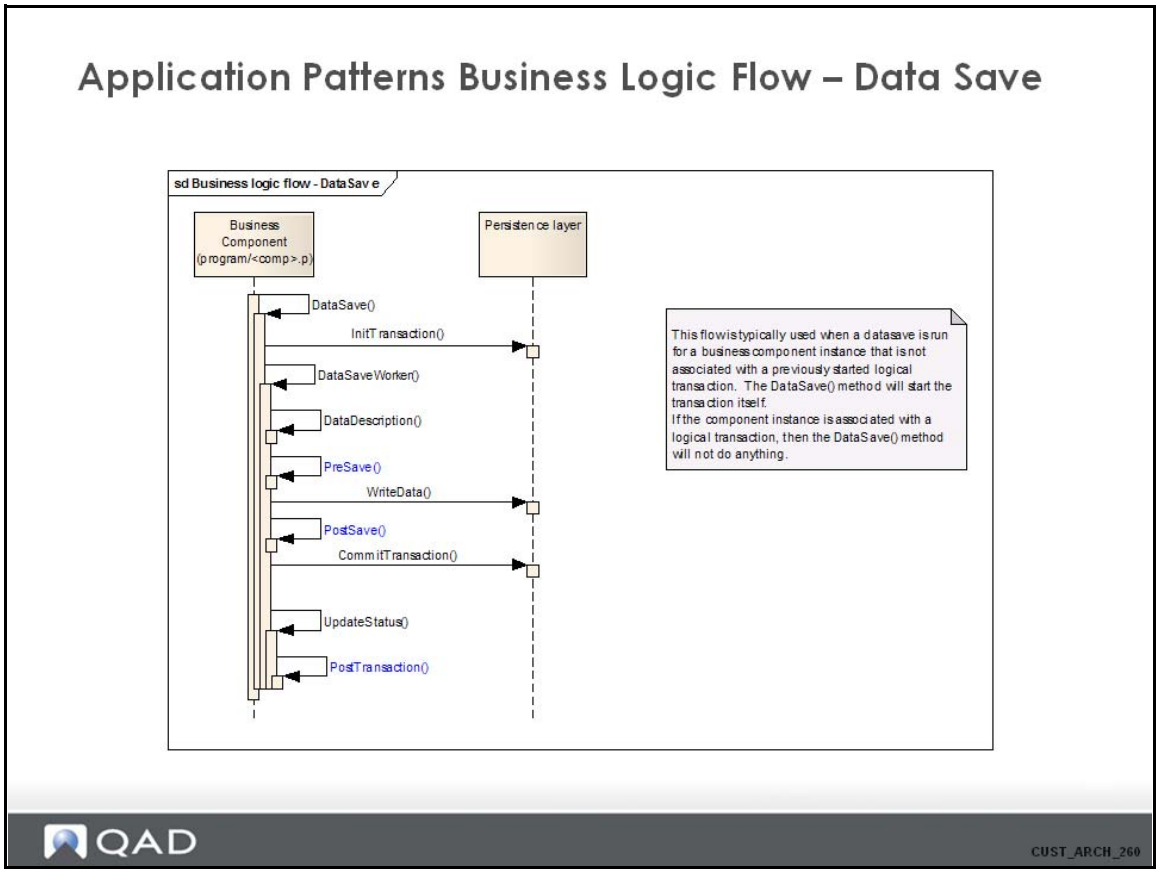
Application Patterns Business Logic Flow 4



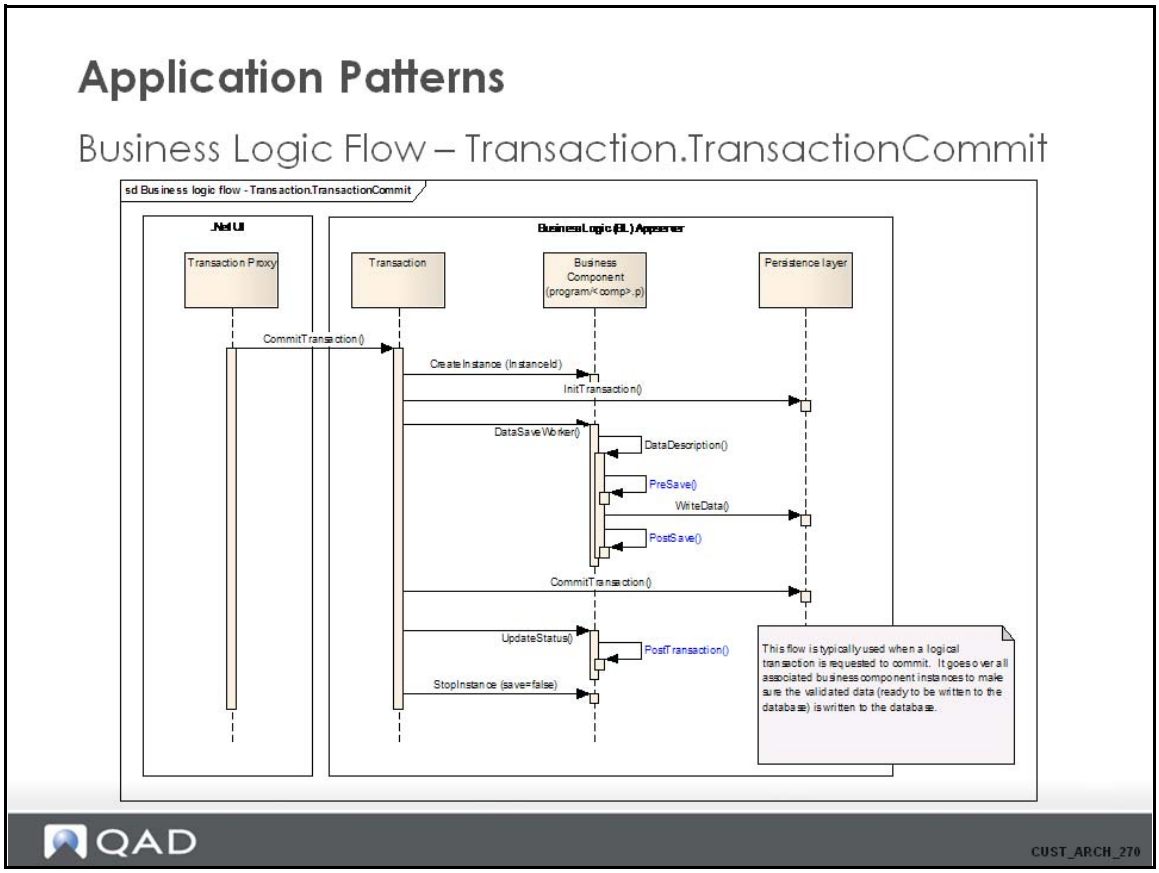
Application Patterns Business Logic Flow 5



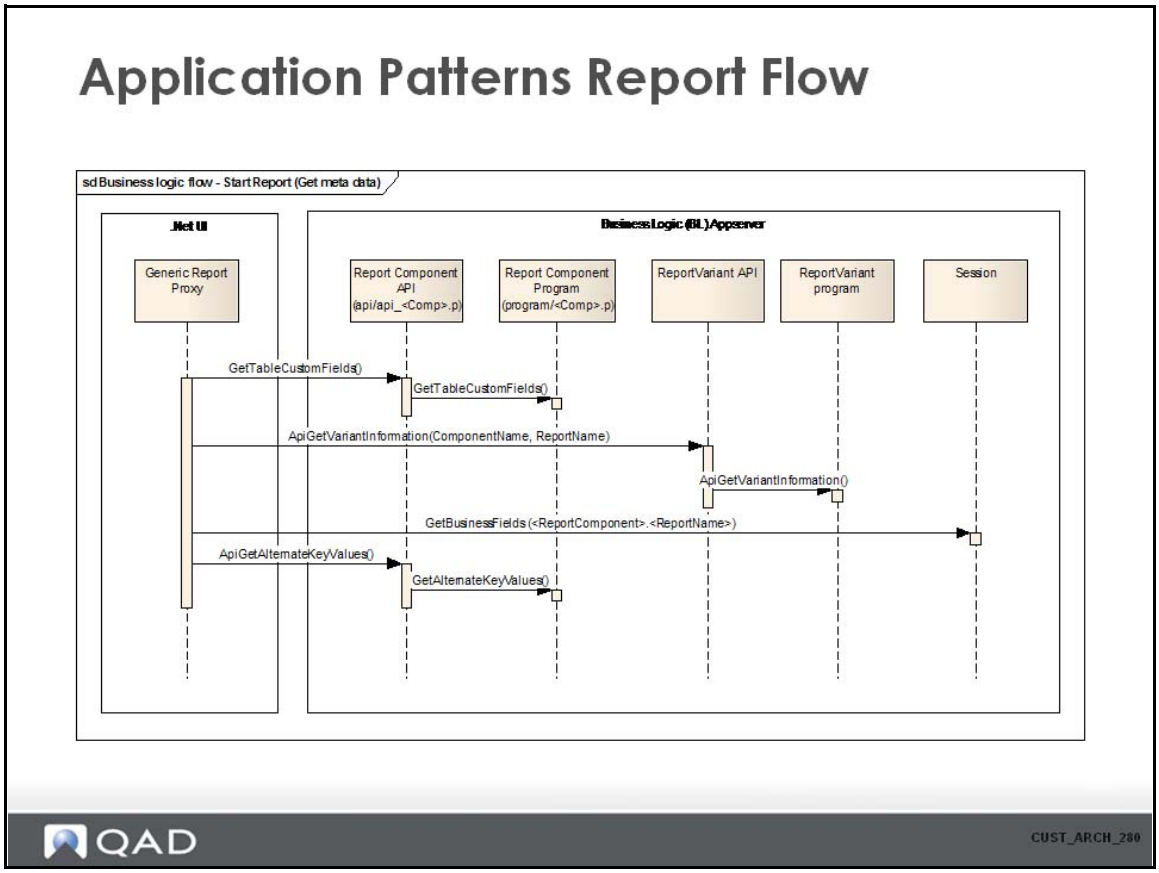
Application Patterns Business Logic Flow 6



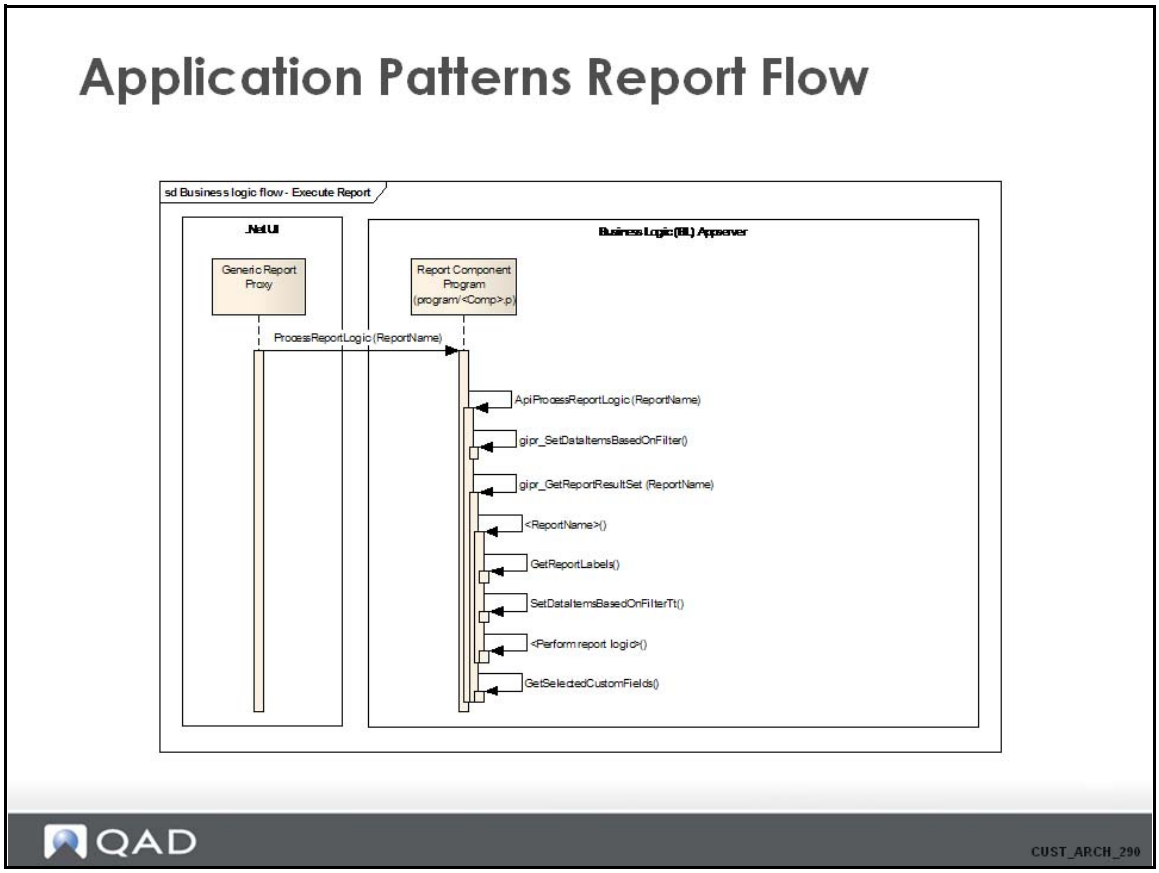
Application Patterns Business Logic Flow 7



Application Patterns Report Flow



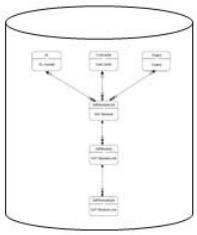
Application Patterns Report Flow 2



Data Handling - Datasets

Data Handling - Datasets

- Widely used in the Financials application layers to keep and manage data
- Imagine a small database in memory
  - A dataset contains one or more temp-tables
  - temp-tables in a dataset are related
  - temp-tables in a dataset can have indexes, and can have constraints defined

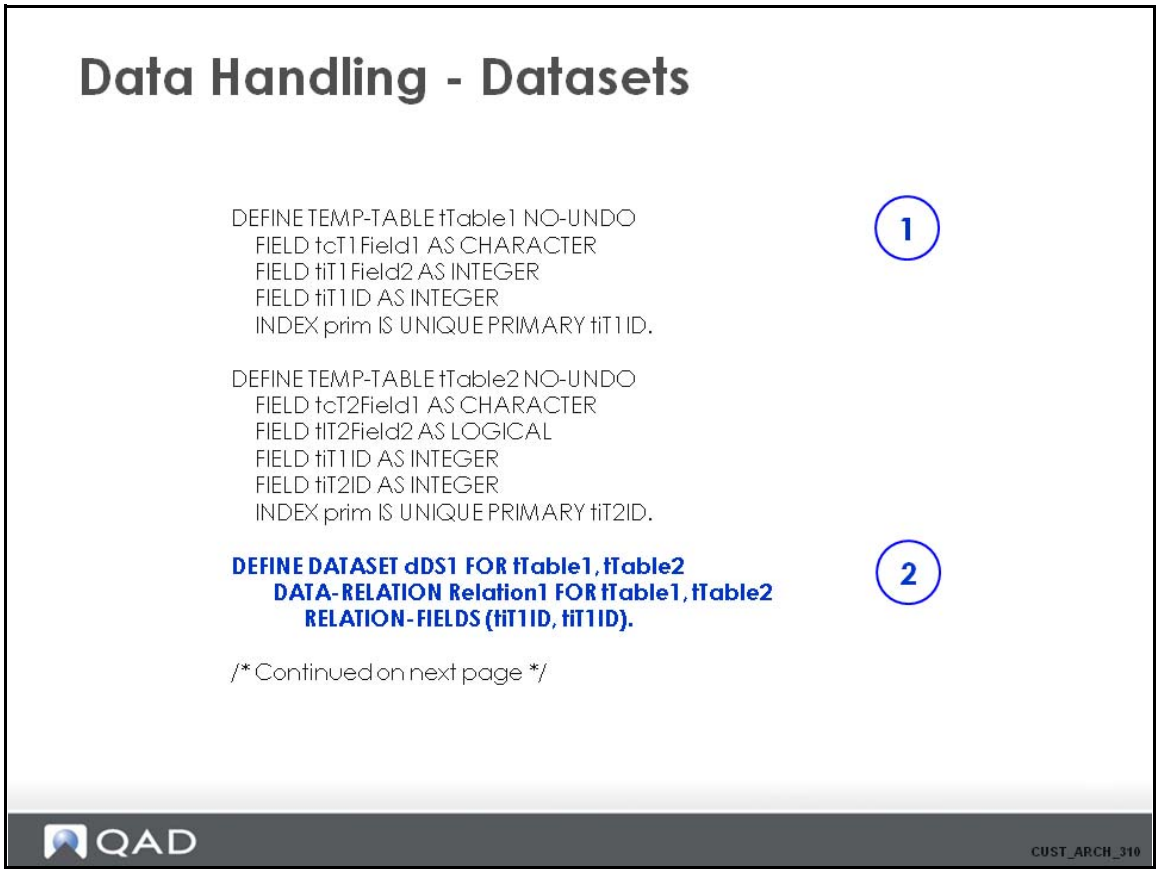


- A dataset can be passed as a parameter to a procedure
- A dataset can easily be serialized to XML format
- A Progress dataset matches a .NET dataset

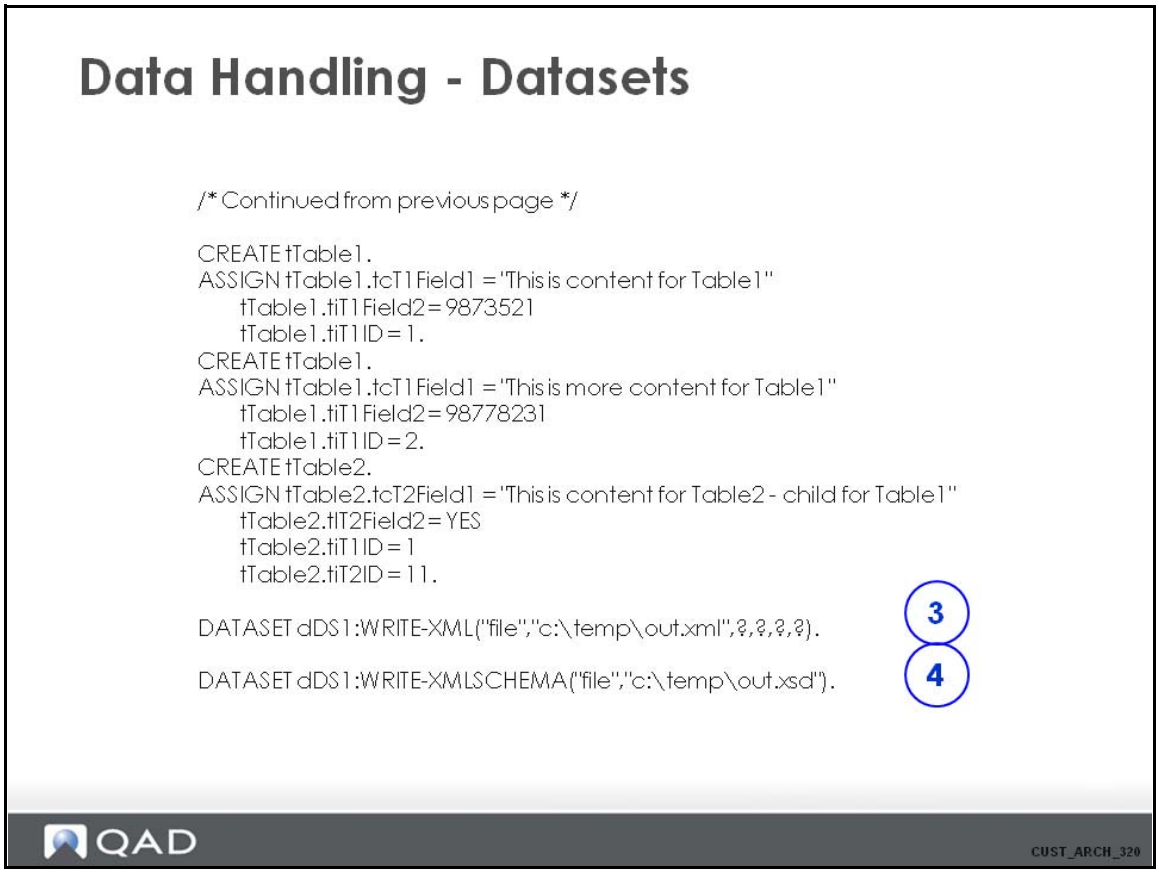
QAD

CUST_ARCH_300

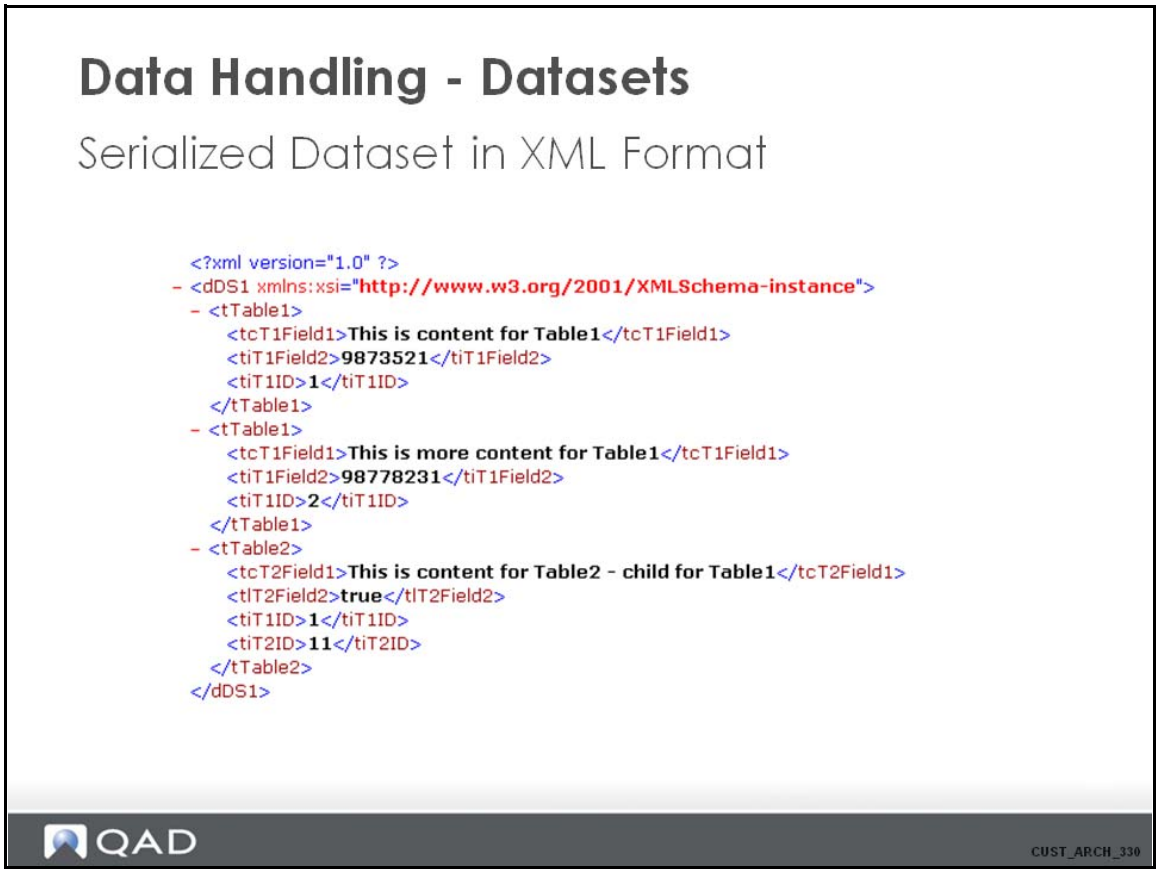
Data Handling - Datasets 2



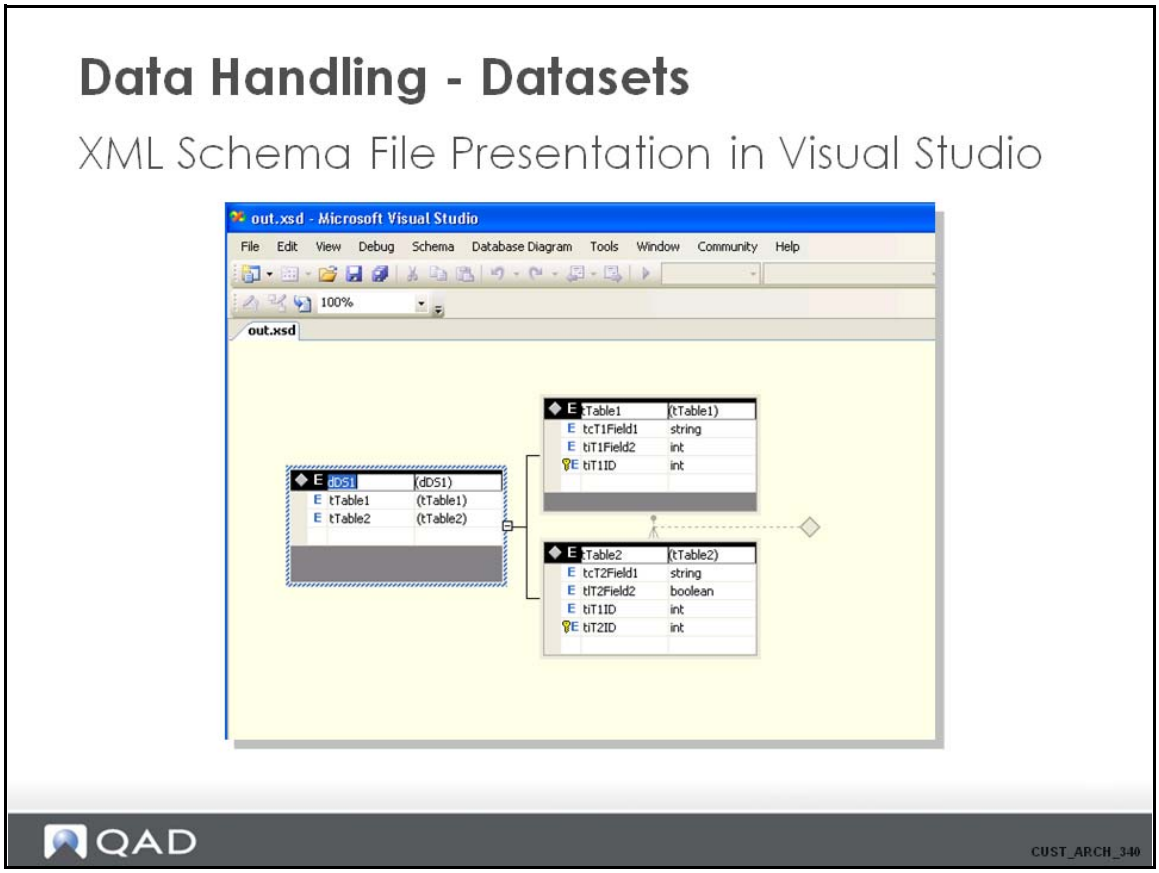
Data Handling - Datasets 3



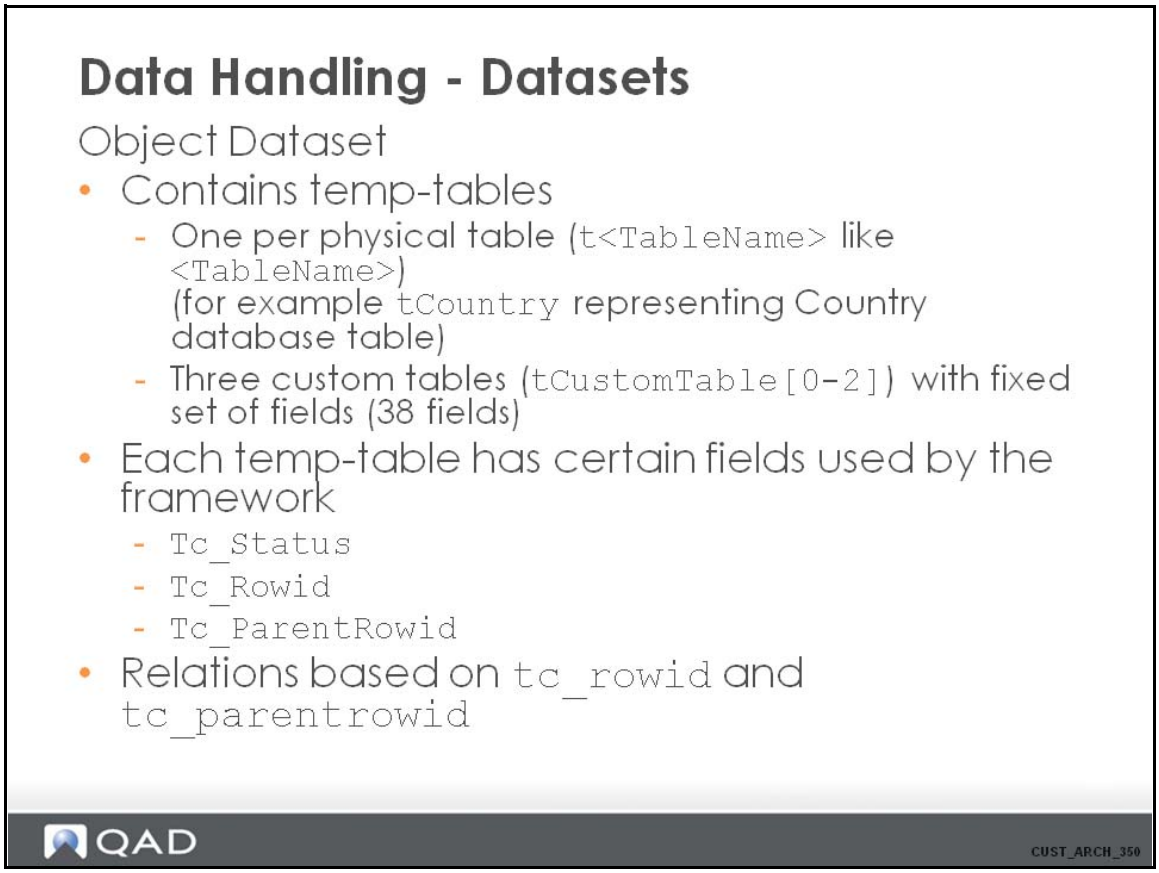
Data Handling - Datasets 4



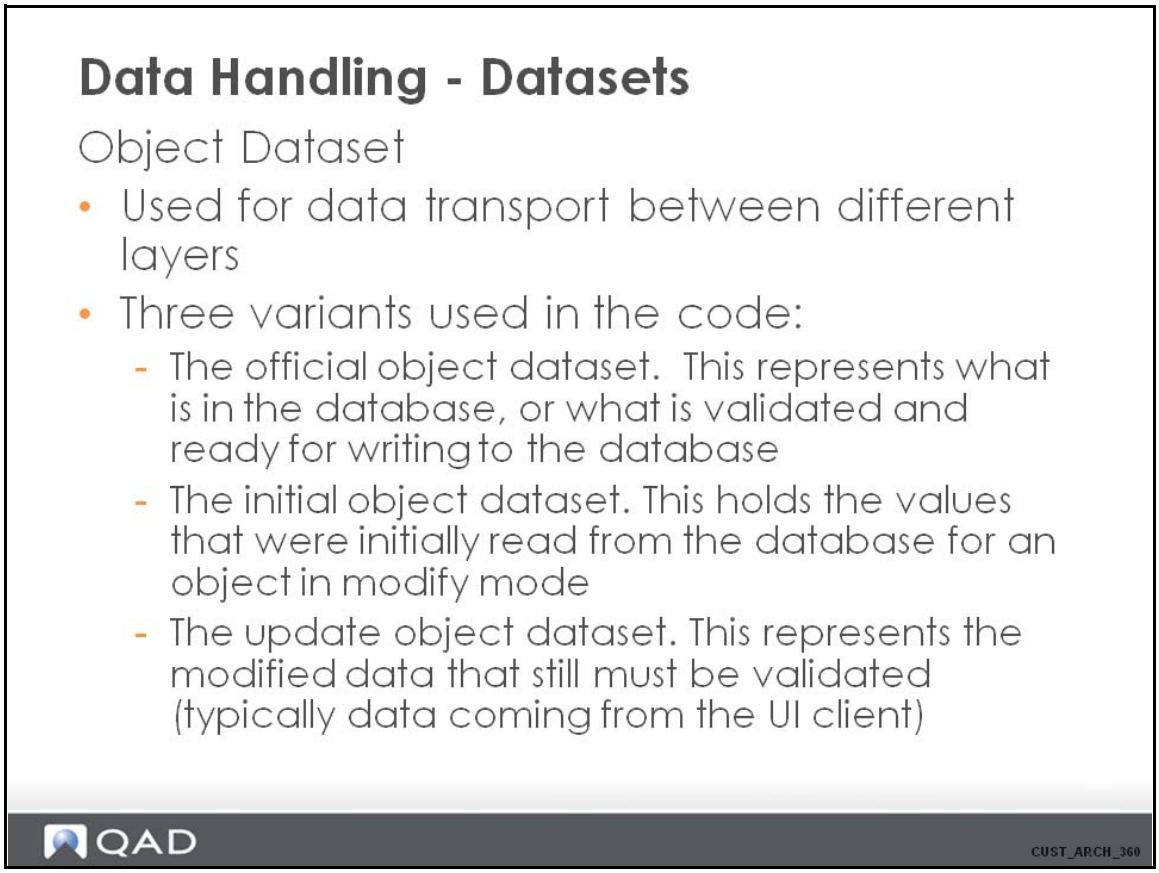
Data Handling - Datasets 5



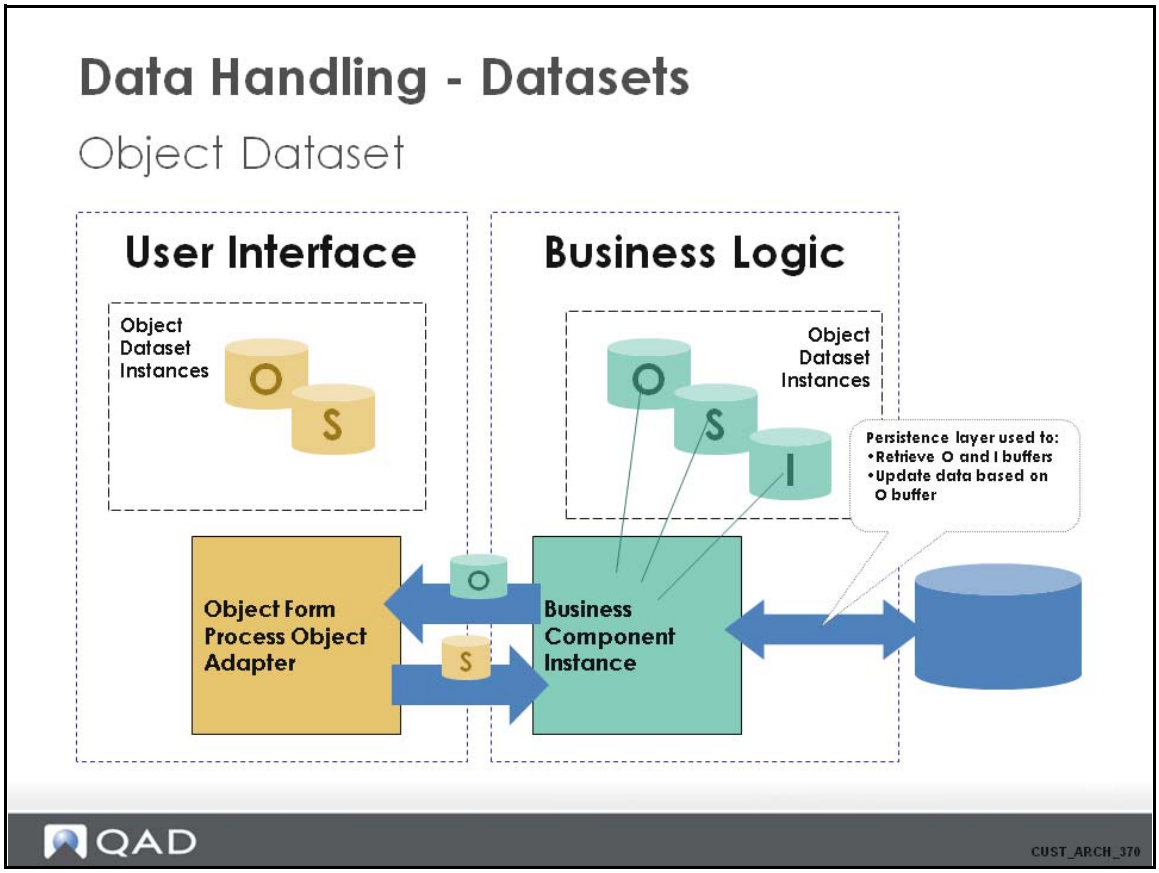
Data Handling - Datasets 6



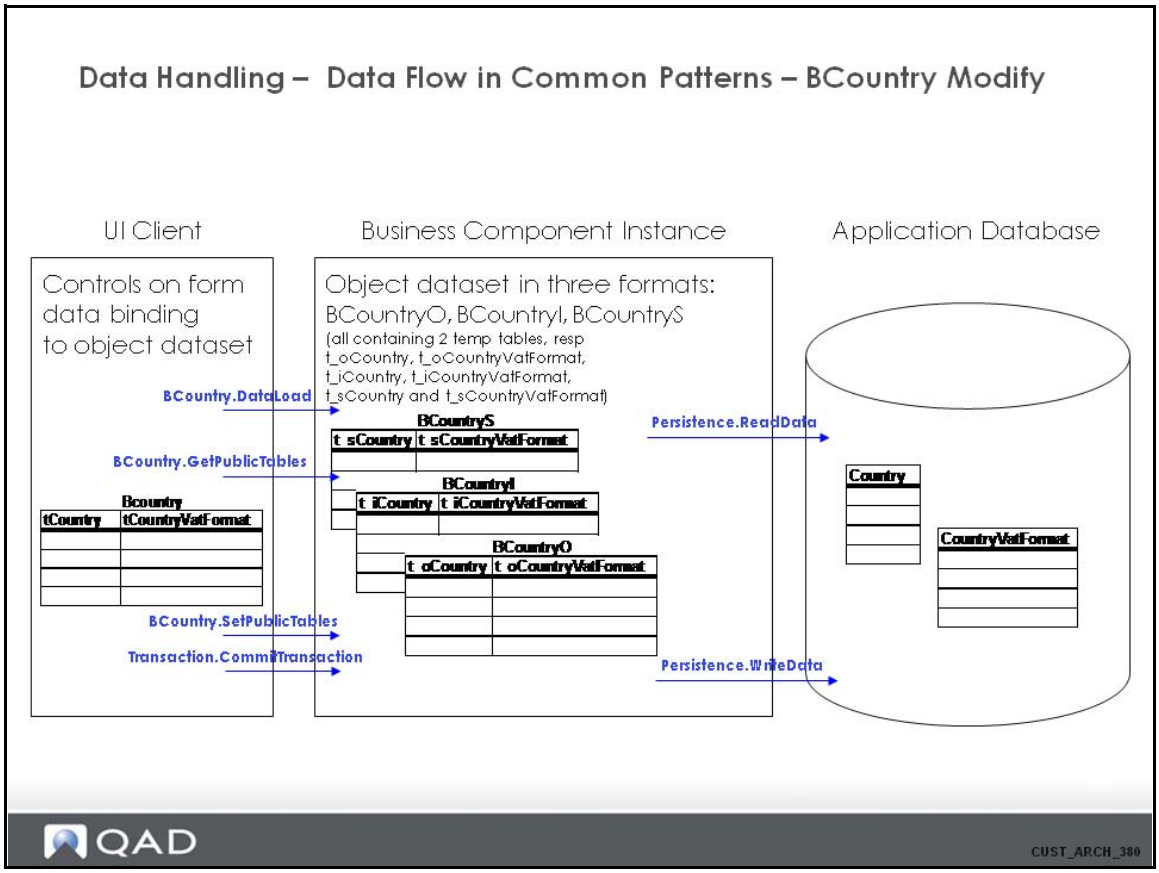
Data Handling - Datasets 7



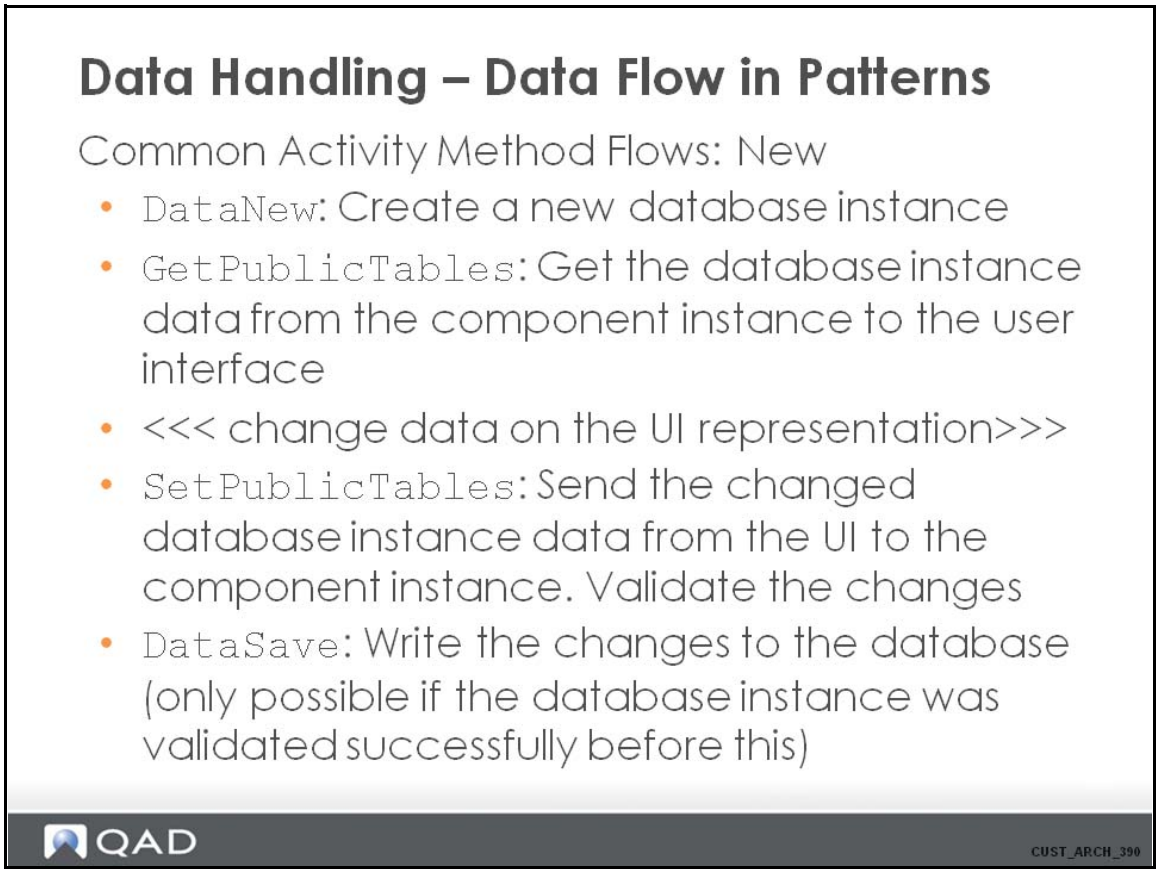
Data Handling - Datasets 8



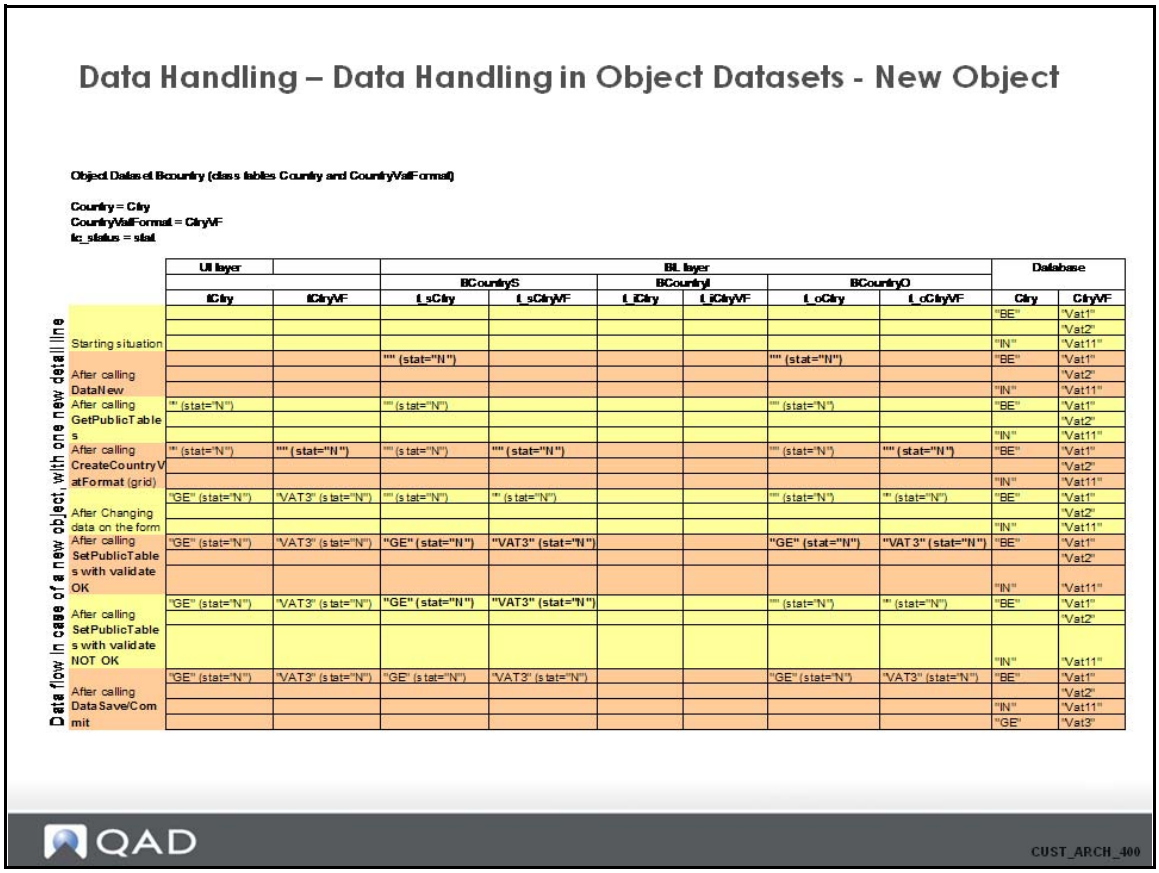
Data Handling - Data Flow In Common Patterns



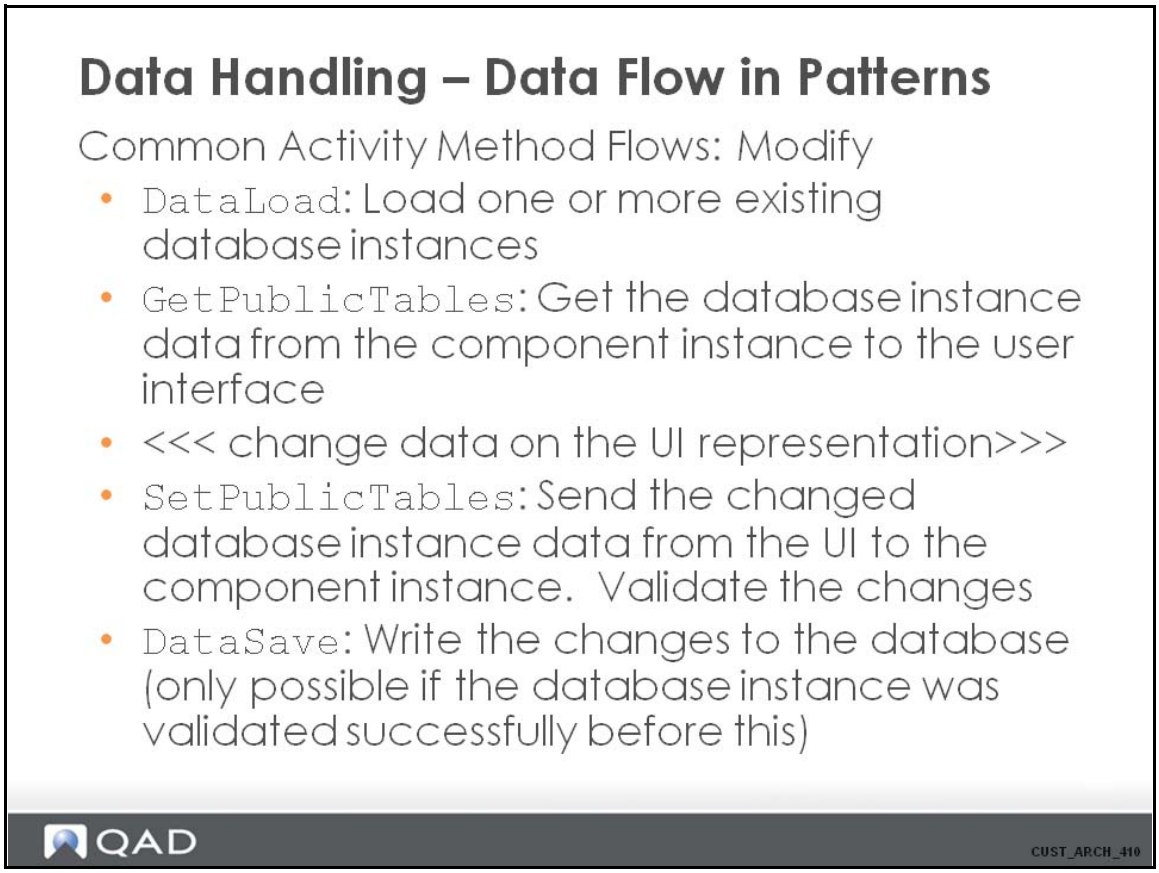
Data Handling - Data Flow in Patterns



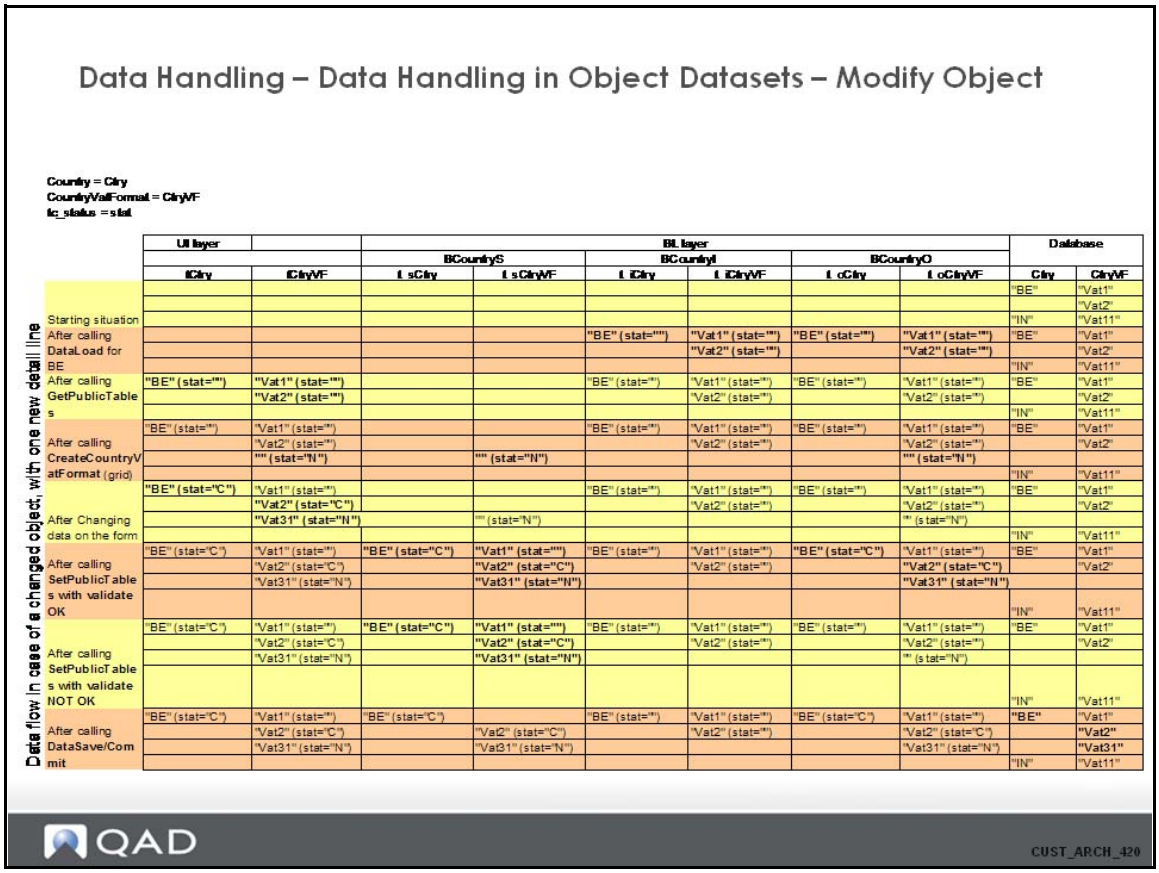
Data Handling - Data Handling in Object Datasets - New Object



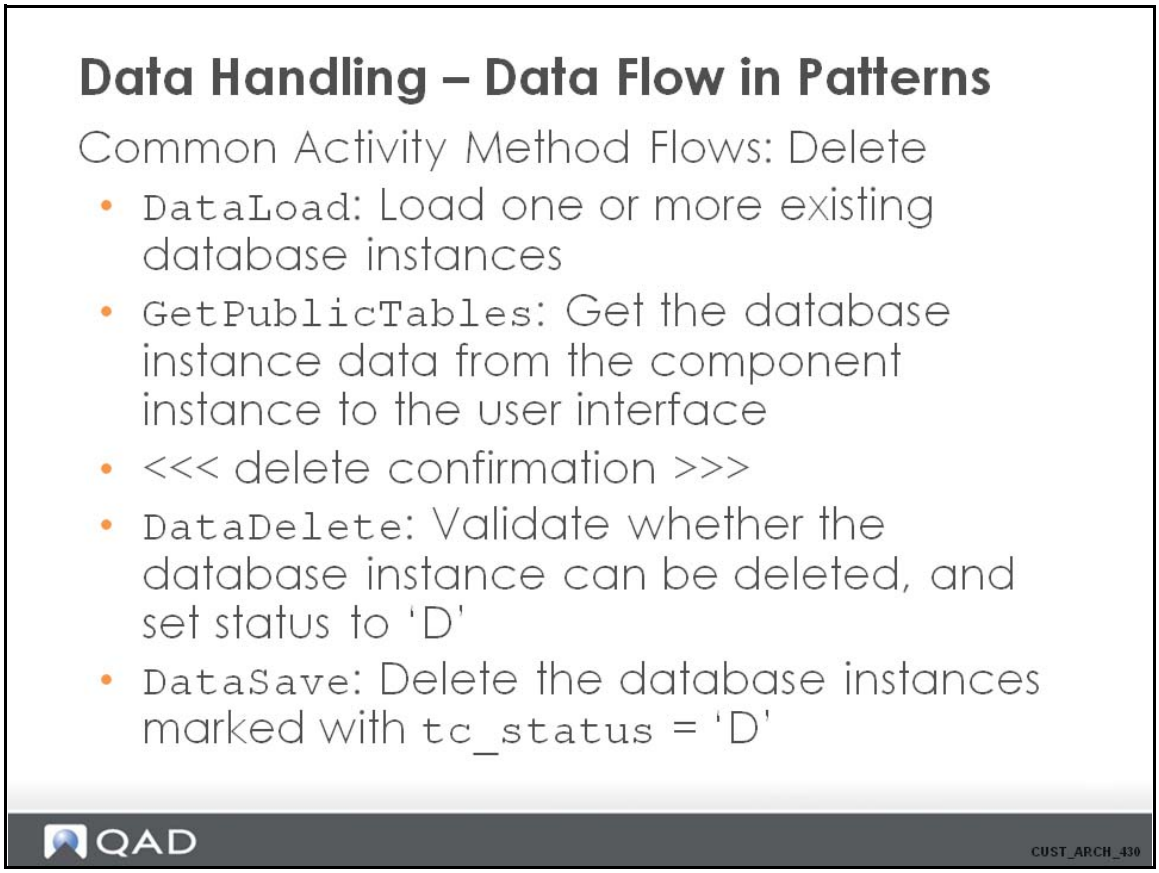
Data Handling - Data Flow in Patterns



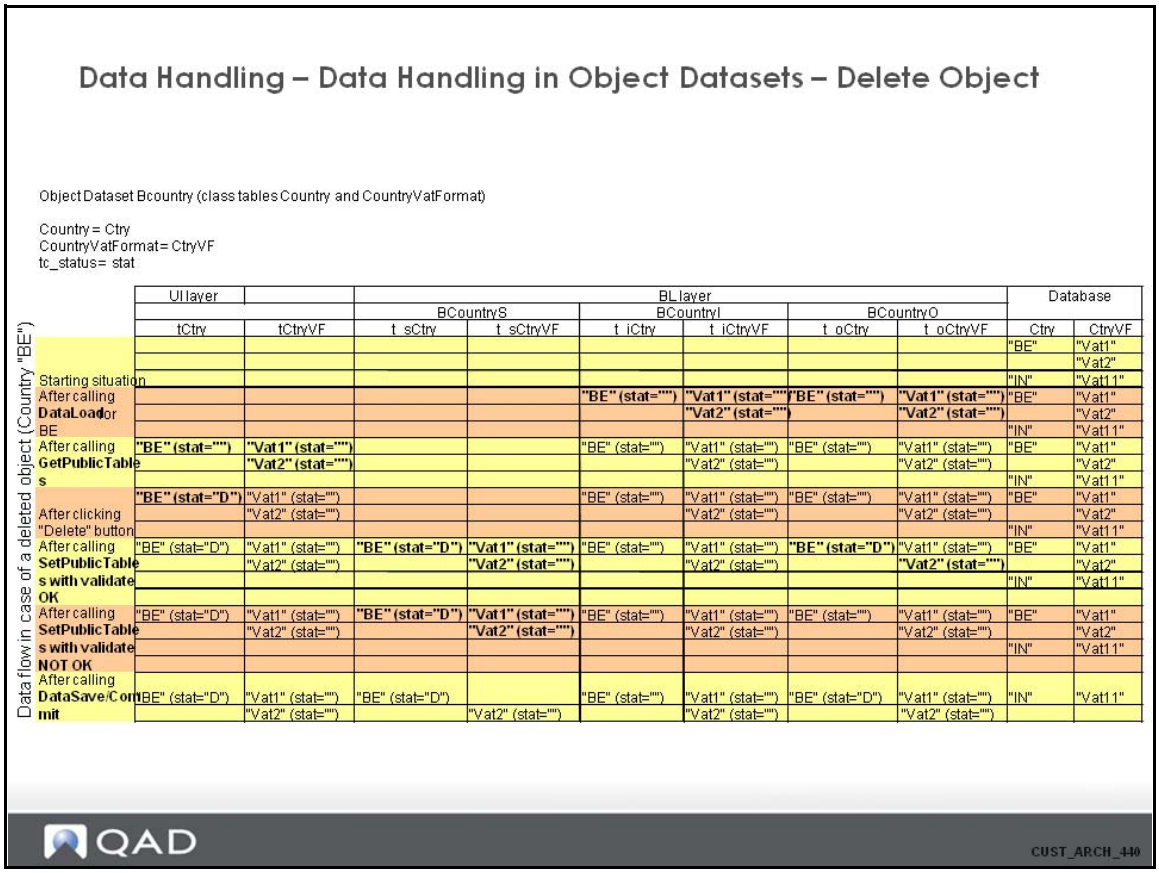
Data Handling - Data Handling in Object Datasets - Modify Object



Data Handling - Data Flow in Patterns



Data Handling - Data Handling in Object Datasets - Delete Objects



Data Handling – Data Handling in Object Datasets – Delete Object

Object Dataset Bcountry (class tables Country and CountryVatFormat)

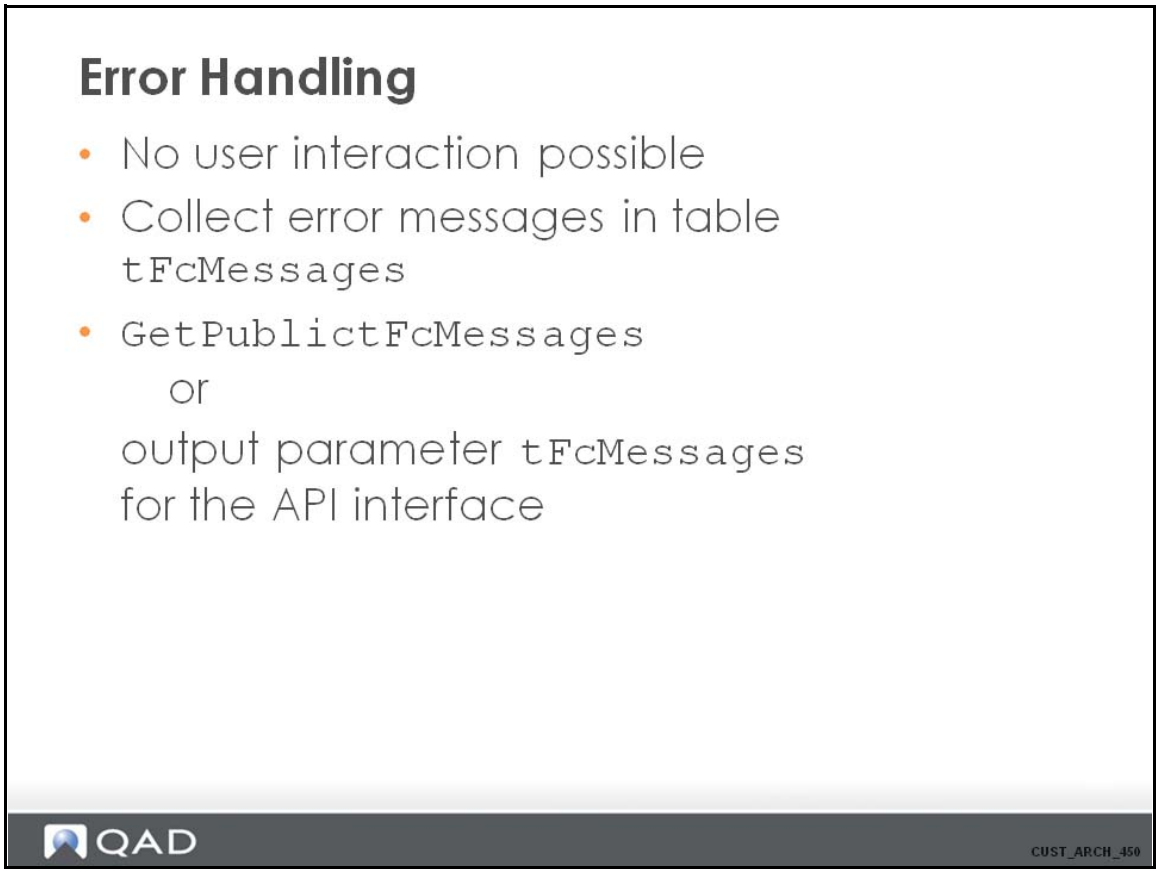
Country = Ctry
CountryVatFormat = CtryVF
tc_status = stat

Data flow in case of a deleted object (Country "BE")

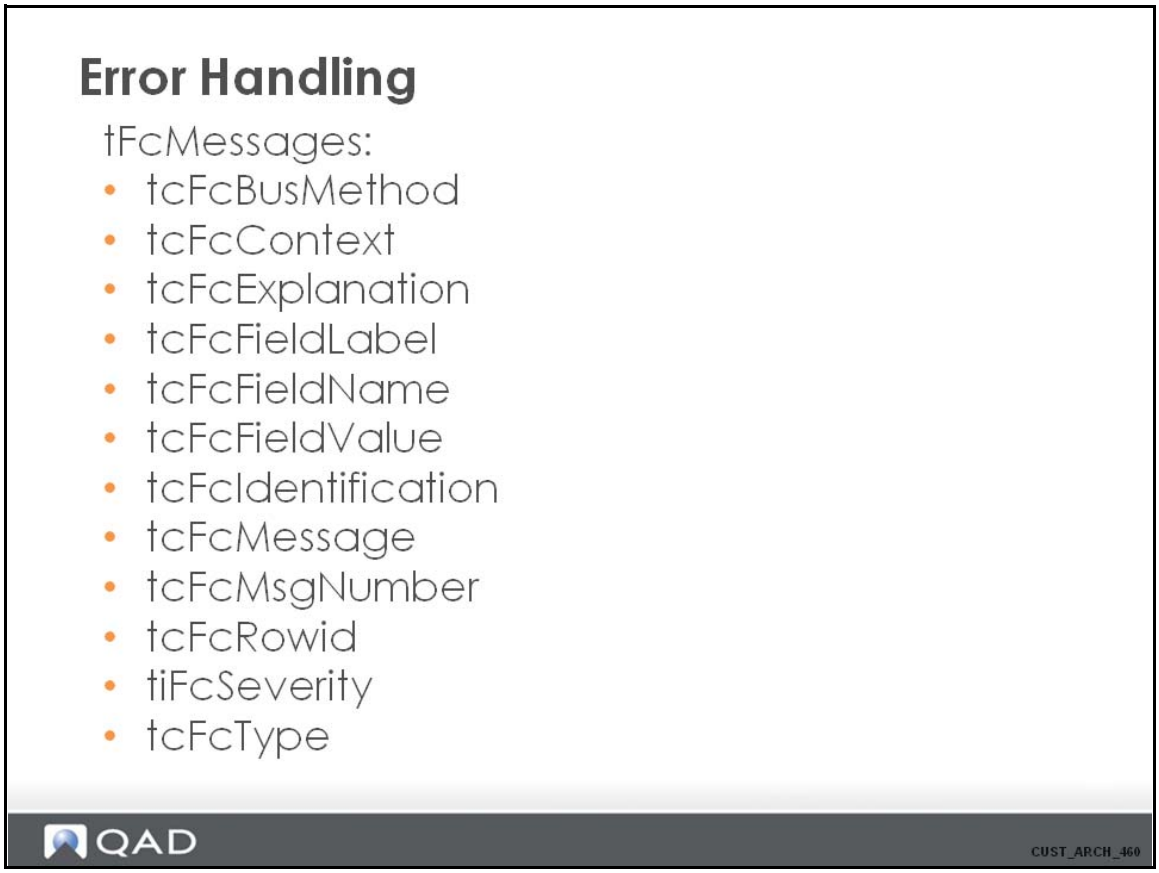
	UI layer		BL layer						Database	
			BCountryS		BCountryI		BCountryO			
	tCtry	tCtryVF	t_sCtry	t_sCtryVF	t_iCtry	t_iCtryVF	t_oCtry	t_oCtryVF	Ctry	CtryVF
									"BE"	"Vat1"
										"Vat2"
Starting situation									"IN"	"Vat11"
After calling **DataLoad**or BE					**"BE" (stat="")**	**"Vat1" (stat="")**	**"BE" (stat="")**	**"Vat1" (stat="")**	"BE"	"Vat1"
						"Vat2" (stat="")		**"Vat2" (stat="")**		"Vat2"
									"IN"	"Vat11"
After calling **GetPublicTable**	**"BE" (stat="")**	**"Vat1" (stat="")**			"BE" (stat="")	"Vat1" (stat="")	"BE" (stat="")	"Vat1" (stat="")	"BE"	"Vat1"
s		**"Vat2" (stat="")**				"Vat2" (stat="")		"Vat2" (stat="")		"Vat2"
									"IN"	"Vat11"
After clicking "Delete" button	**"BE" (stat="D")**	"Vat1" (stat="")			"BE" (stat="")	"Vat1" (stat="")	"BE" (stat="")	"Vat1" (stat="")	"BE"	"Vat1"
		"Vat2" (stat="")				"Vat2" (stat="")		"Vat2" (stat="")		"Vat2"
									"IN"	"Vat11"
After calling **SetPublicTable**	"BE" (stat="D")	"Vat1" (stat="")	**"BE" (stat="D")**	**"Vat1" (stat="")**	"BE" (stat="")	"Vat1" (stat="")	**"BE" (stat="D")**	"Vat1" (stat="")	"BE"	"Vat1"
s with validate		"Vat2" (stat="")		**"Vat2" (stat="")**		"Vat2" (stat="")		**"Vat2" (stat="")**		"Vat2"
OK									"IN"	"Vat11"
After calling **SetPublicTable**	"BE" (stat="D")	"Vat1" (stat="")	**"BE" (stat="D")**	**"Vat1" (stat="")**	"BE" (stat="")	"Vat1" (stat="")	"BE" (stat="")	"Vat1" (stat="")	"BE"	"Vat1"
s with validate		"Vat2" (stat="")		**"Vat2" (stat="")**		"Vat2" (stat="")		"Vat2" (stat="")		"Vat2"
NOT OK									"IN"	"Vat11"
After calling **DataSave/Com**	"BE" (stat="D")	"Vat1" (stat="")	"BE" (stat="D")		"BE" (stat="")	"Vat1" (stat="")	"BE" (stat="D")	"Vat1" (stat="")	"IN"	"Vat11"
mit		"Vat2" (stat="")		"Vat2" (stat="")		"Vat2" (stat="")		"Vat2" (stat="")		

QAD CUST_ARCH_440

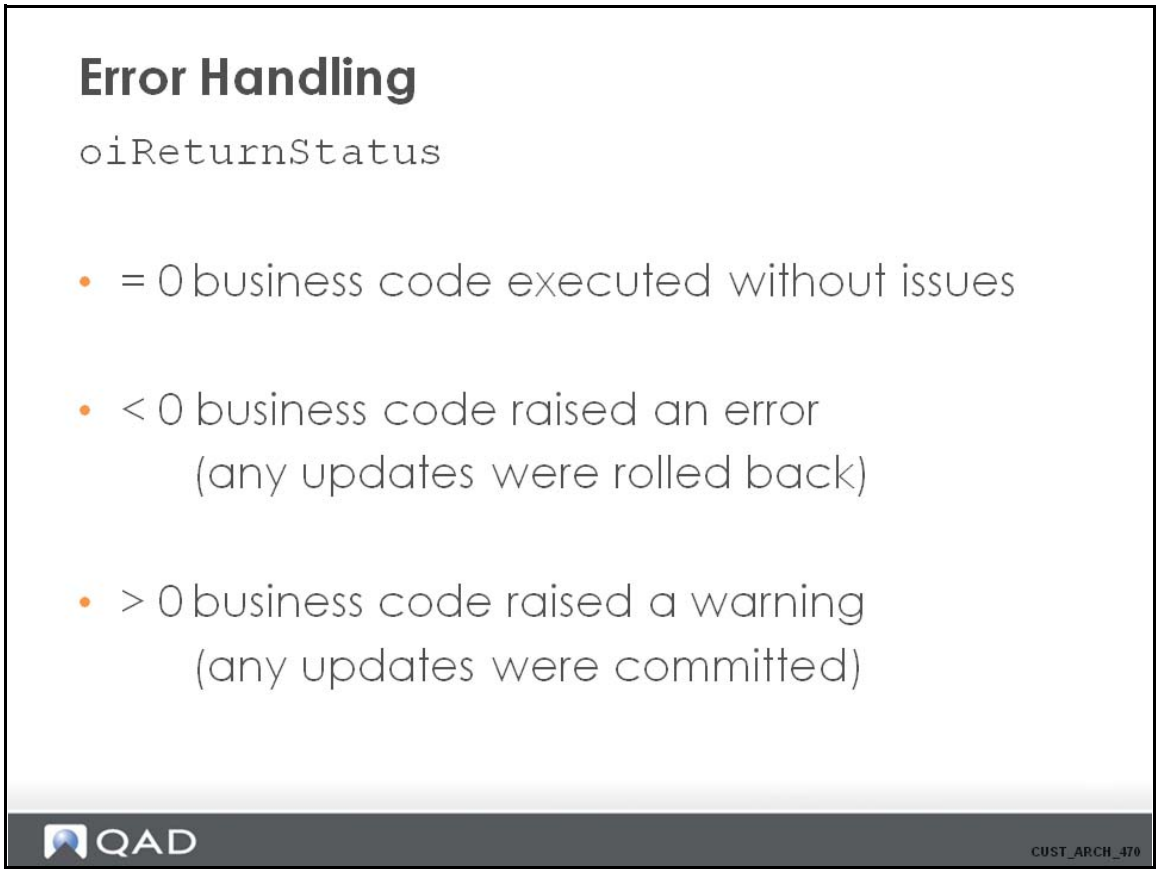
Error Handling



Error Handling 2

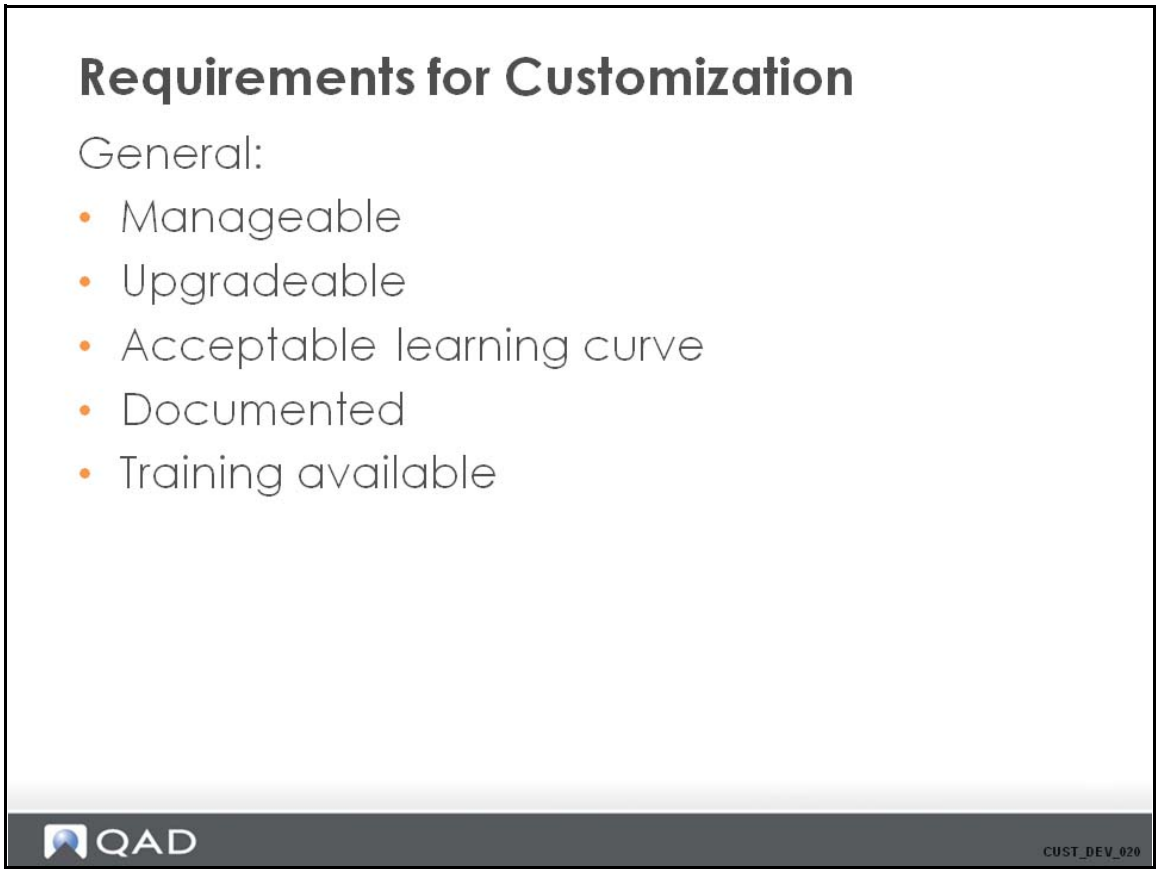


Error Handling 3

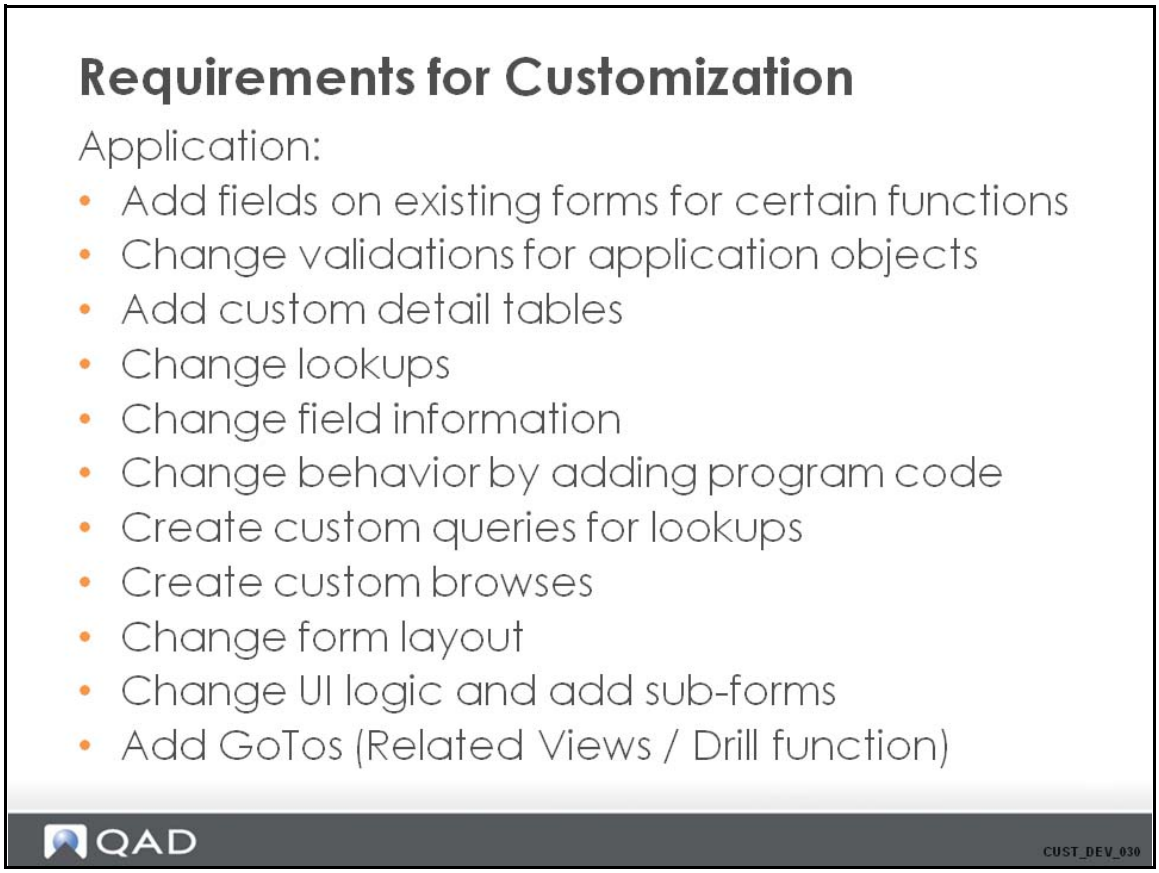


Non-intrusive Customization: Development

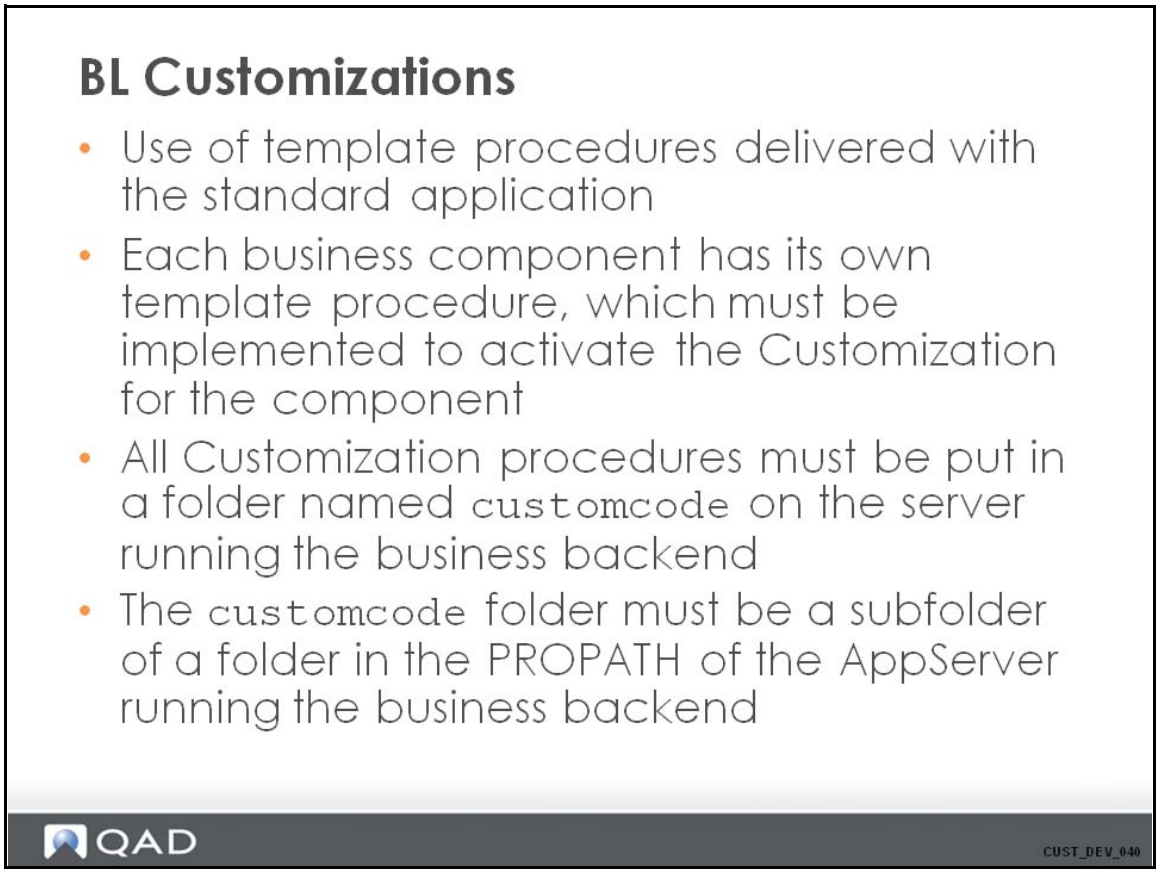
Requirements for Customization



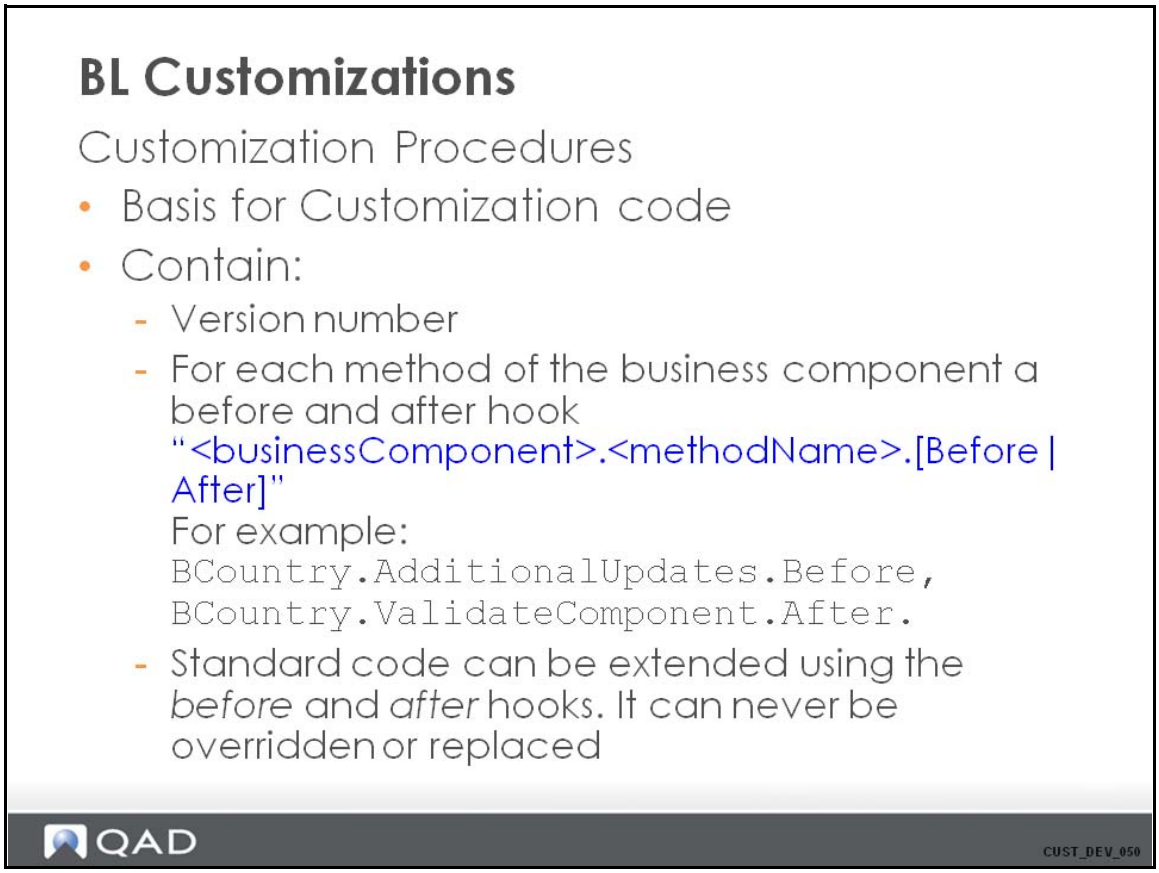
Requirements for Customization 2



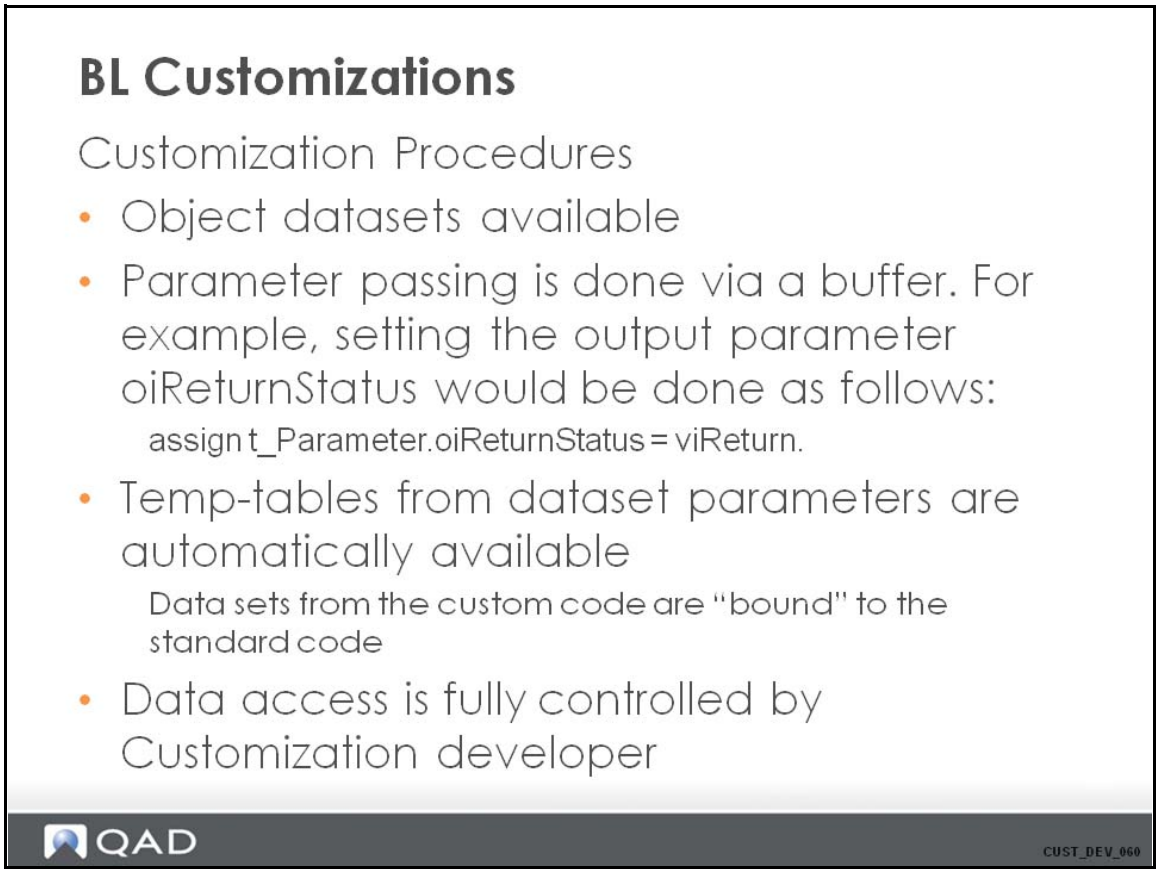
BL Customizations



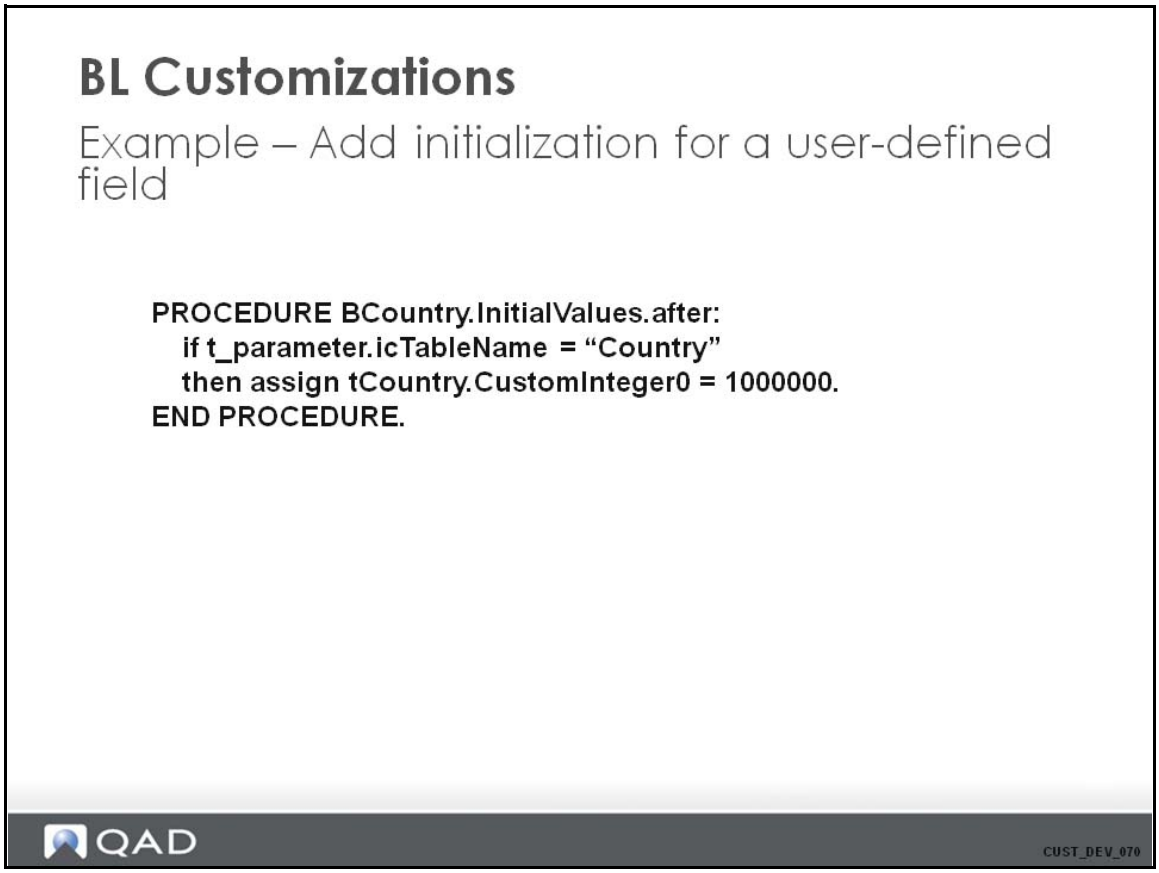
BL Customizations 2



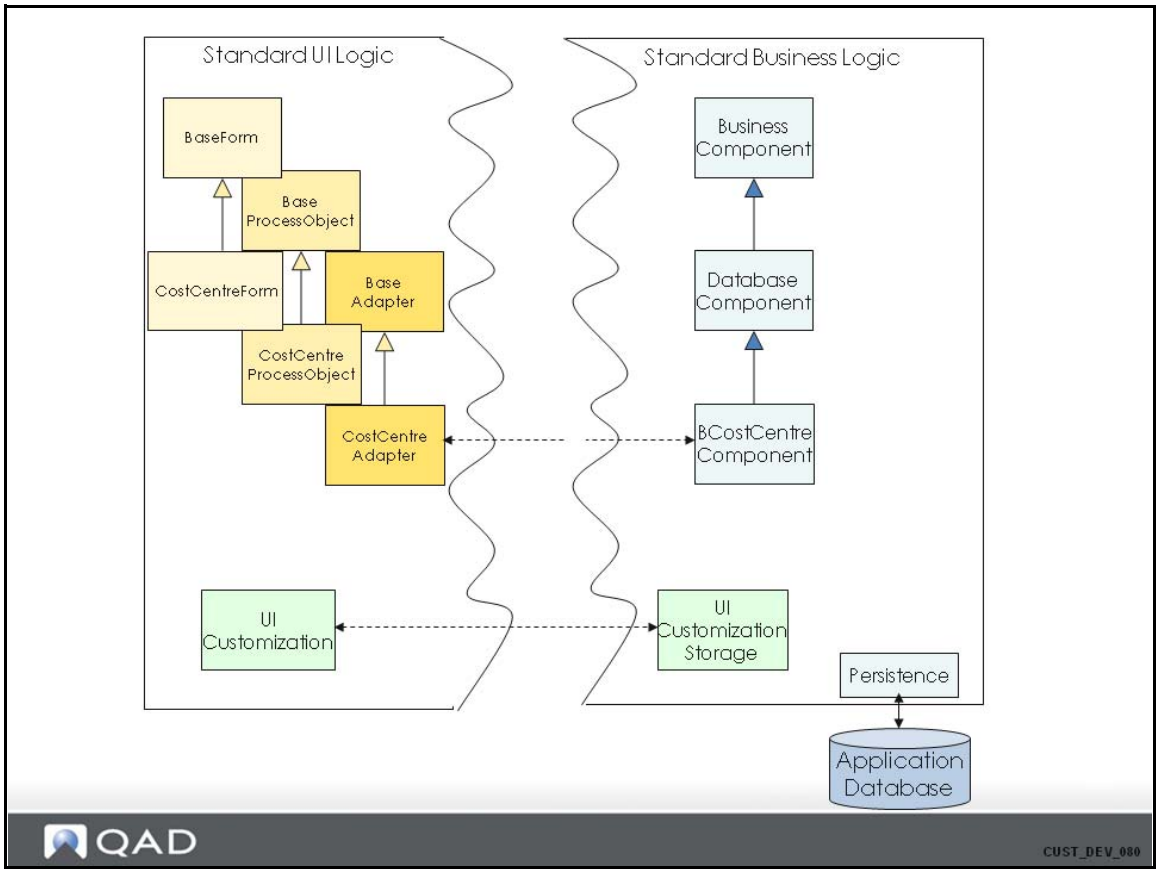
BL Customizations 3



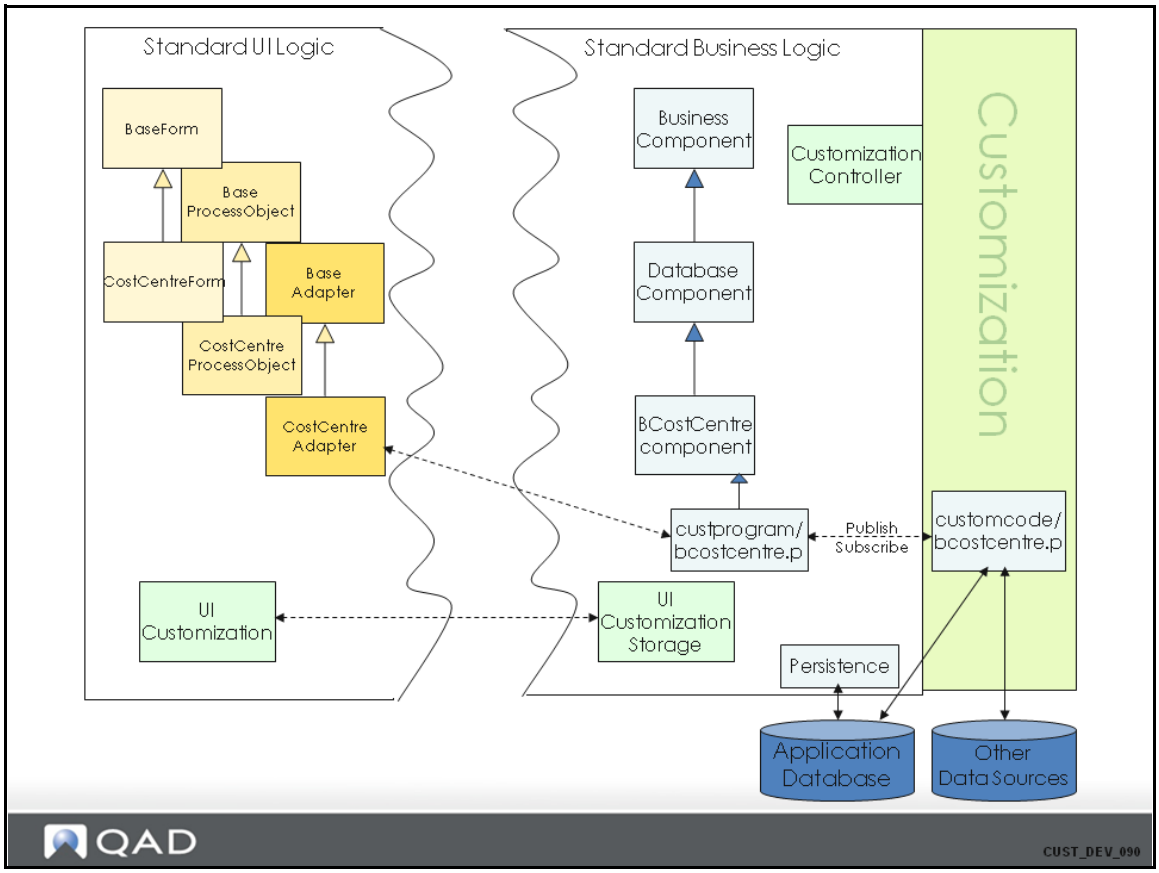
BL Customizations 4



Customization Diagram

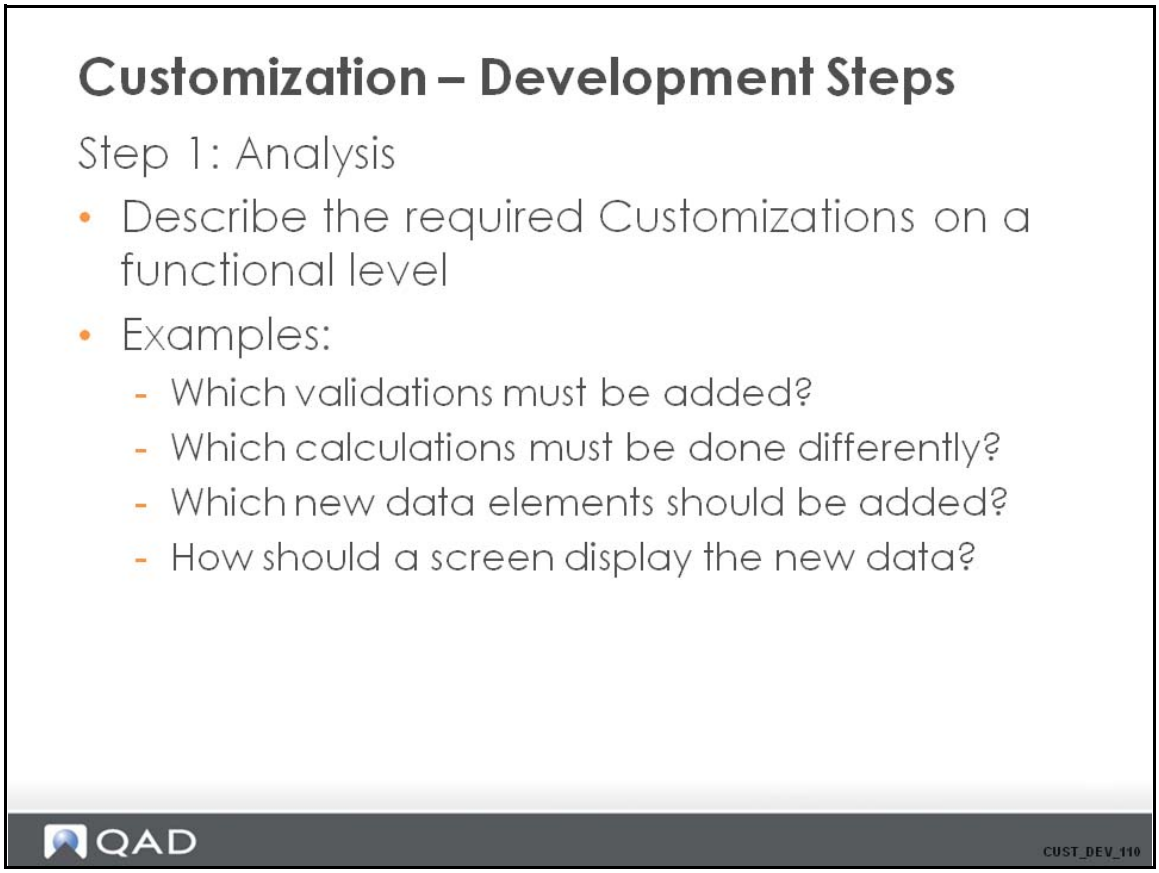


Customization Diagram 2



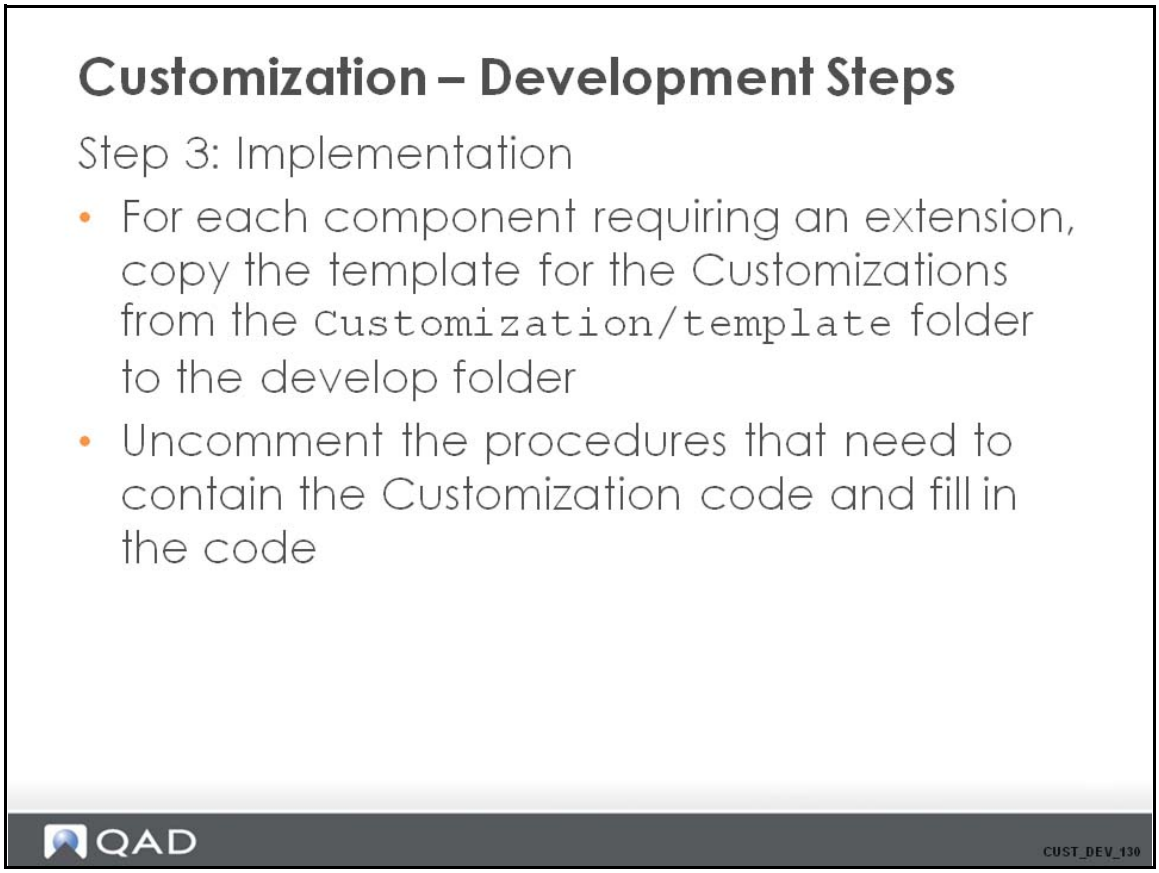
Customization - Development Steps

Customization - Development Steps 2

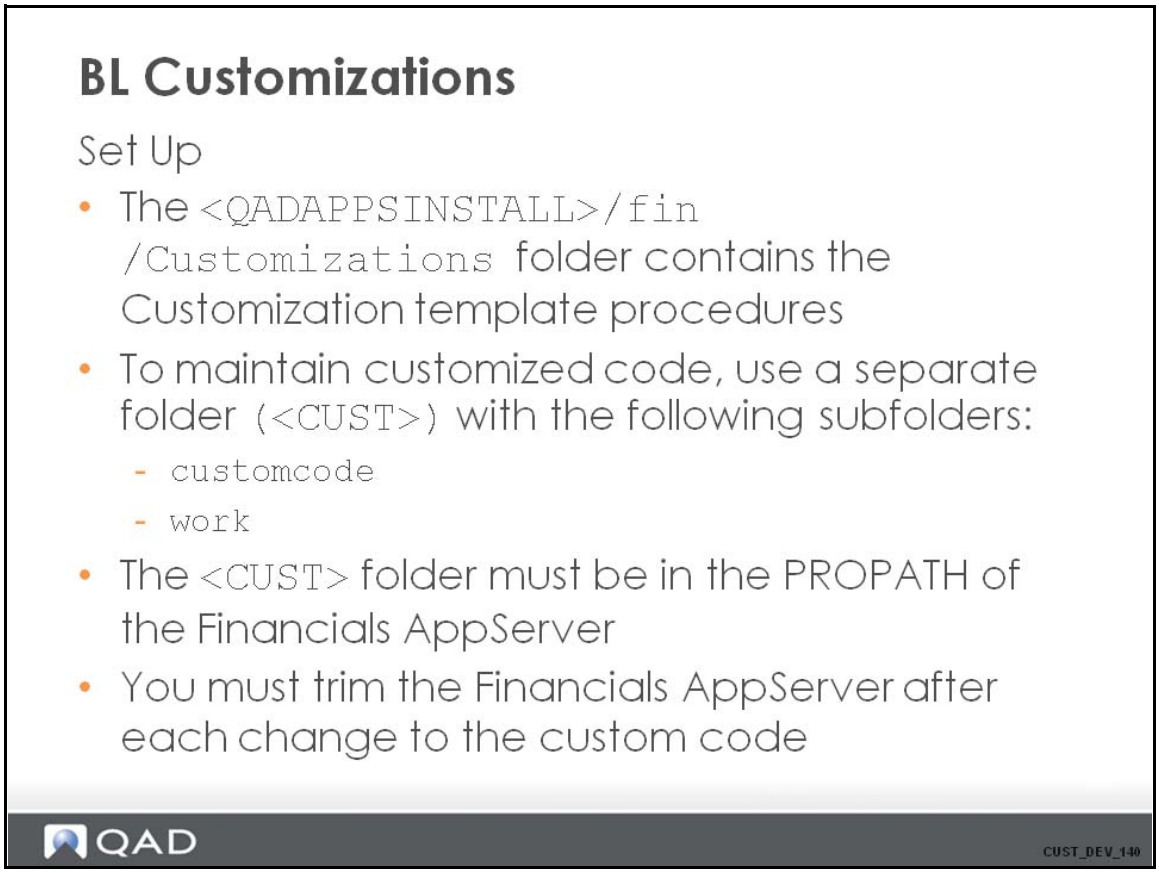


Customization - Development Steps 3

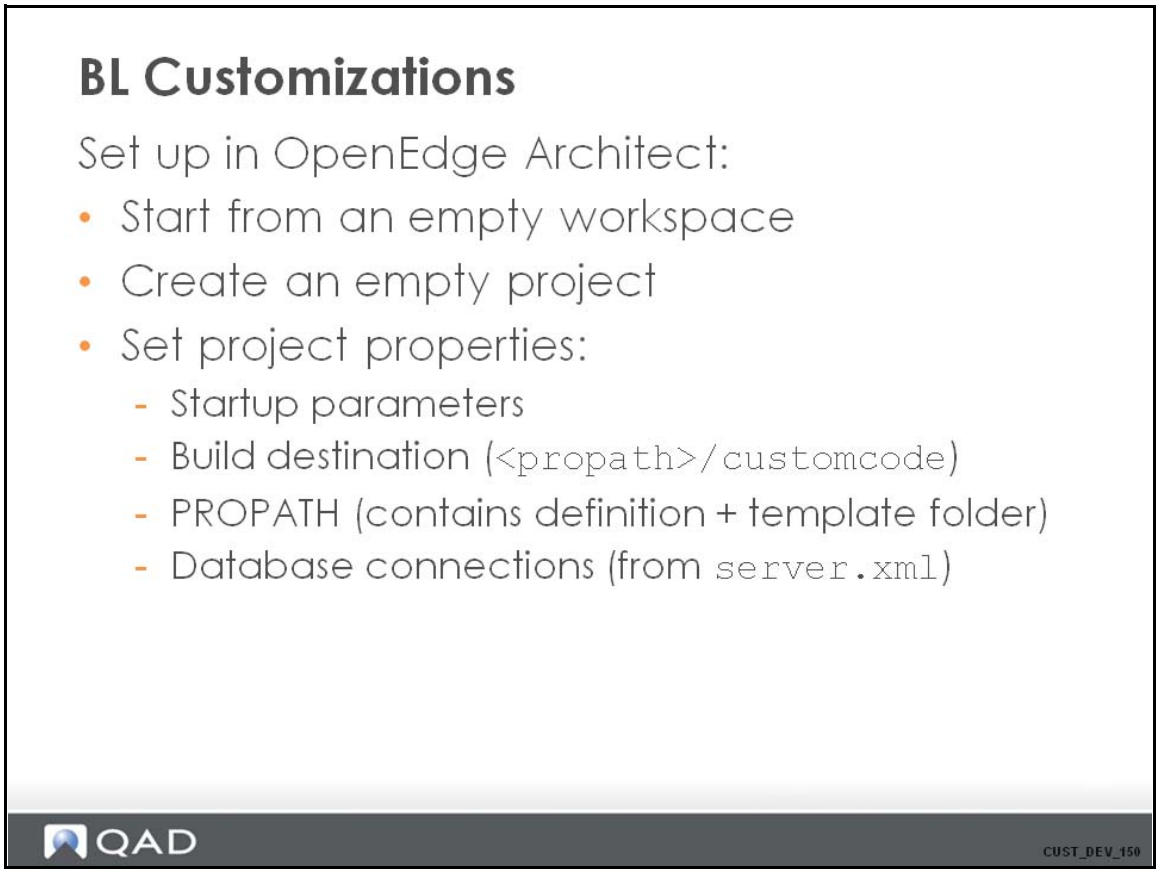
Customization - Development Steps 4



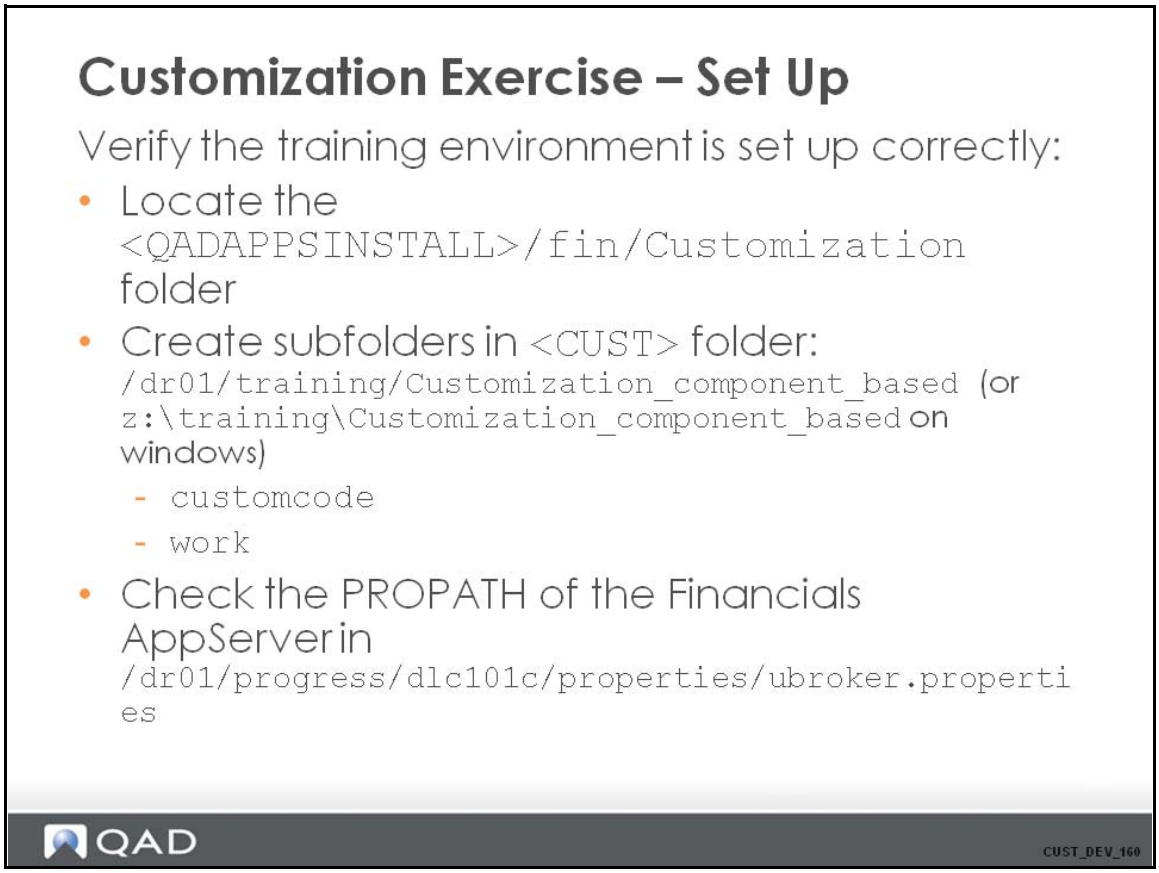
BL Customizations



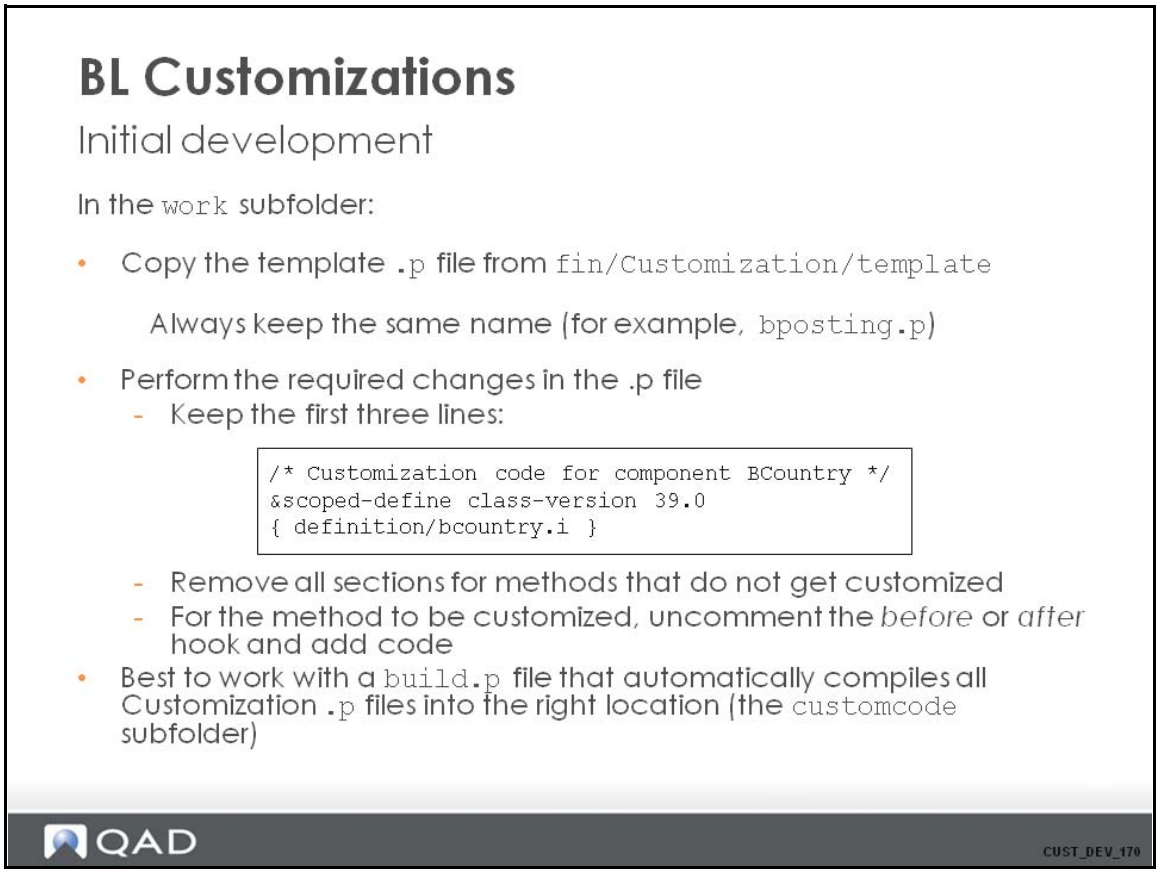
BL Customizations 2



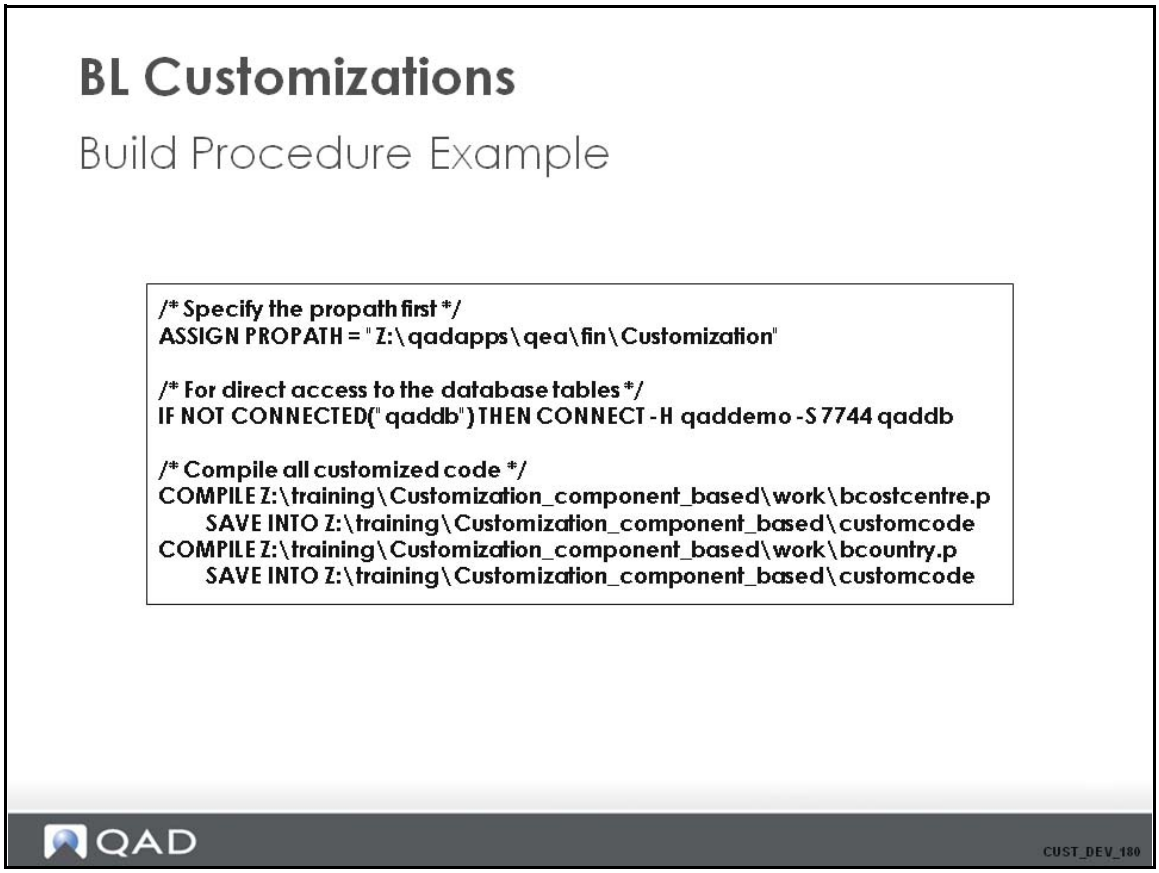
Customization Exercise - Set Up



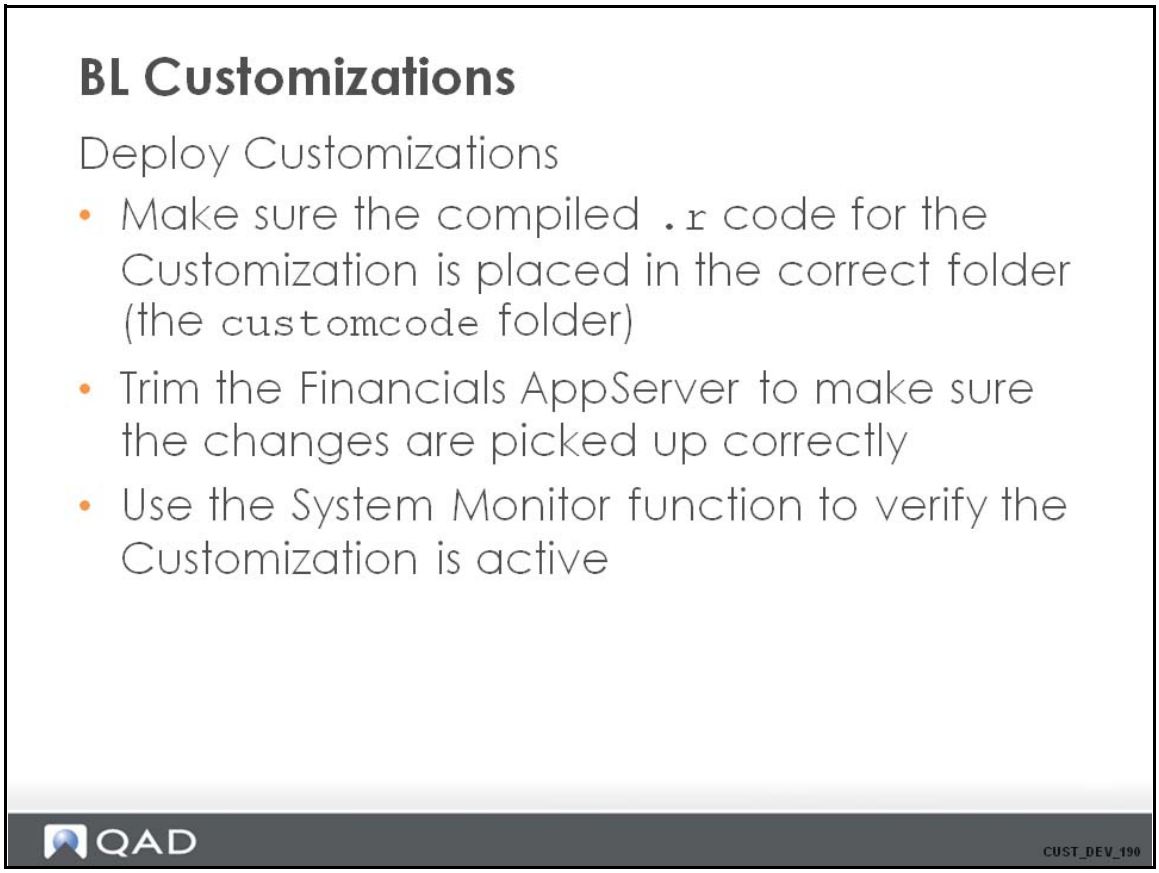
BL Customizations



BL Customizations 2



BL Customizations 3

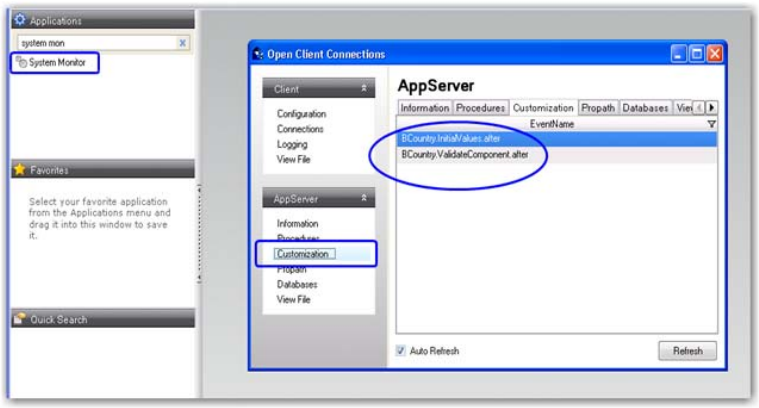


BL Customizations 4

BL Customizations

Use System Monitor to check:

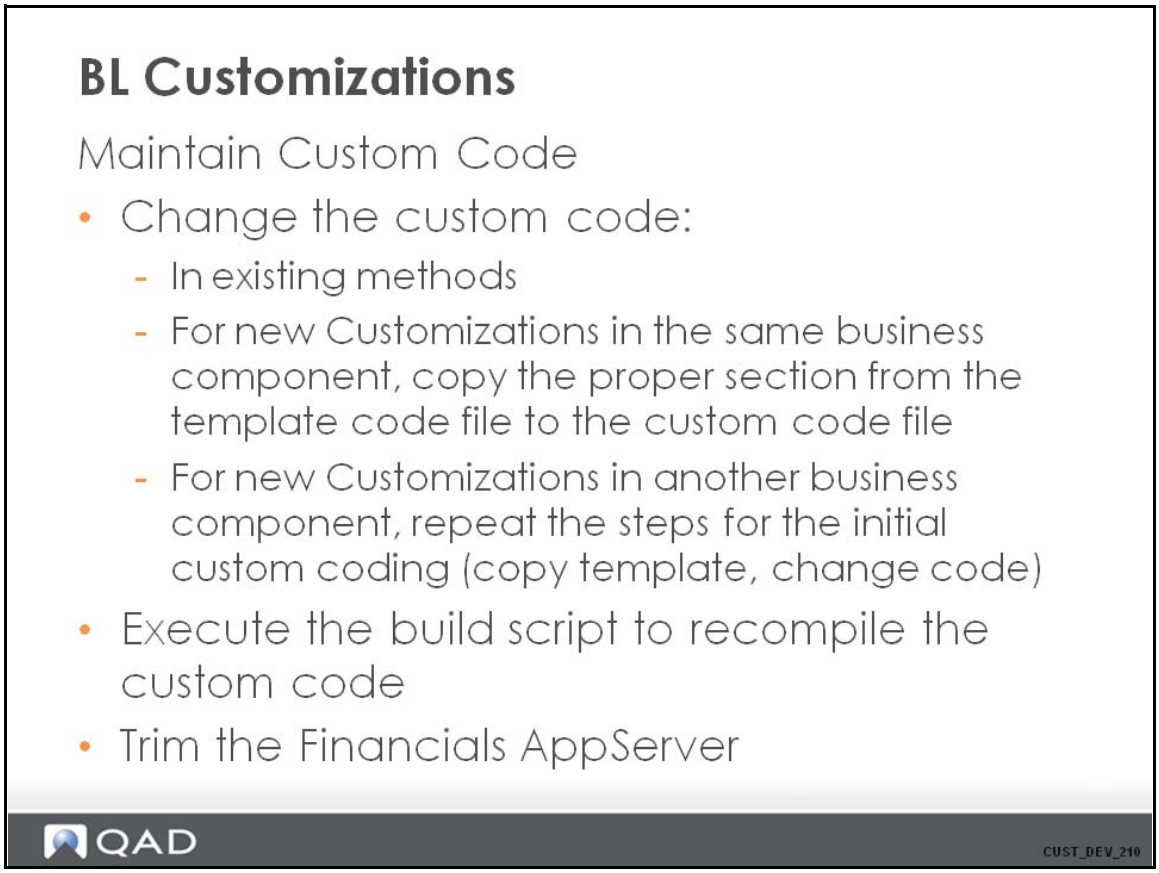
- If Customization is active
- Which hooks are implemented by the backend Customization



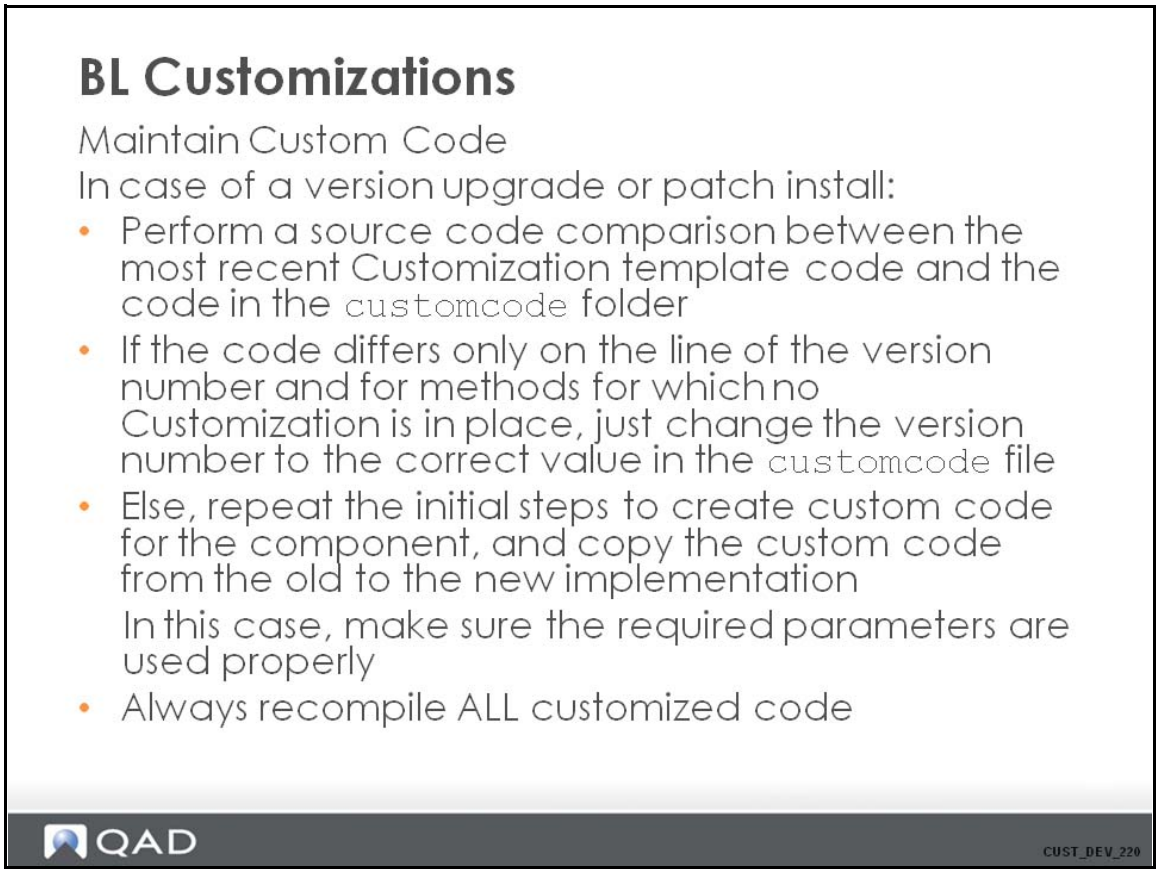
QAD

CUST_DEV_200

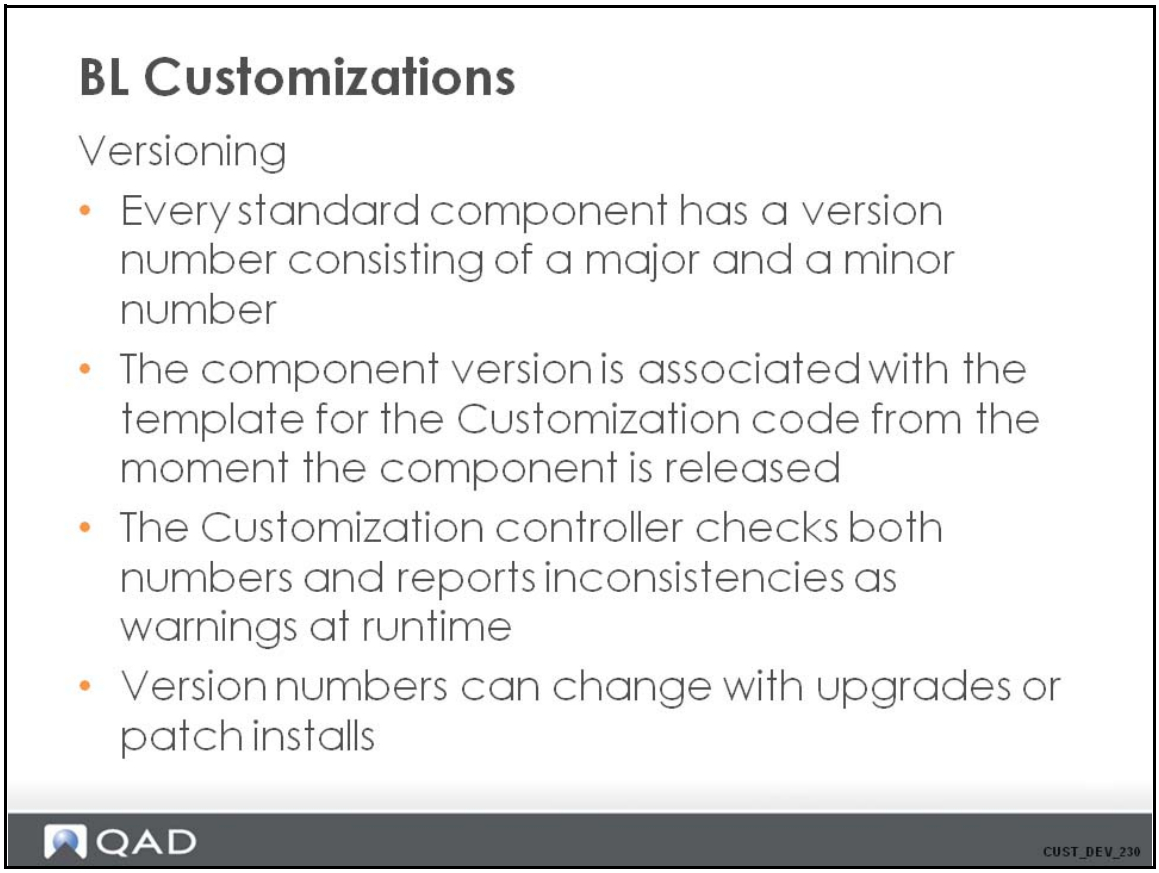
BL Customizations 5



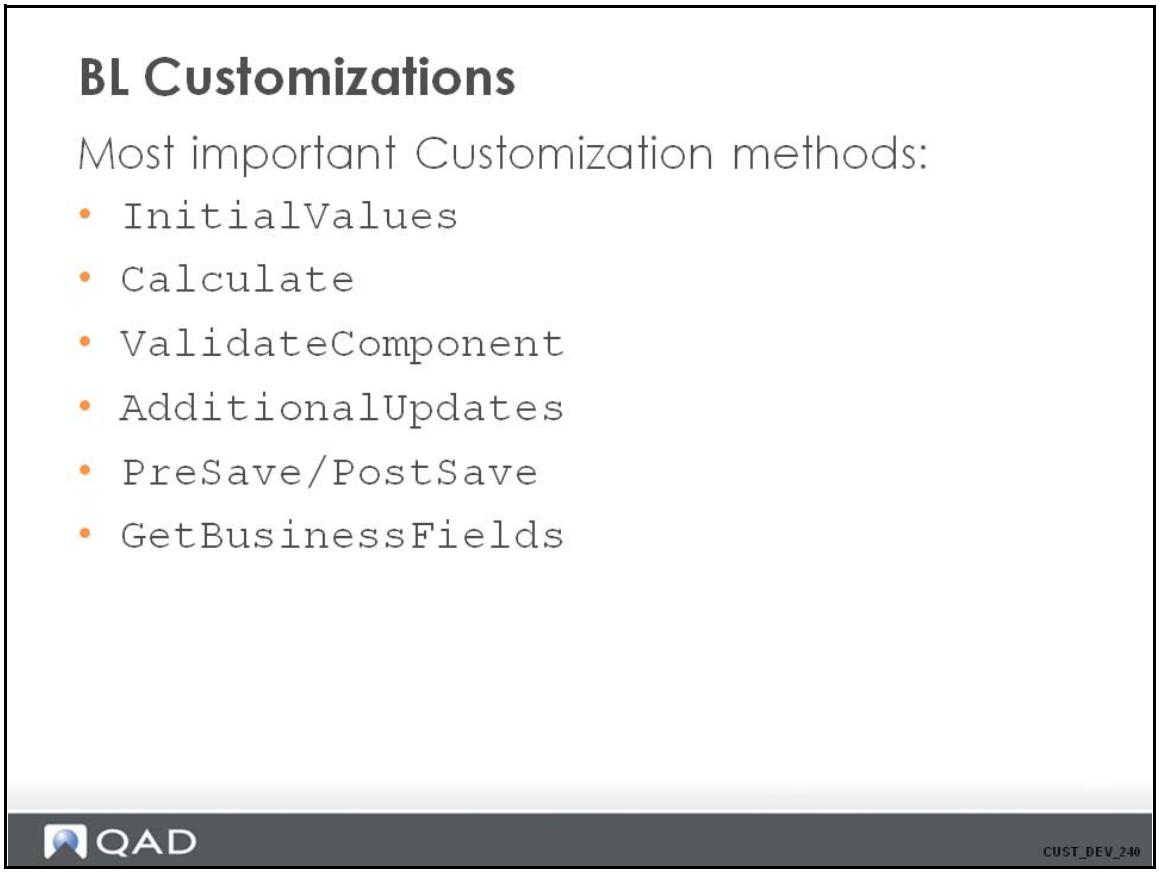
BL Customizations 6



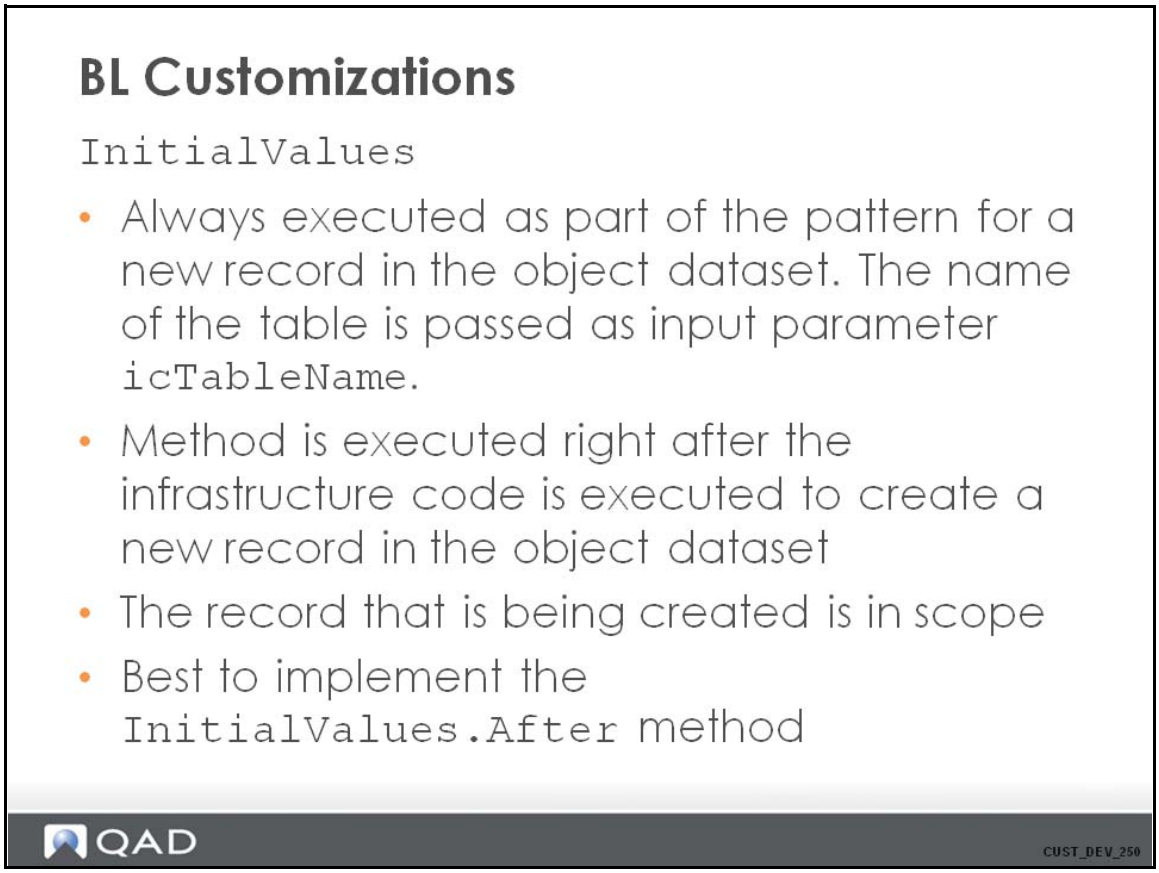
BL Customizations 7



BL Customizations 8



BL Customizations - InitialValues



BL Customizations - InitialValues 2

BL Customizations

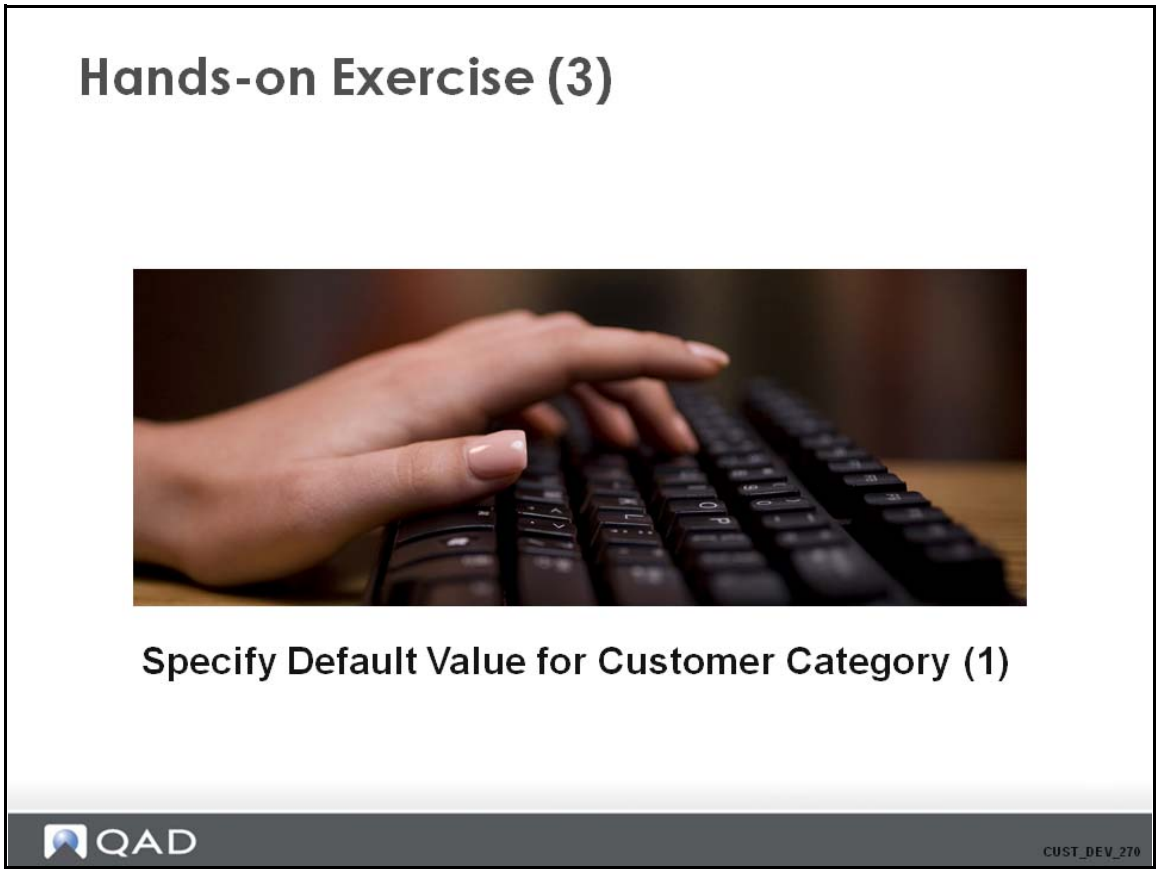
InitialValues - Example

```
/*
Procedure: InitialValues
Description:
Add code here to initialize the calculated fields of a 'new' record (= a record that must be created in the
application database) in a class temp-table.
Parameters:
input icTableName
    (Name of the database table of which a record is created in the class temp-table.)
output oiReturnStatus
    ()
*/

PROCEDURE BCountry.InitialValues.after:
    if t_parameter.icTableName = "Country"
    then assign tCountry.CustomInteger0 = 1000000.
END PROCEDURE.
```

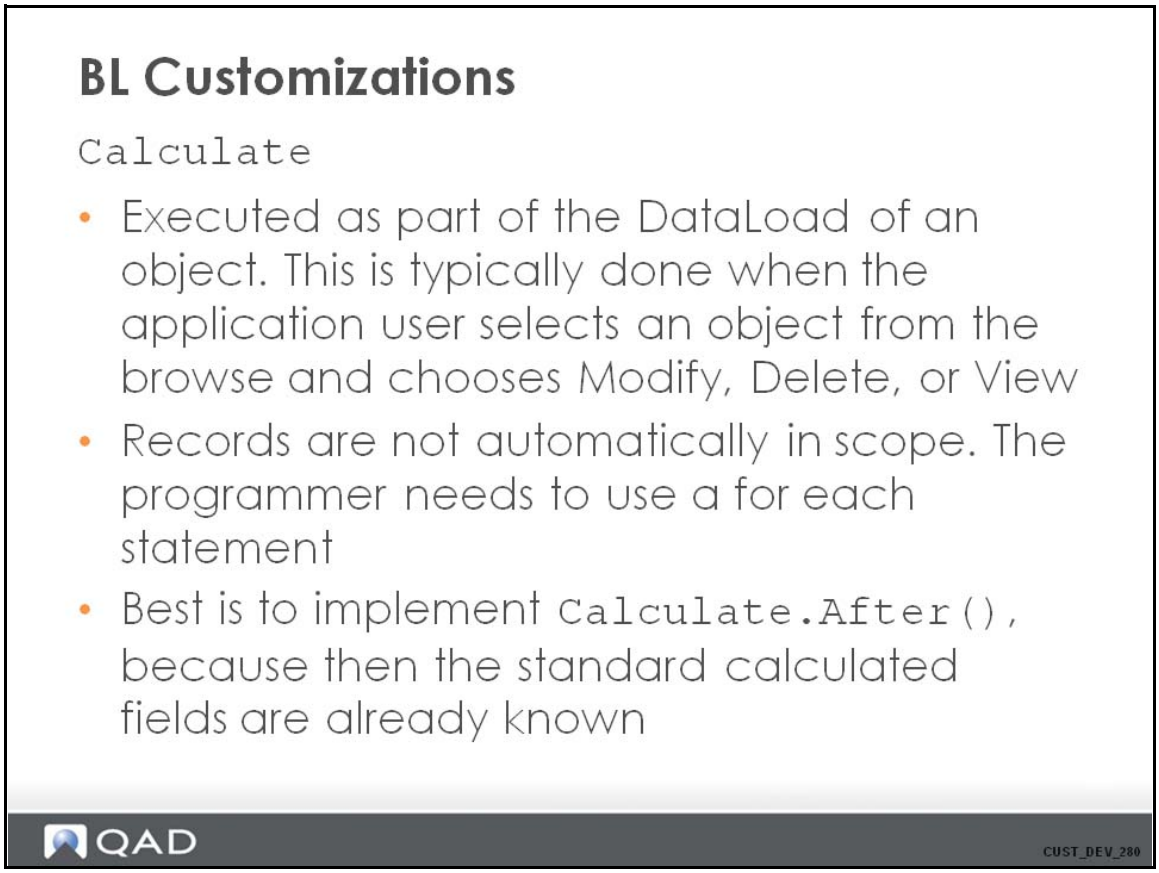
QAD

CUST_DEV_260

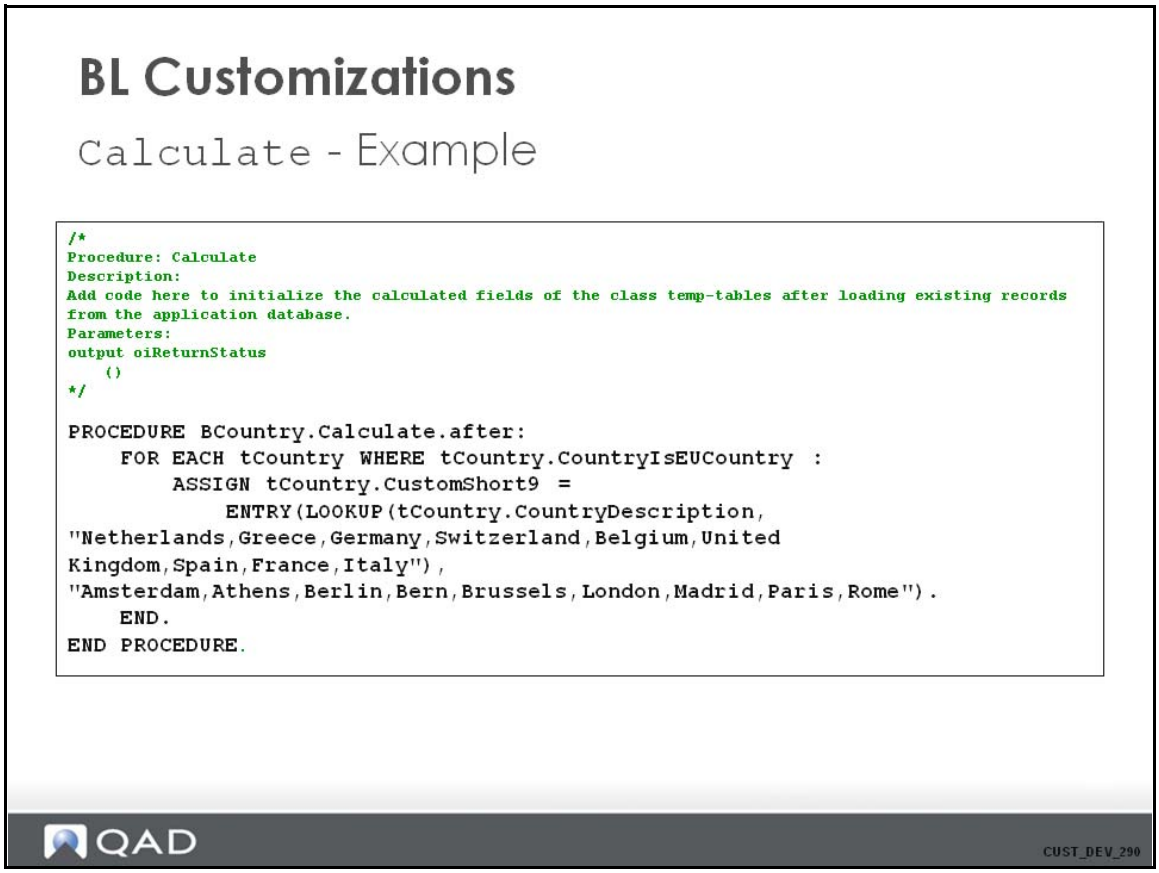
Hands-on Exercise (3)



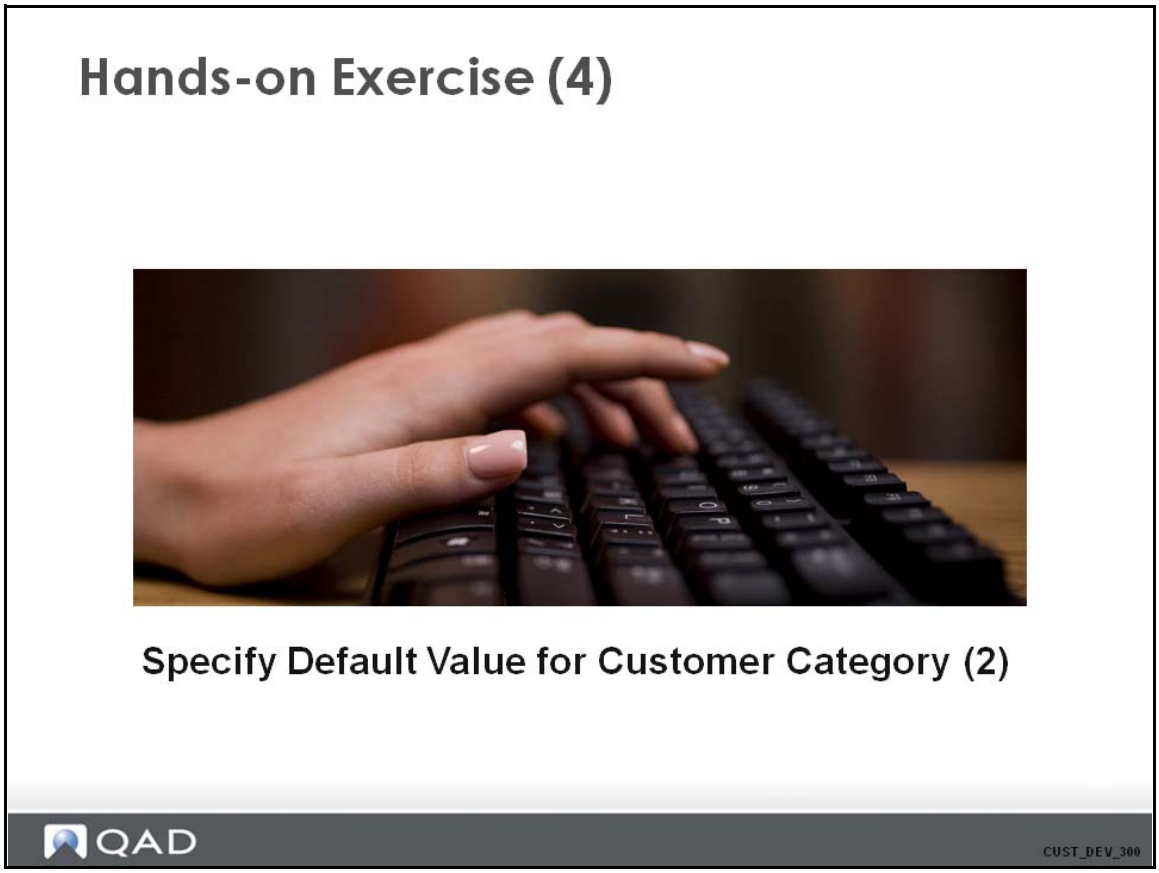
BL Customizations - Calculate



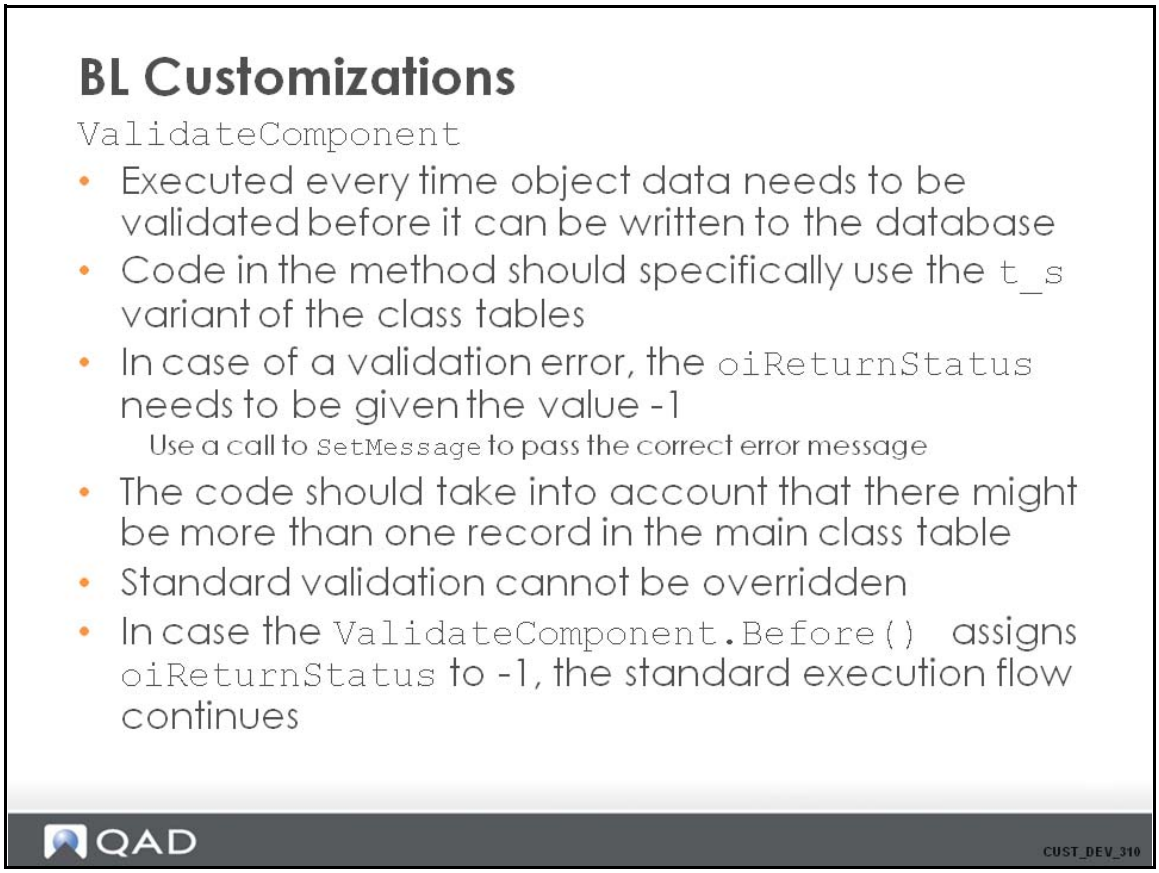
BL Customizations - Calculate 2



Hands-on Exercise (4)



BL Customizations - ValidateComponent



BL Customizations - ValidateComponent 2

BL Customizations

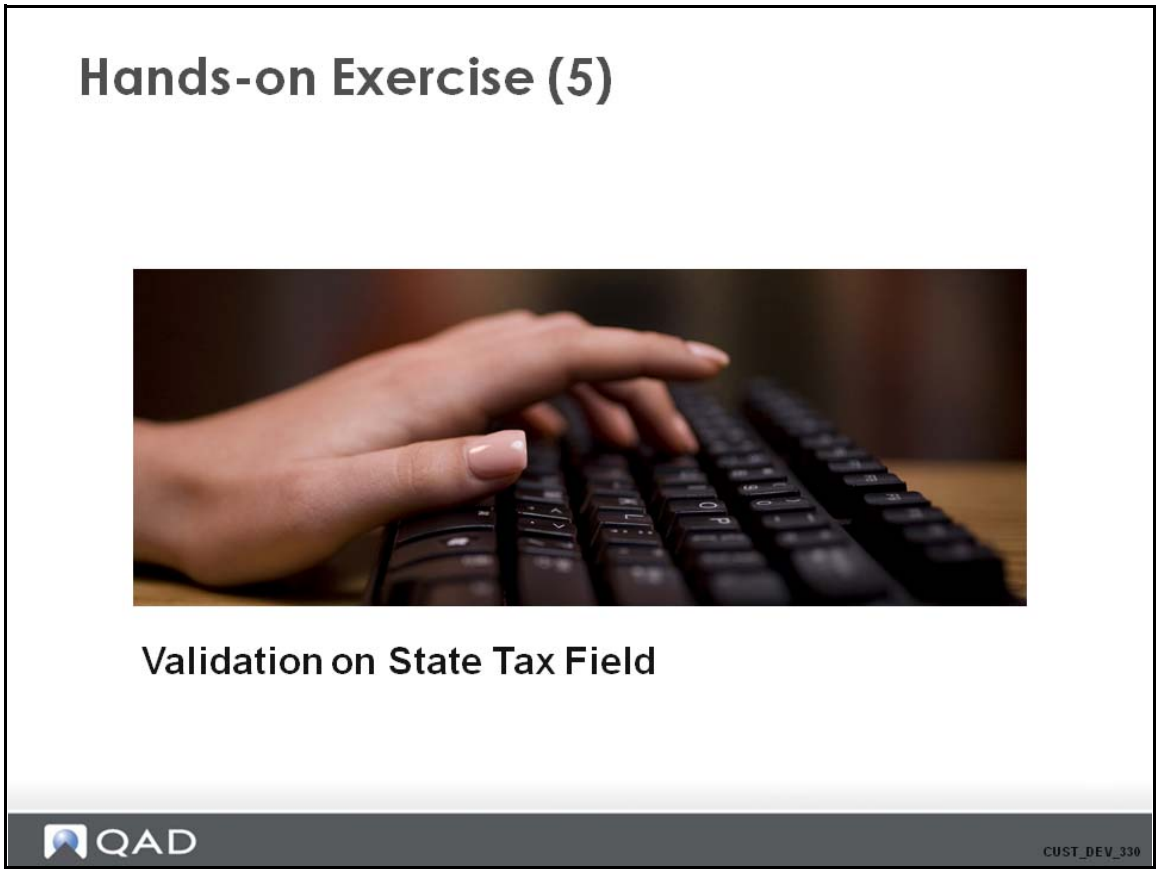
ValidateComponent - Example

```
/*
Procedure: ValidateComponent
Description: …
*/
PROCEDURE BCountry.ValidateComponent.After:
    DEFINE VARIABLE viReturn AS INTEGER NO-UNDO.
    FOR EACH t_sCountry:
        IF t_sCountry.CustomInteger0 = 0 THEN DO:
            ASSIGN t_parameter.oiReturnStatus = -102.
            run SetMessage ('The number of habitants ($1) must be greater than 0.',
                STRING(t_sCountry.CustomInteger0),'t_sCountry.CustomInteger0',
                string(t_sCountry.CustomInteger0),
                'E', 3, t_sCountry.tc_Rowid, 'CUSTOM-001', '', '', '', output viReturn).
            RETURN.
        END.
    END.
END PROCEDURE.
```

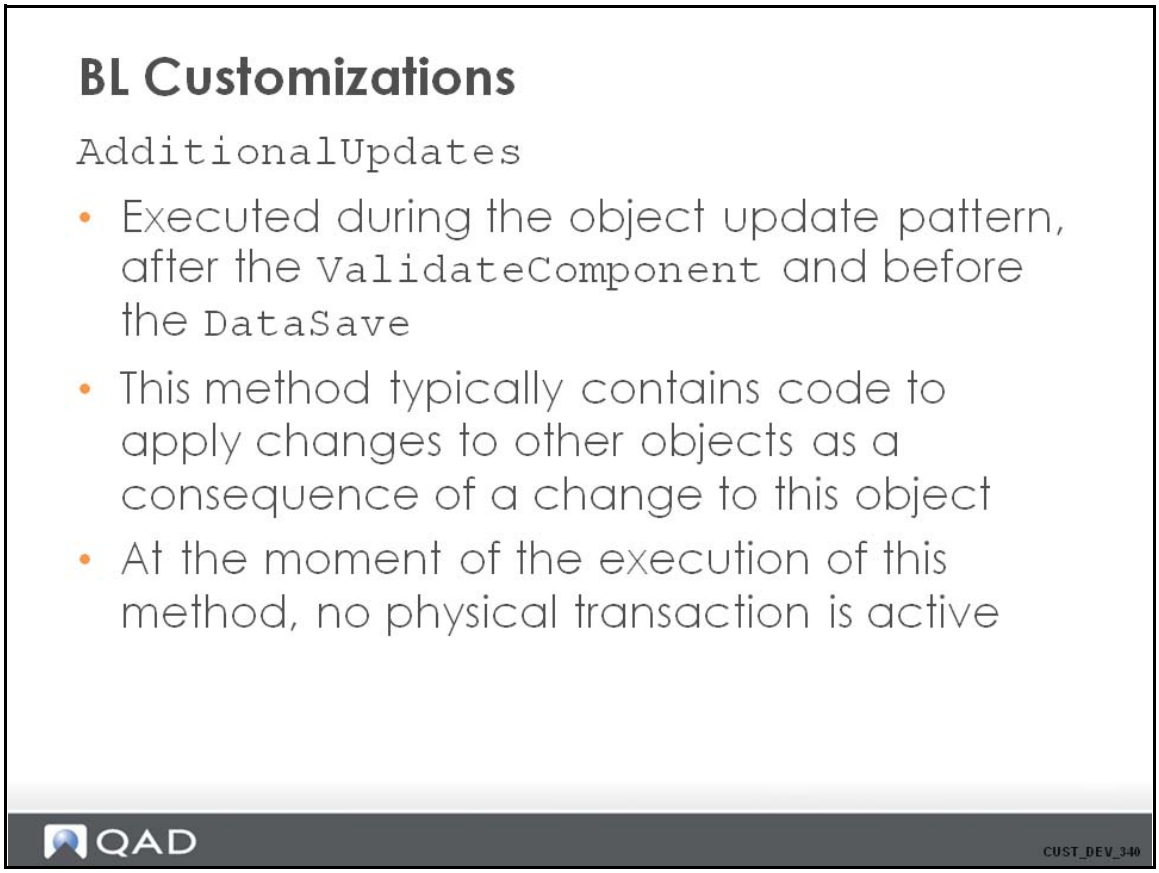
QAD

CUST_DEV_320

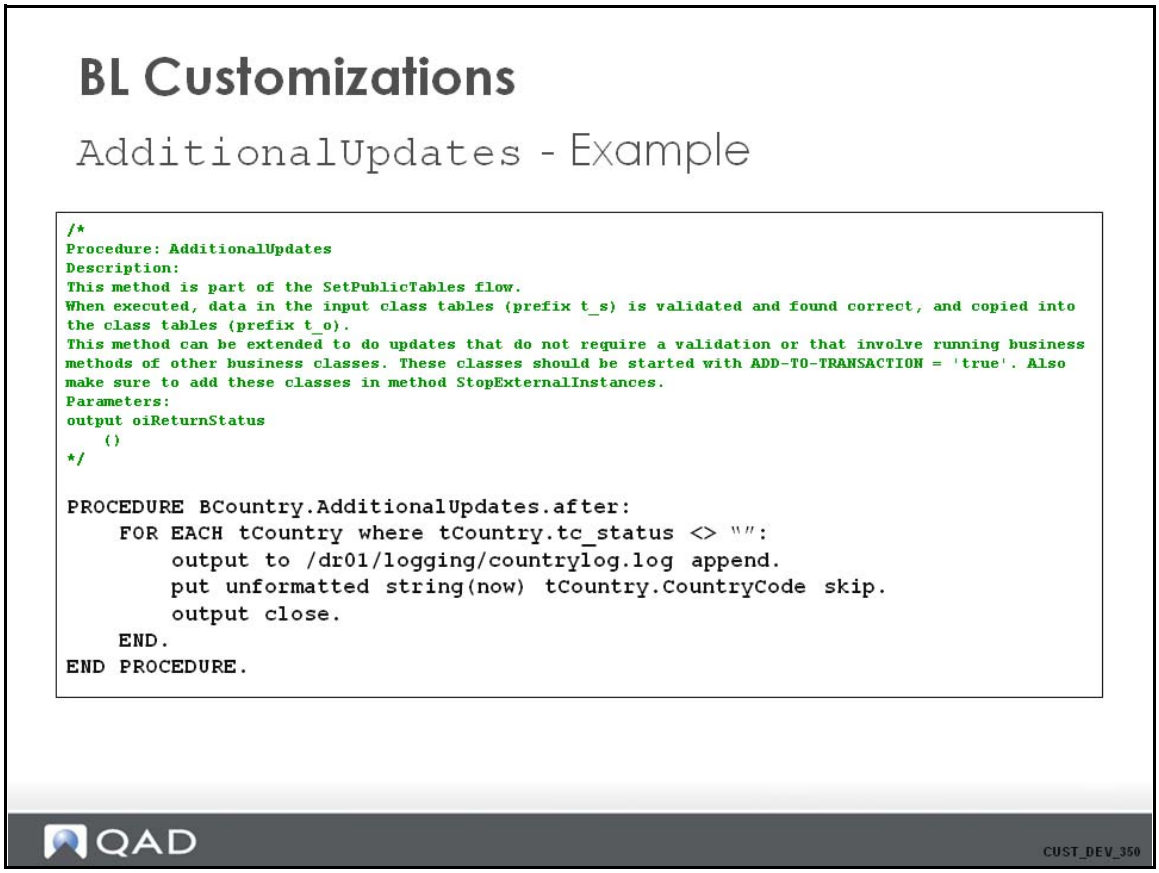
Hands-on Exercise (5)



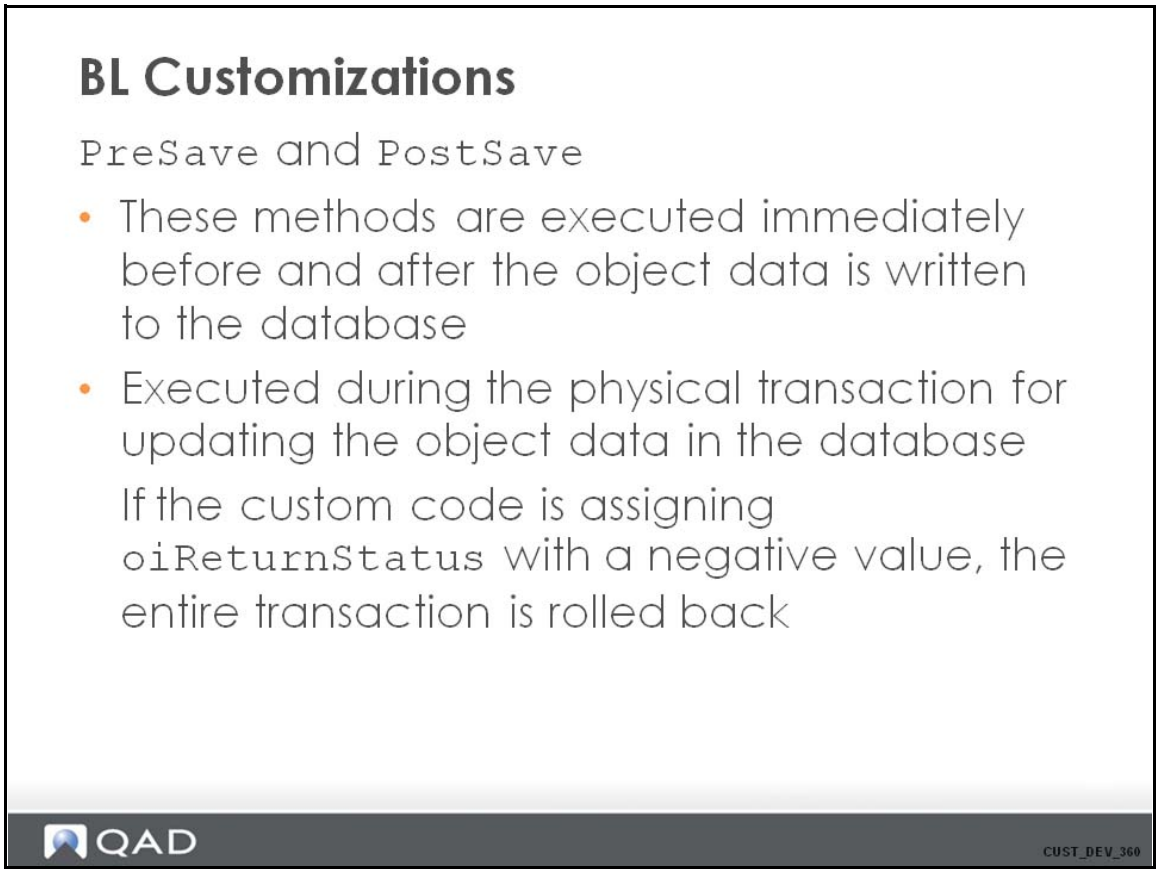
BL Customizations - AdditionalUpdates



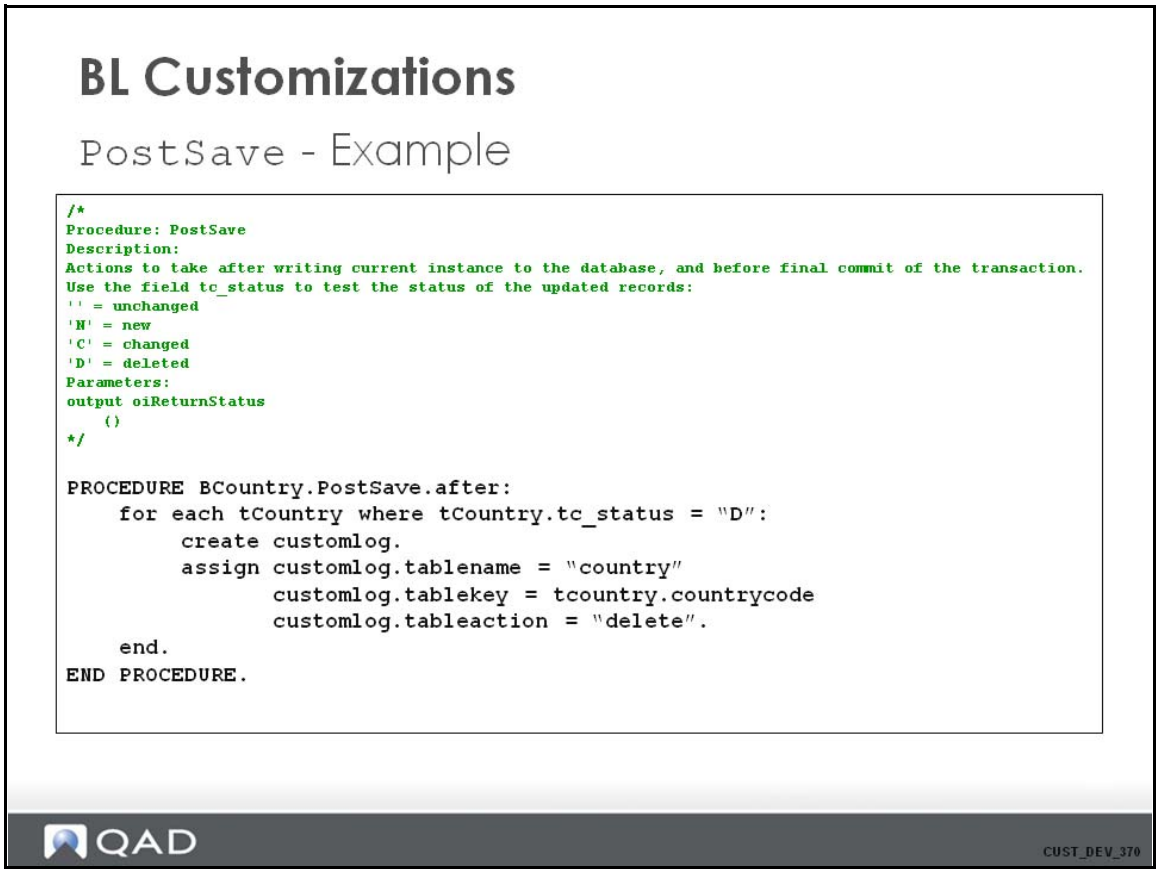
BL Customizations - Additional Updates 2



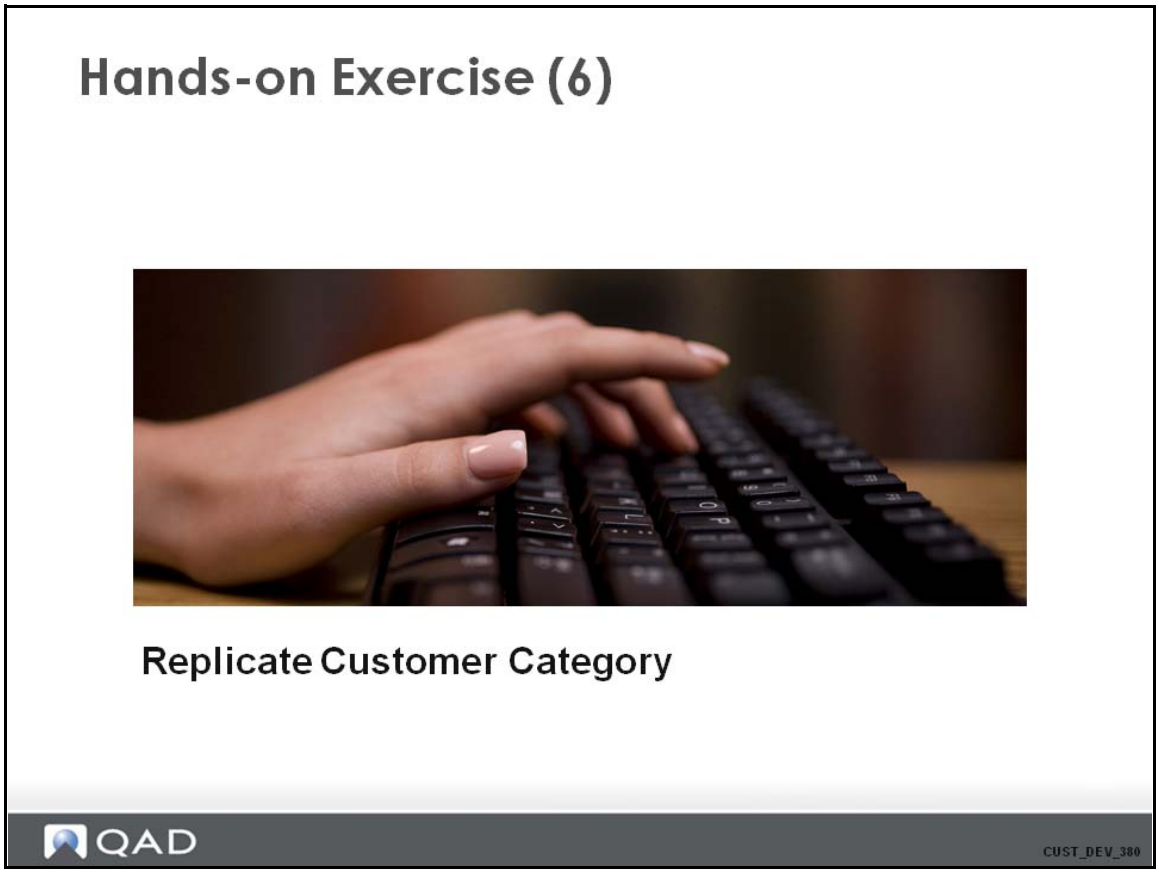
BL Customizations - PreSave and PostSave



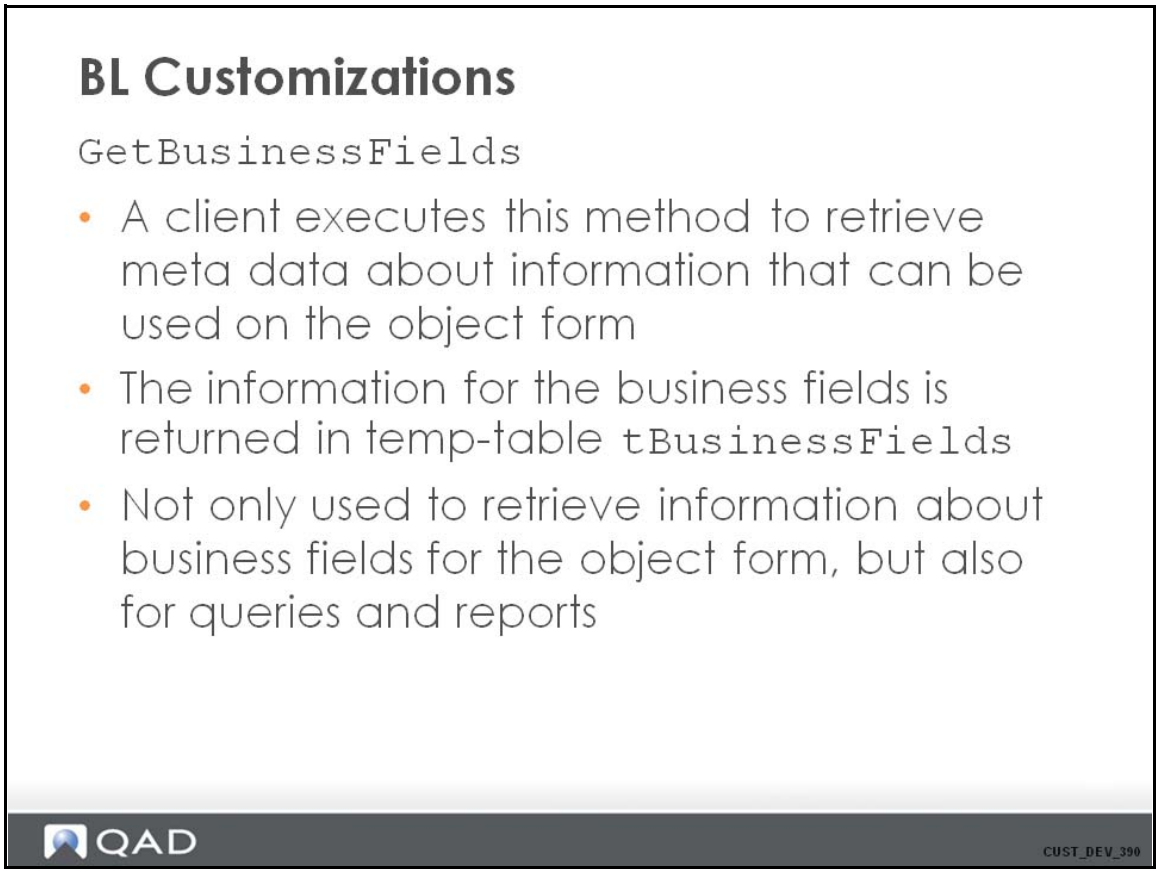
BL Customizations - PostSave



Hands-on Exercise (6)



BL Customizations - GetBusinessFields



BL Customizations - Business Fields Mechanism

BL Customizations

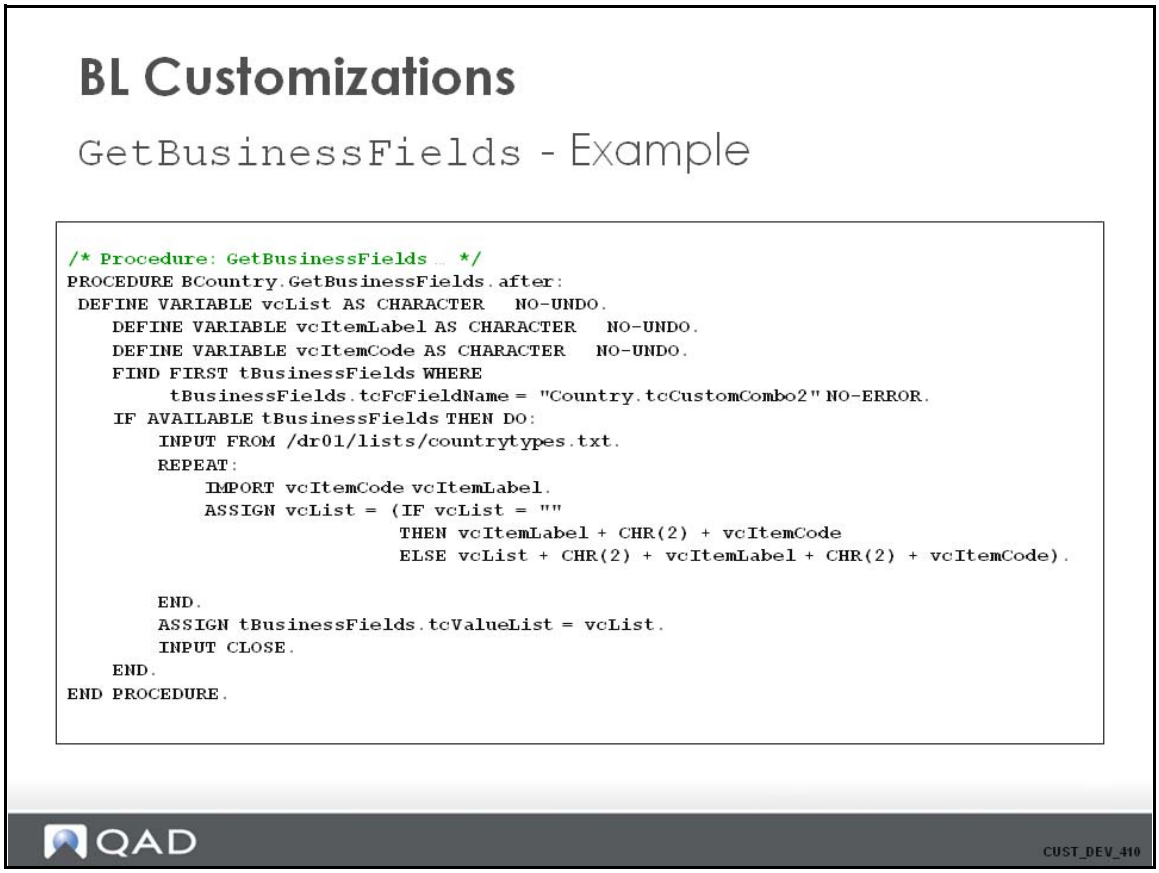
Business Fields Mechanism

The dataset (with one temp table) `tBusinessFields`, which is returned by the `GetBusinessFields()` method, contains the following fields:

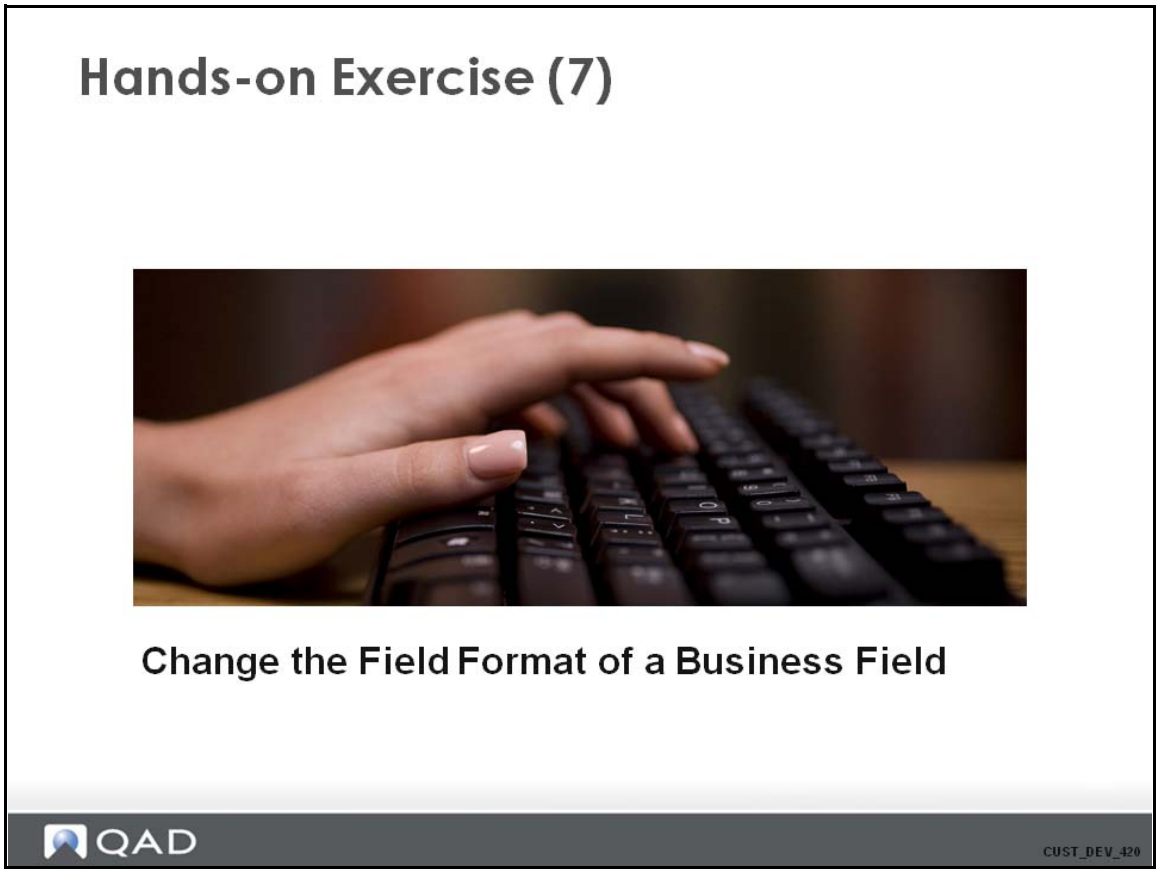
tcColumnLabel	character	Column label
tcControlType	character	ControlType
tcDataType	character	Can be any of the following values c,d,i,l (resp. character, decimal, date, integer or logical)
tcDisabledForActivities	character	business field is disabled for any of these activity codes (comma separated list of activity codes)
tcDisplayFormat	character	DisplayFormat (progress display format like "x(40)")
tcFcDescription	character	Description of the field (currently not used by the UI)
tcFcFieldName	character	The name of the business field.
tcFcFieldType	character	can be "B", "F", "C" (resp. business, filter or custom)
tcFilterOperators	character	This is the list of possible filter operators. This is used in case of "F" typed business fields.
tcHiddenForActivities	character	business field is hidden for any of these activity codes (comma separated list of activity codes)
tcLookupFilterField	character	This is the filter field in the query that is referenced by tcLookupQuery. It receives the value of the current business field upon calling the lookup with the tcLookupQuery reference.
tcLookupQuery	character	The query that should be used when the lookup is chosen for the business field.
tcLookupReturnField	character	This is the business field from the result set from the tcLookupQuery for which the value is returned and filled in into the current business field.
tcRelatedObject	character	Business component to which the business field refers. This activates the "goto" link.
tcSideLabel	character	side label
tiStoredSearch_ID	integer	Linked stored search (this is only available for user-defined business fields)
tcValidationMask	character	The name of the validation mask for the field (this is not used by the UI at the moment)
tcValueList	character	The list of discrete values the business field can have. This is used to populate the combo-box.
tcInitialValue	character	The initial value of the business field. Only used in the "create" activity.
tiSequence	integer	A sequence of the business field. Only used as unique key in the tBusinessFields temp-table.

QAD CUST_DEV_400

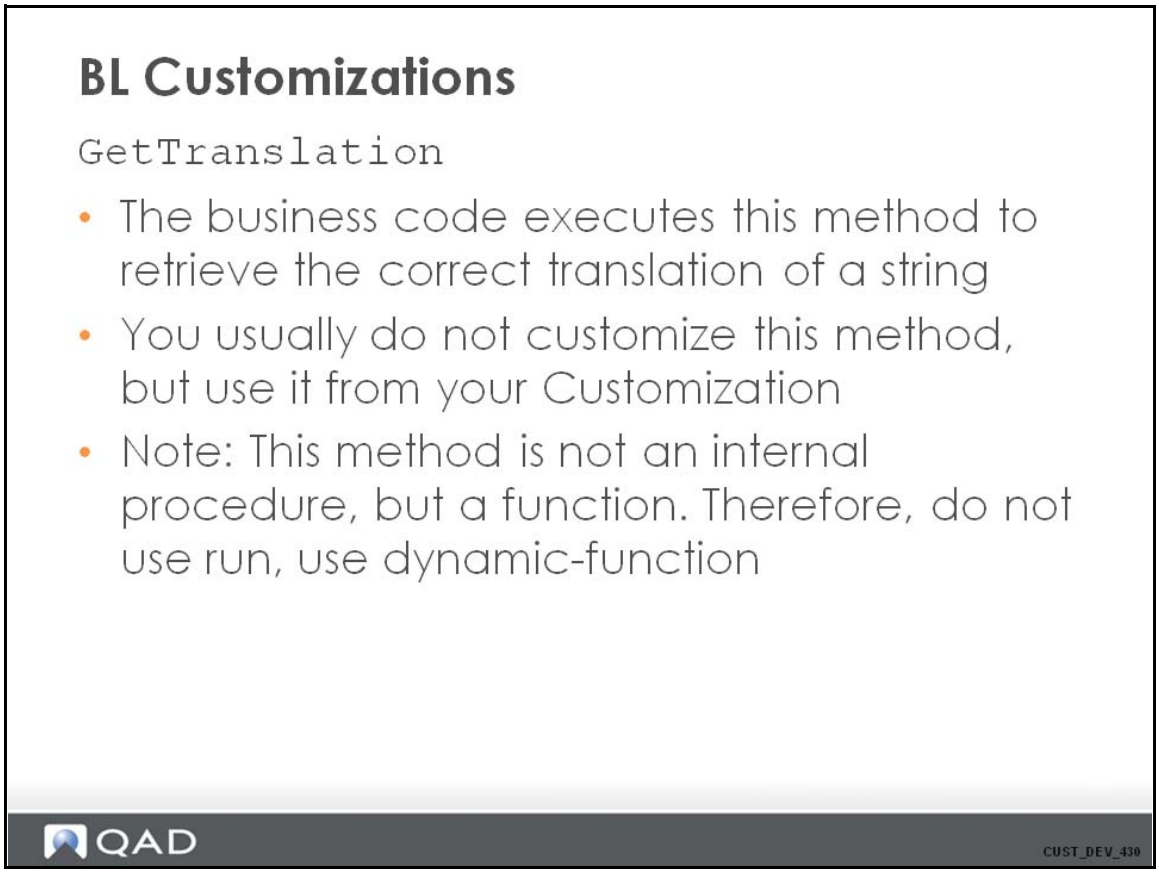
BL Customizations - GetBusinessFields



Hands-on Exercise (7)



BL Customizations - GetTranslation



BL Customizations - GetTranslation 2

BL Customizations

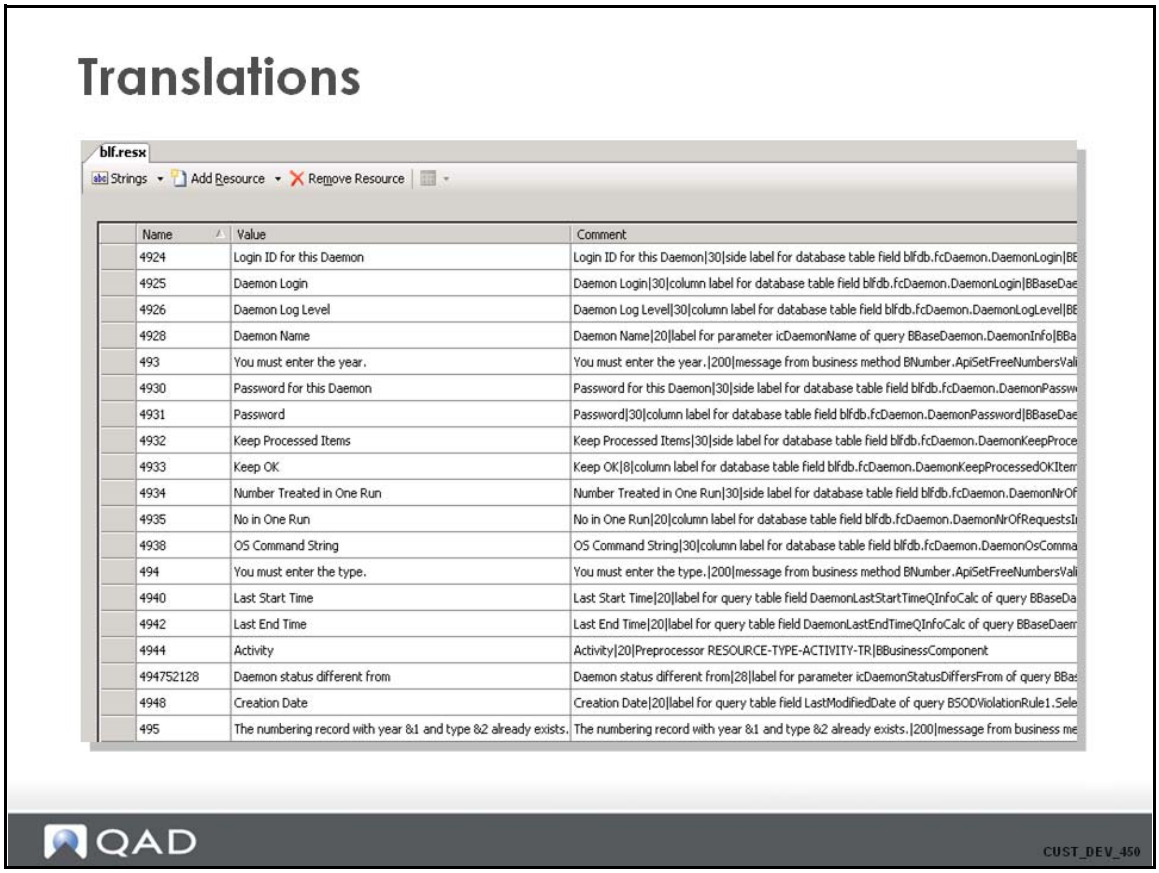
GetTranslation - Example

```
PROCEDURE BCountry.ValidateComponent.before:
    DEFINE VARIABLE viReturn AS INTEGER NO-UNDO.
    DEFINE VARIABLE vcMessage AS CHARACTER NO-UNDO.
    FOR EACH t_sCountry:
        IF t_sCountry.CustomInteger0 = 0 THEN DO:
            ASSIGN t_parameter.oiReturnStatus = -102.
            ASSIGN vcMessage = DYNAMIC-FUNCTION
                 ('GetTranslation' in {&TARGETPROCEDURE}, 12345, PCode',
                  'The number of habitants ($1) must be greater than 0.').
            run SetMessage
               (vcMessage,
                STRING(t_sCountry.CustomInteger0),
                't_sCountry.CustomInteger0',
                string(t_sCountry.CustomInteger0),
                'E', 3, t_sCountry.tc_Rowid,
                'CUSTOM-001', '', '', '', output viReturn).
            RETURN.
        END.
    END.
END PROCEDURE.
```

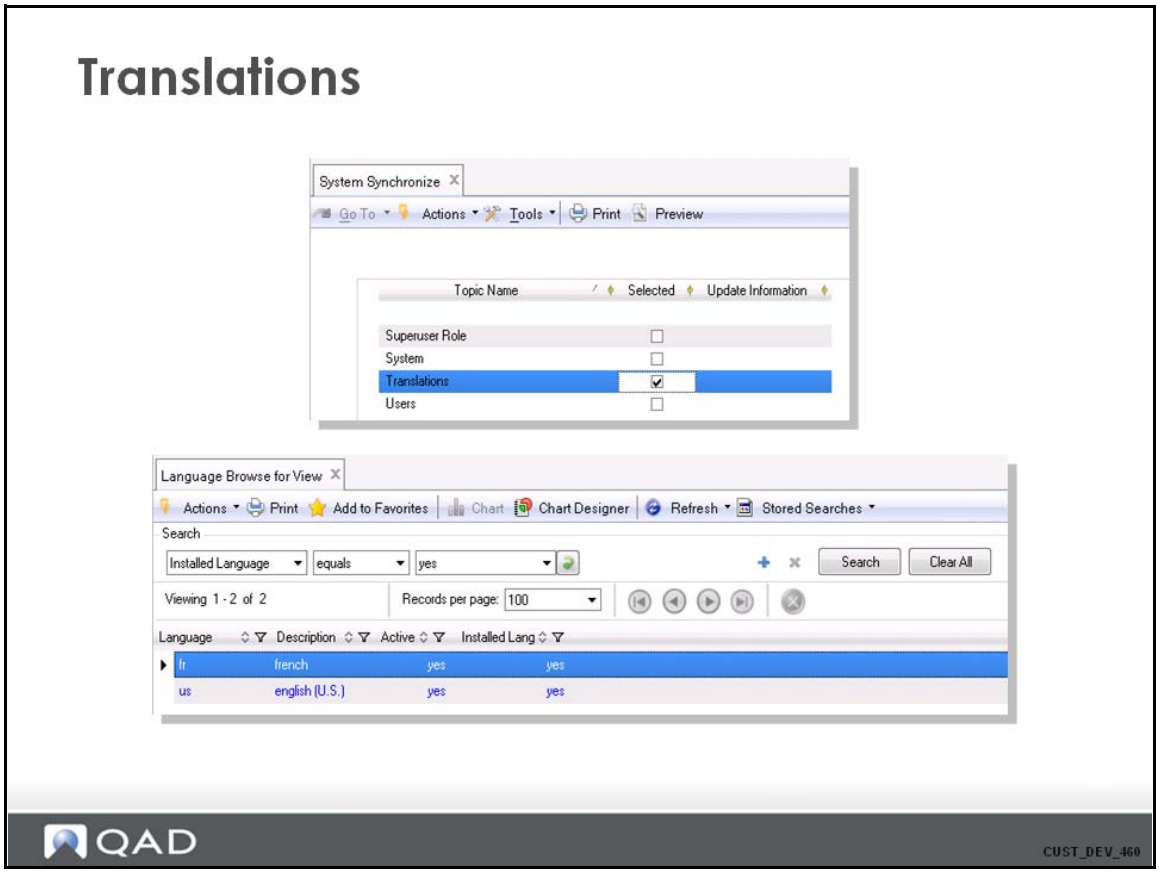
QAD

CUST_DEV_440

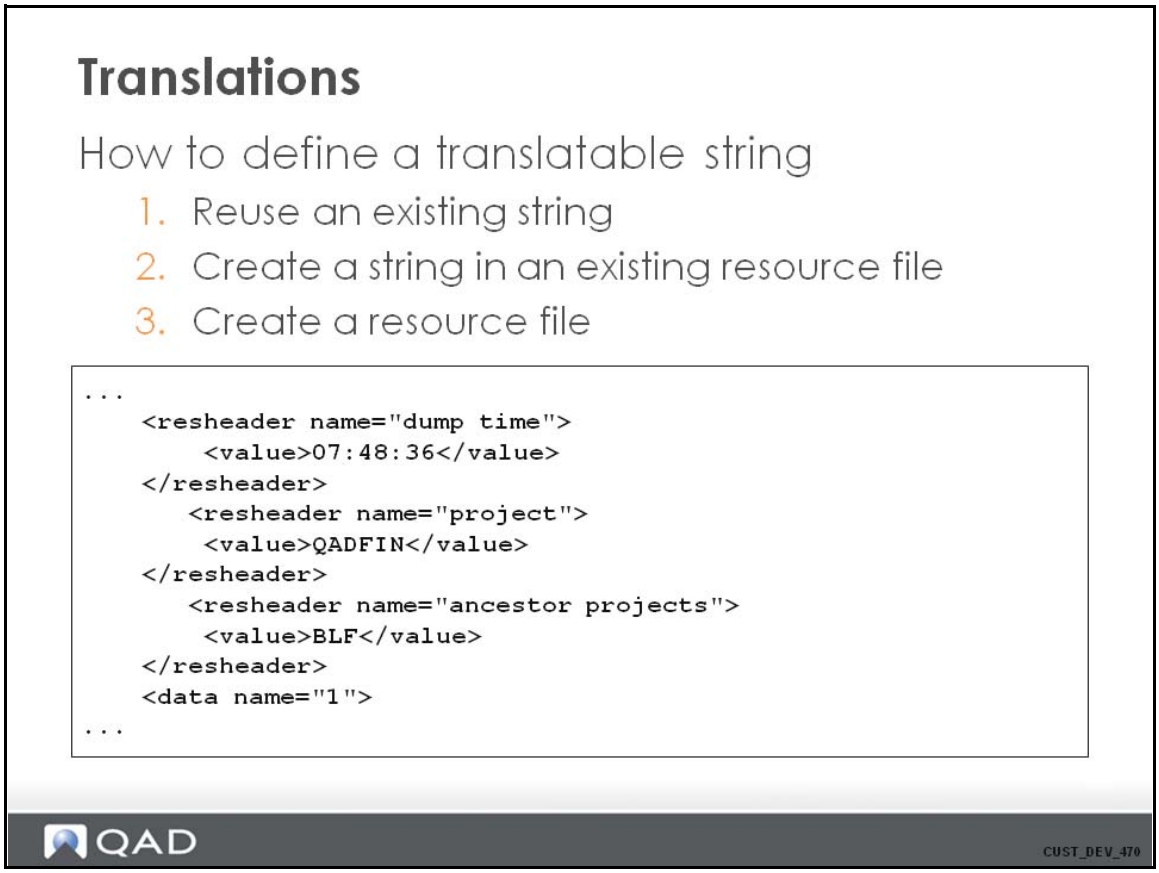
Translations



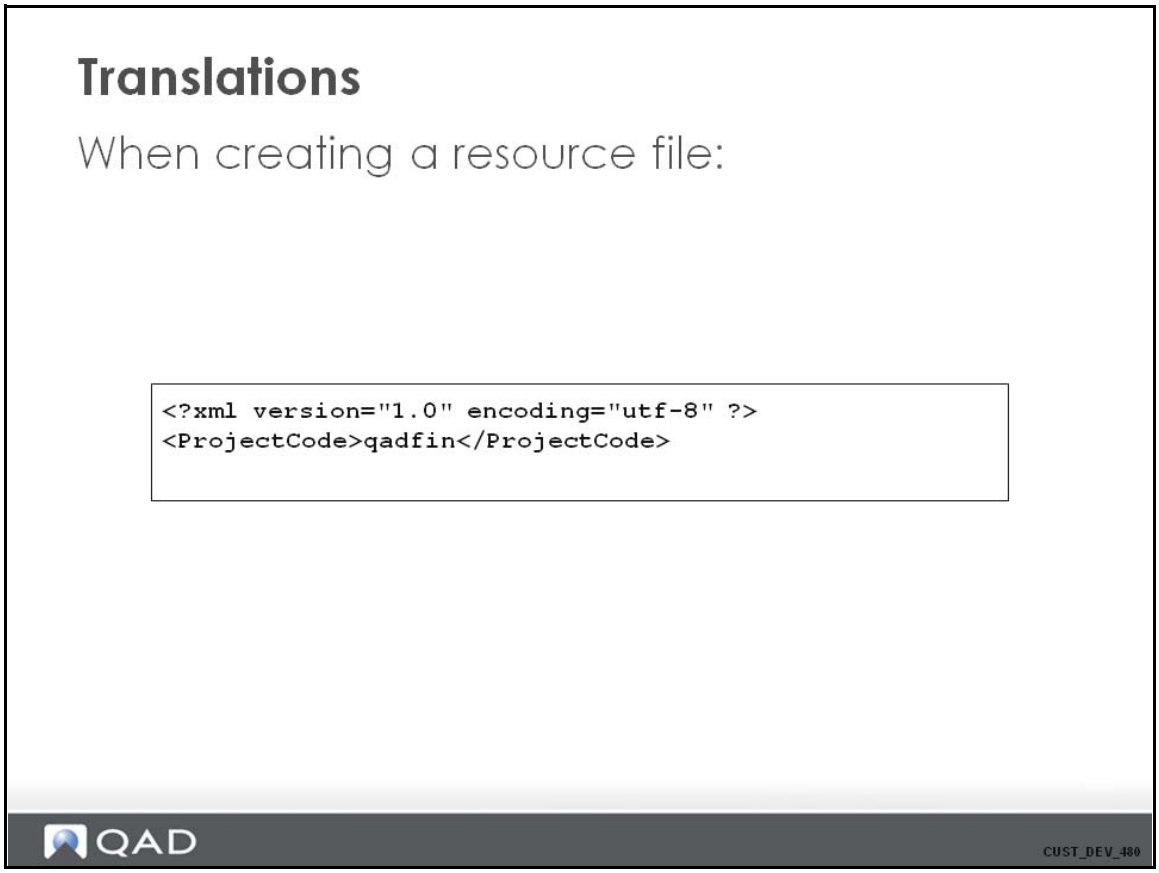
Translations 2



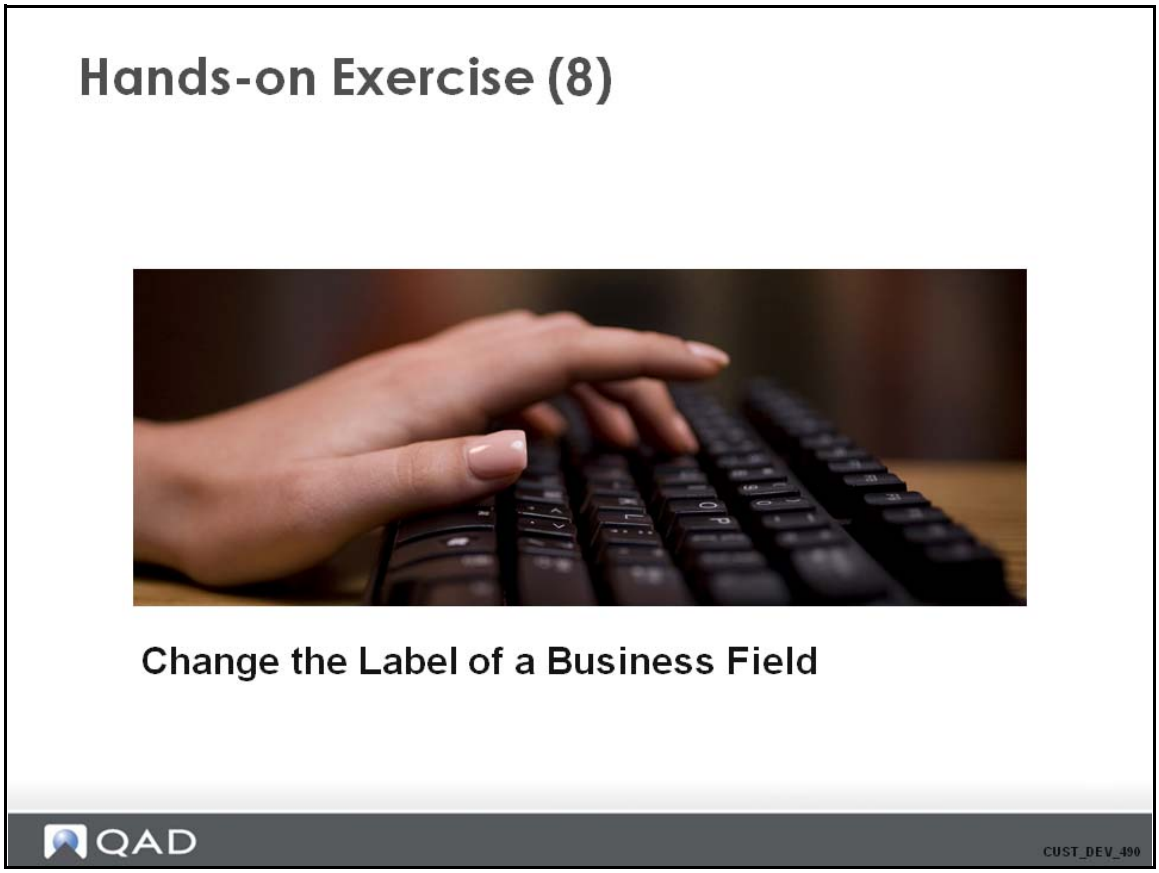
Translations 3



Translations 4



Hands-on Exercise (8)

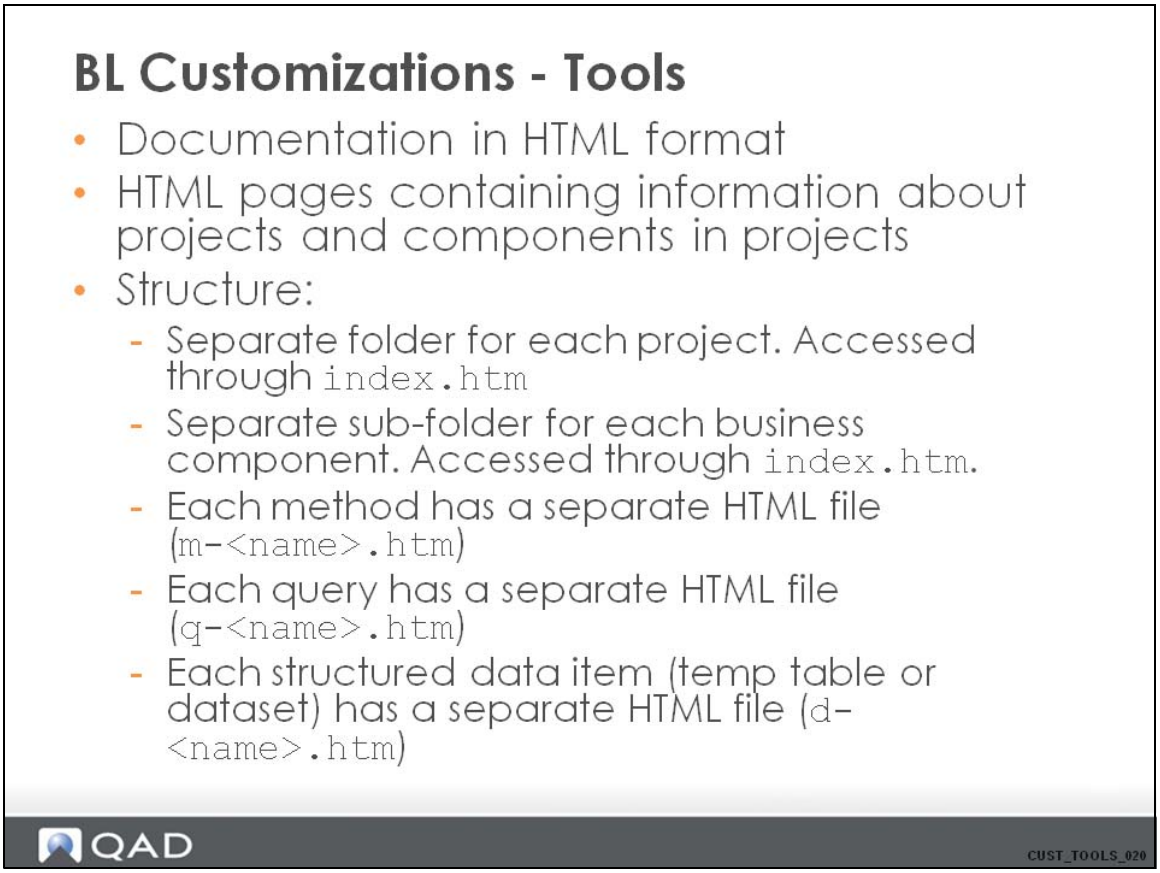


Chapter 5

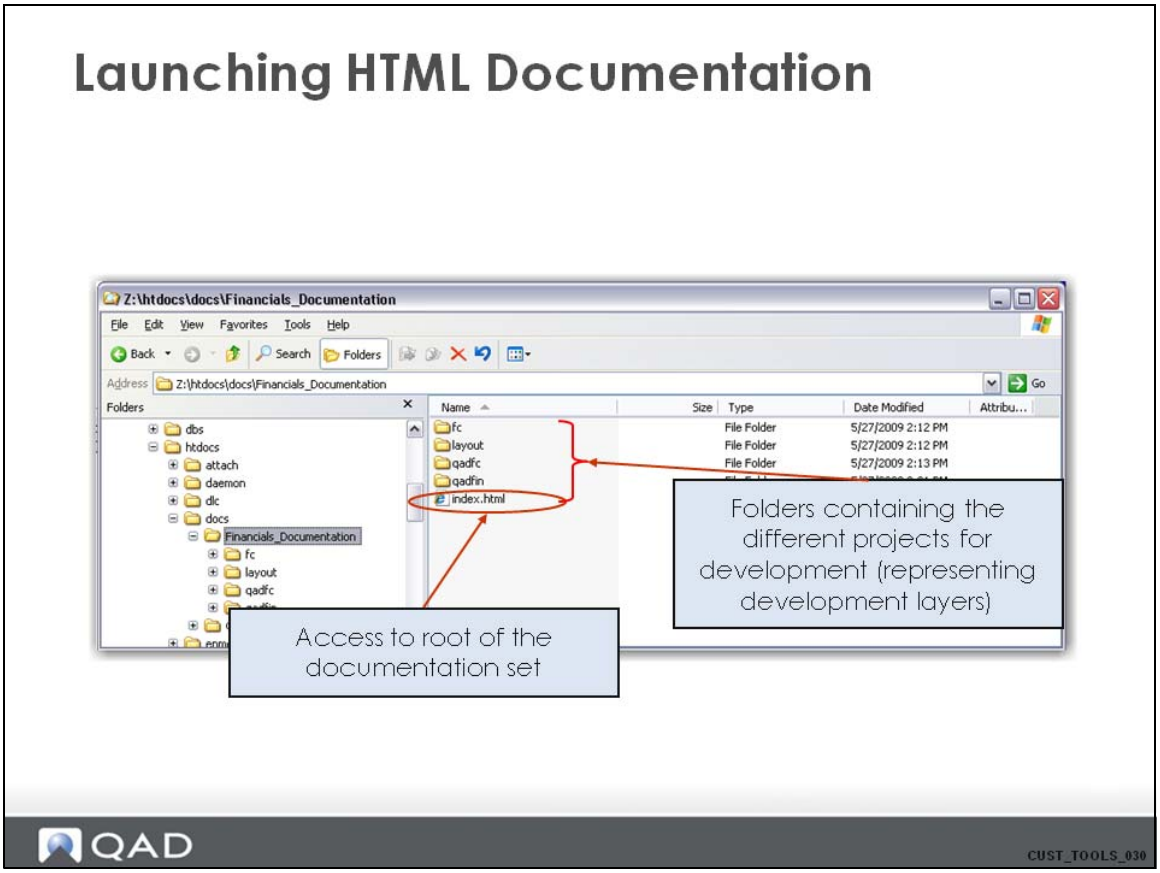
Non-intrusive Customization: Supporting Tools

Non-intrusive Customization: Supporting Tools

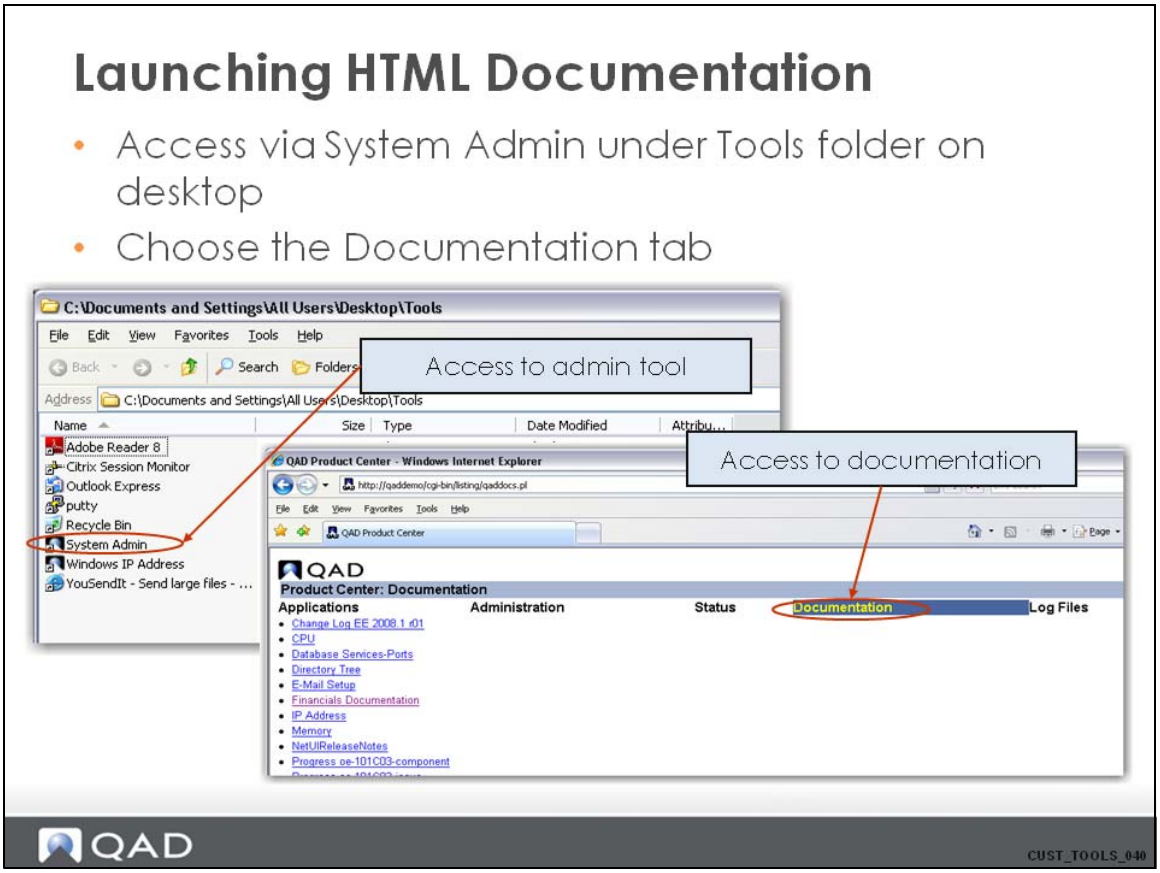
BL Customization - Tools



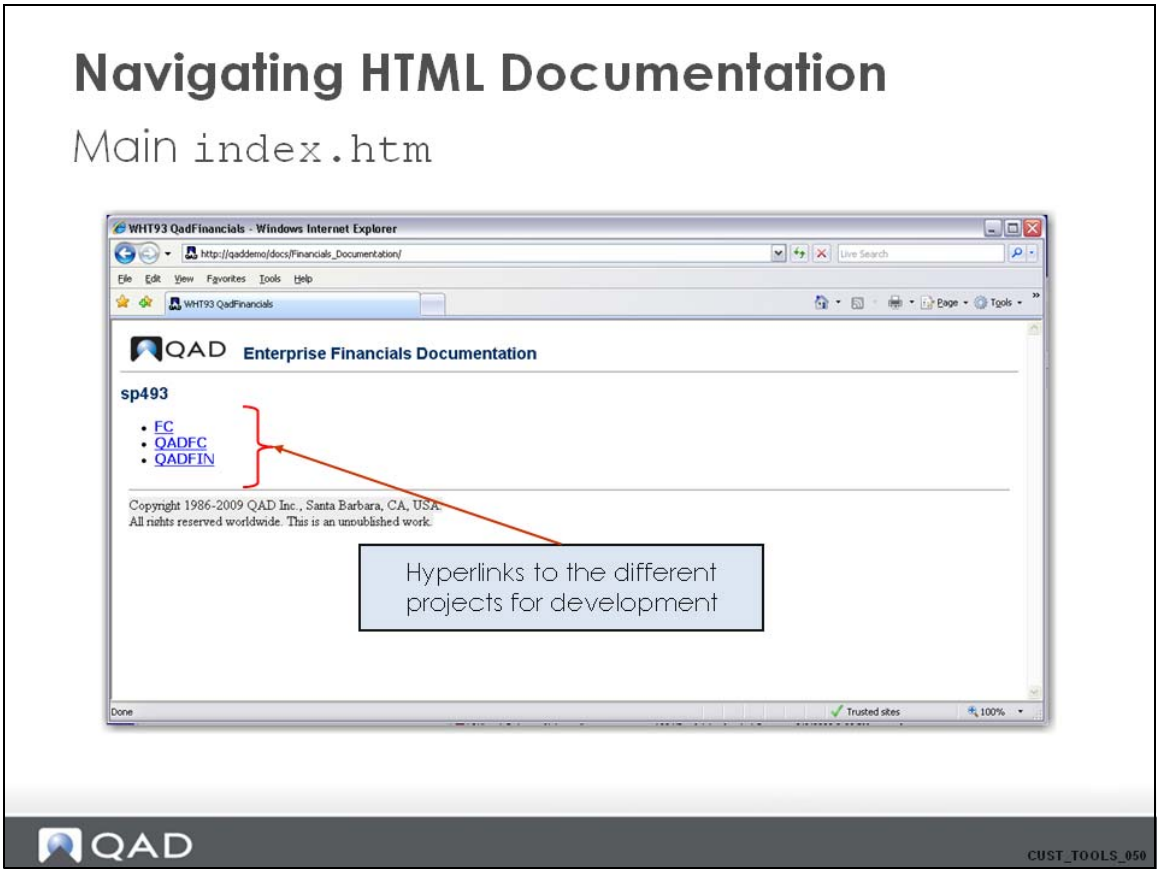
Launching HTML Documentation



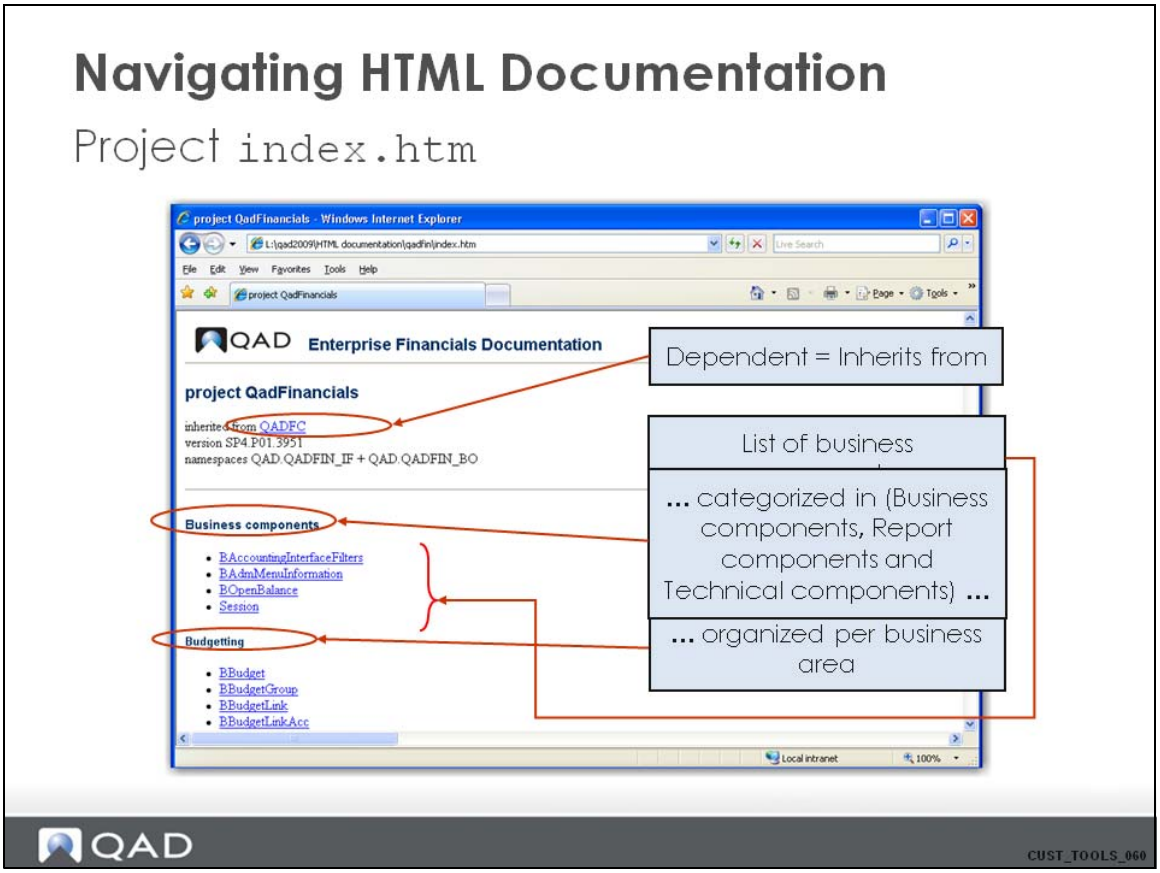
Launching HTML Documentation 2



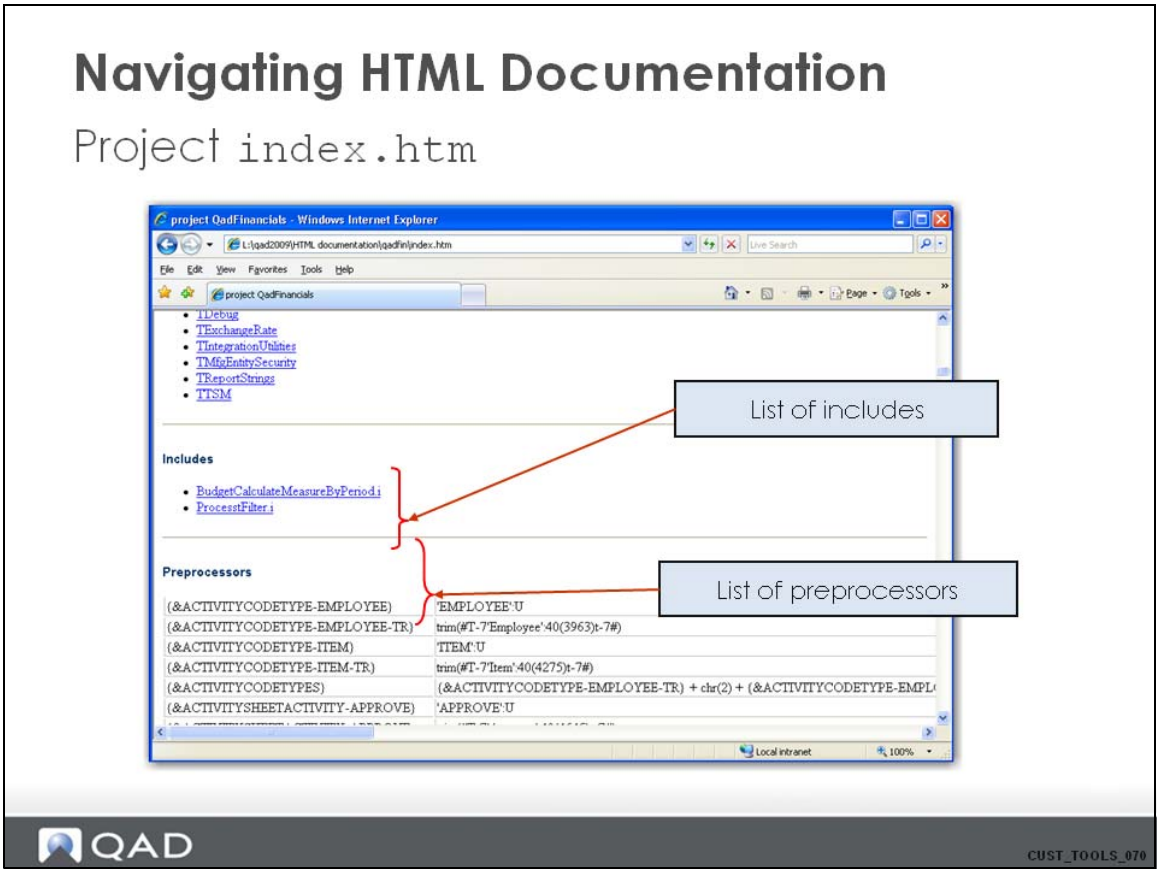
Navigating HTML Documentation 3



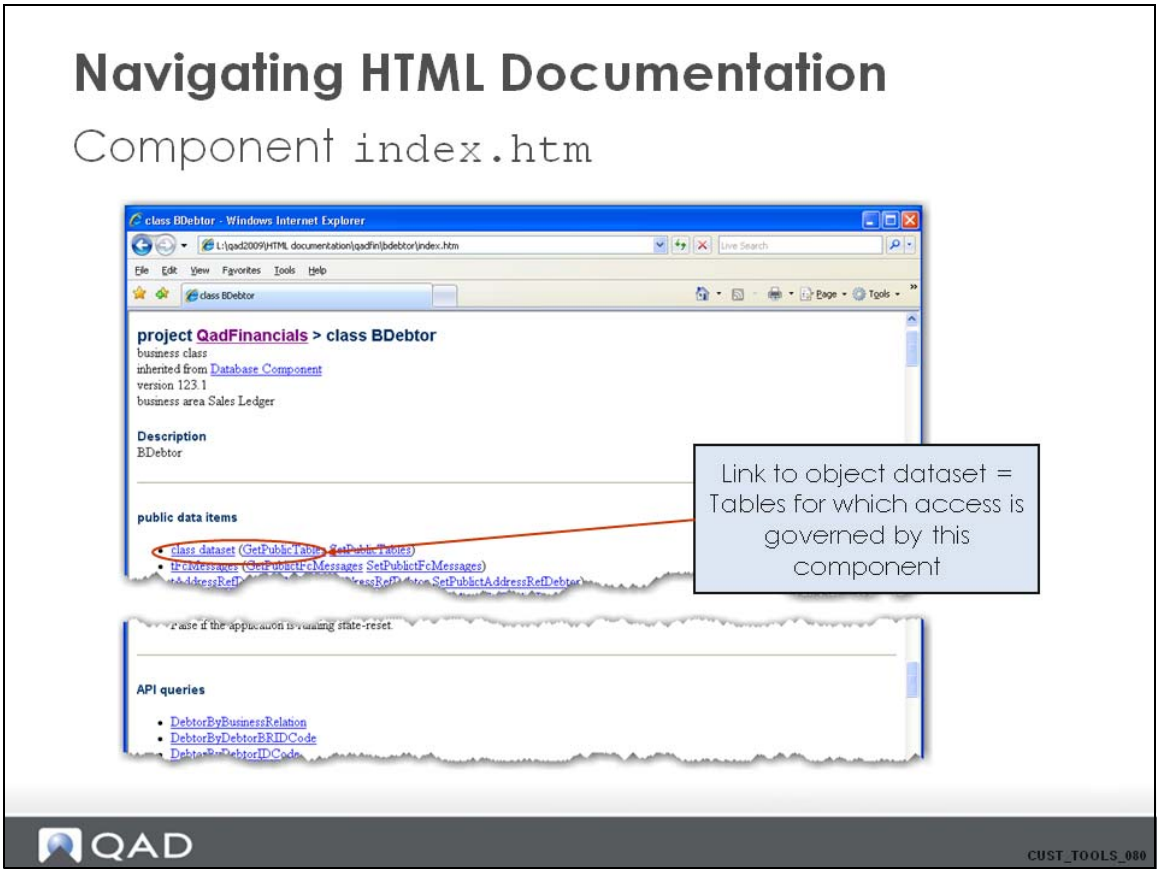
Navigating HTML Documentation 4



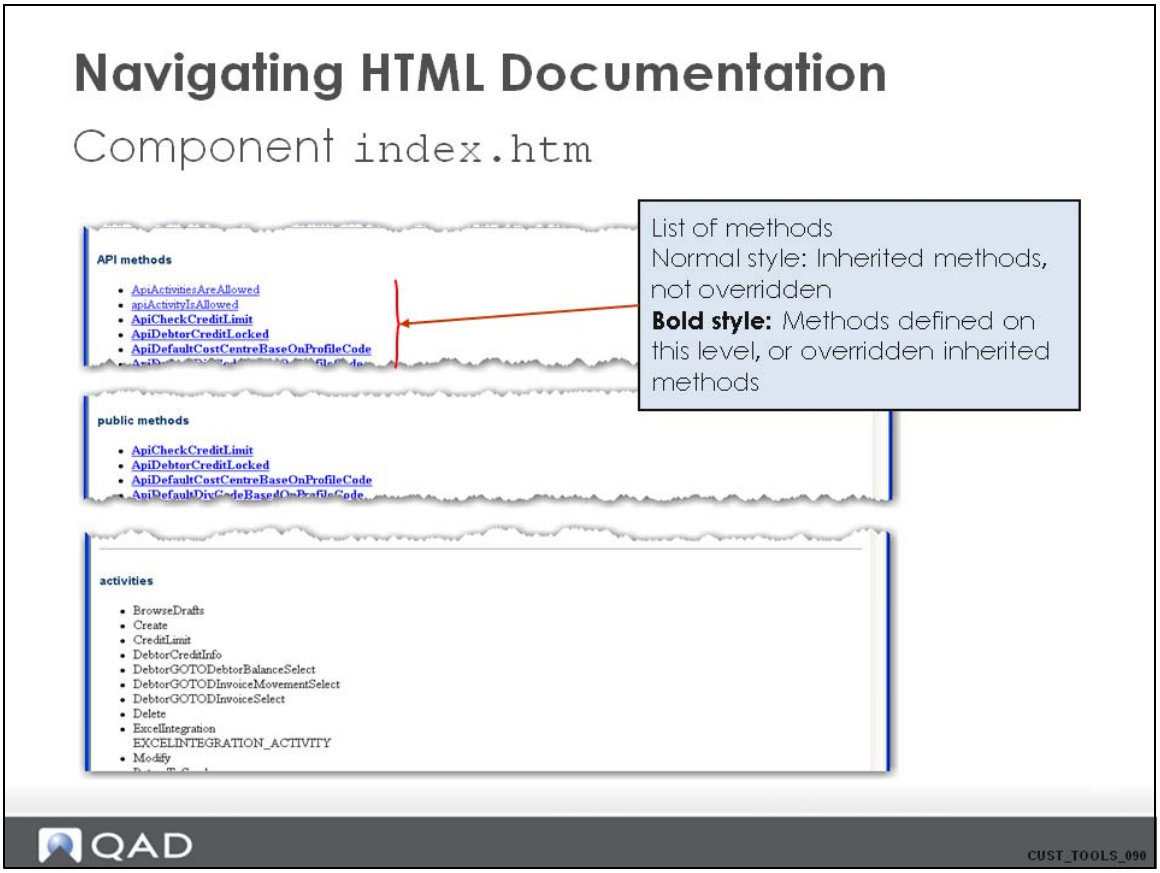
Navigating HTML Documentation 5



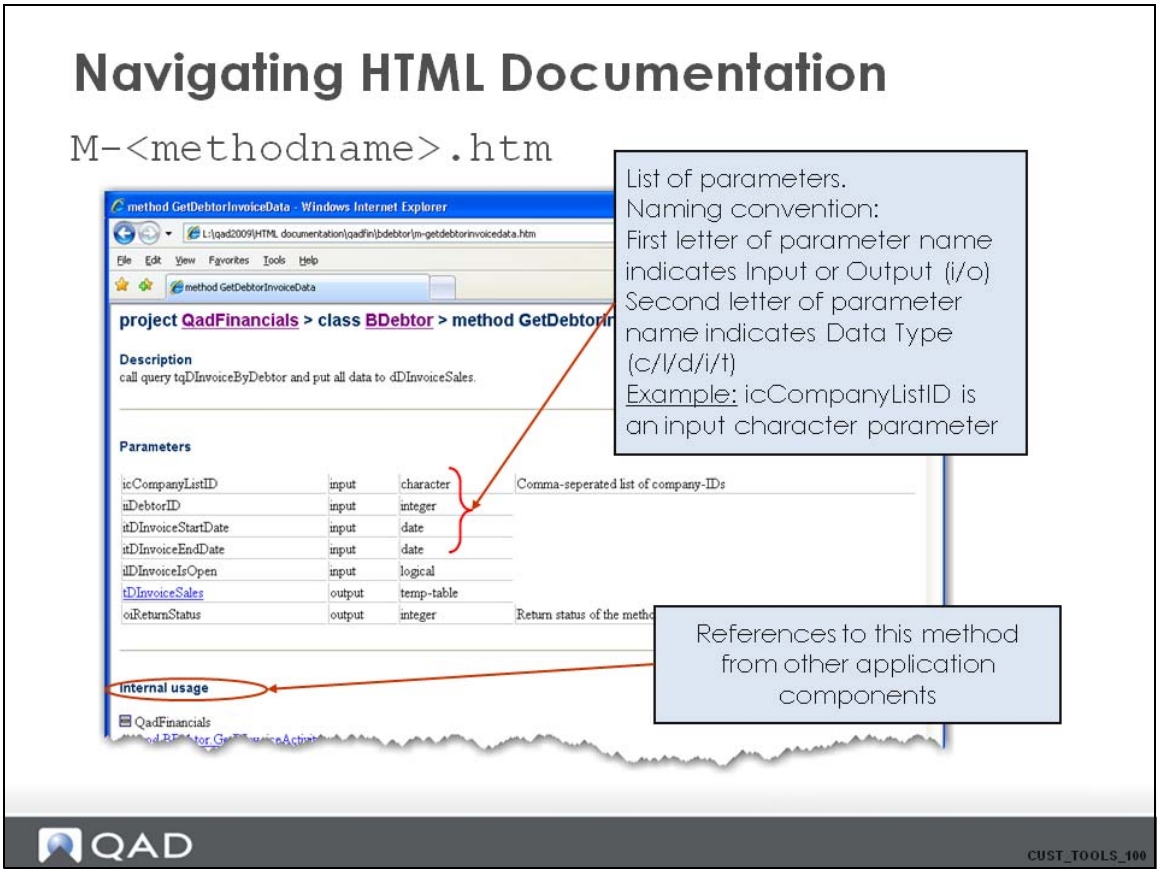
Navigating HTML Documentation 6



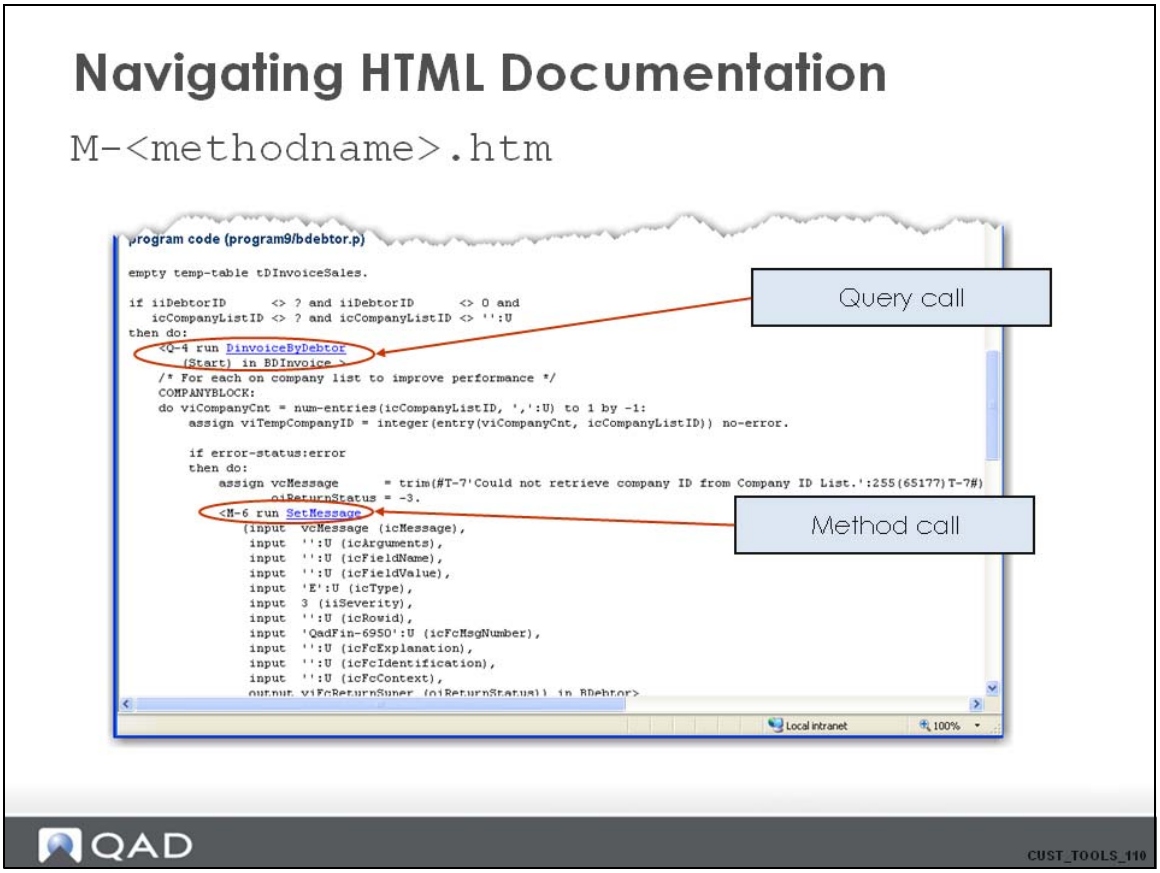
Navigating HTML Documentation 7



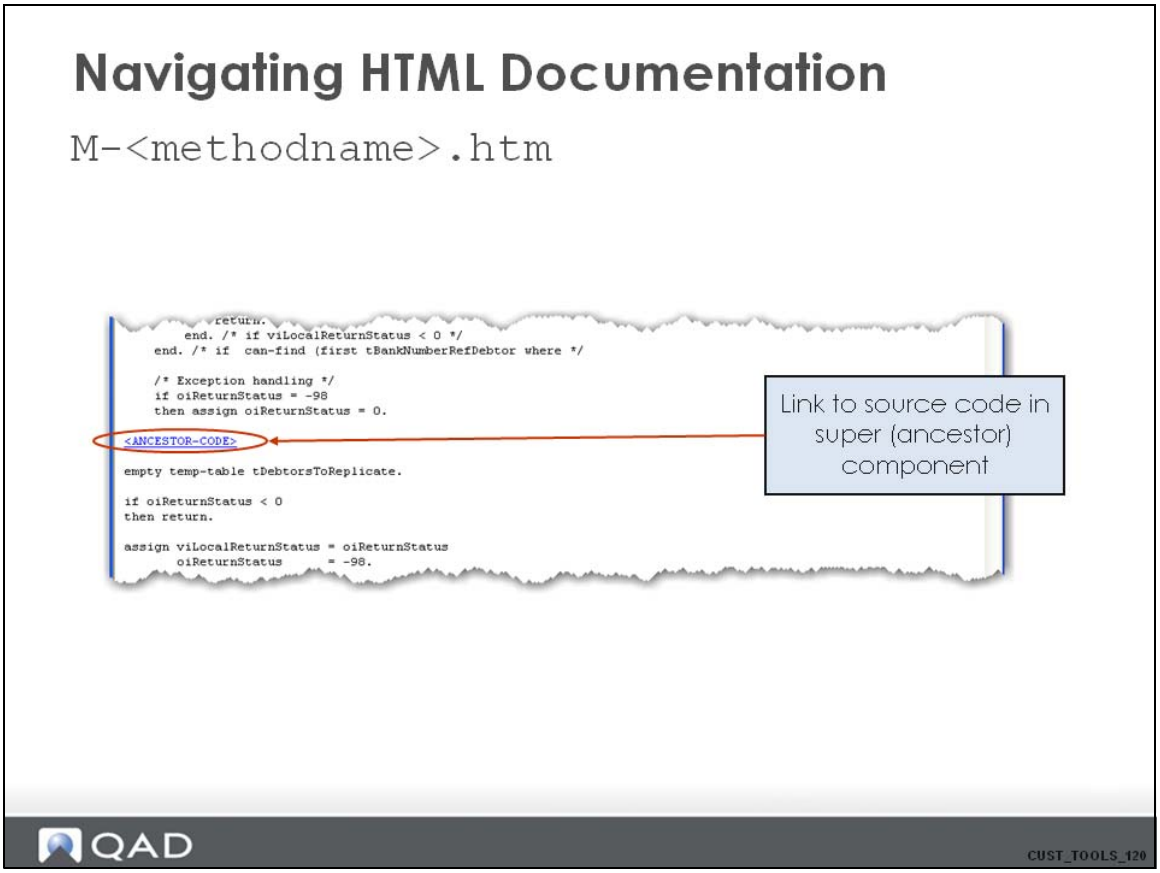
Navigating HTML Documentation 8



Navigating HTML Documentation 9



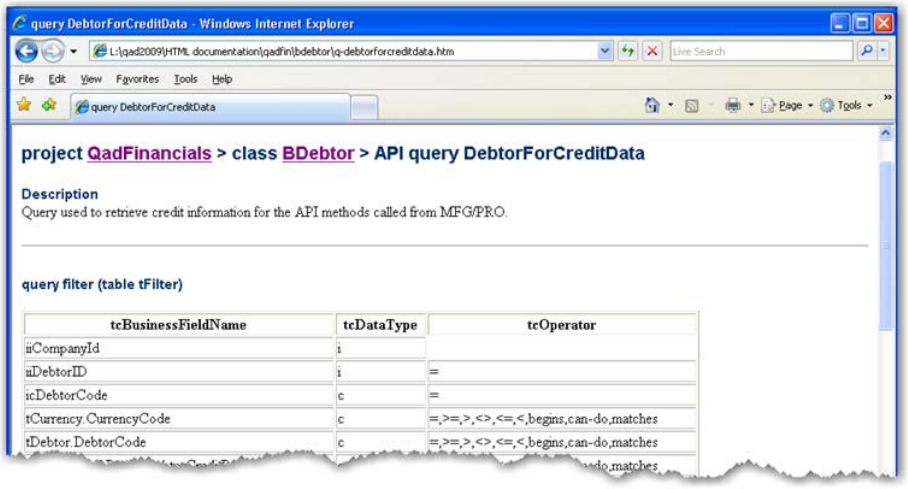
Navigating HTML Documentation 10



Navigating HTML Documentation 11

Navigating HTML Documentation

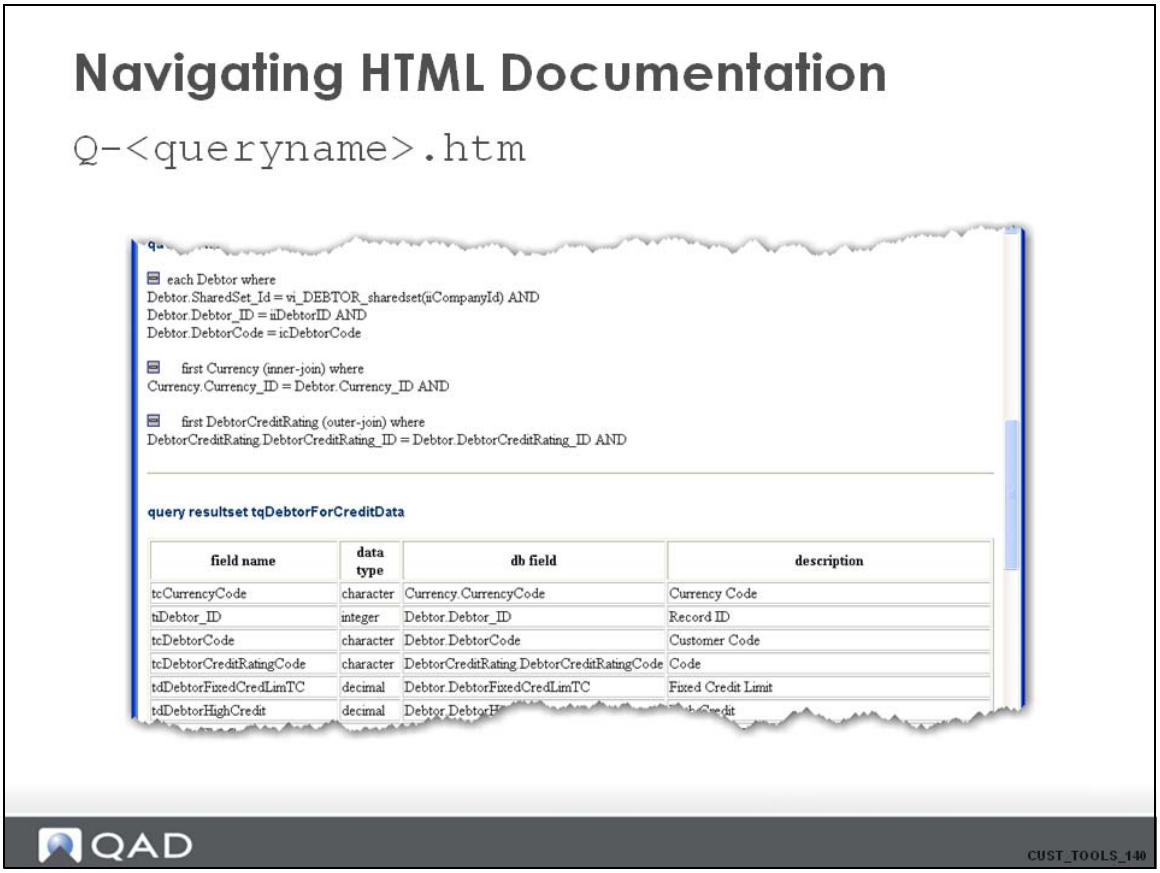
Q-<queryname>.htm



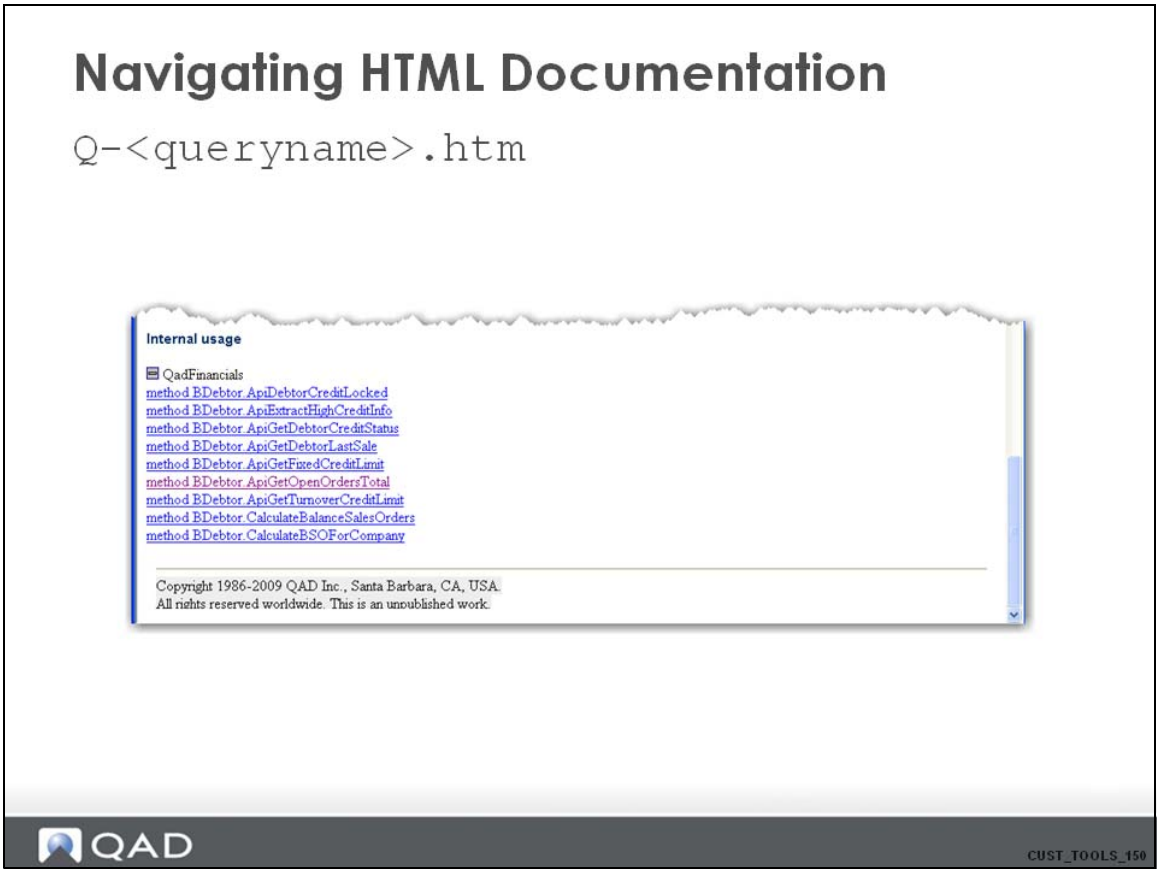
QAD

CUST_TOOLS_130

Navigating HTML Documentation 12



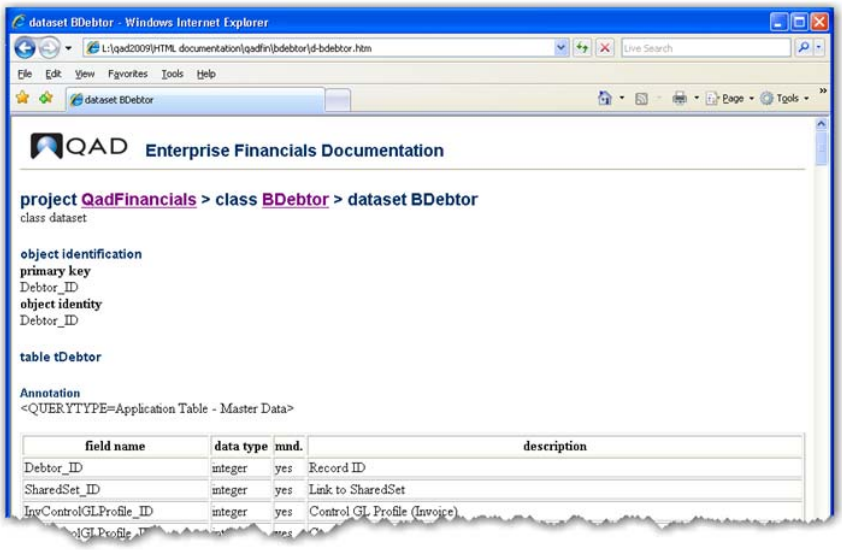
Navigating HTML Documentation 13



Navigating HTML Documentation 14

Navigating HTML Documentation

D-<datasetName>.htm



QAD

CUST_TOOLS_160

Navigating HTML Documentation 15

Navigating HTML Documentation

D-<datasetName>.htm

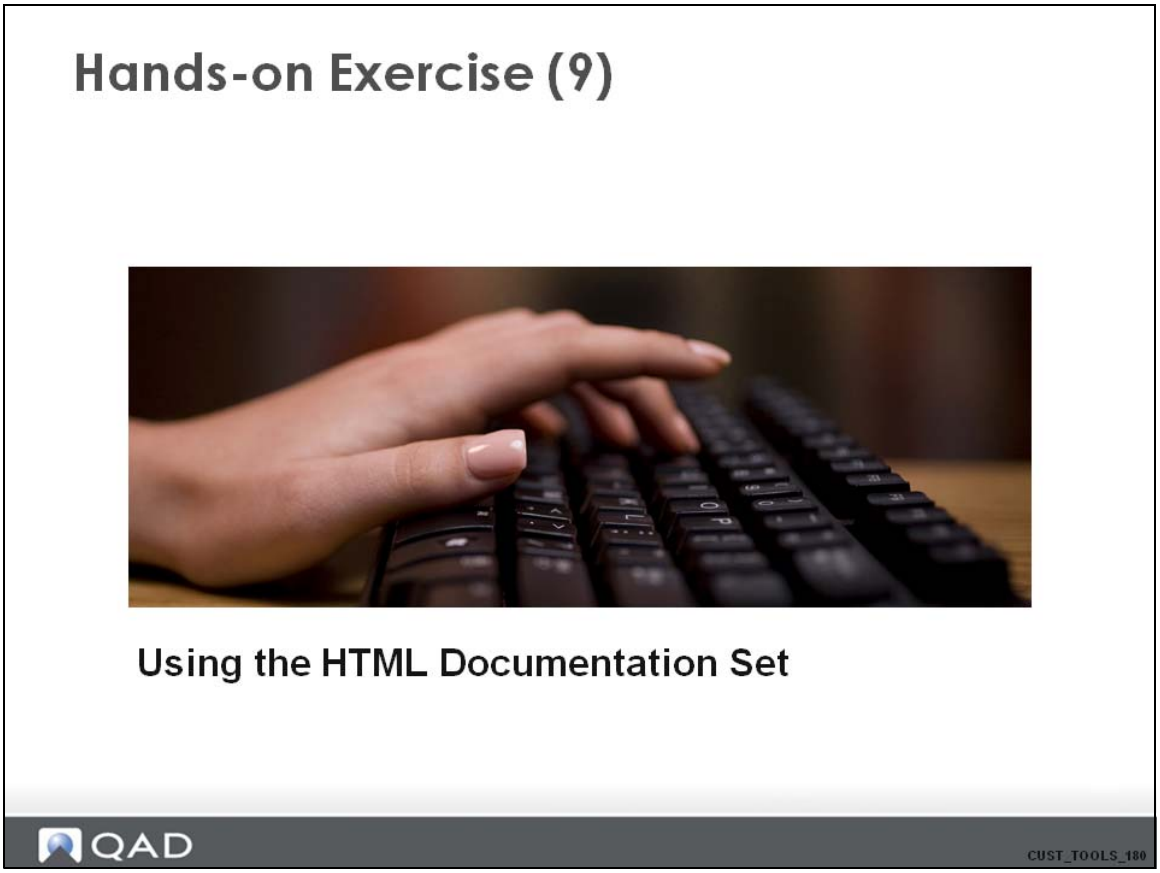
table tDebtorSafDefault

field name	data type	mnd.	description
DebtorSafDefault_ID	integer	yes	Record ID
Debtor_ID	integer	yes	Link to Debtor
SafConcept_ID	integer	yes	SAF Concept Code
Saf_ID	integer	yes	SAF Code
tcSafConceptCode	character	no	SAF Concept Code
tcSafCode	character	no	SAF Code
LastModifiedDate	date	no	Last Modified Date
LastModifiedTime	integer	no	Last Modified Time
LastModifiedUser	character	no	Last Modified User
tc_Rowid	character	yes	primary index
tc_ParentRowid	character	yes	= tDebtor.tc_Rowid
tc_Status	character	no	update status

QAD

CUST_TOOLS_170

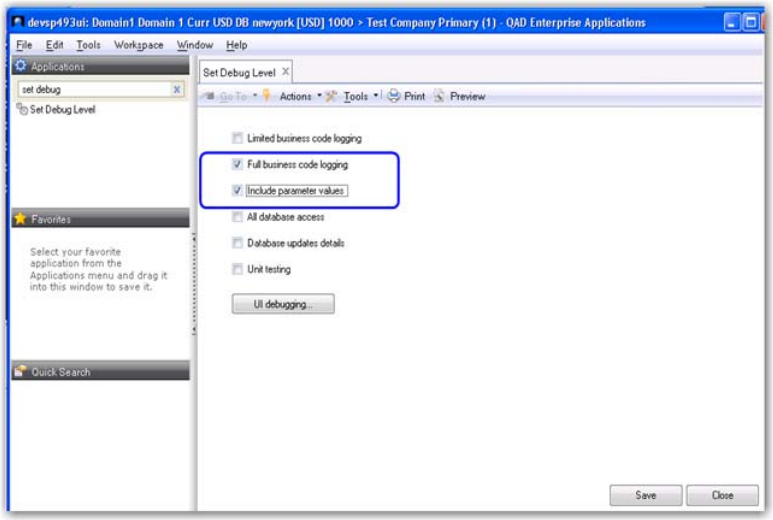
Hands-on Exercise (9)



BL Customizations - Tools

BL Customizations – Tools

- BL Code Tracing (CT Log file)
- Use Set Debug Level



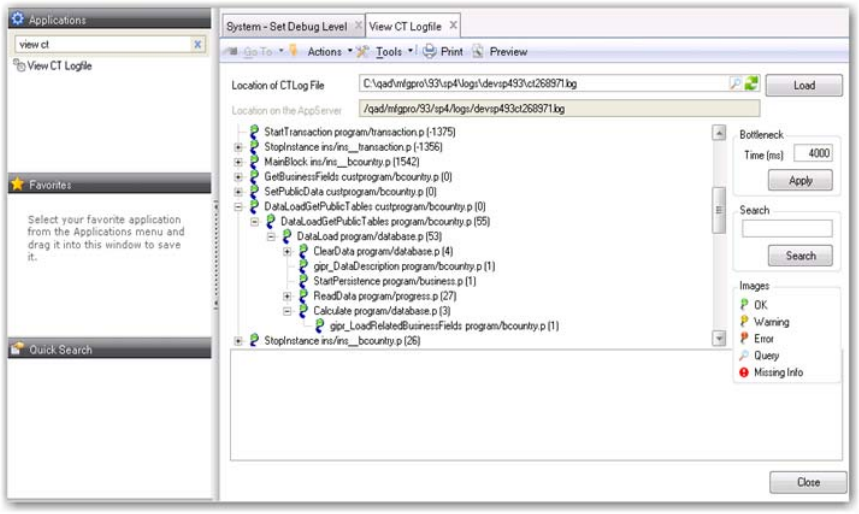
QAD

CUST_TOOLS_190

BL Customizations - BL Code Tracing

BL Customizations – BL Code Tracing

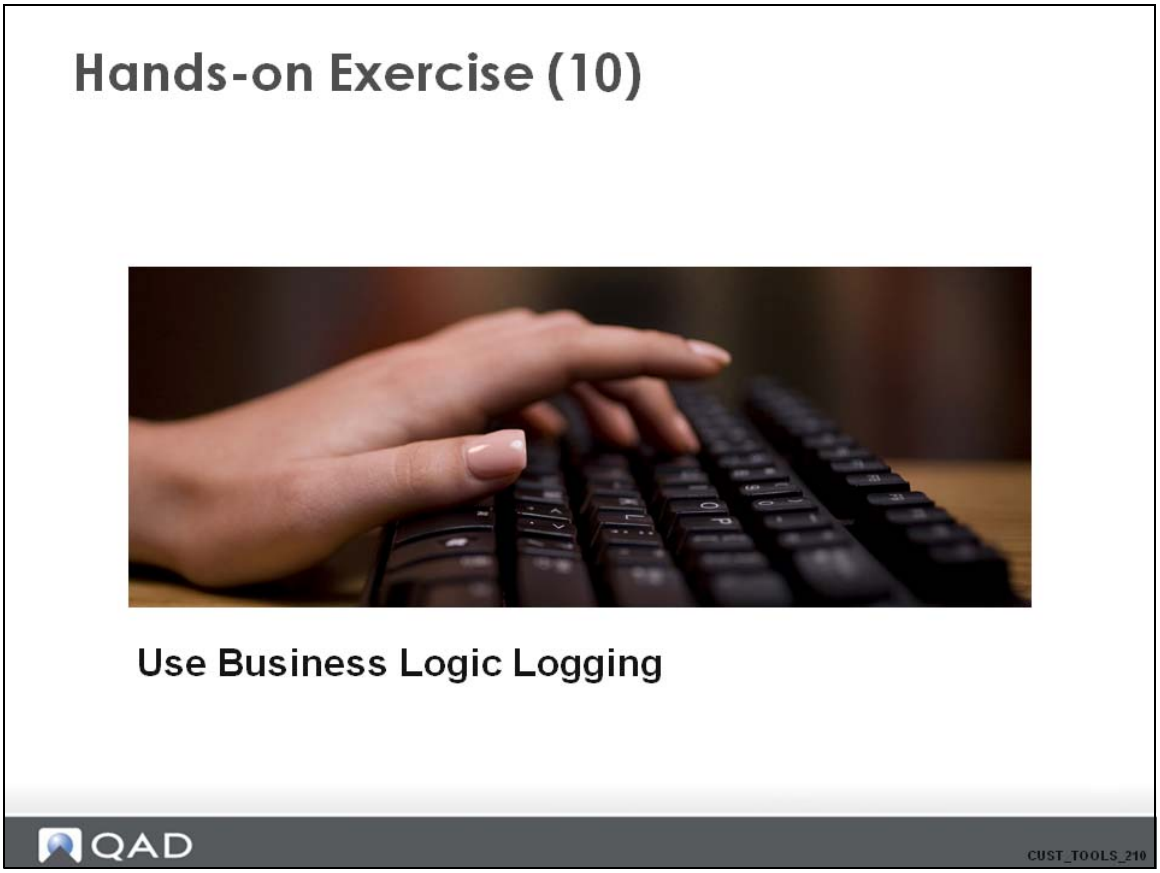
Use View CT Logfile



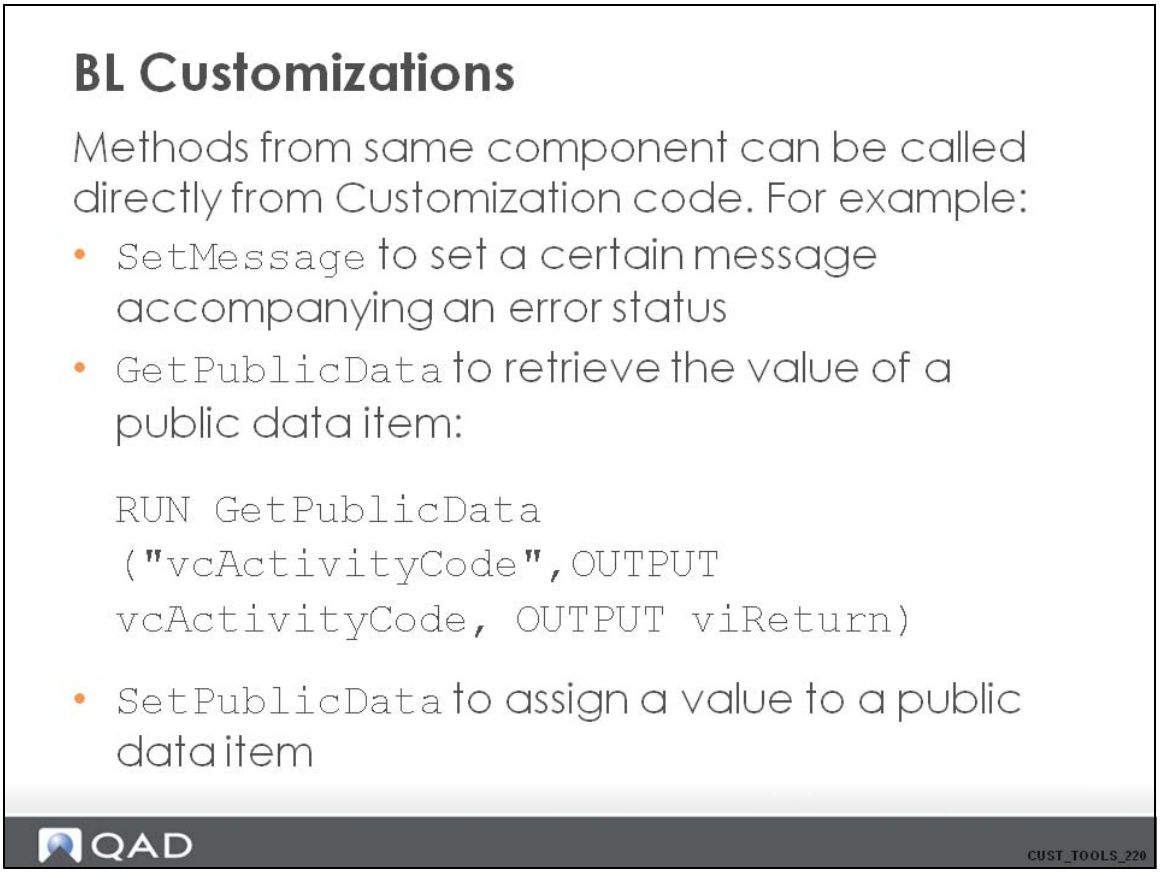
QAD

CUST_TOOLS_200

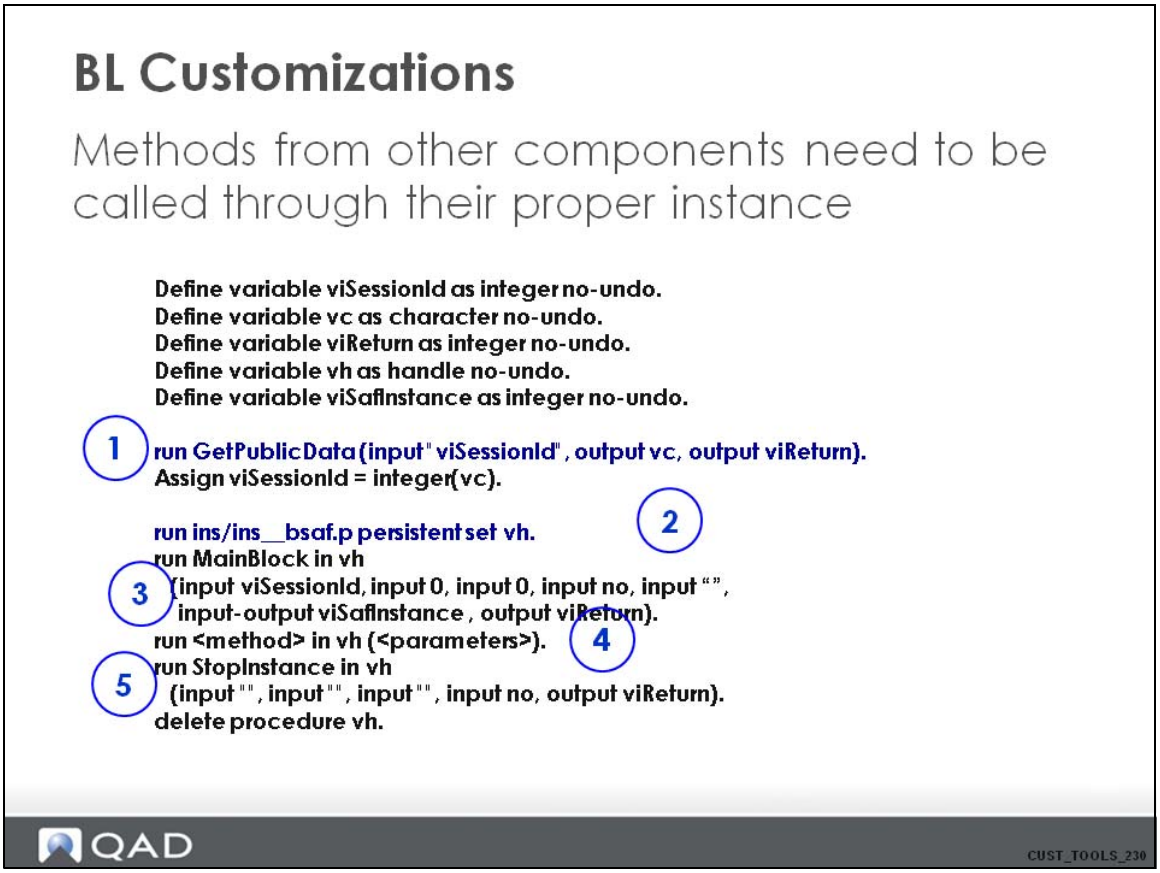
Hands-on Exercise (10)



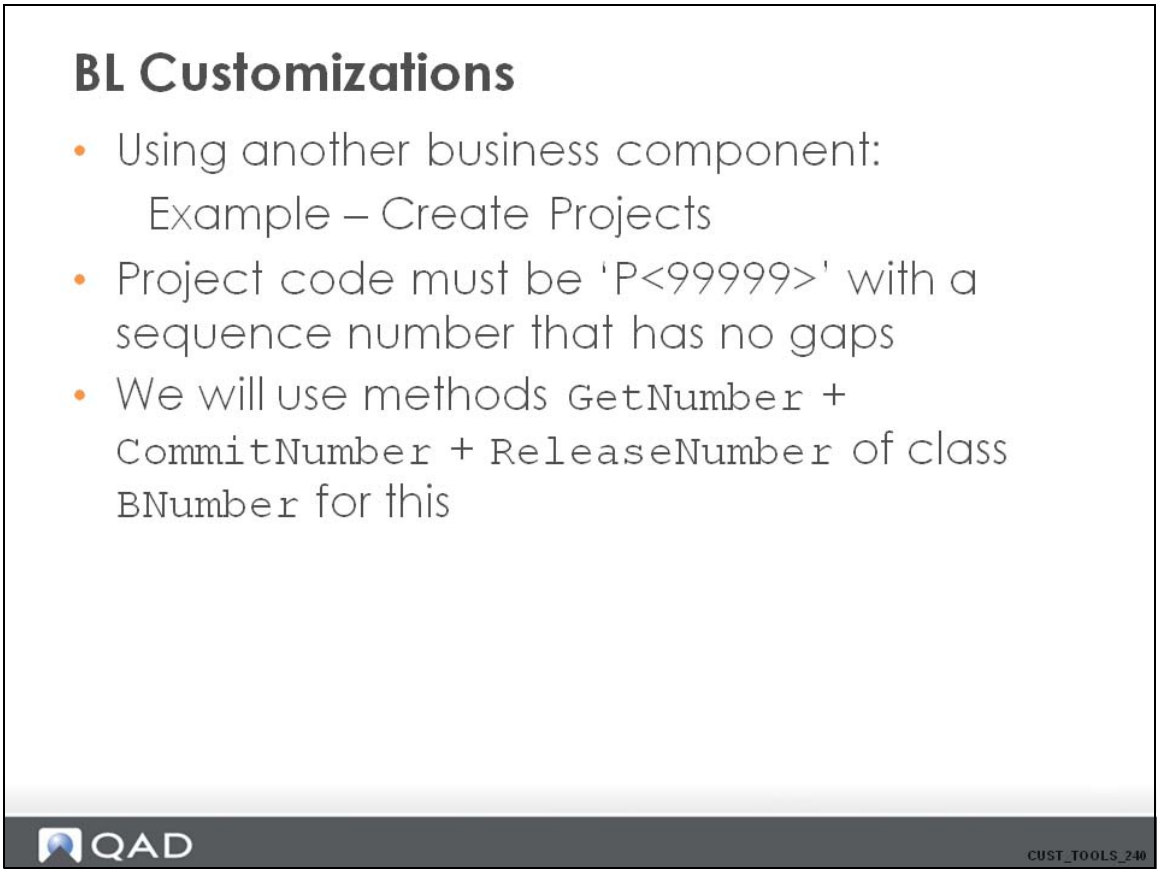
BL Customizations



BL Customizations 2

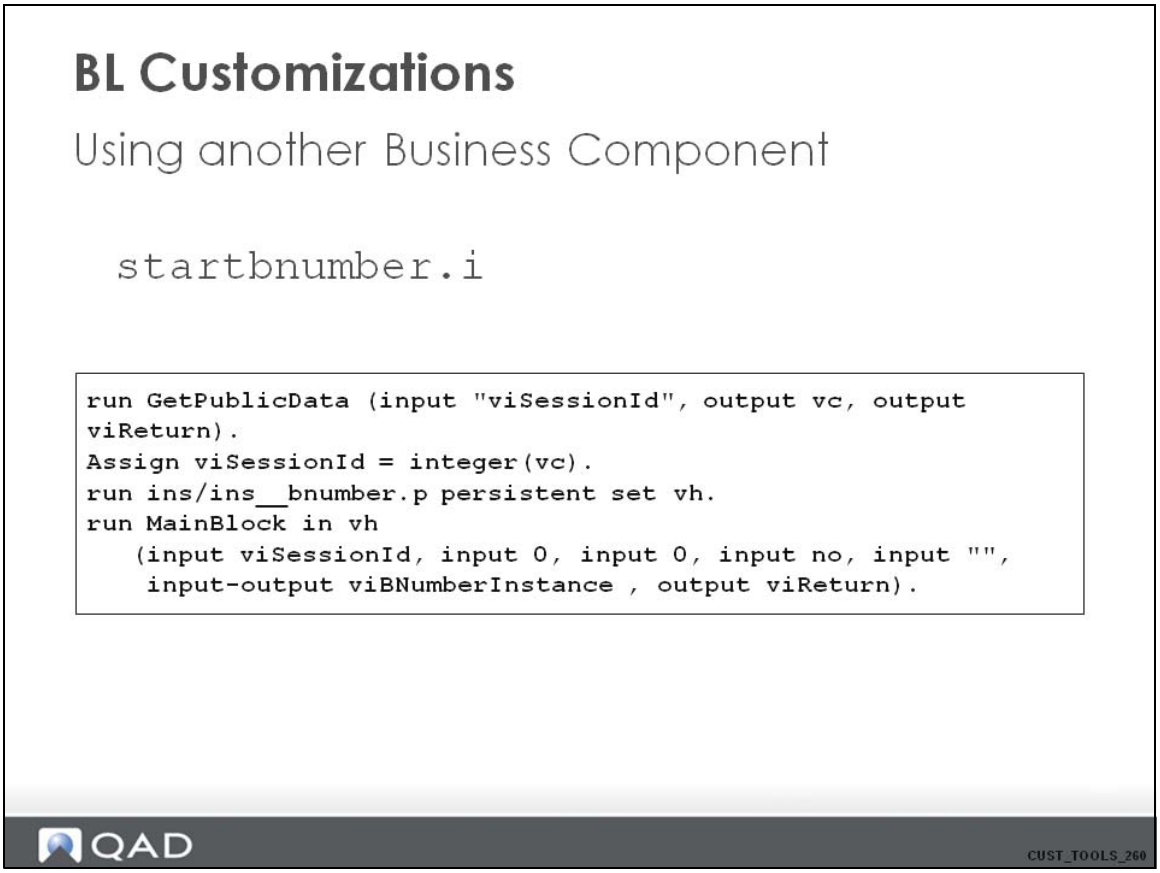


BL Customizations 3

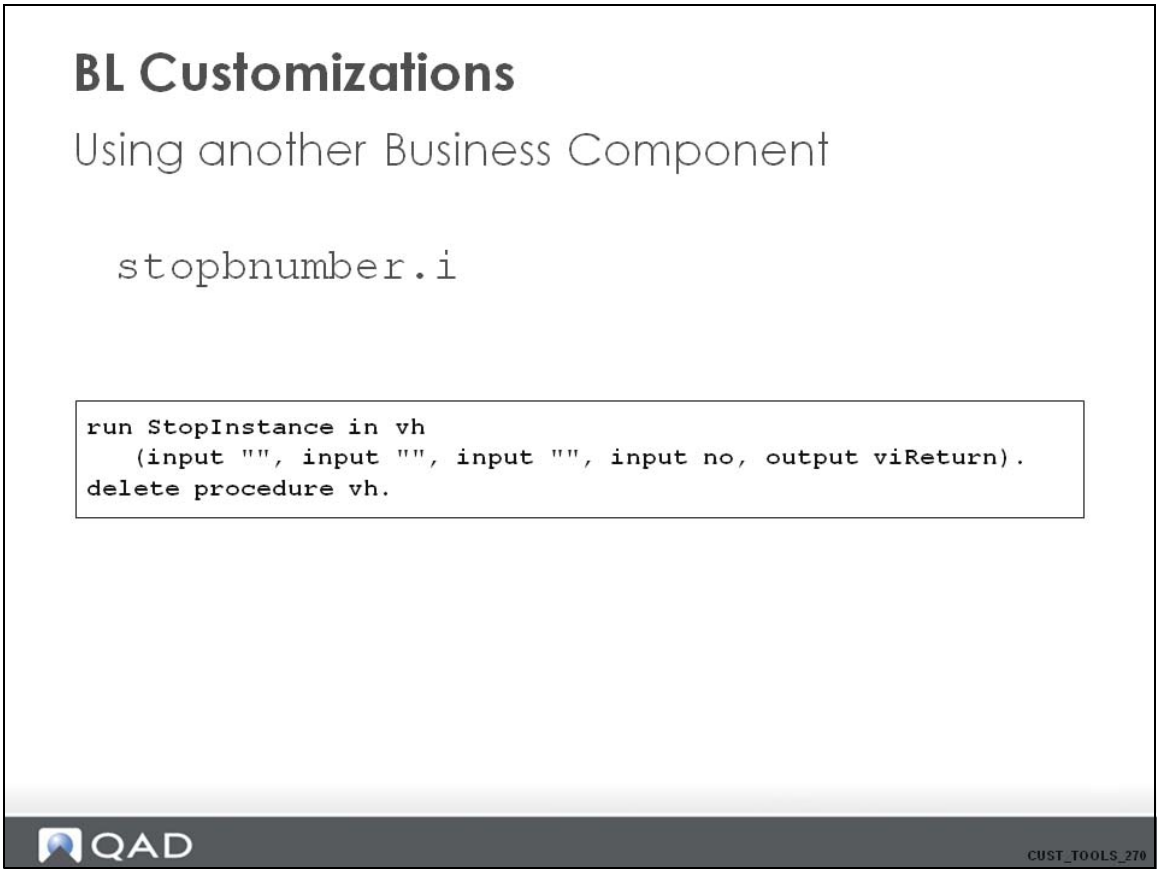


BL Customizations 4

BL Customizations 5



BL Customizations 6



BL Customizations 7

BL Customizations

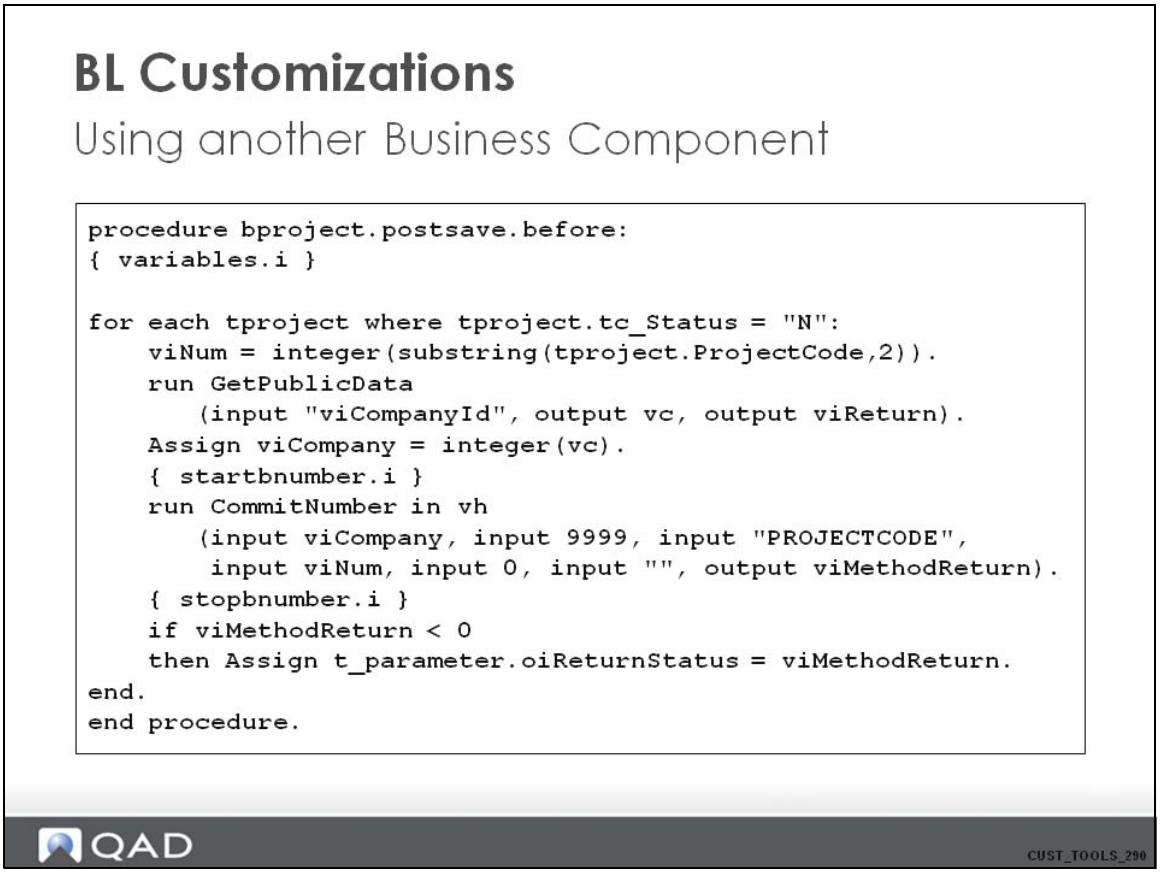
Using another Business Component

```
procedure bproject.initialvalues.before:
{ variables.i }

if t_parameter.icTableName = "project" then do:
    run GetPublicData
       (input "viCompanyId", output vc, output viReturn).
    Assign viCompany = integer(vc).
    { startbnumber.i }
    run GetNumber in vh
       (input viCompany, input 9999, input "PROJECTCODE",
        output viNum, input 0, input "", output viMethodReturn).
    { stopbnumber.i }
    if viMethodReturn < 0
    then Assign t_parameter.oiReturnStatus = viMethodReturn.
    else Assign tproject.projectcode = "P" + string(viNum,"99999").
end.
end procedure.
```

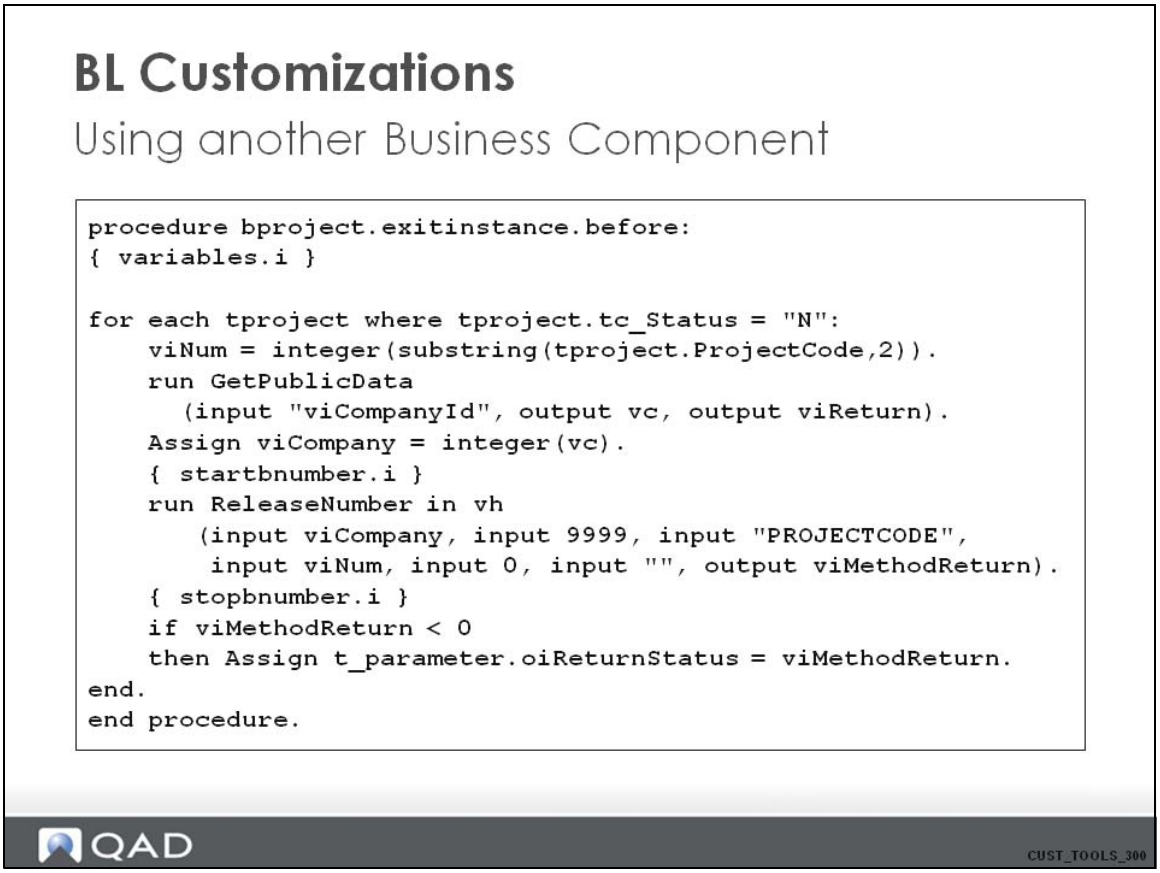
QAD

CUST_TOOLS_280

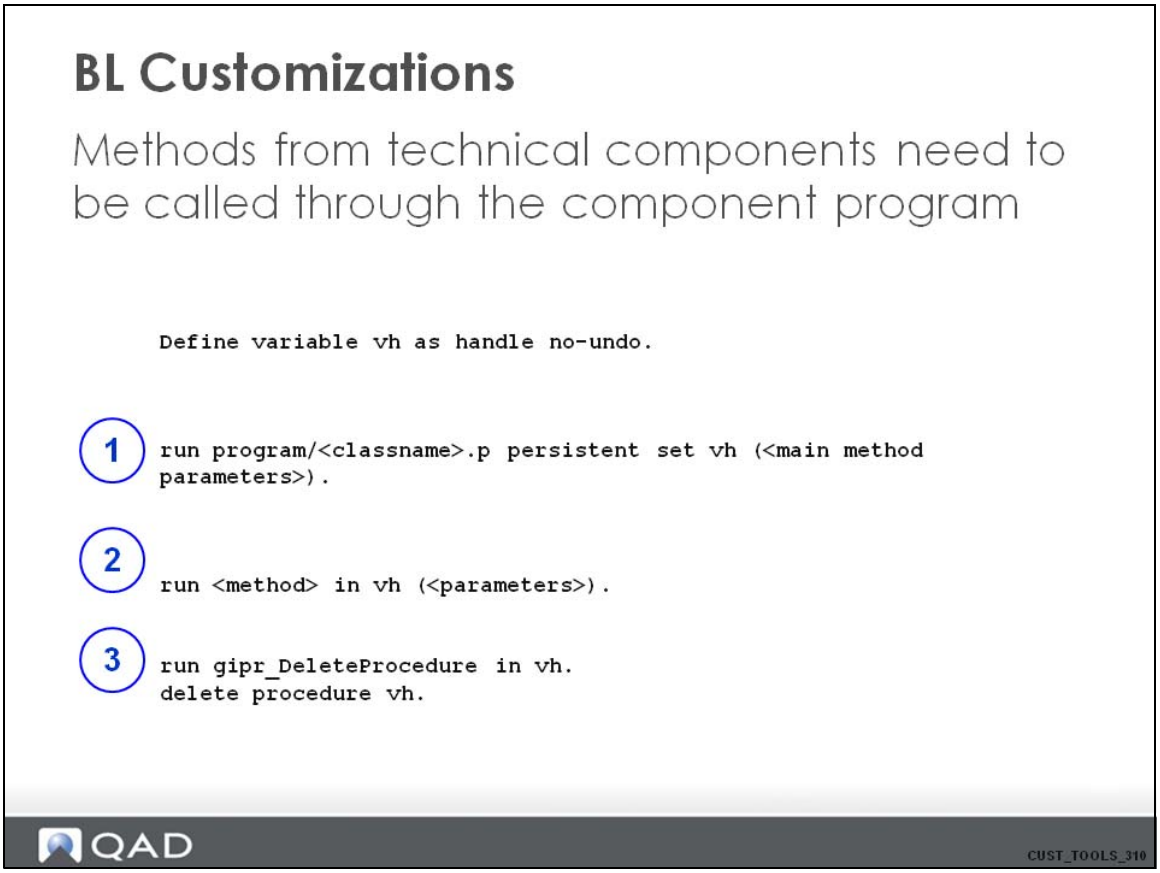
BL Customizations 8



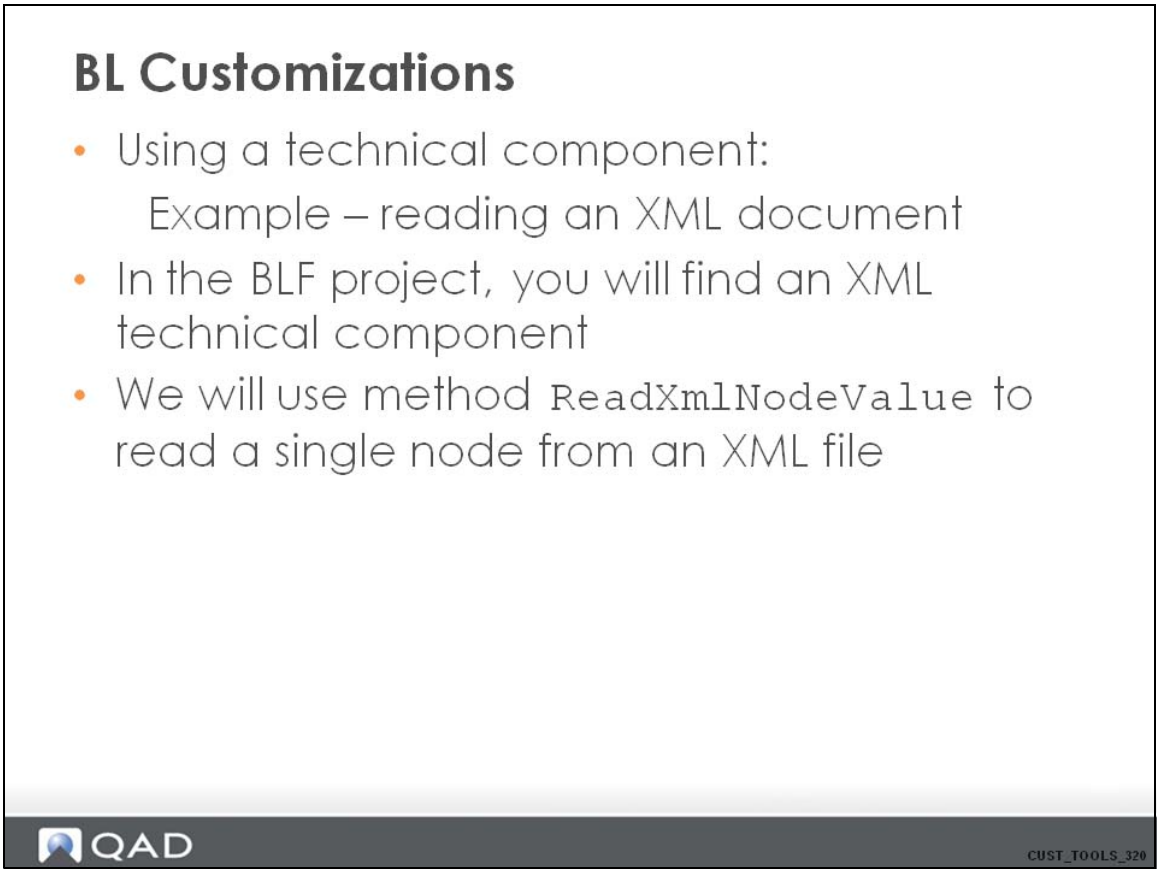
BL Customizations 9



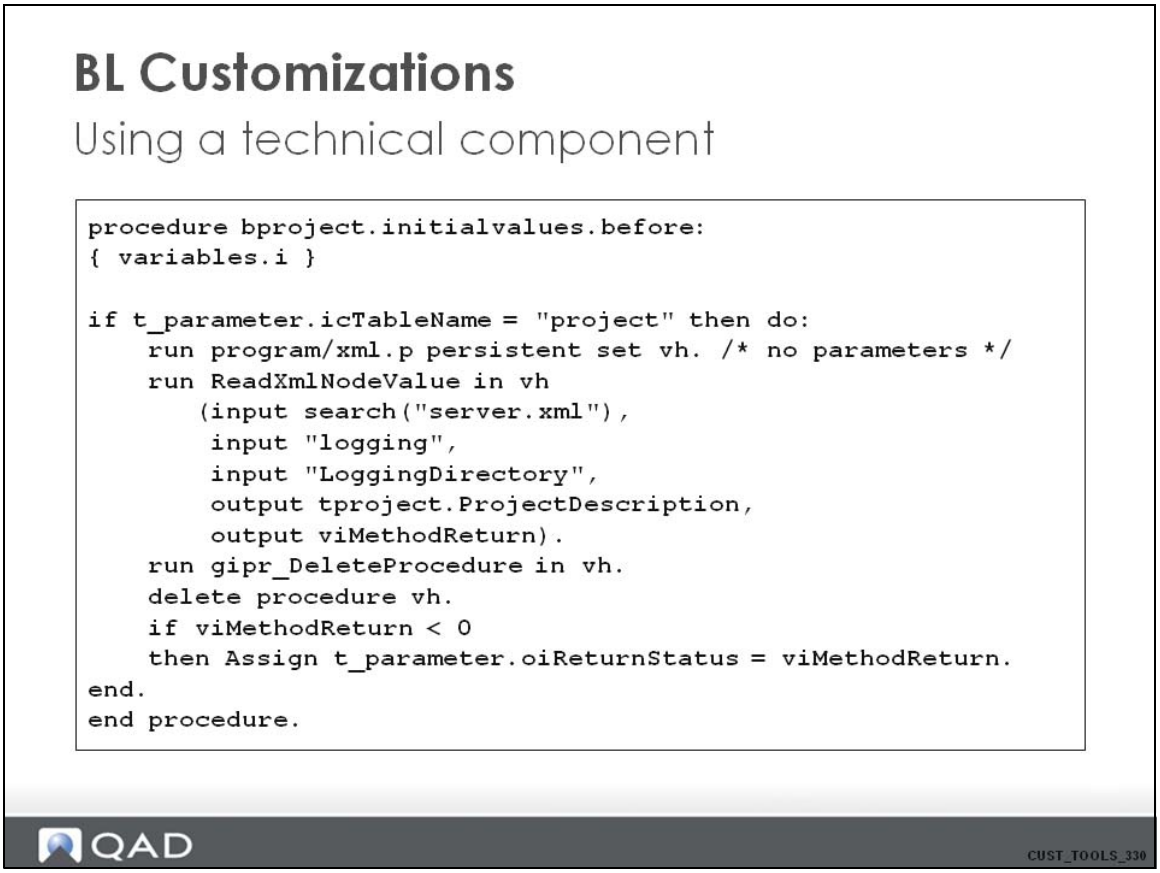
BL Customizations 10



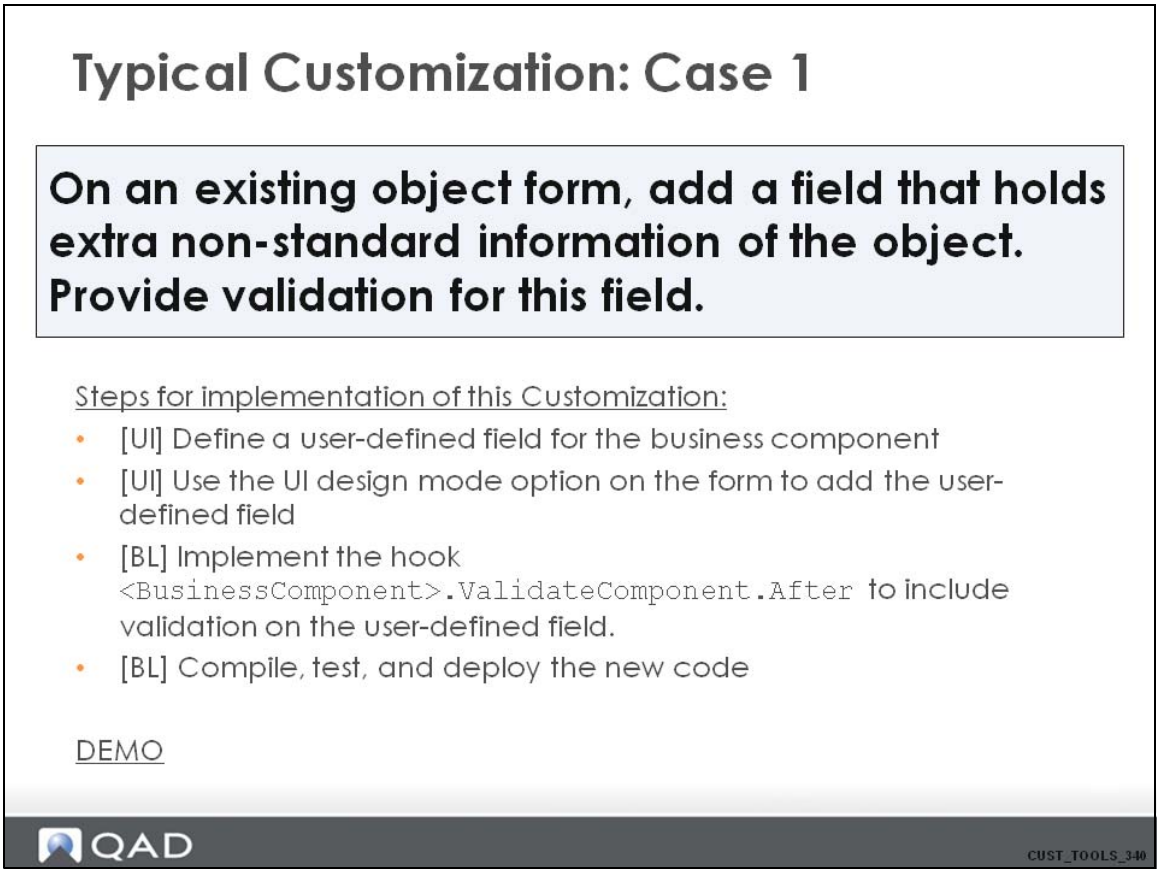
BL Customizations 11



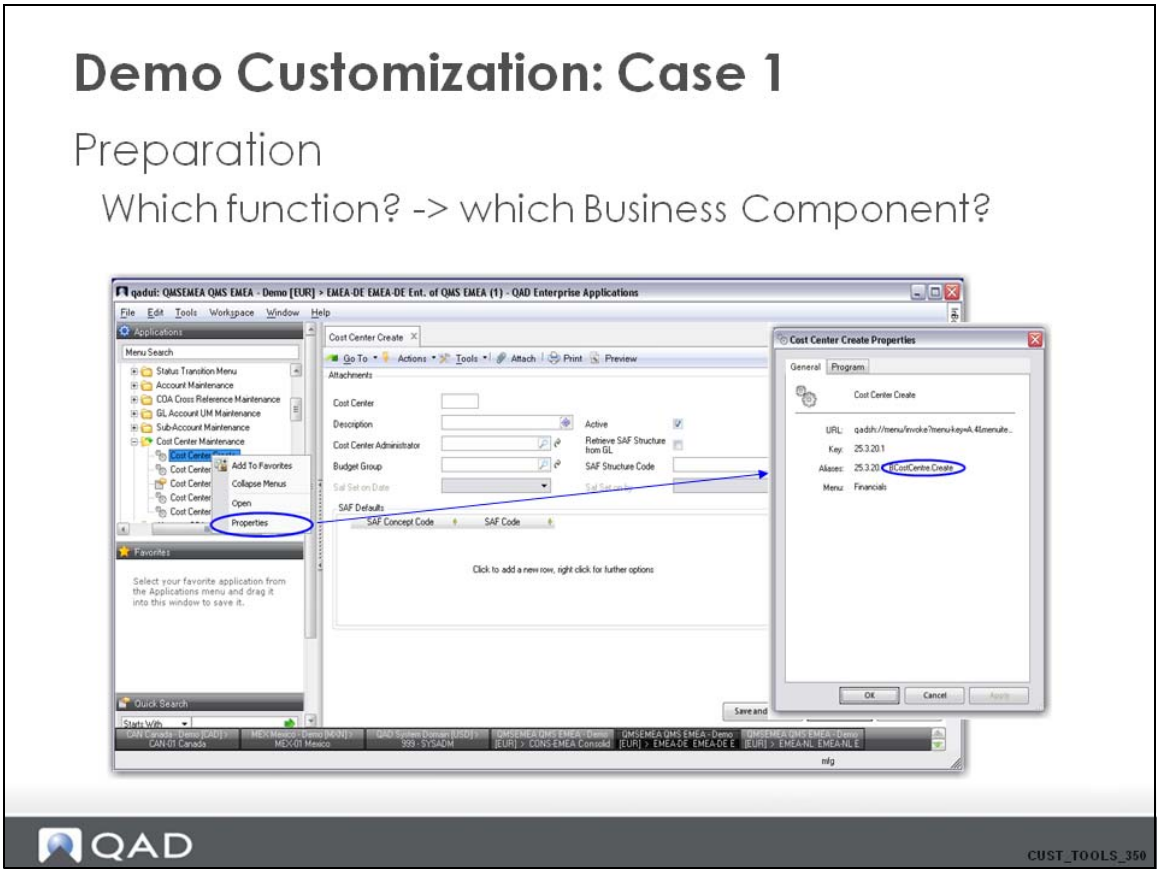
BL Customizations 12



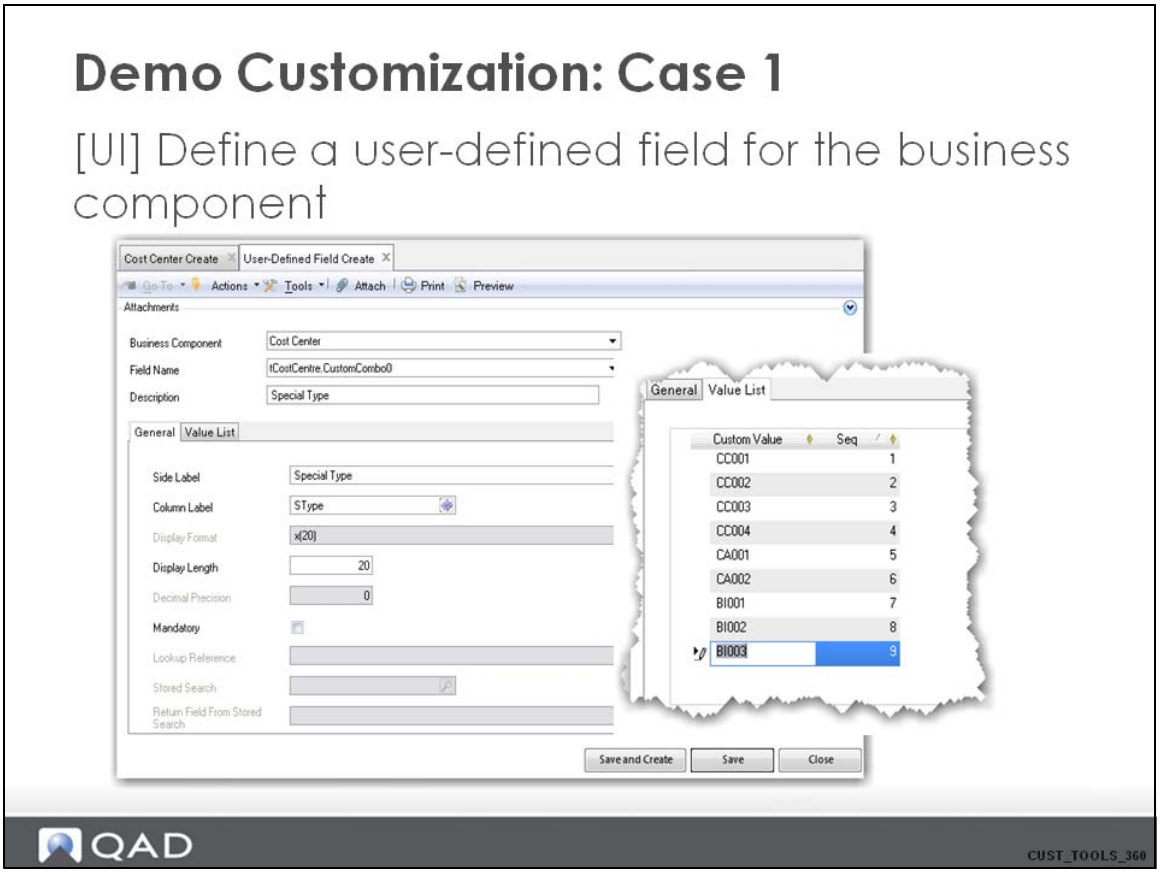
Typical Customization: Case 1



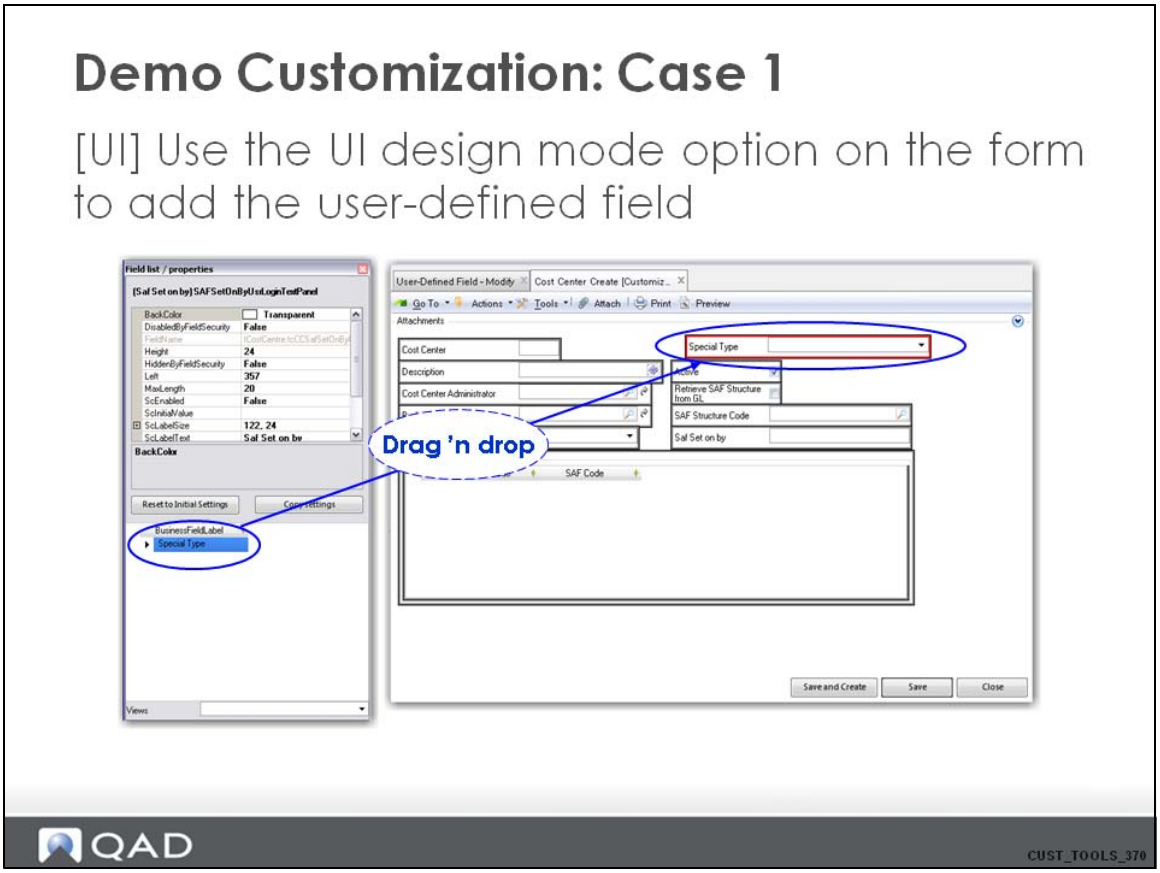
Demo Customization: Case 1



Demo Customization: Case 1 (2)



Demo Customization: Case 1 (3)

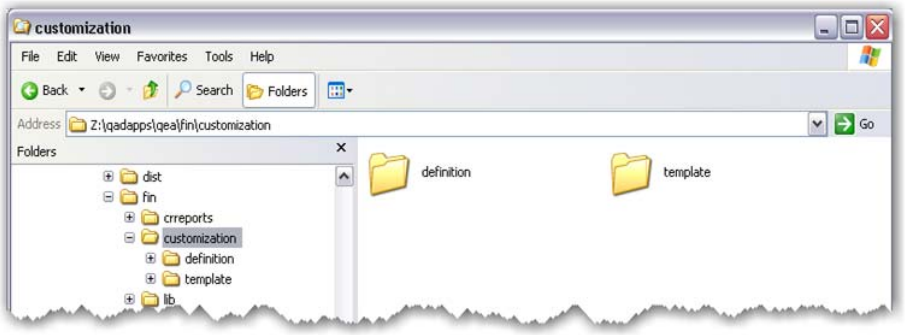


Demo Customization: Case 1 (4)

Demo Customization: Case 1

Preparation

- Fin AppServer PROPATH should include a folder with `customcode` as sub-folder
- Find the `Customization` folder in the installation folder of the Financials business layer



QAD

CUST_TOOLS_380

Demo Customization: Case 1 (5)

Demo Customization: Case 1

[BL] Implement the hook `<BusinessComponent>.ValidateComponent.After` to include validation on the user-defined field

Copy the file `bcostcentre.p` from the `Customization/template` folder to the folder that will contain the Customization source code

QAD

CUST_TOOLS_390

Demo Customization: Case 1 (6)

Demo Customization: Case 1

Edit `bcostcentre.p` and add code in `ValidateComponent`

```
END PROCEDURE.
**/
PROCEDURE BCostCentre.ValidateComponent.after:
    DEFINE VARIABLE viReturn AS INTEGER NO-UNDO.
    FOR EACH t_sCostCentre WHERE can-do("N,C",t_sCostCentre.tc_status):
        IF t_sCostCentre.CustomCombo0 = "CC004" AND
           t_sCostCentre.tcUsrLogin <> "mfg"
        THEN DO :
            ASSIGN t_parameter.oiReturnStatus = -1.
            run SetMessage
               (input "The specific type ($1) can only be set for cost centers with admi
                input t_sCostCentre.CustomCombo0 + CHR(2) + t_sCostCentre.tcUsrLogin,
                input "Special Type",
                input t_sCostCentre.CustomCombo0,
                input "E",
                input 3,
                input t_sCostCentre.tc_Rowid,
                input "CUST-001",
                input "",
                input "",
                input "",
                output viReturn).
        END.
    END.
END PROCEDURE.
```

1
2
3

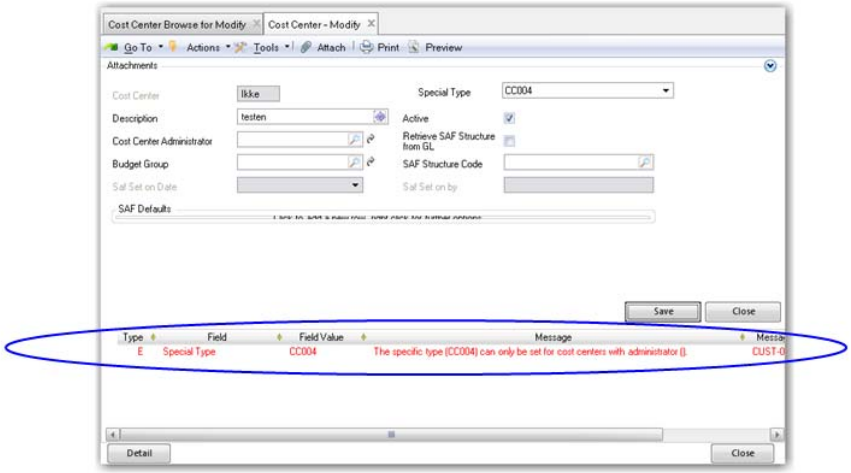
QAD

CUST_TOOLS_400

Demo Customization: Case 1 (7)

Demo Customization: Case 1

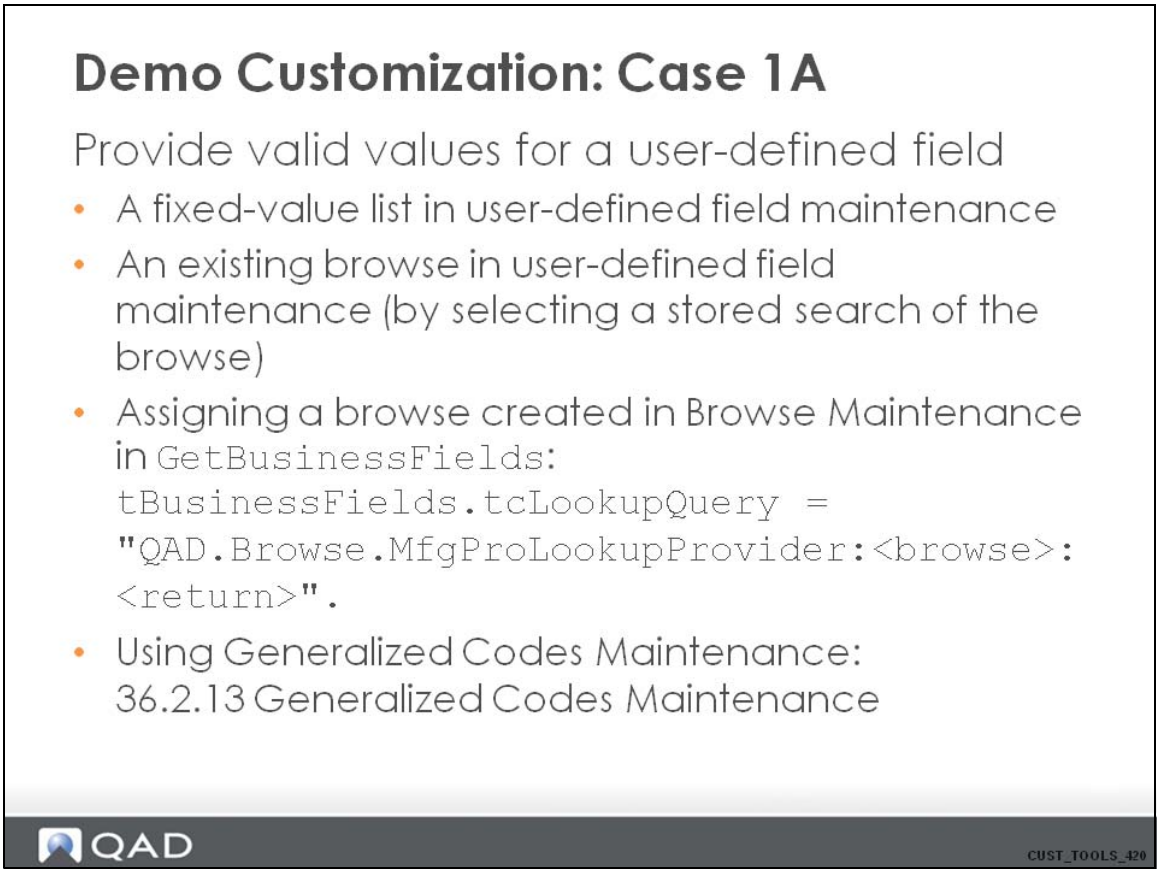
Compile, deploy customization, trim the AppServer, and run the Cost Center Create or Modify again



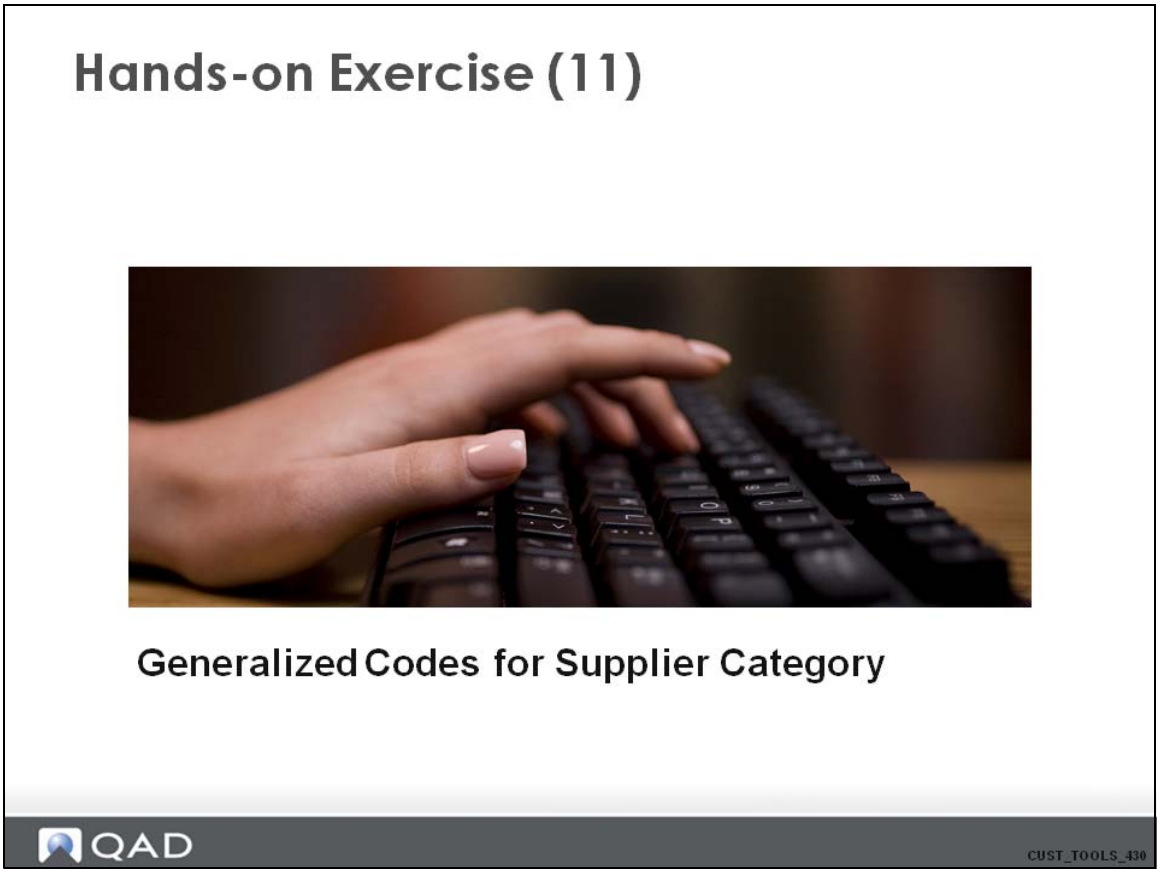
QAD

CUST_TOOLS_410

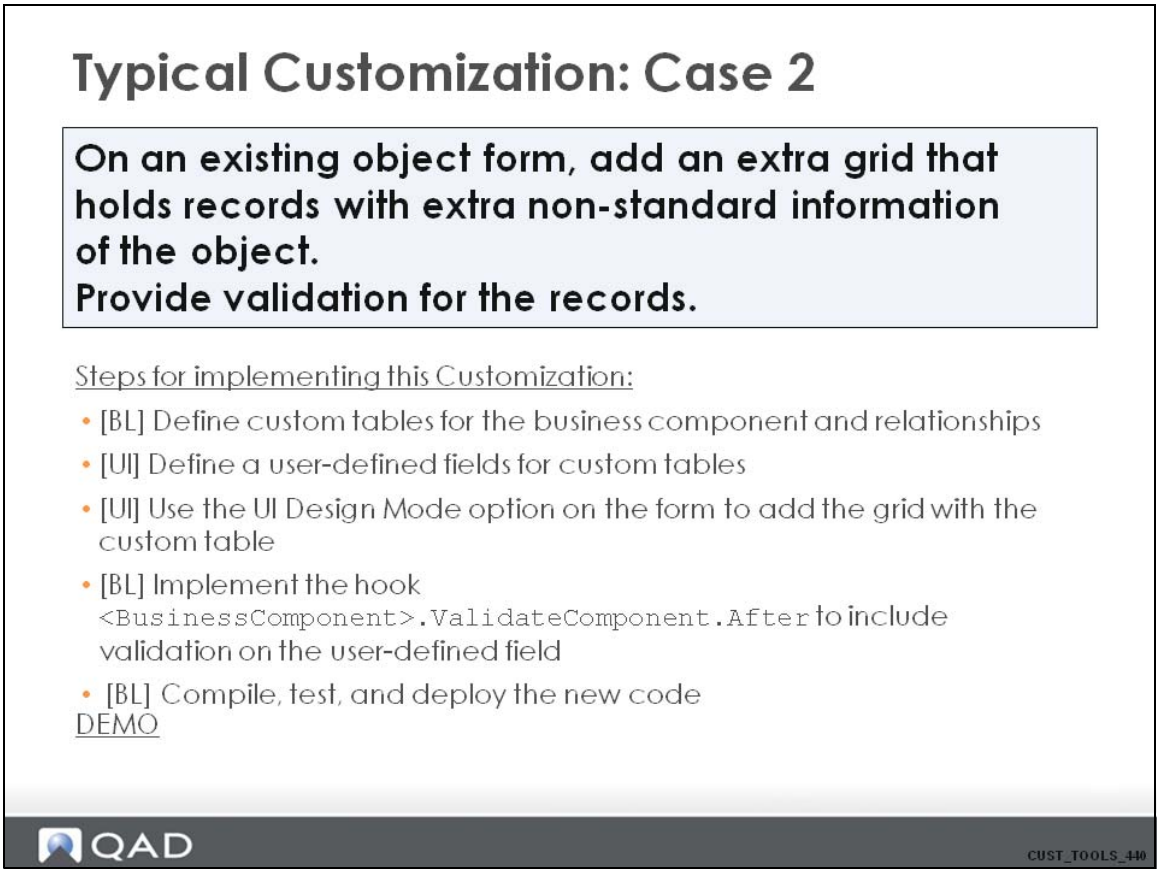
Demo Customization: Case 1A



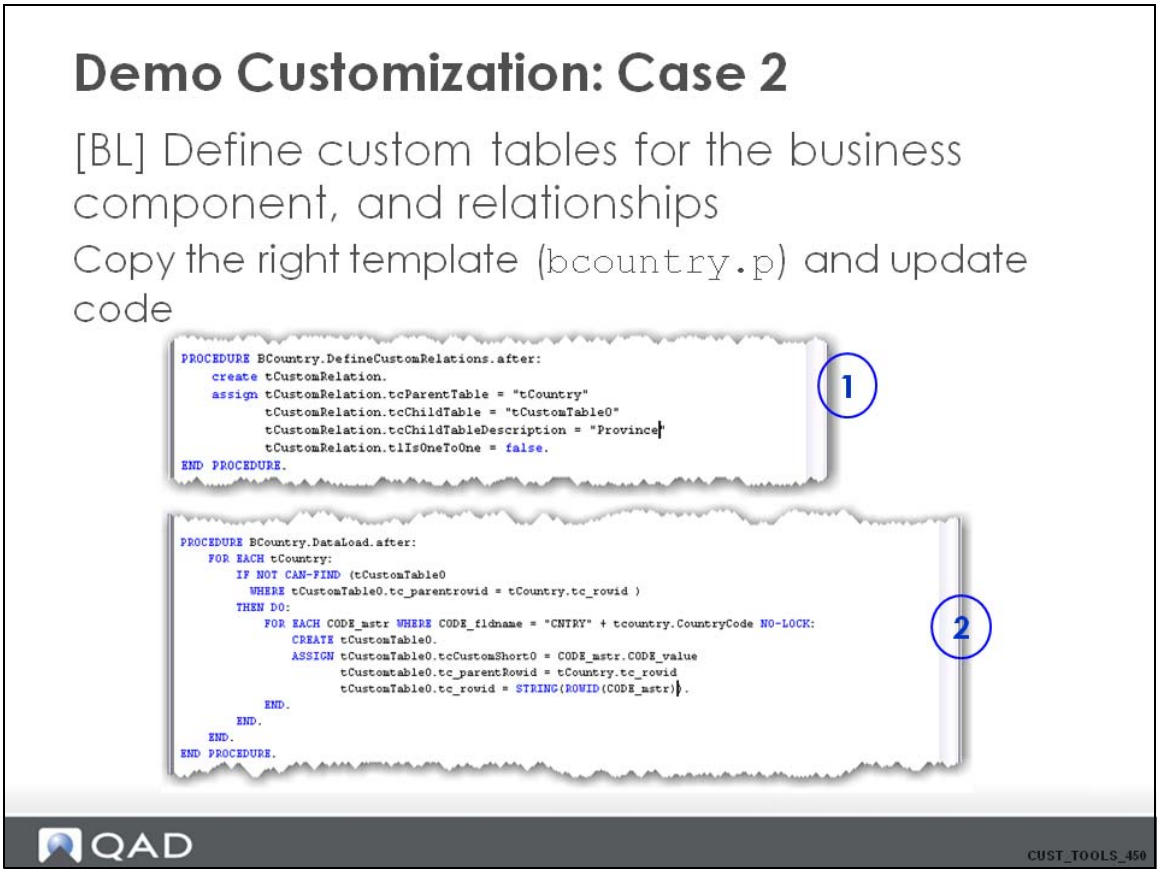
Hands-on Exercise (11)



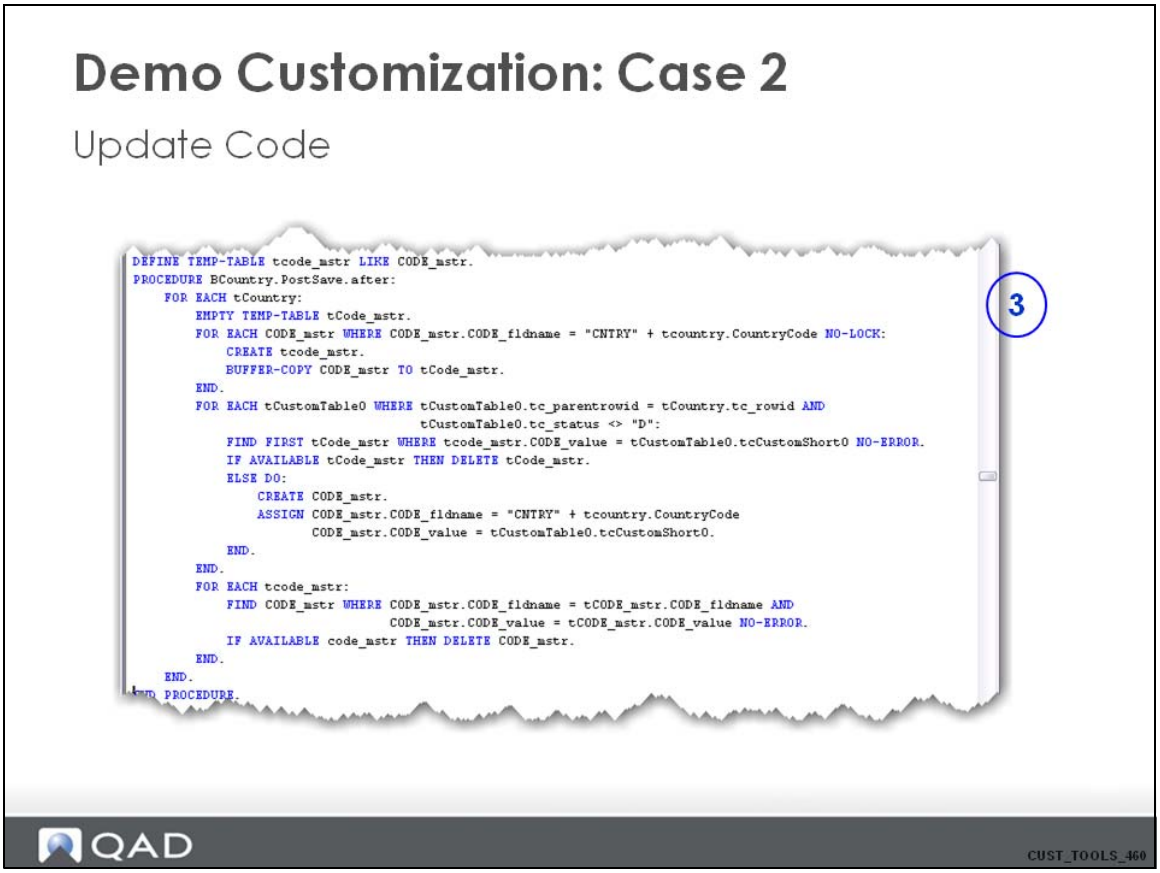
Typical Customization: Case 2



Demo Customization: Case 2



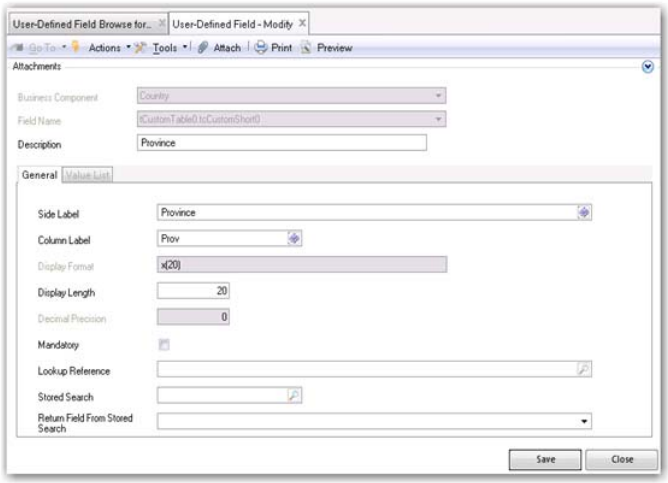
Demo Customization: Case 2 (2)



Demo Customization: Case 2 (3)

Demo Customization: Case 2

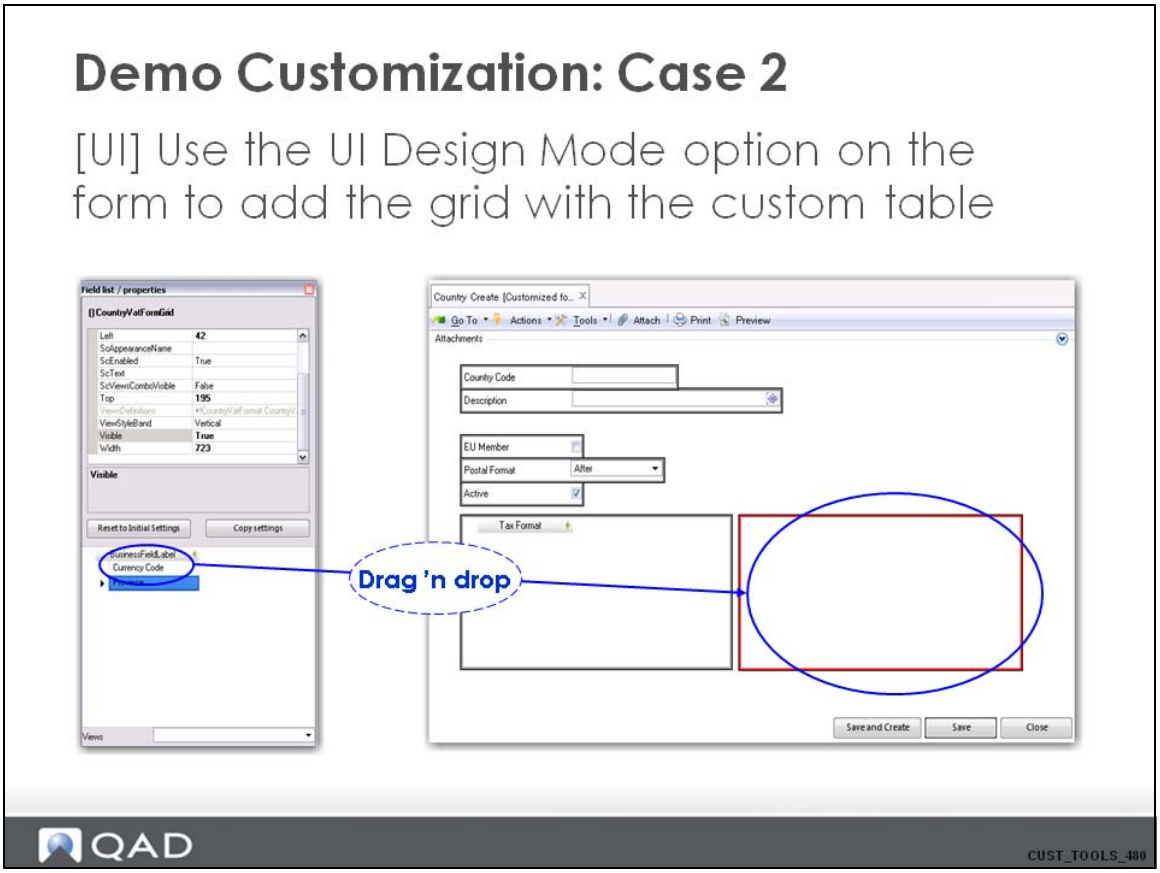
- Compile, deploy Customization, and trim the AppServer
- [UI] Define a user-defined fields for custom tables



QAD

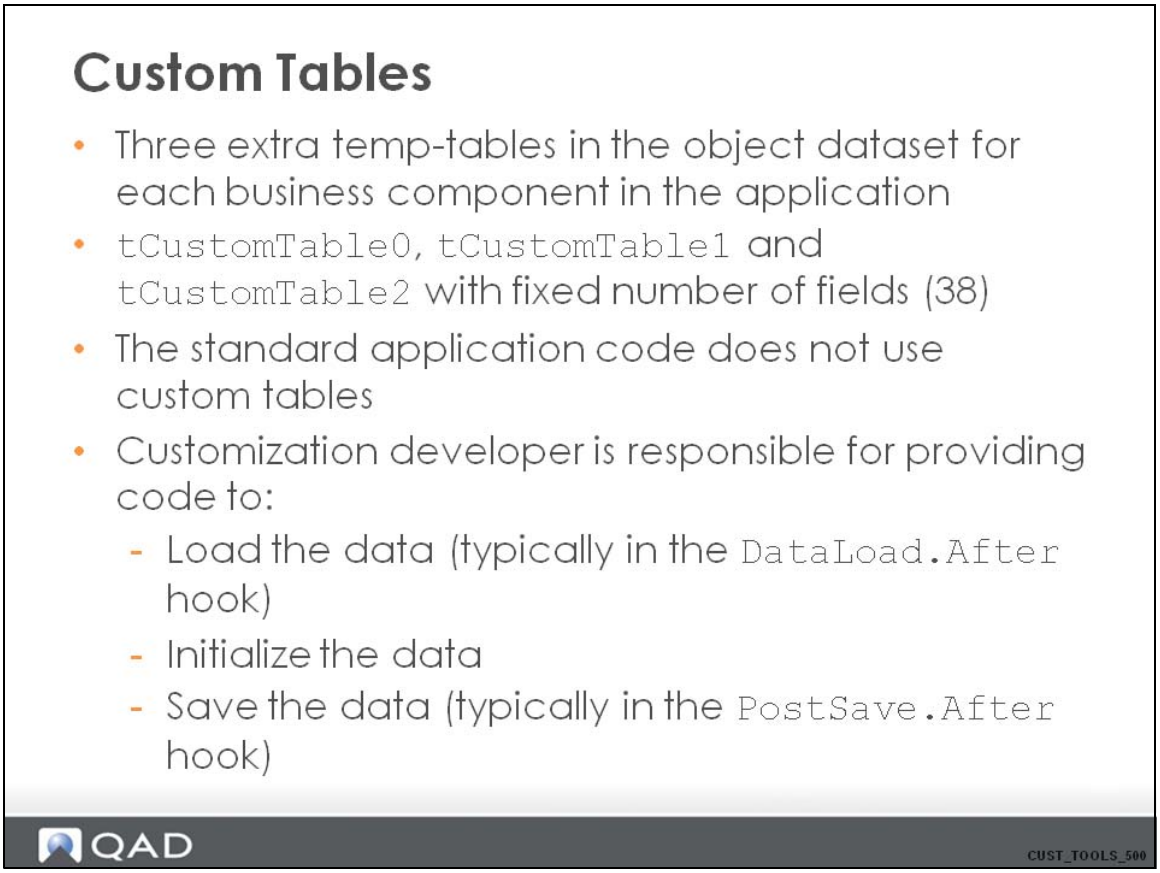
CUST_TOOLS_470

Demo Customization: Case 2 (4)



Demo Customization: Case 2 (5)

Custom Tables



Custom Tables 2

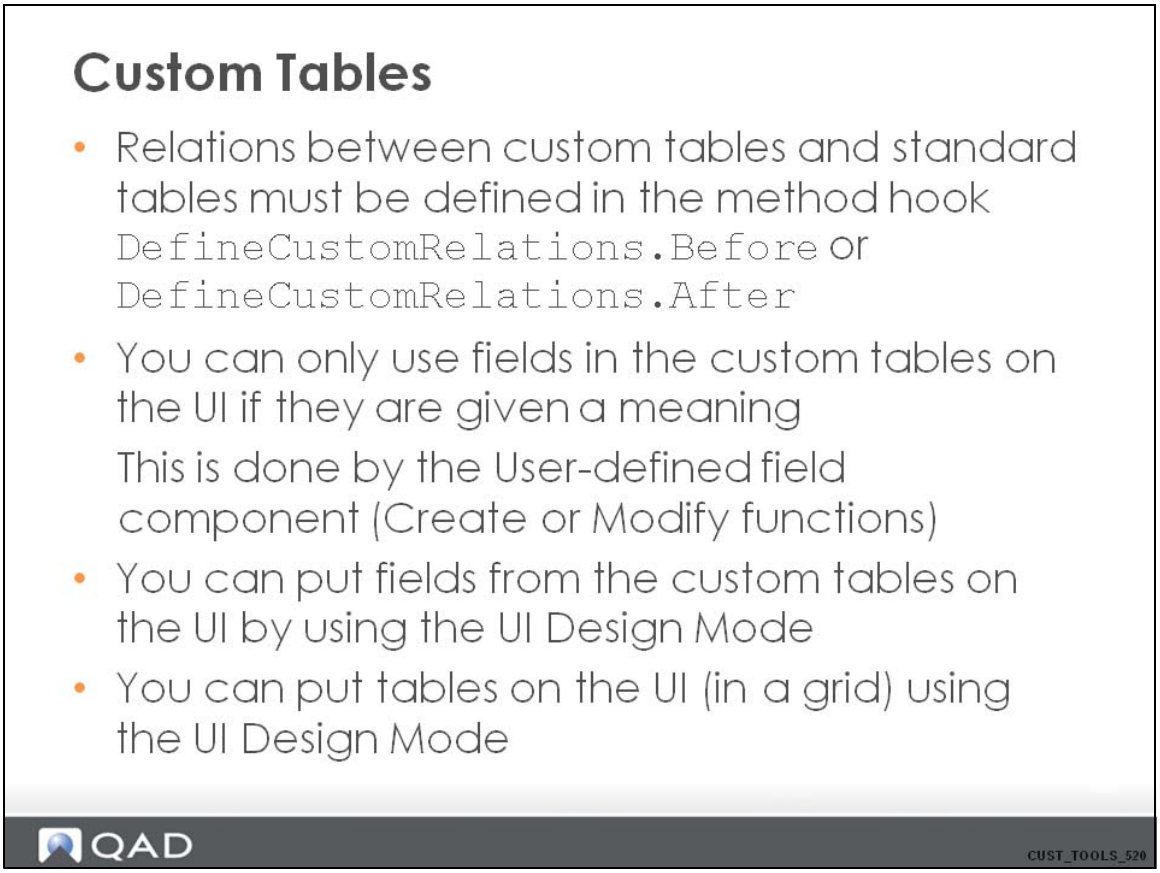
Custom Tables

- Three extra temp-tables:

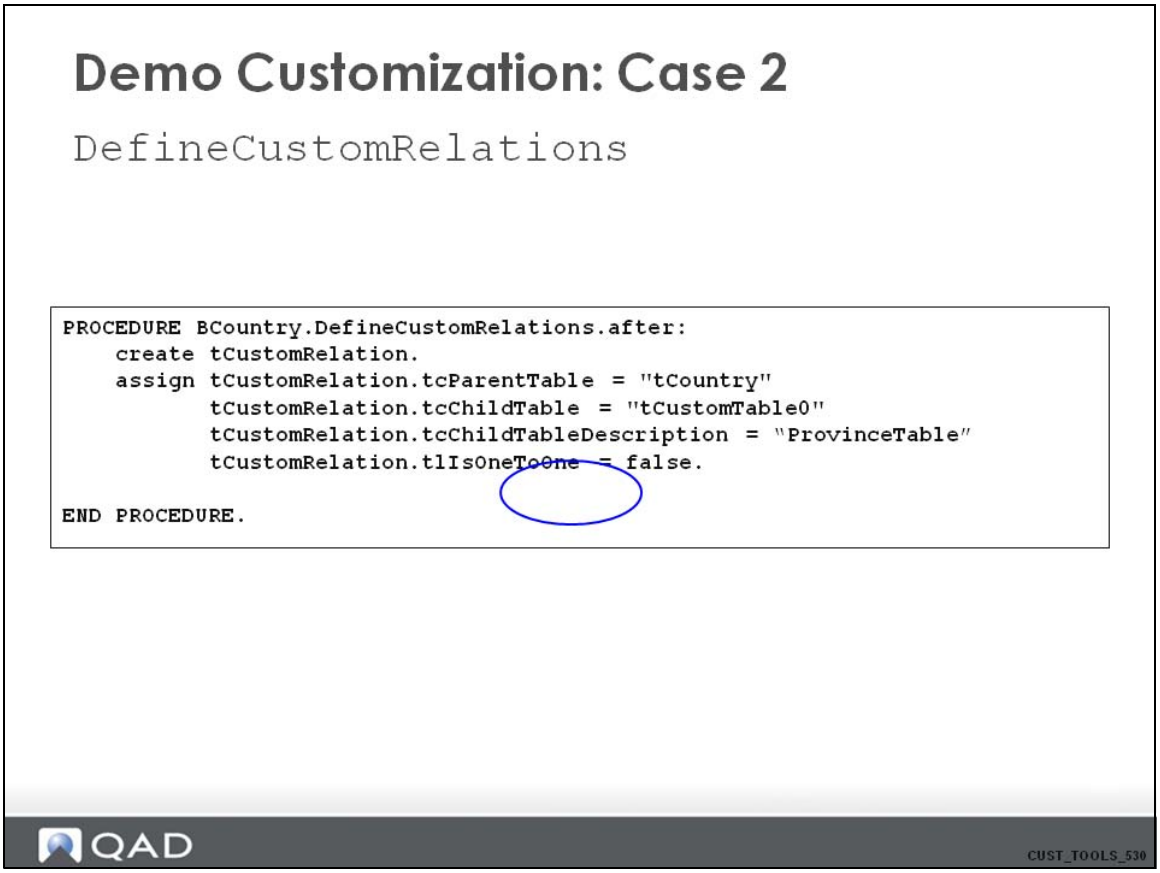
- Fixed list of fields:
  - tcCustomShort[0-9]
  - tcCustomCombo[0-9]
  - tcCustomLong[0-1]
  - ttCustomDate[0-4]
  - tiCustomInteger[0-4]
  - tdCustomDecimal[0-4]
  - tcCustomNote

QAD CUST_TOOLS_510

Custom Tables 3



Demo Customization: Case 2 - DefineCustomRelations



Demo Customization: Case 2 - DataLoad

Demo Customization: Case 2

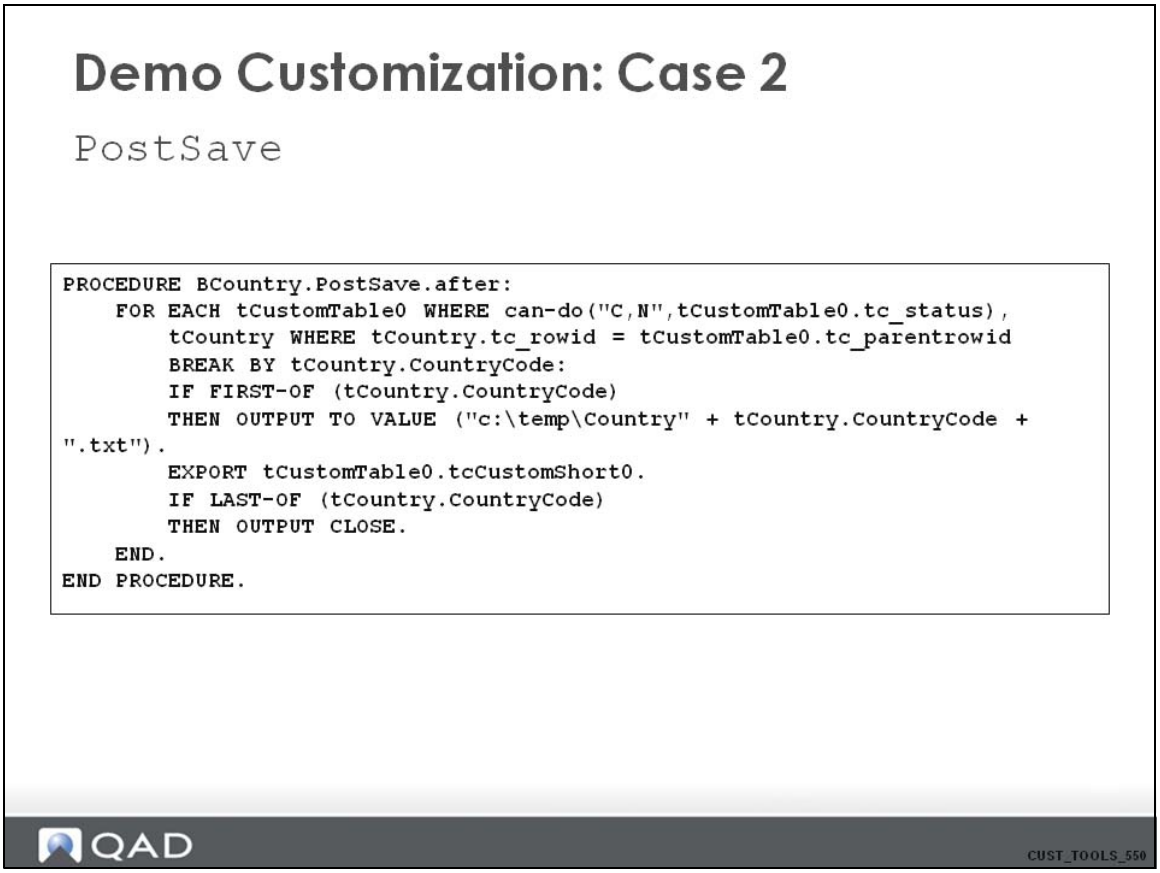
DataLoad

```
PROCEDURE BCountry.DataLoad.after:
    DEFINE VARIABLE vcProvince AS CHARACTER   NO-UNDO.
    FOR EACH tCountry:
        FILE-INFO:FILE-NAME = "c:\temp\Country" + tCountry.CountryCode +
".txt"
                              NO-ERROR.
        IF FILE-INFO:FILE-CREATE-DATE <> ?
        THEN DO:
            INPUT FROM VALUE ("c:\temp\Country" + tCountry.CountryCode +
".txt").
                REPEAT:
                    IMPORT vcProvince.
                    CREATE tCustomTable0.
                    ASSIGN tCustomTable0.tcCustomShort0 = vcProvince
                           tCustomTable0.tc_parentRowid = tCountry.tc_rowid
                           tCustomTable0.tc_rowid = "rowid" + string(TIME).
            END.
            INPUT CLOSE.
        END.
    END.
END PROCEDURE.
```

QAD

CUST_TOOLS_540

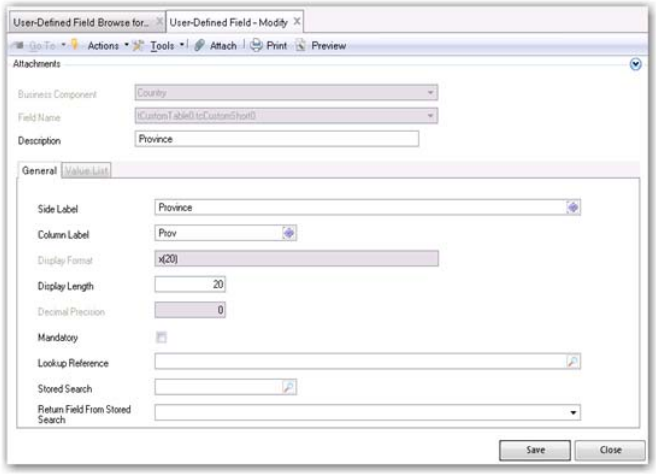
Demo Customization: Case 2 - PostSave



Demo Customization: Case 2 - Compile and Deploy

Demo Customization: Case 2

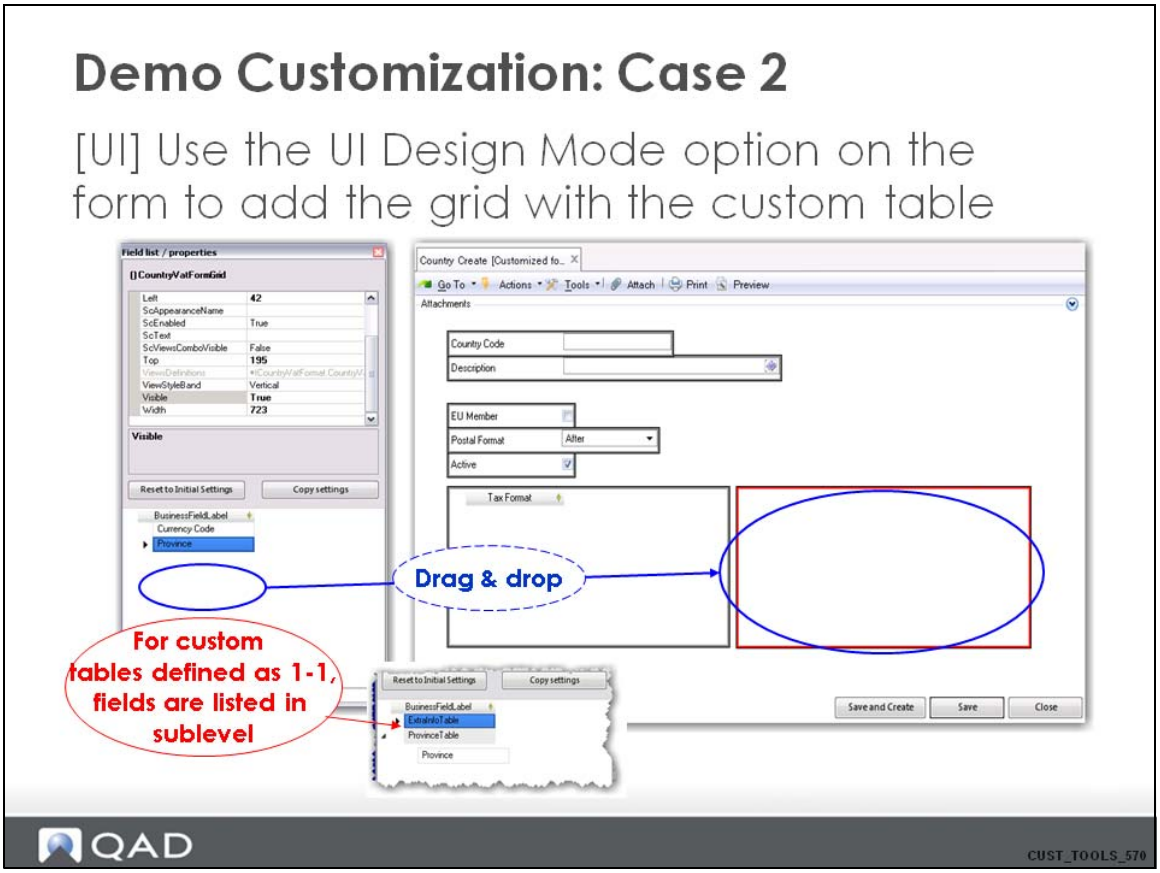
- Compile, deploy Customization, and trim the AppServer
- [UI] Define a user-defined field for custom tables



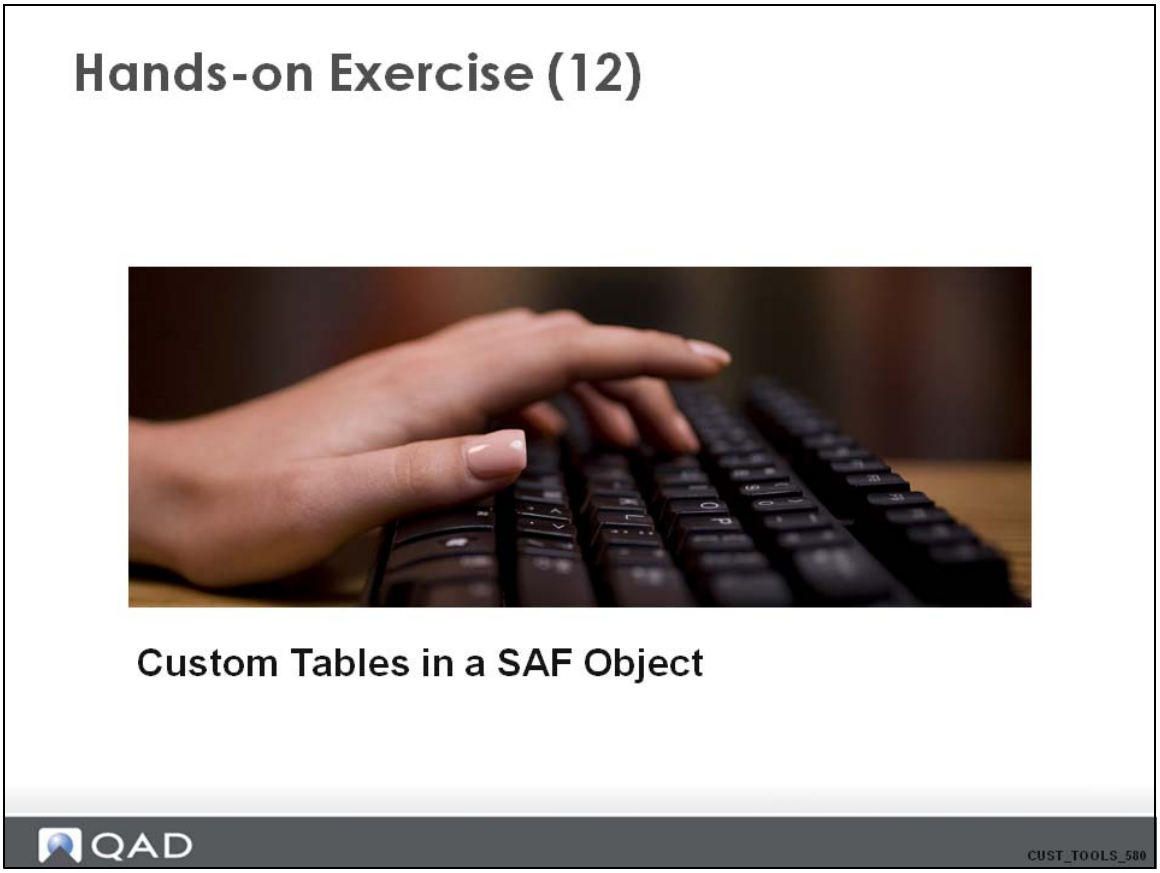
QAD

CUST_TOOLS_560

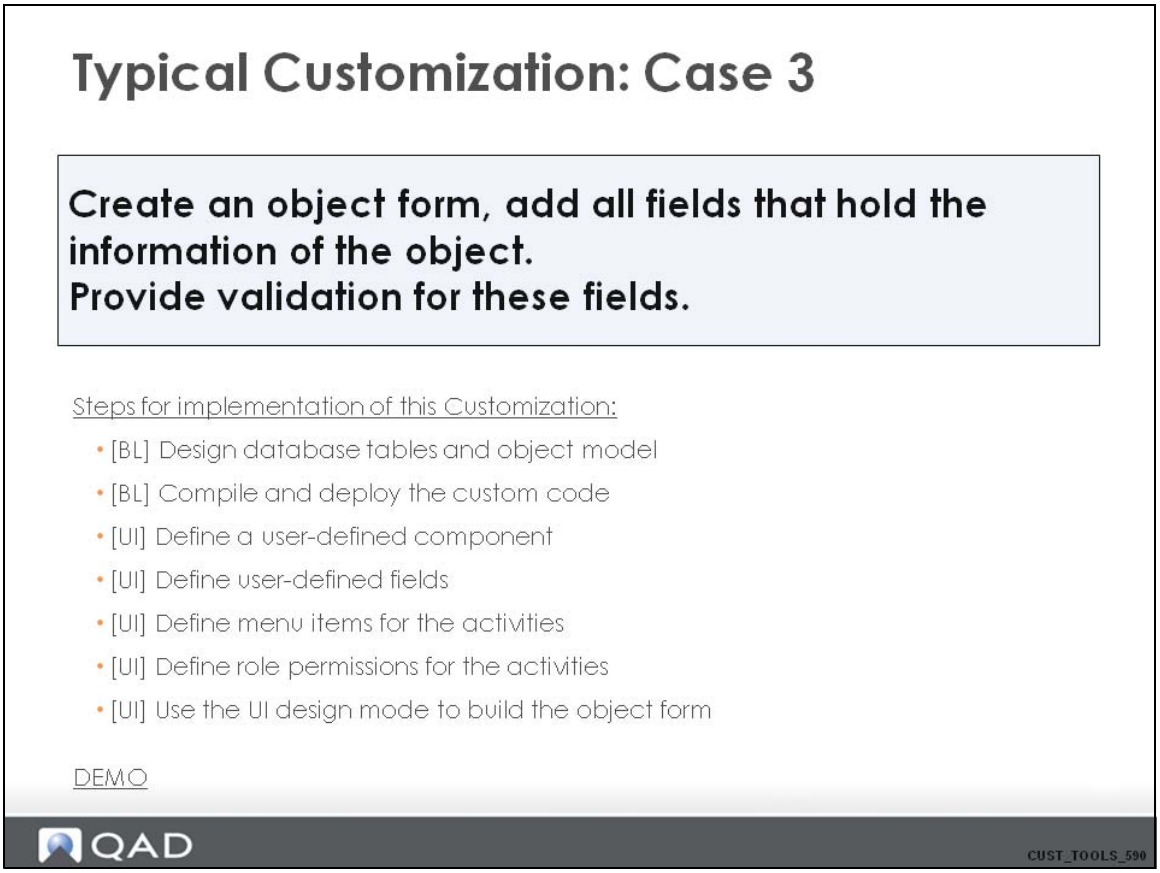
Demo Customization: Case 2 - Add Grid



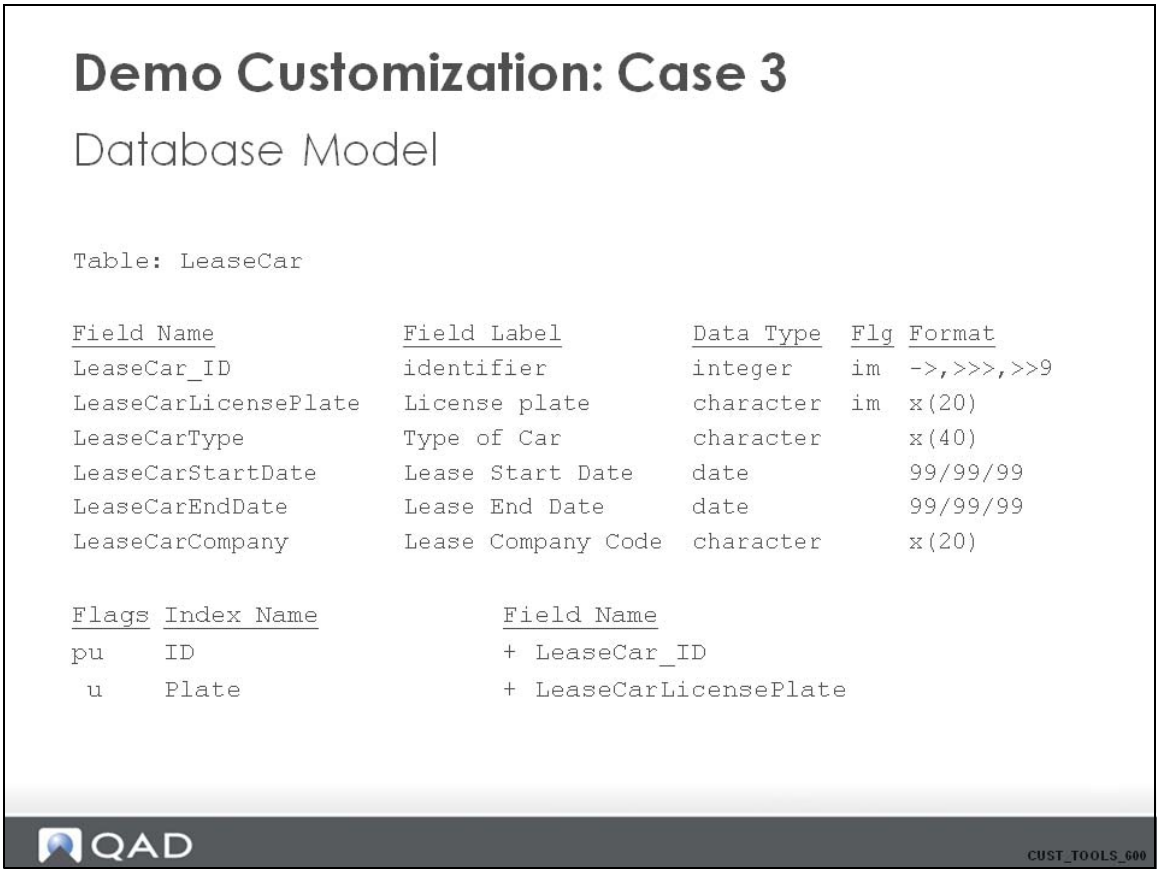
Hands-on Exercise (12)



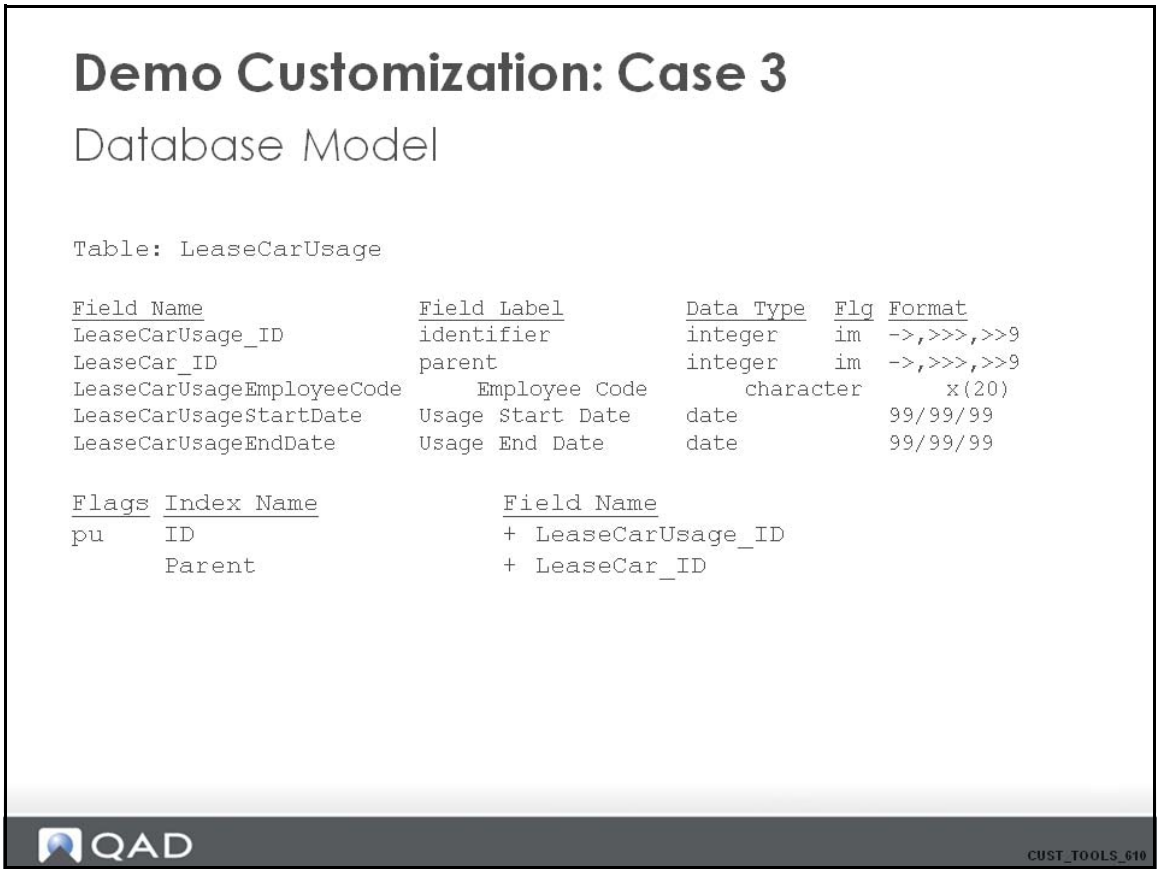
Typical Customization: Case 3



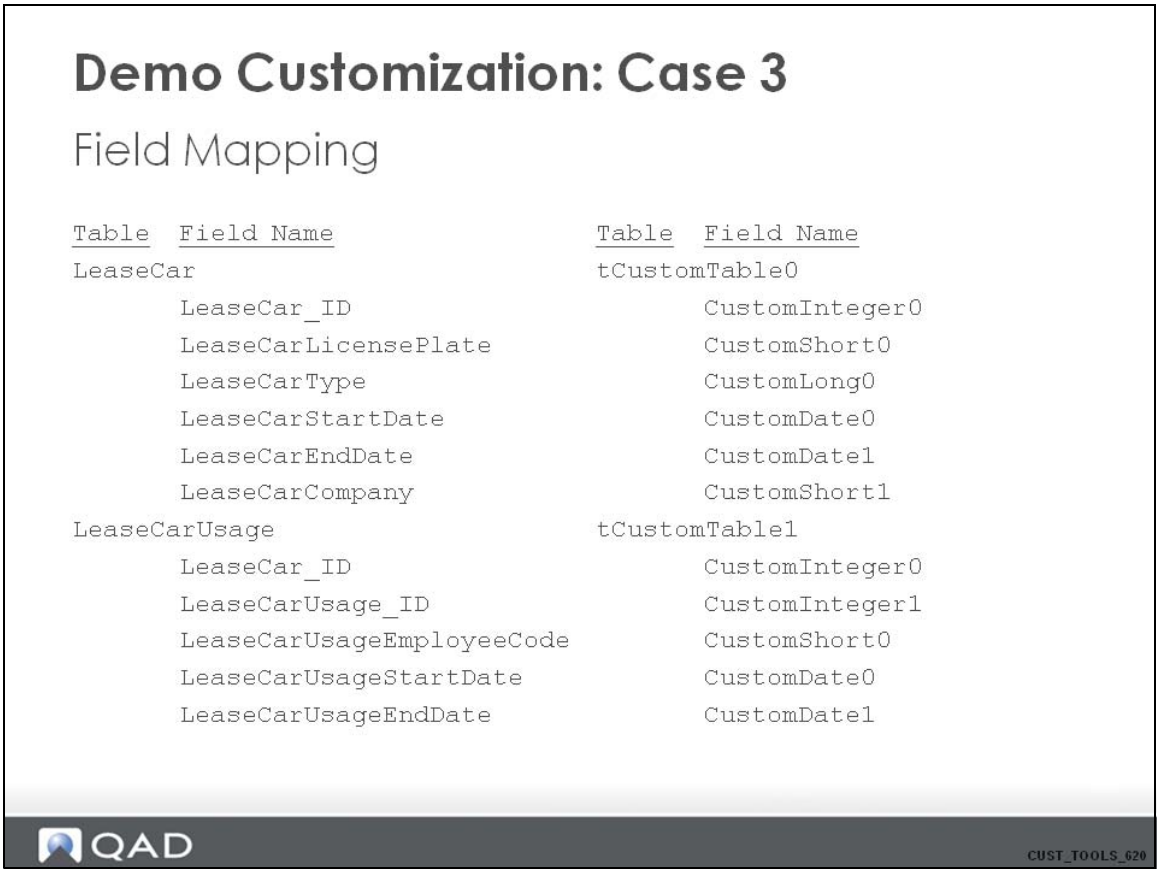
Demo Customization: Case 3 - Database Model



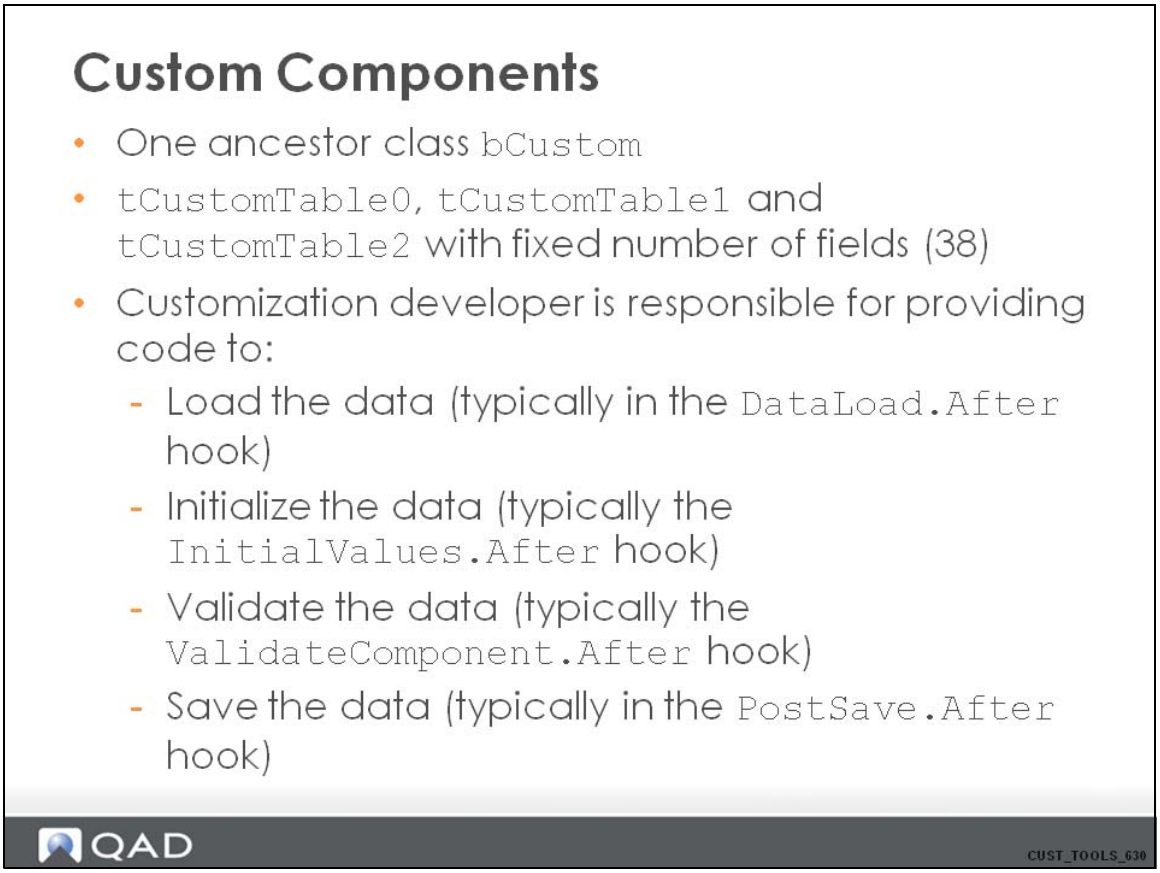
Demo Customization: Case 3 - Database Model 2



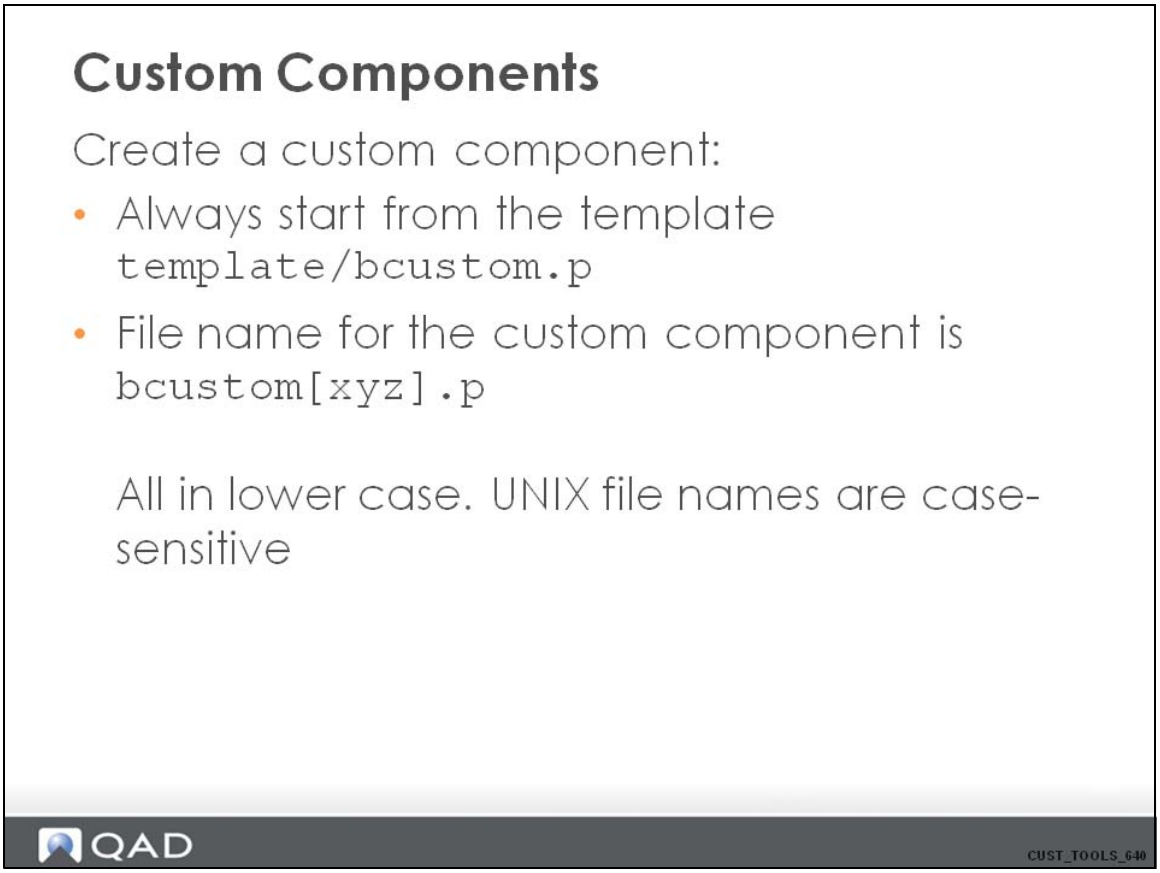
Demo Customization: Case 3 - Field Mapping



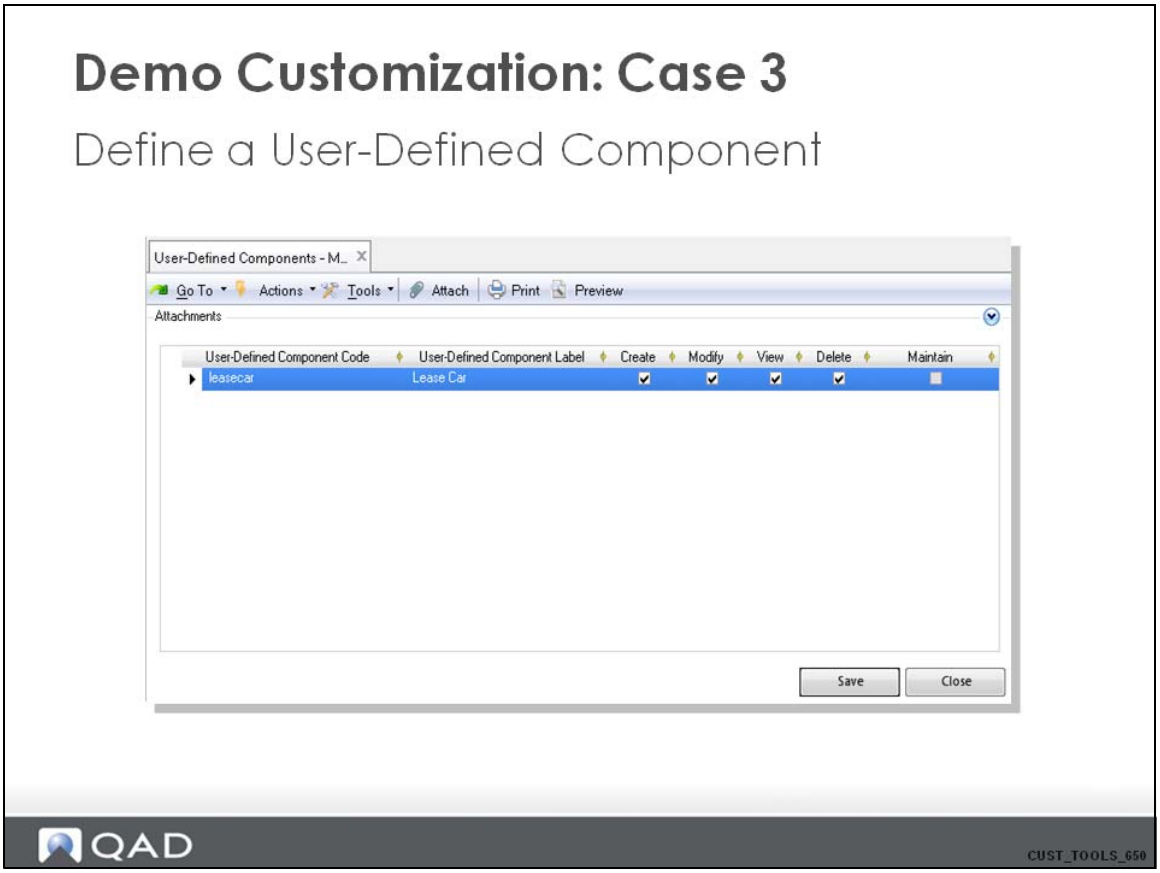
Custom Components



Custom Components 2

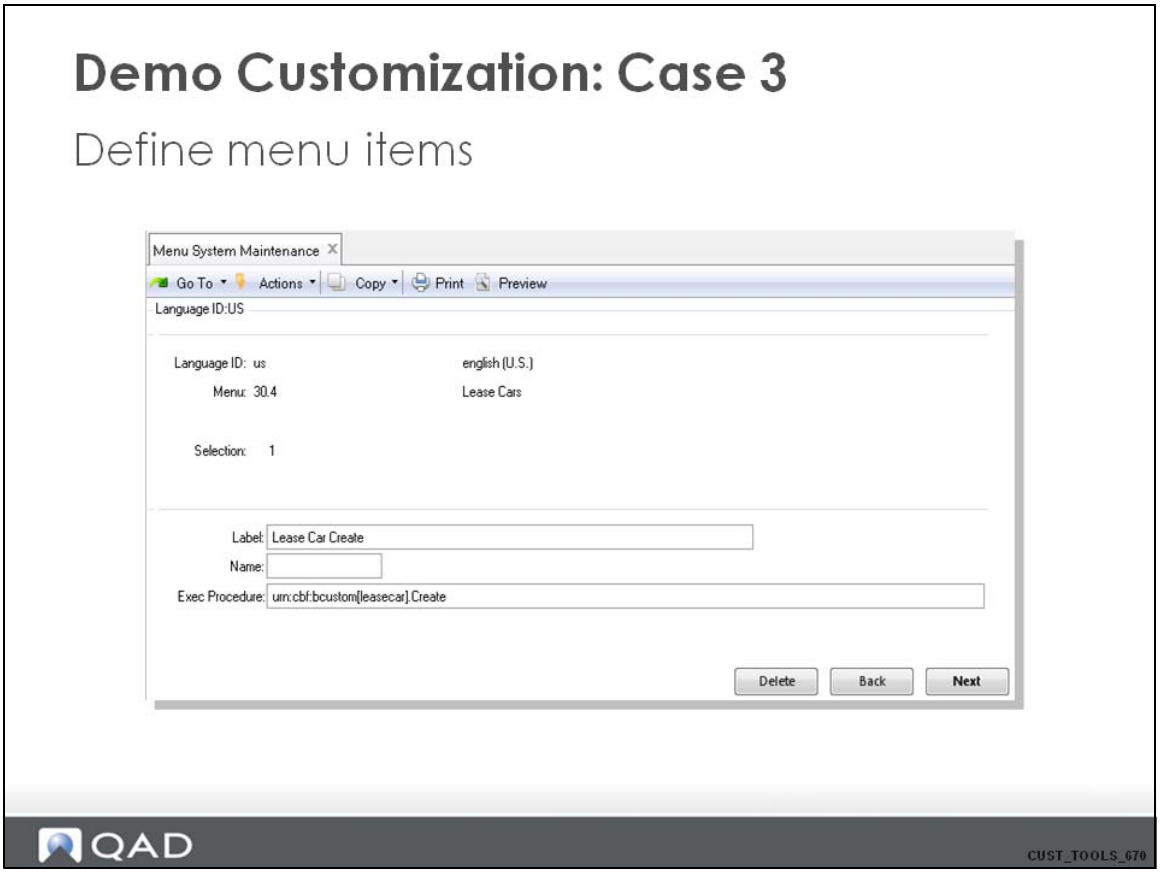


Demo Customization: Case 3 - Define User-Defined Component

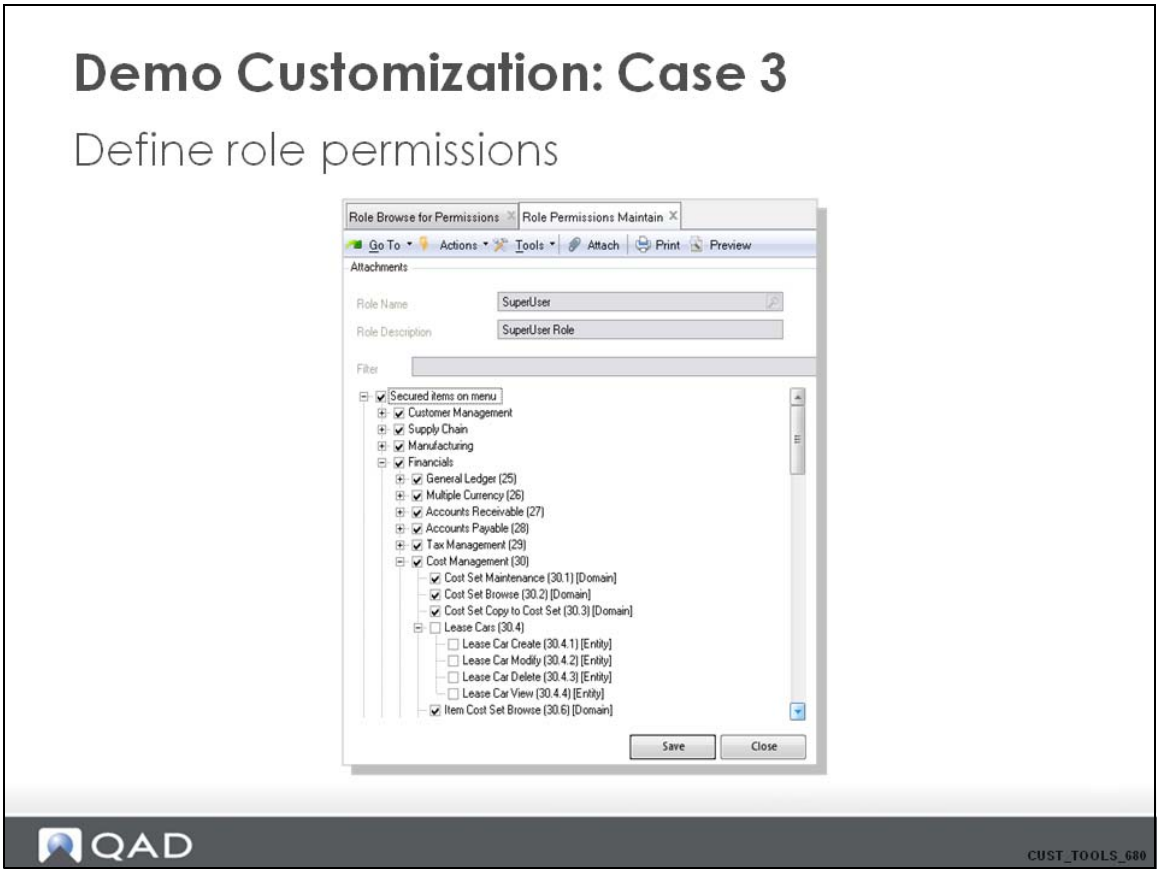


Demo Customization: Case 3 - Define User-Defined Fields

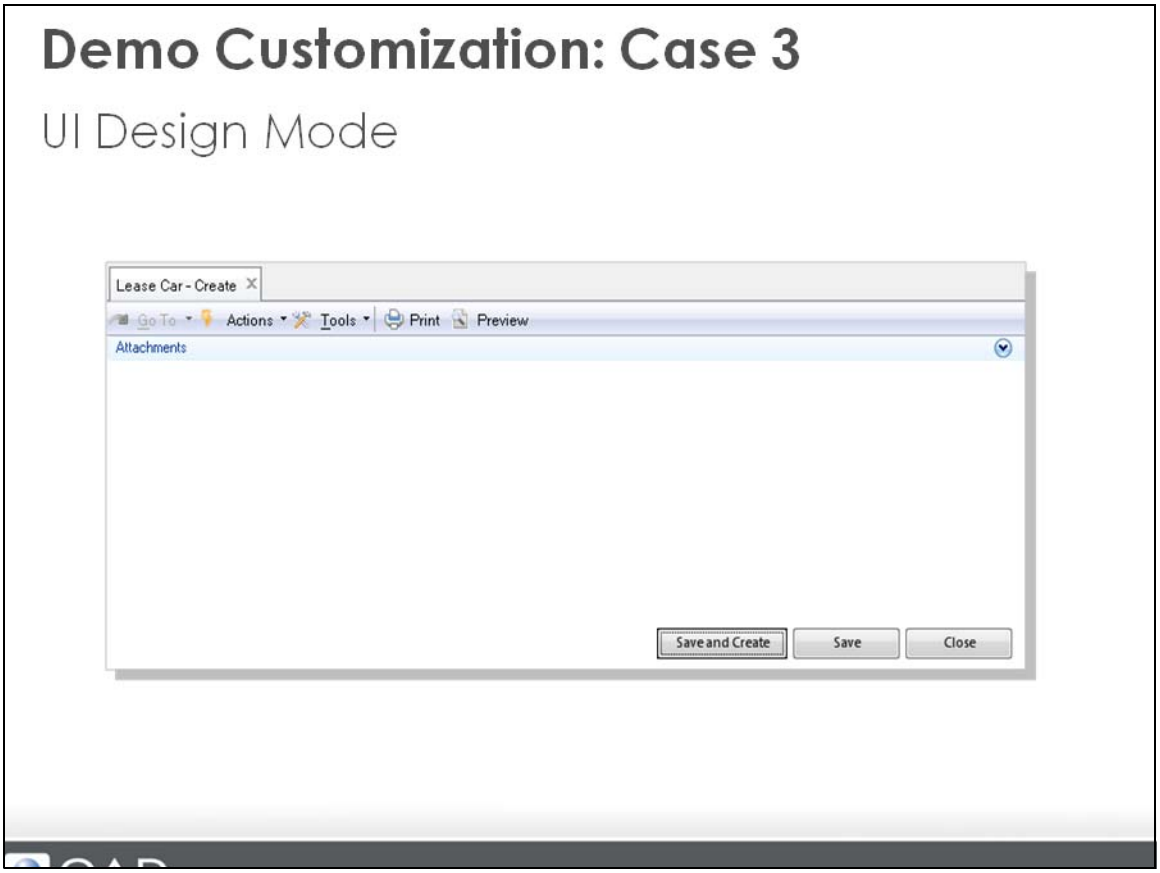
Demo Customization: Case 3 - Define Menu Items



Demo Customization: Case 3 - Define Role Permissions

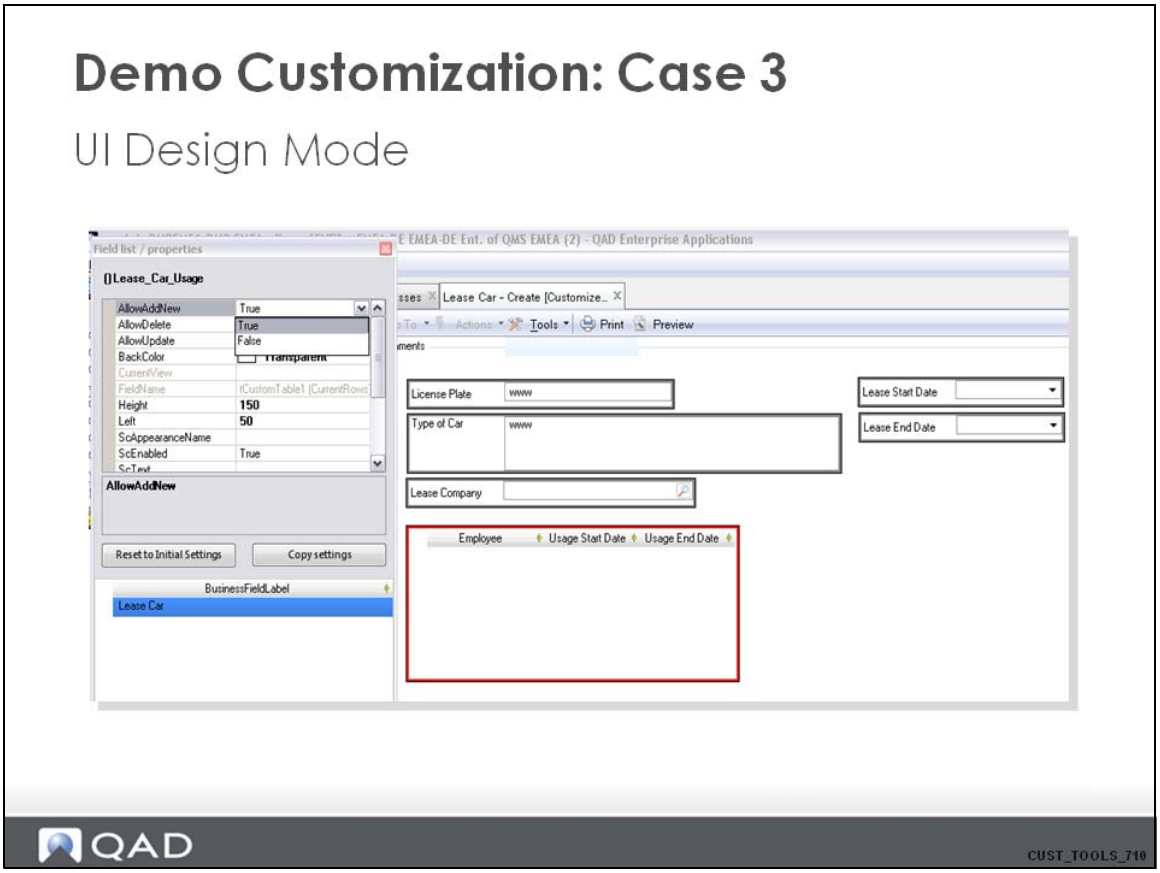


Demo Customization: Case 3 - UI Design Mode

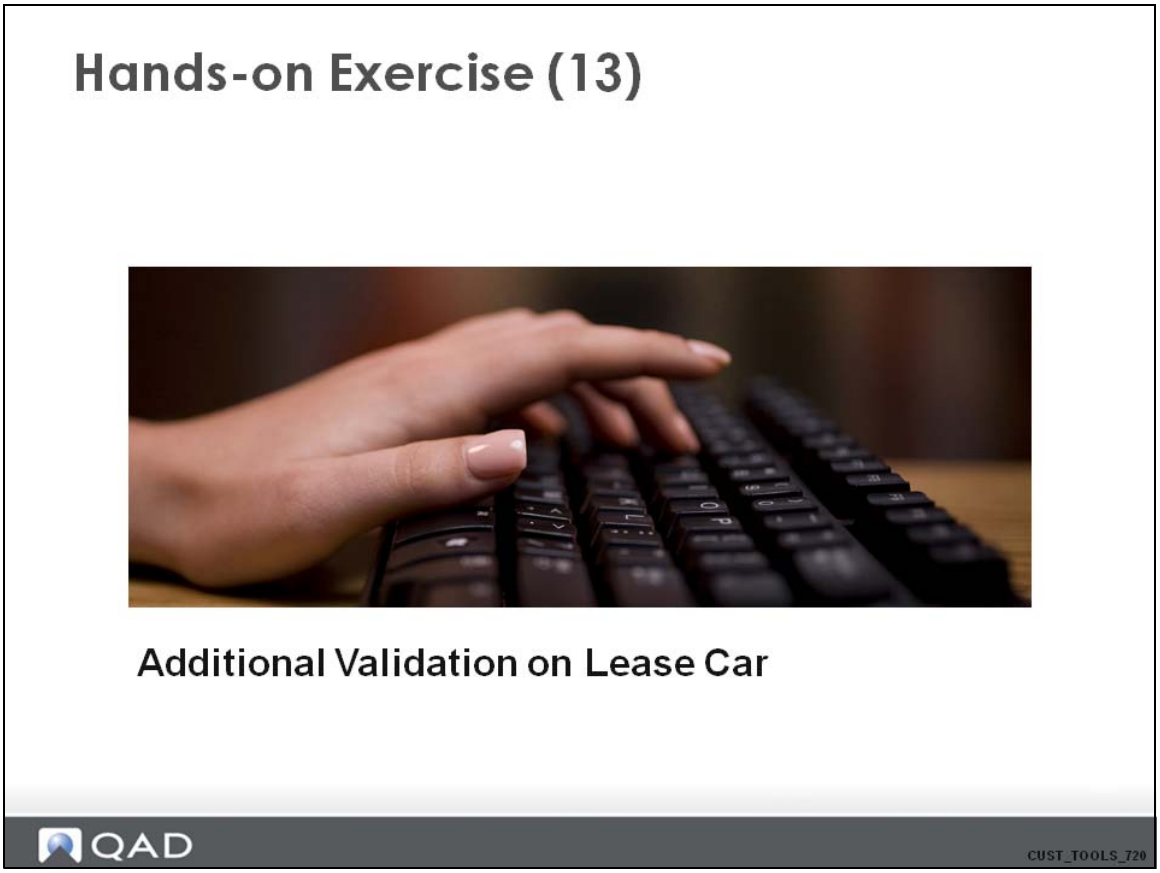


Demo Customization: Case 3 - UI Design Mode 2

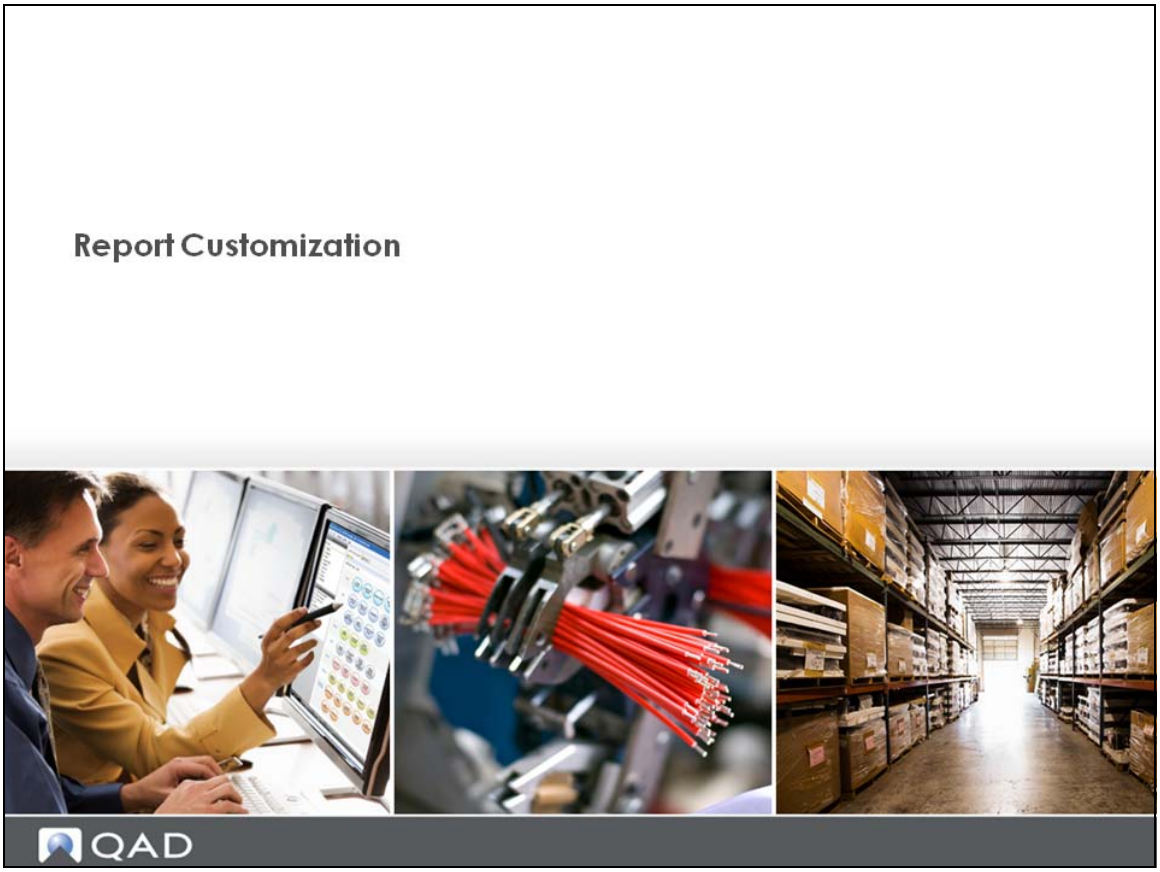
Demo Customization: Case 3 - UI Design Mode 3



Hands-on Exercise (13)



Report Customization



Report Framework for Financials Reports

Report Framework for Financials Reports 2

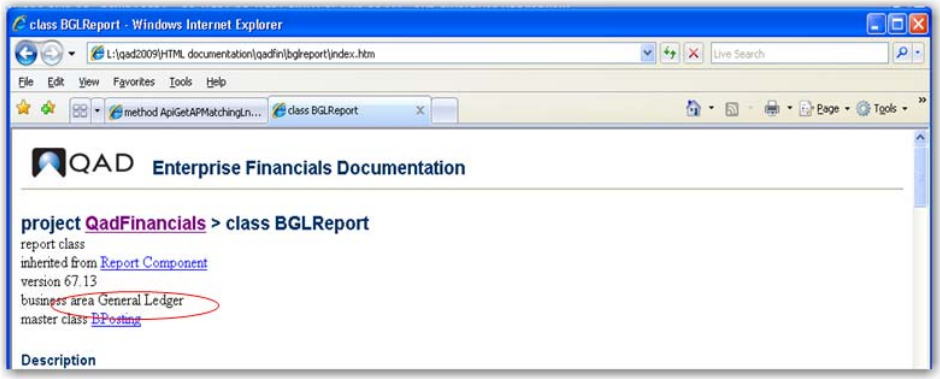
Report Framework for Financials Reports 3

Report Framework for Financials Reports

Report business component development

Inheritance from abstract foundation component

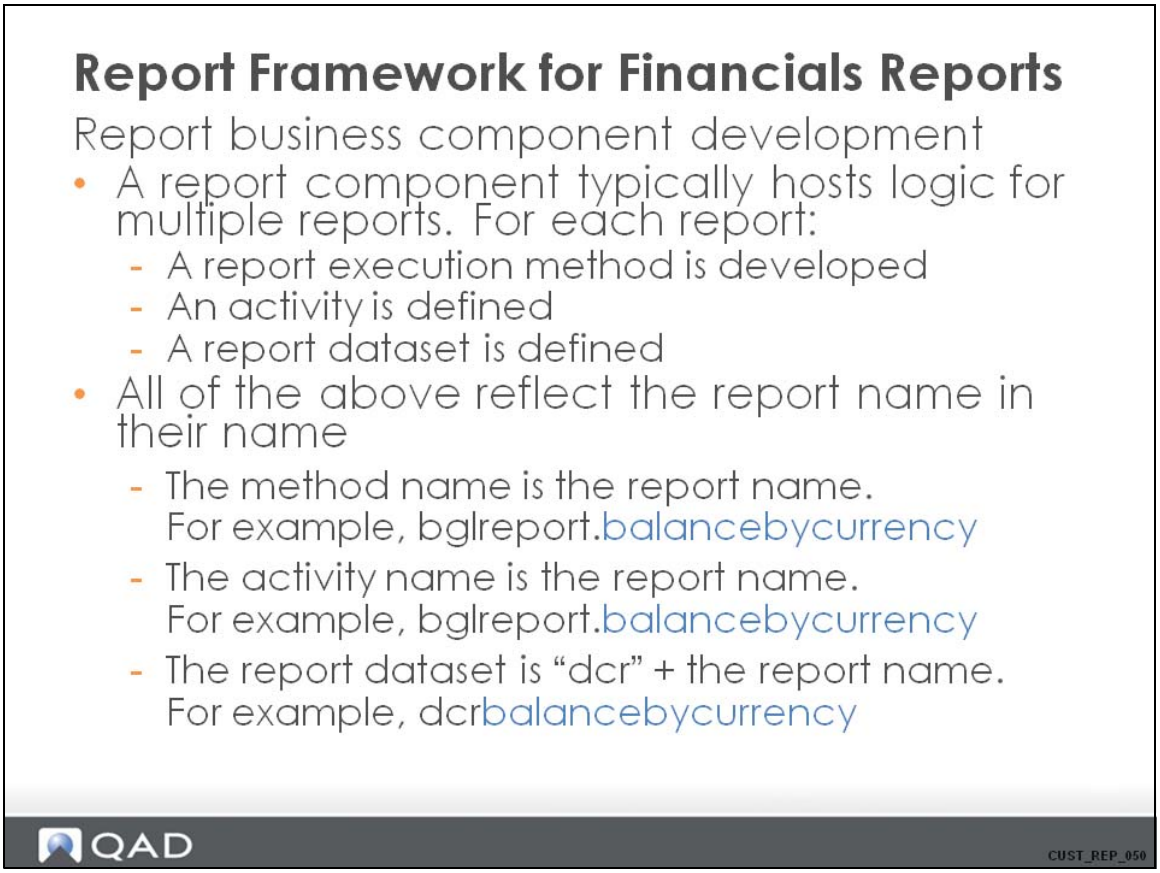
Report Component



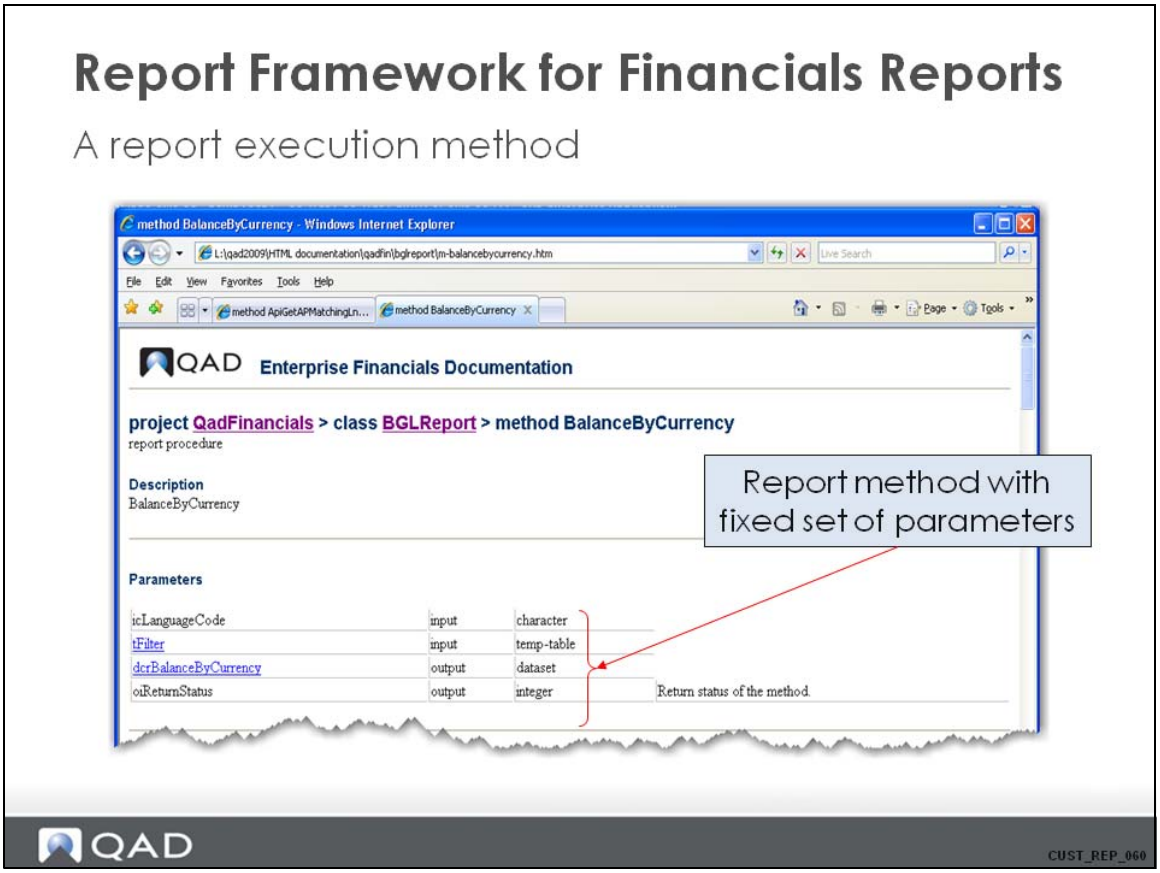
QAD

CUST_REP_040

Report Framework for Financials Reports 4



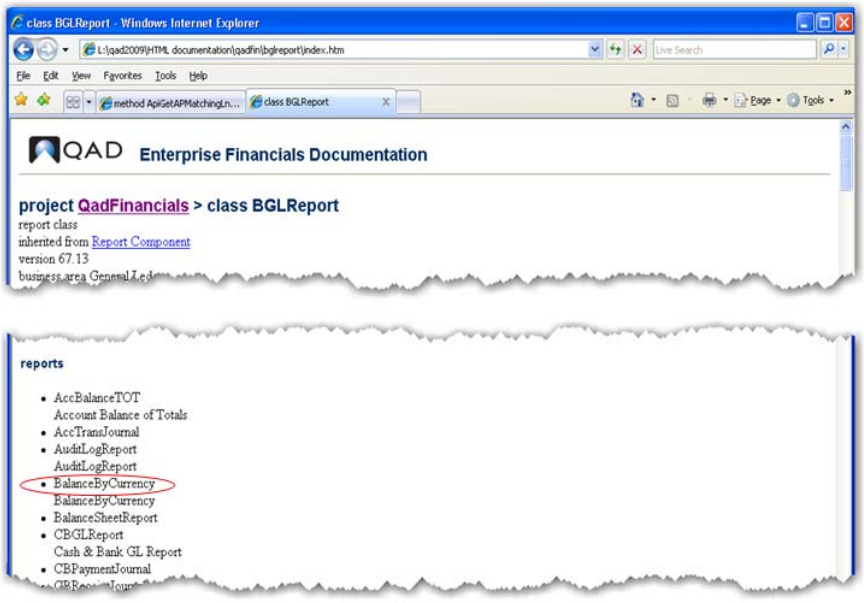
Report Framework for Financials Reports 5



Report Framework for Financials Reports 6

Report Framework for Financials Reports

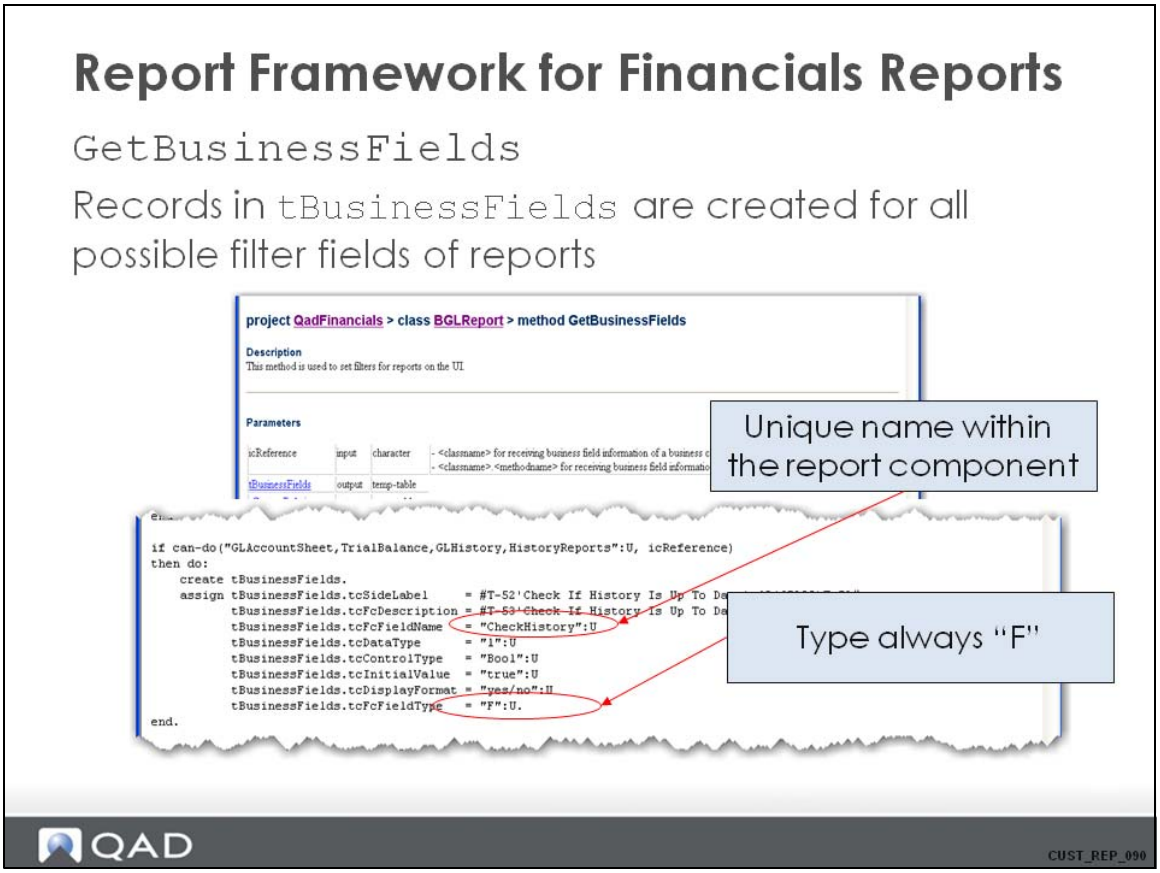
An activity



QAD CUST_REP_070

Report Framework for Financials Reports 7

Report Framework for Financials Reports 8



Report Framework for Financials Reports 9

Report Framework for Financials Reports

`SetDataItemsBasedOnFilterTT`

Read the records from the `tqFilter` and fill in the correct data items (variables)

project QadFinancials > class BGLReport > method SetDataItemsBasedOnFilterTT

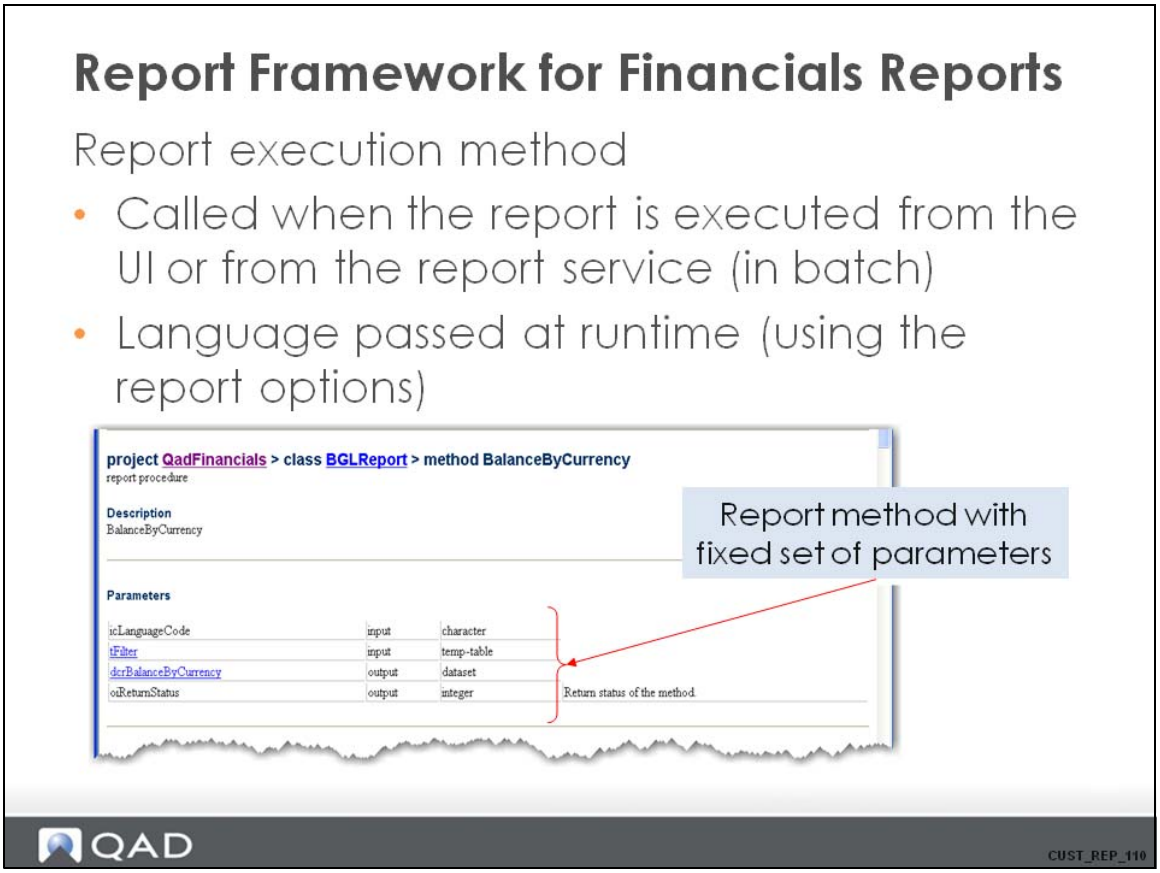
Description

```
for each tFilter:
    case tFilter.tcBusinessFieldName:
        when "SortBy":U
        then do:
            if tFilter.tcParameterValue = "Posting Date":U
            then assign vcSortByFilter = "1" no-error.

            if tFilter.tcParameterValue = "Sequence Number":U
            then assign vcSortByFilter = "2" no-error.
        end.

        when "AccYear":U
        then assign viFromAccYearFilter = int(tFilter.tcParameterValue) no-error.

        when "AccPeriod":U
        then assign viFromAccPeriodFilter = int(tFilter.tcParameterValue) no-error.
```

QAD

CUST_REP_100

Report Framework for Financials Reports 10

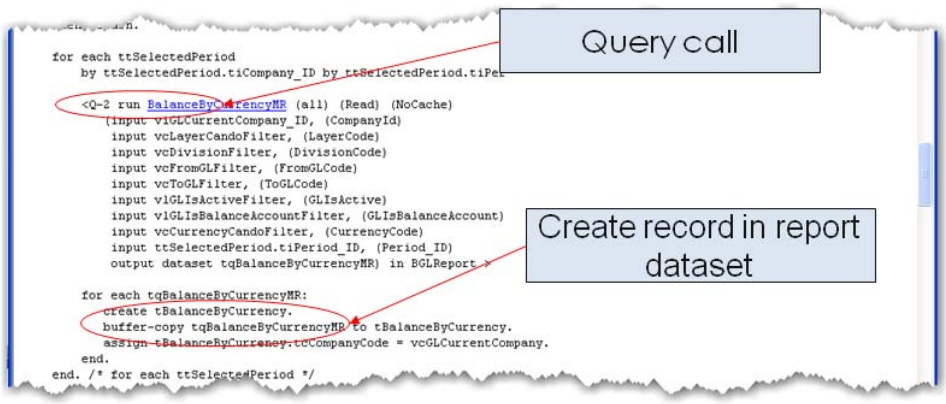


Report Framework for Financials Reports 11

Report Framework for Financials Reports

Report execution method

Typically is code that calls one or more queries and based on that data populates the report dataset



QAD CUST_REP_120

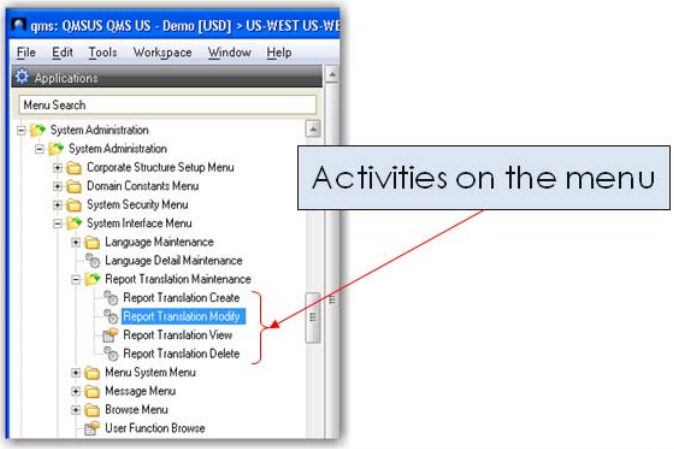
Report Framework for Financials Reports 12

Report Framework for Financials Reports

Languages and Translations

`BReportTranslation` component

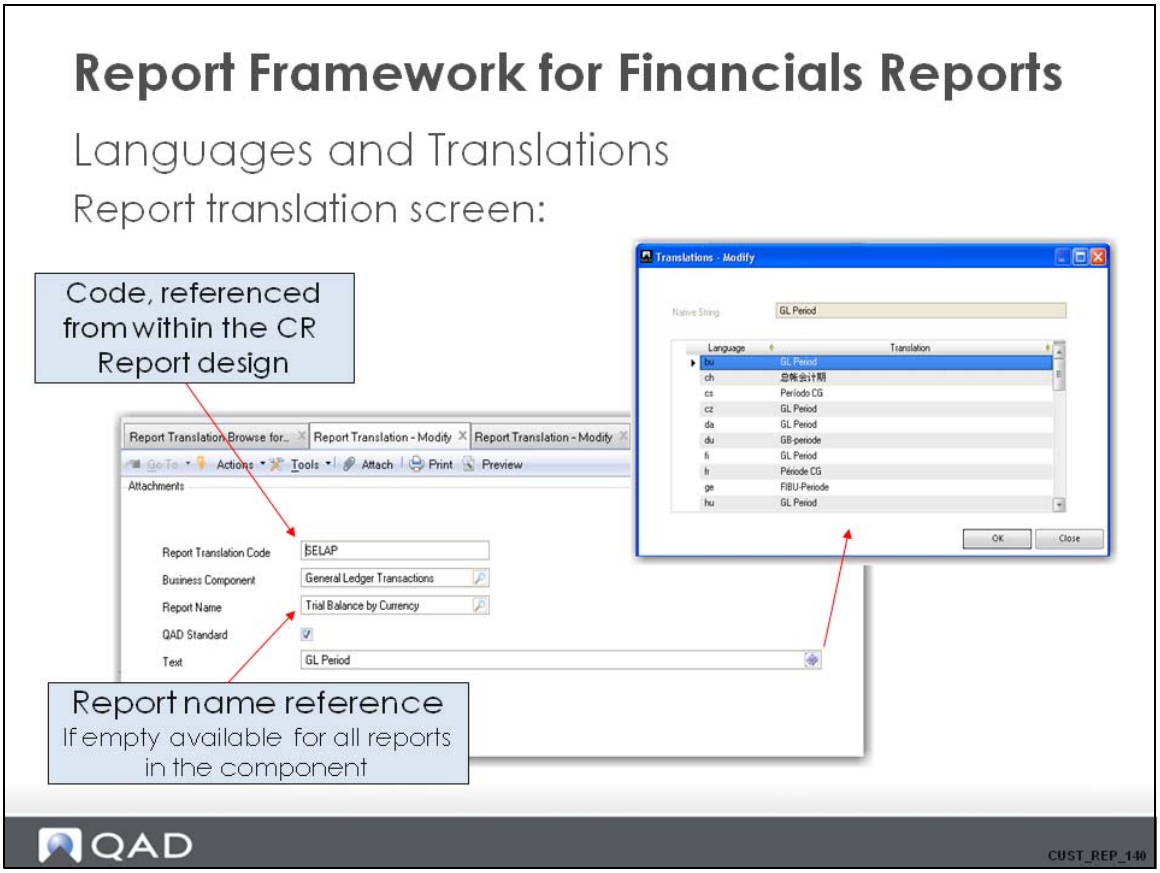
- This component is used to maintain the labels and translations on specific reports
- Menu items:



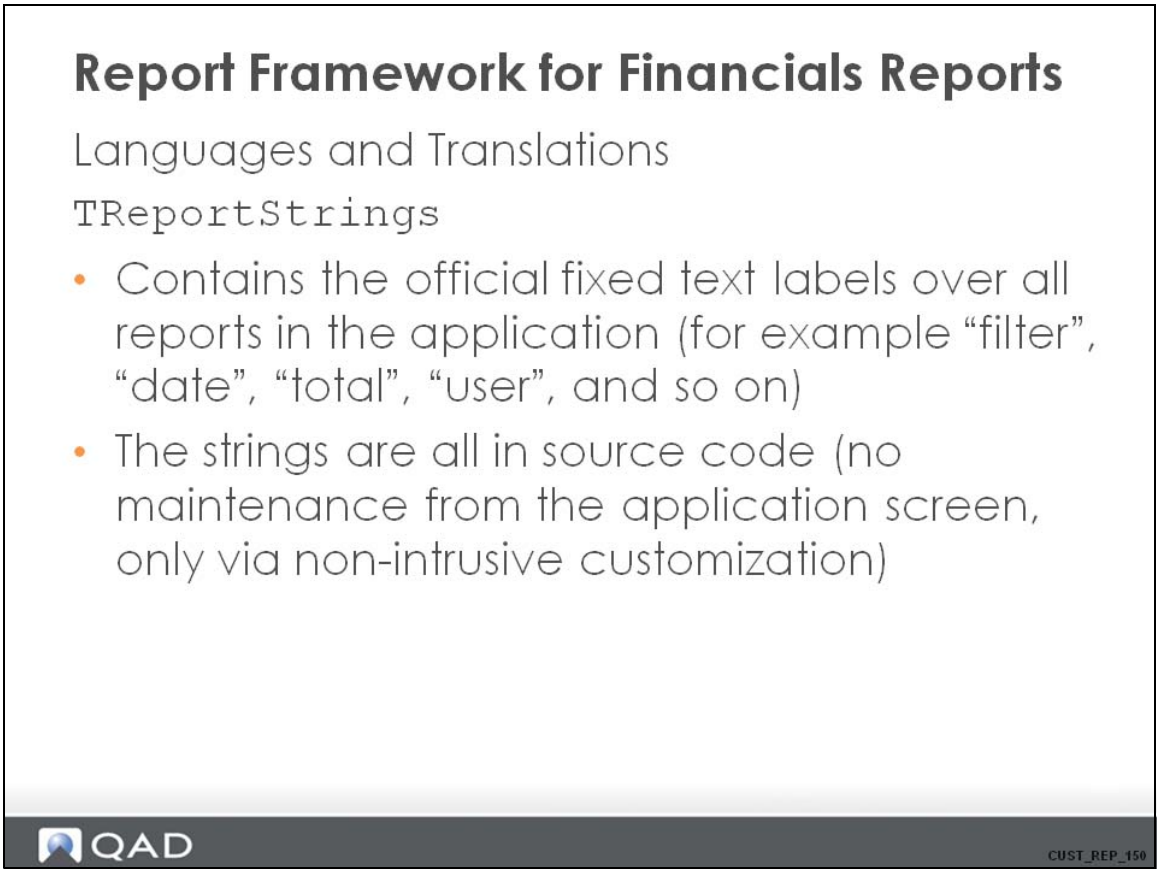
QAD

CUST_REP_130

Report Framework for Financials Reports 13



Report Framework for Financials Reports 14



Report Framework for Financials Reports 15

Report Framework for Financials Reports

Languages and Translations

- `TReportStrings.MainBlock` method contains the code
- Translation is fixed and comes with the QAD application installation
- Strings are referenced from Crystal reports in a similar way as the specific report labels

```
create tReportStrings.
assign tReportStrings.tcStringText = #T-3'Date':10(358)t-3#
       tReportStrings.tcStringCode = "O-DATE":U.

create tReportStrings.
assign tReportStrings.tcStringText = #T-4'DB':2(359)t-4#
       tReportStrings.tcStringCode = "O-DB":U.

create tReportStrings.
assign tReportStrings.tcStringText = #T-5'End of Report':20(360)t-5#
       tReportStrings.tcStringCode = "O-EOR":U.

create tReportStrings.
assign tReportStrings.tcStringText = #T-6'Filter Values for this Report':100(361)T-6#
       tReportStrings.tcStringCode = "O-FILTER":U.

create tReportStrings.
assign tReportStrings.tcStringText = #T-7'No Data':20(362)t-7#
       tReportStrings.tcStringCode = "O-ND":U.

create tReportStrings.
assign tReportStrings.tcStringText = #T-8'Page':10(363)t-8#
```

QAD

CUST_REP_160

Report Framework for Financials Reports 16

Report Framework for Financials Reports

For all of the strings defined in `TReportStrings` component, do the following:

1. System Synchronize (36.24.3.2) for Report Strings
2. Log out from the application and log in again

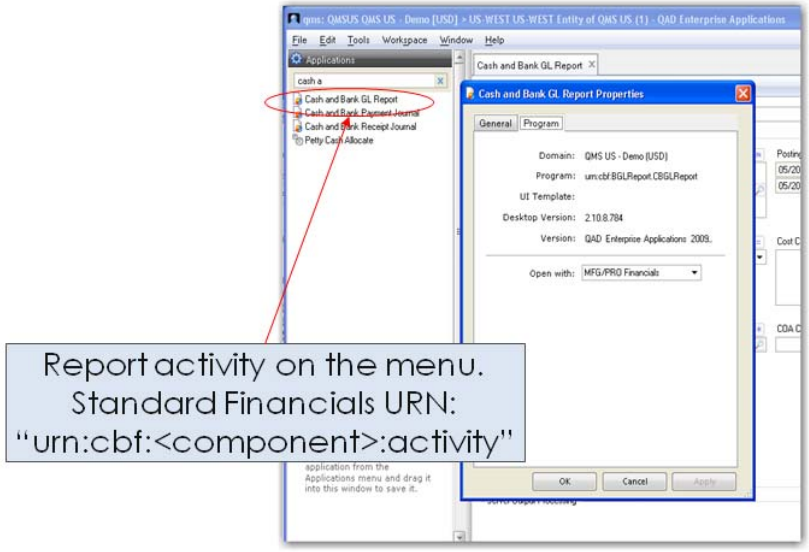
QAD

CUST_REP_170

Report Framework for Financials Reports 17

Report Framework for Financials Reports

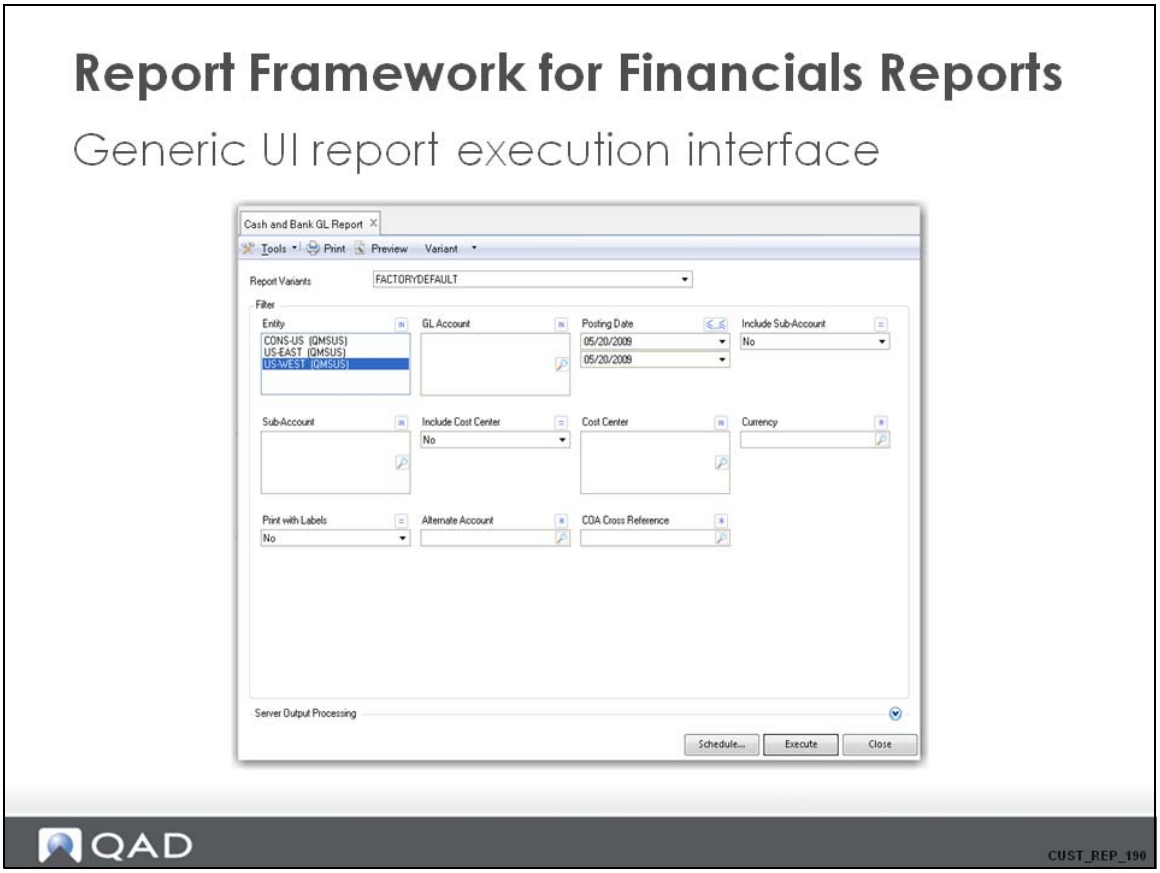
Report execution, report activities



QAD

CUST_REP_180

Report Framework for Financials Reports 18

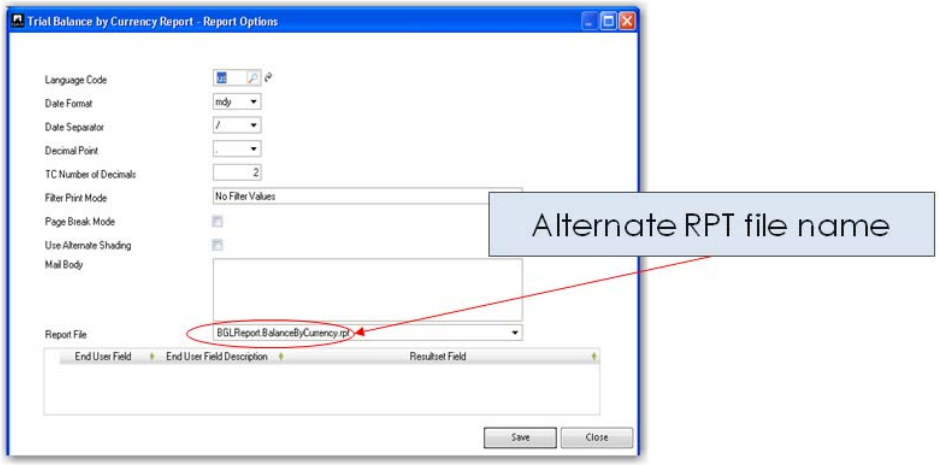


Report Framework for Financials Reports 19

Report Framework for Financials Reports

Generic UI report execution interface

Use Tools->Report options... to pass settings for the report execution



QAD

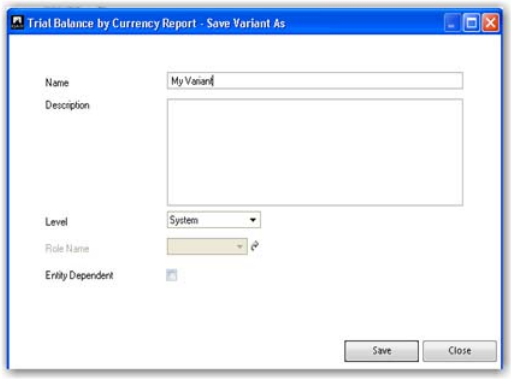
CUST_REP_200

Report Framework for Financials Reports 20

Report Framework for Financials Reports

Generic UI report execution interface

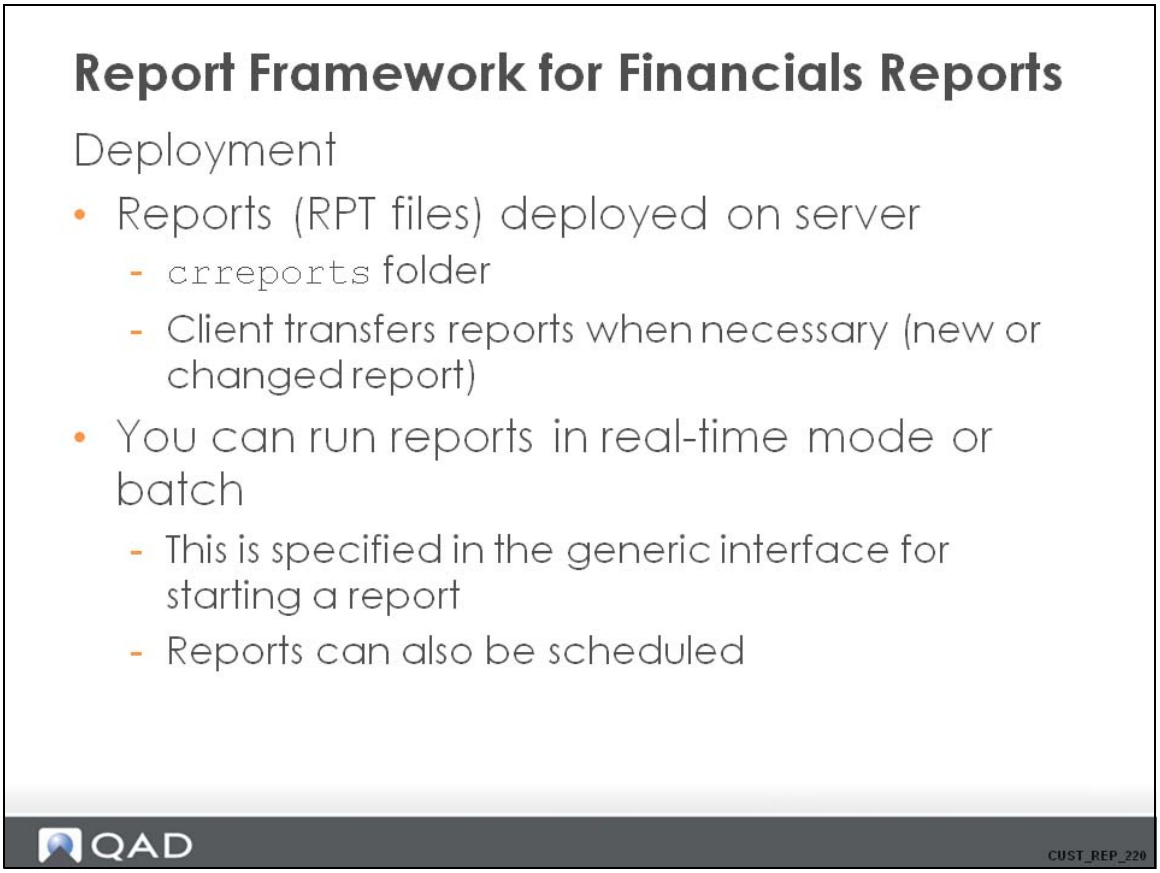
- Settings can be saved as variants.
  (Variant->Save as…)
  Saved data: Report options and filter field settings
- System comes with factory default variant for each report



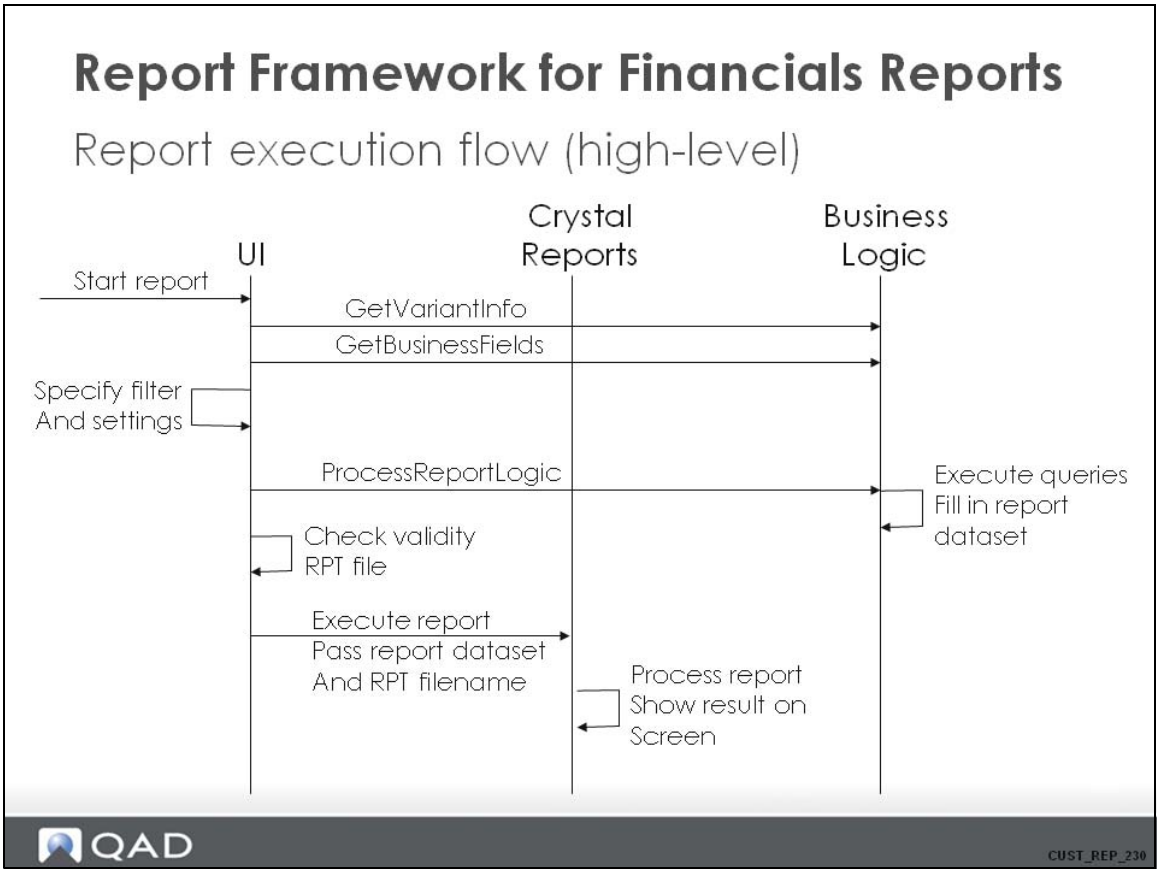
QAD

CUST_REP_210

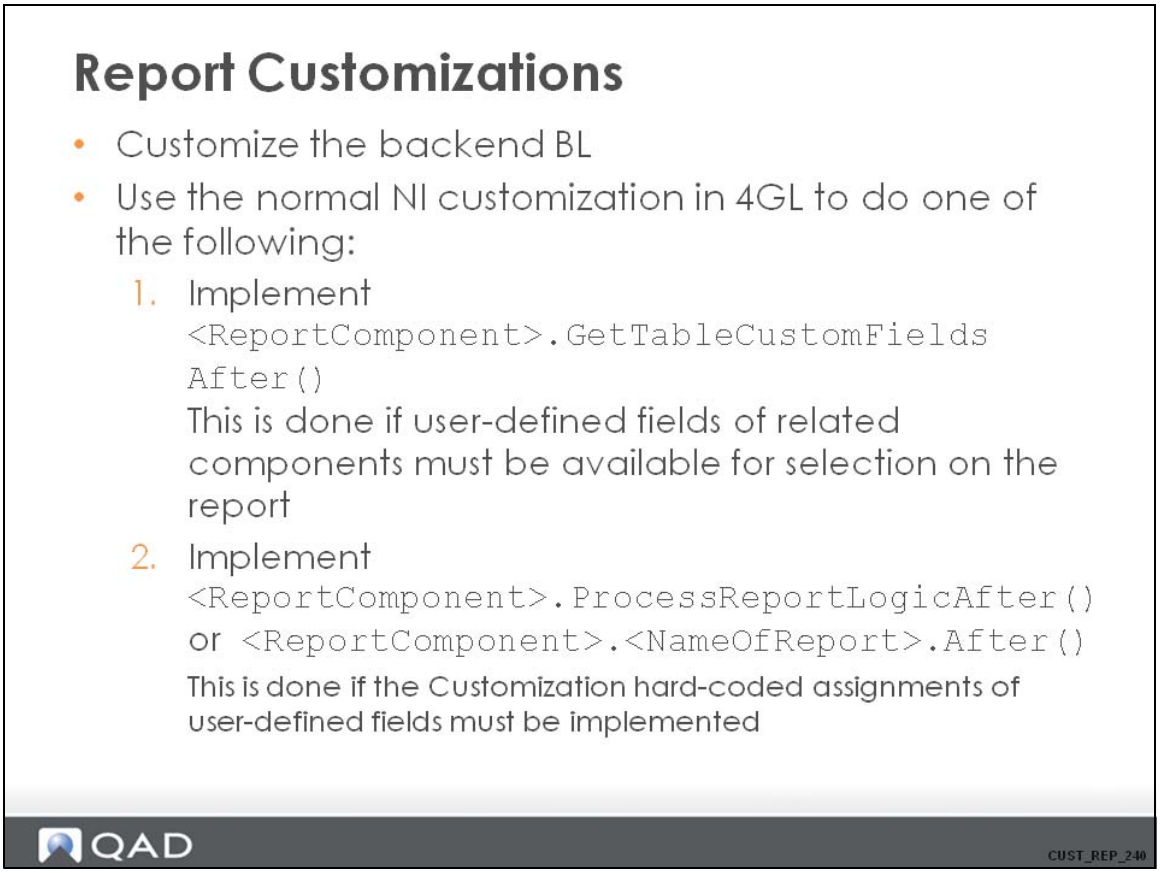
Report Framework for Financials Reports 21



Report Framework for Financials Reports 22



Report Customizations

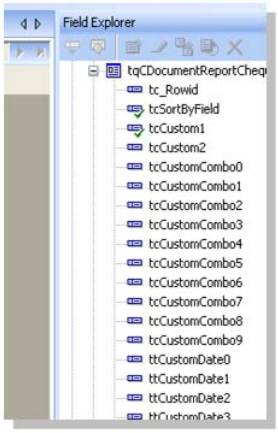


Report Customization - User-Defined Fields

Reports Customization

User-defined fields (Custom Fields)

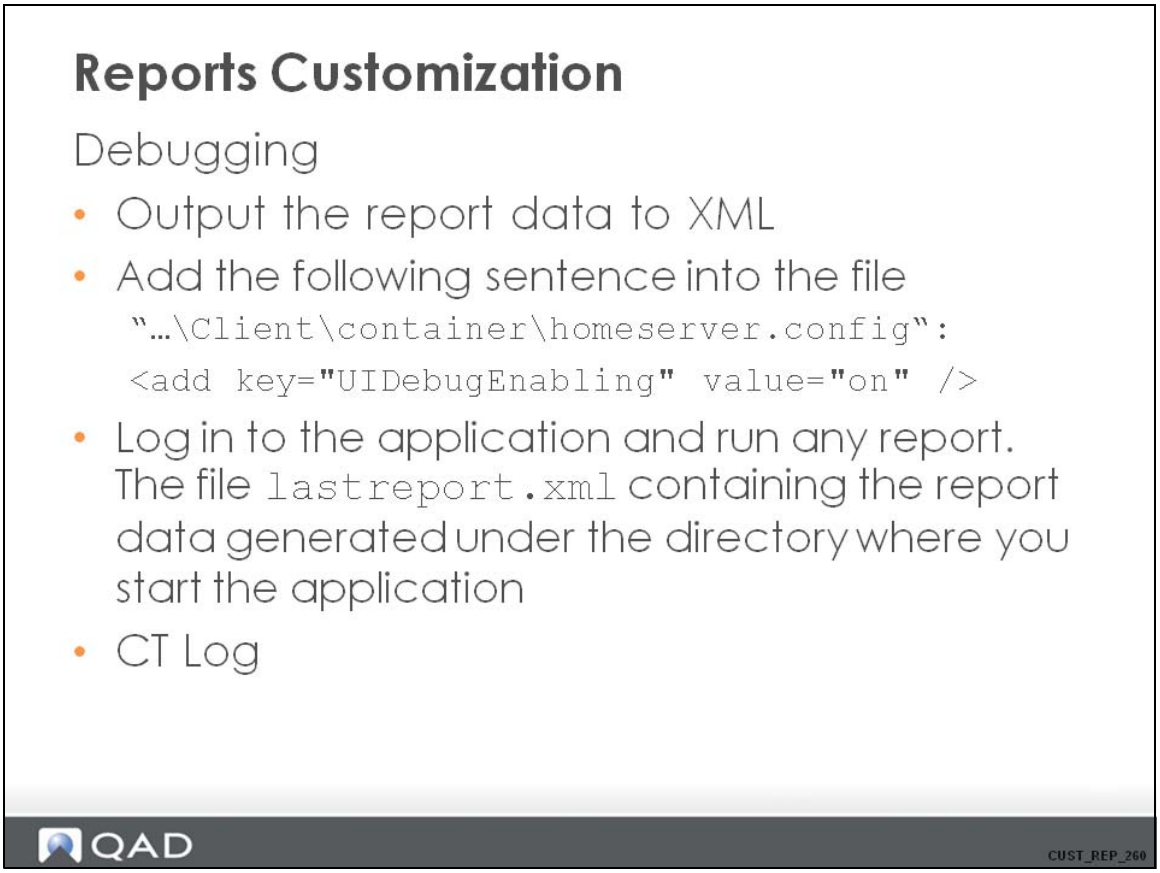
- The custom fields are just used as normal data fields in reports
- On BL side, ensure the correct values are assigned for custom fields



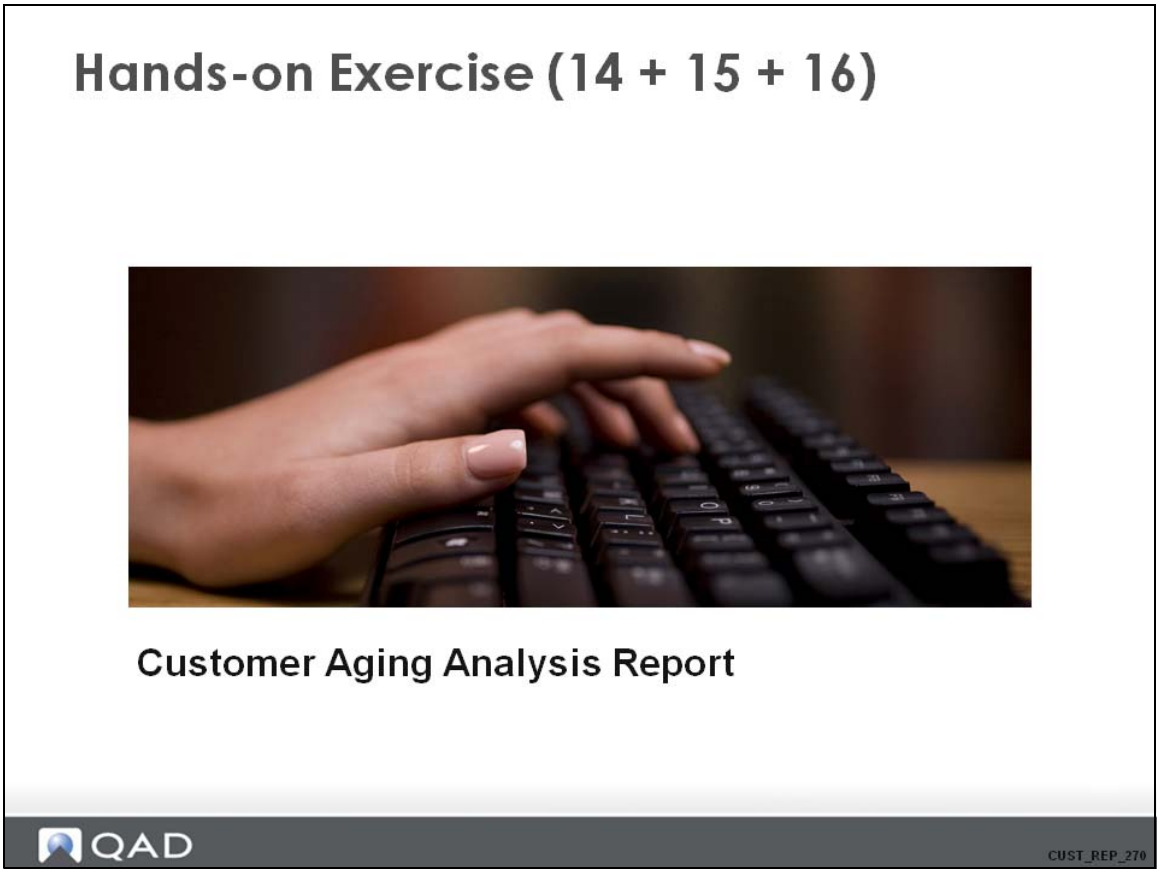
QAD

CUST_REP_250

Report Customization - Debugging



Hands-on Exercise (14 + 15 + 16)

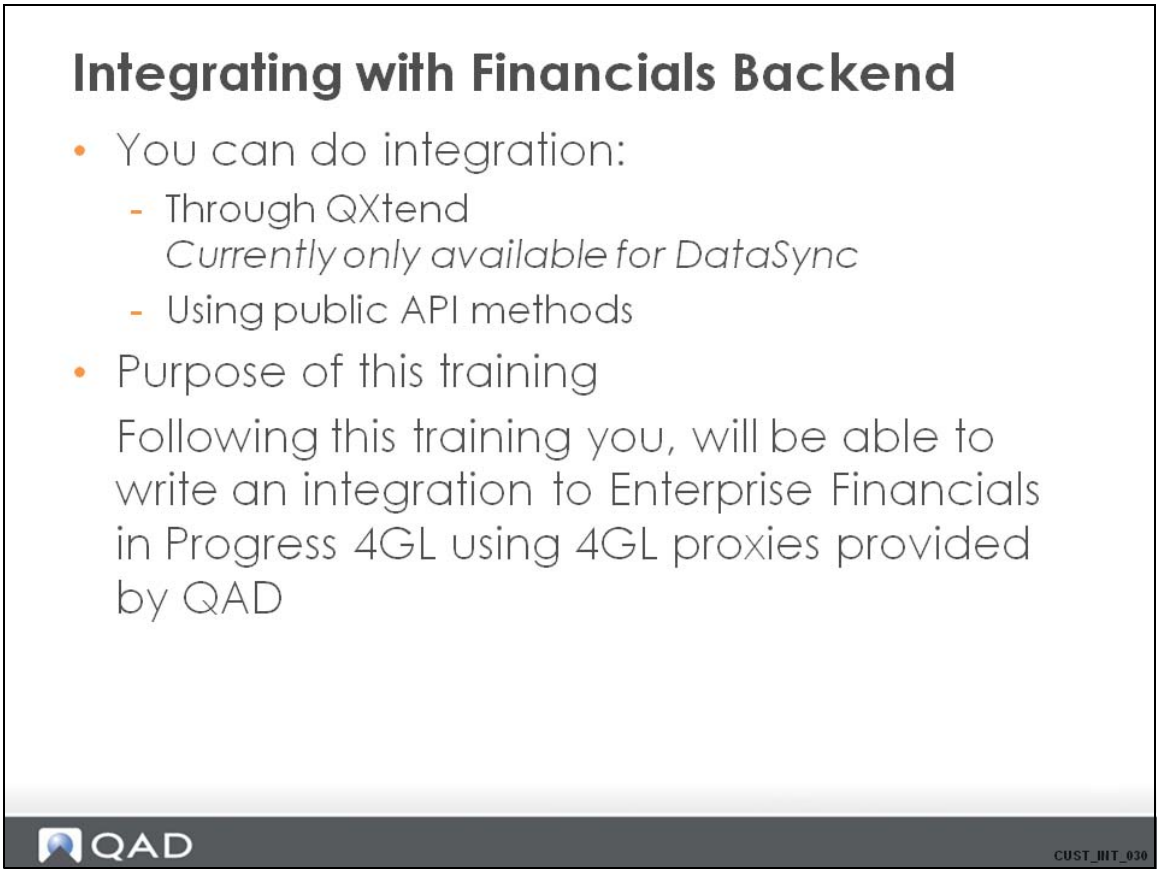


Chapter 6

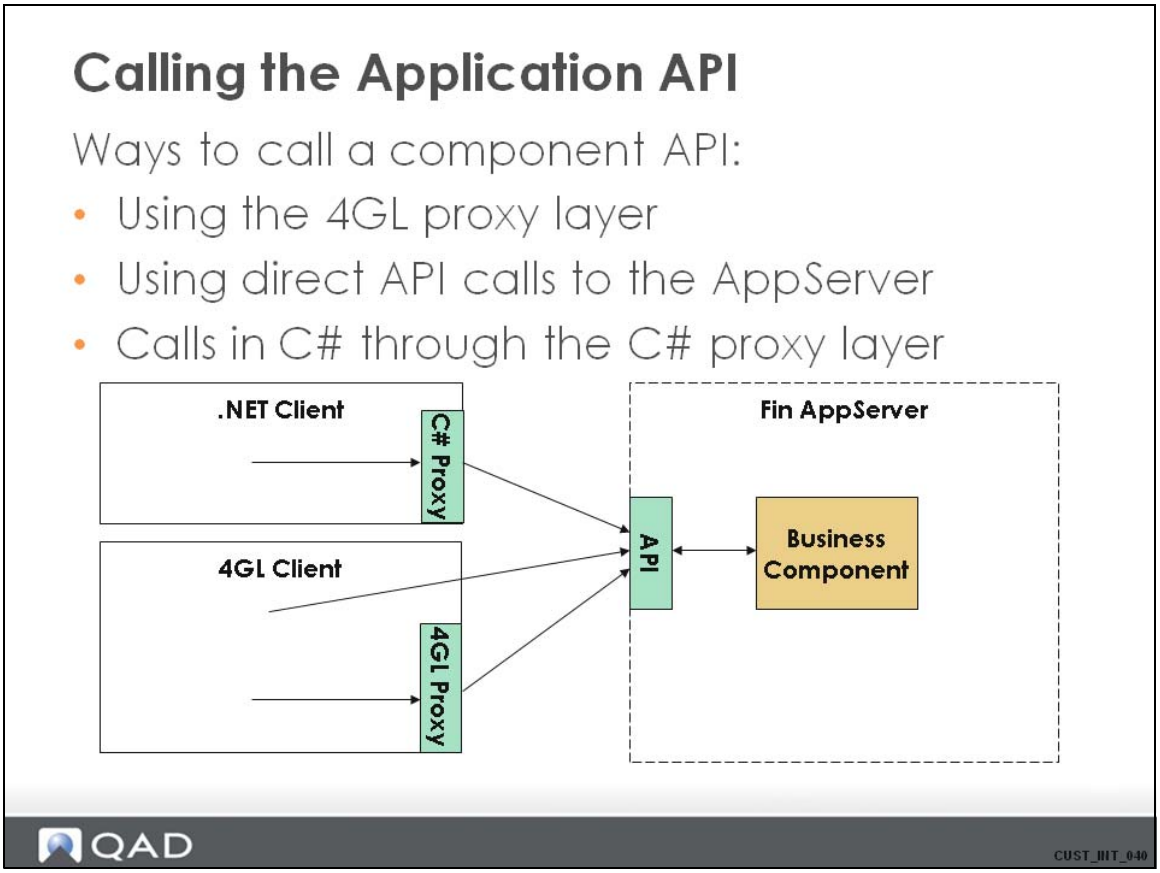
Integration with Enterprise Financials: Backend

Integration with Enterprise Financials: Backend

Integrating with Financials Backend



Calling the Application API 1

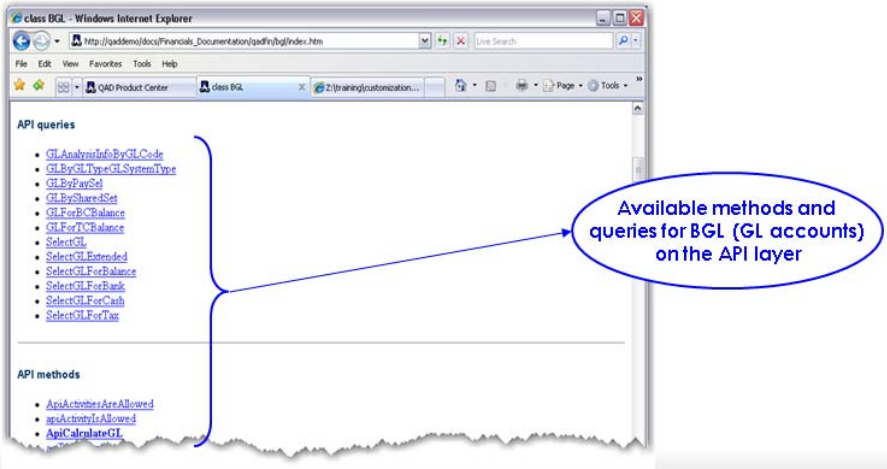


Calling the Application API 2

Calling the Application API

Available API methods:

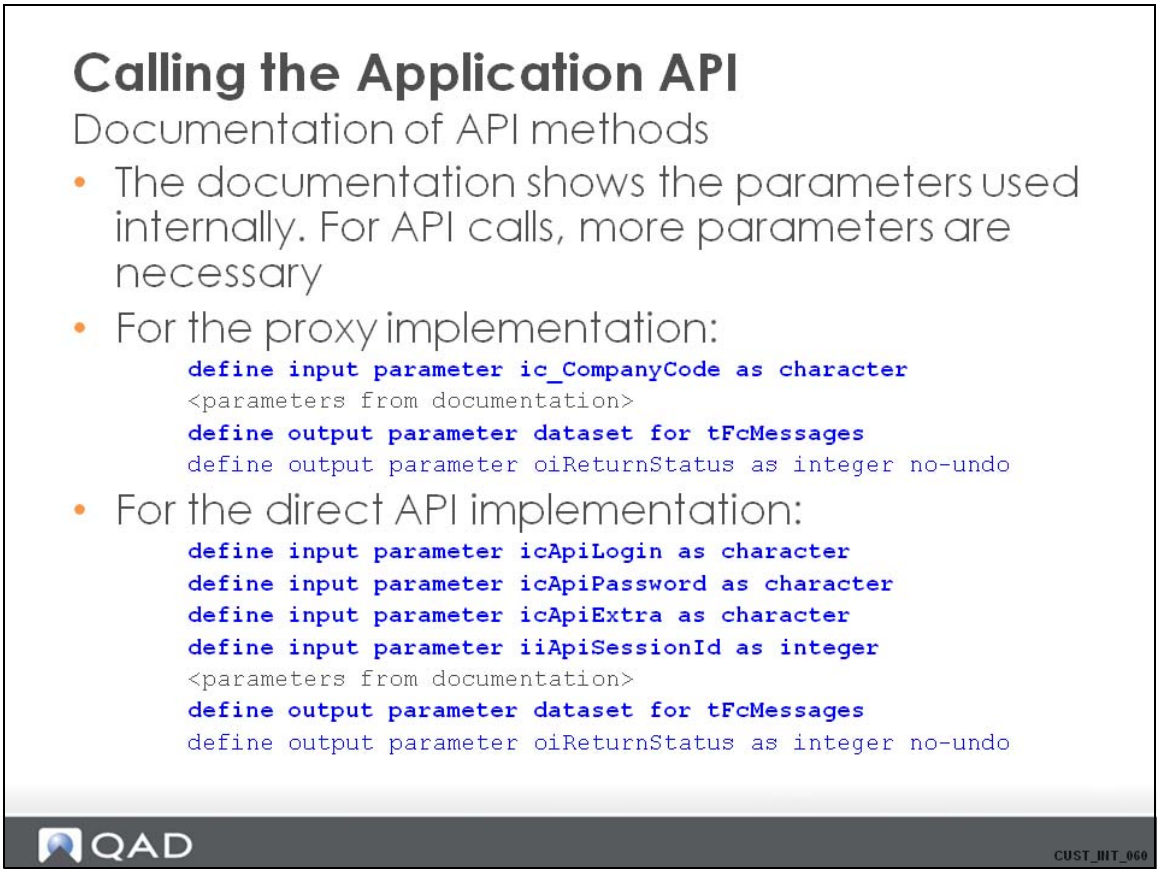
- Use HTML documentation
- Look for API queries and methods on the main page of a business component



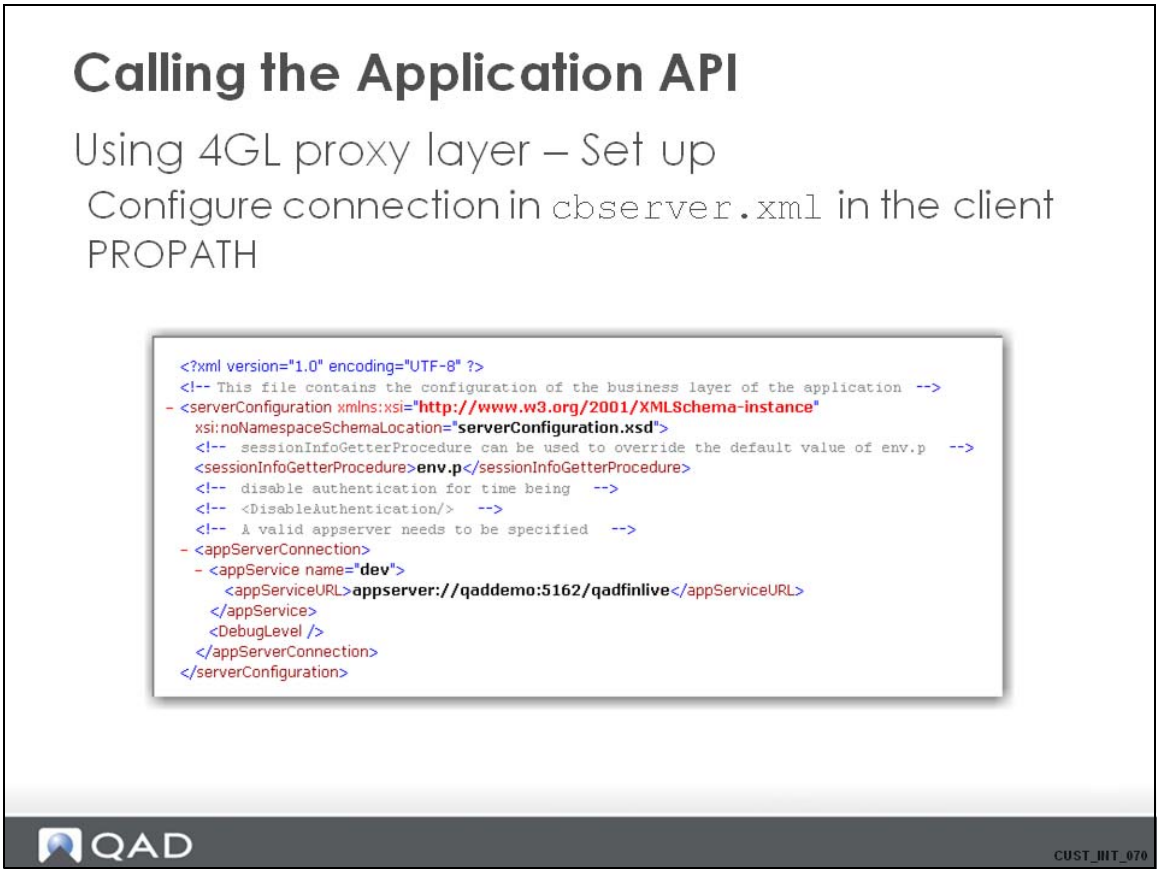
QAD

CUST_INT_050

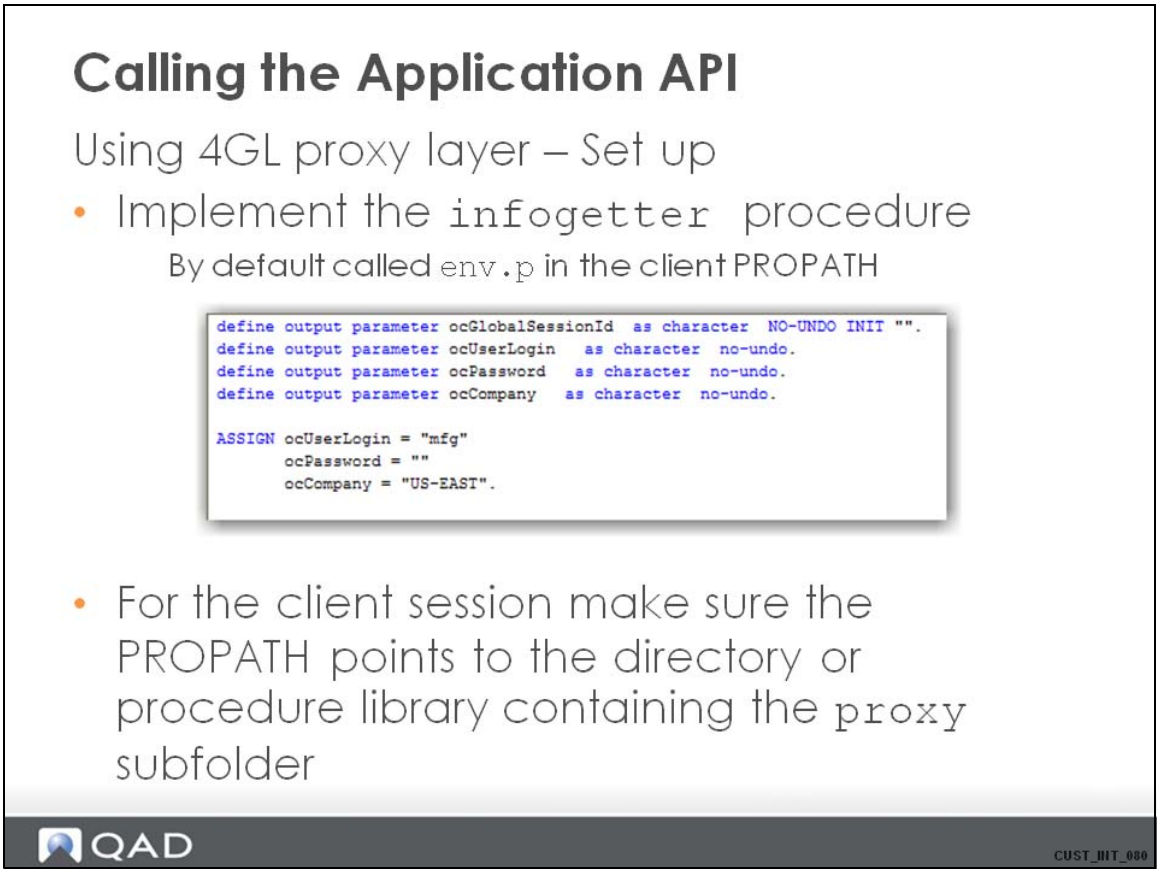
Calling the Application API 3



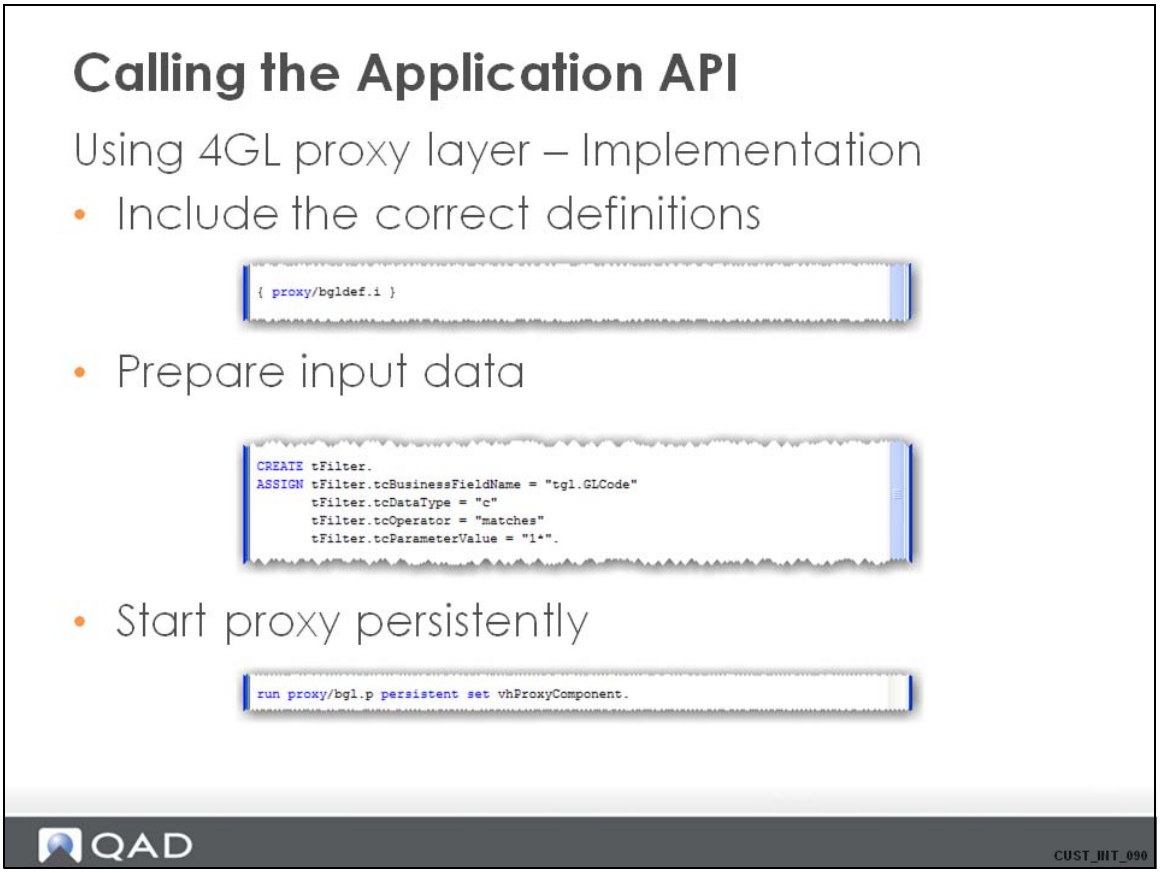
Calling the Application API 4



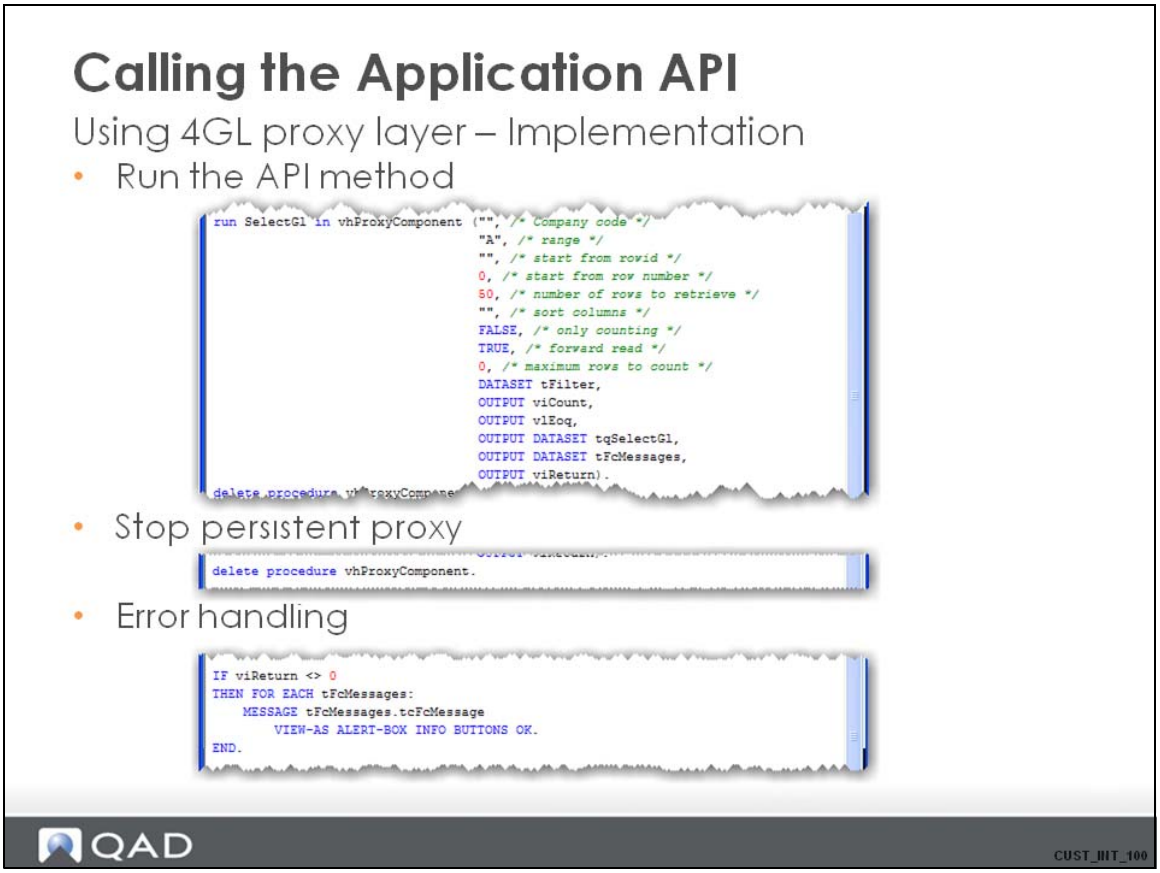
Calling the Application API 5



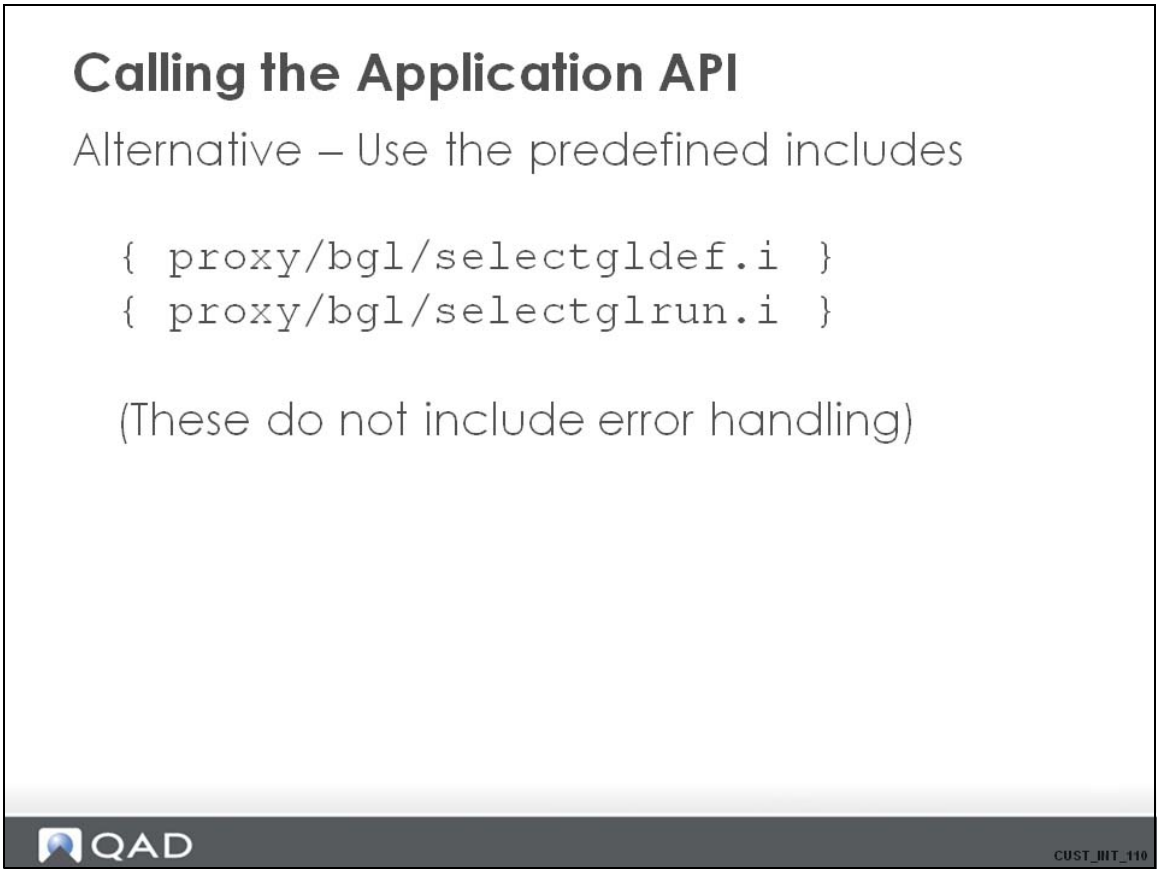
Calling the Application API 6



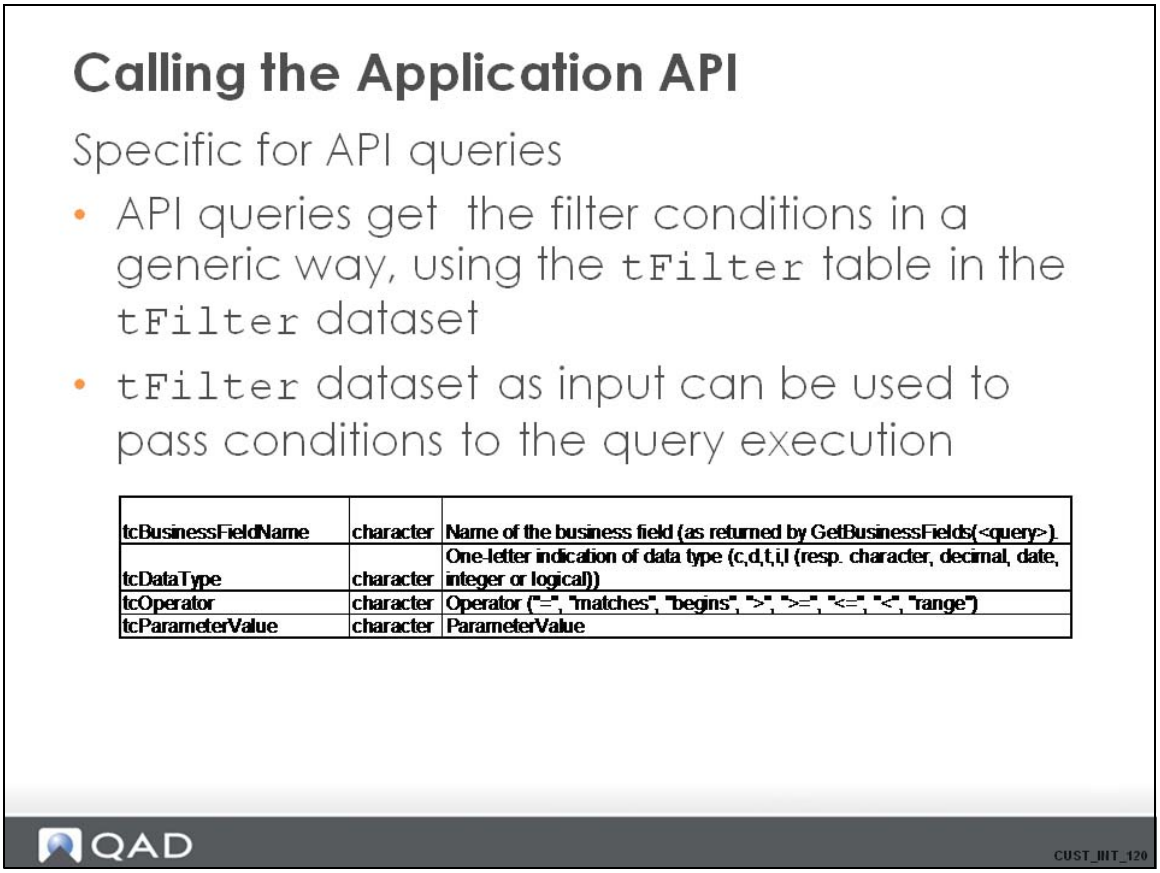
Calling the Application API 7



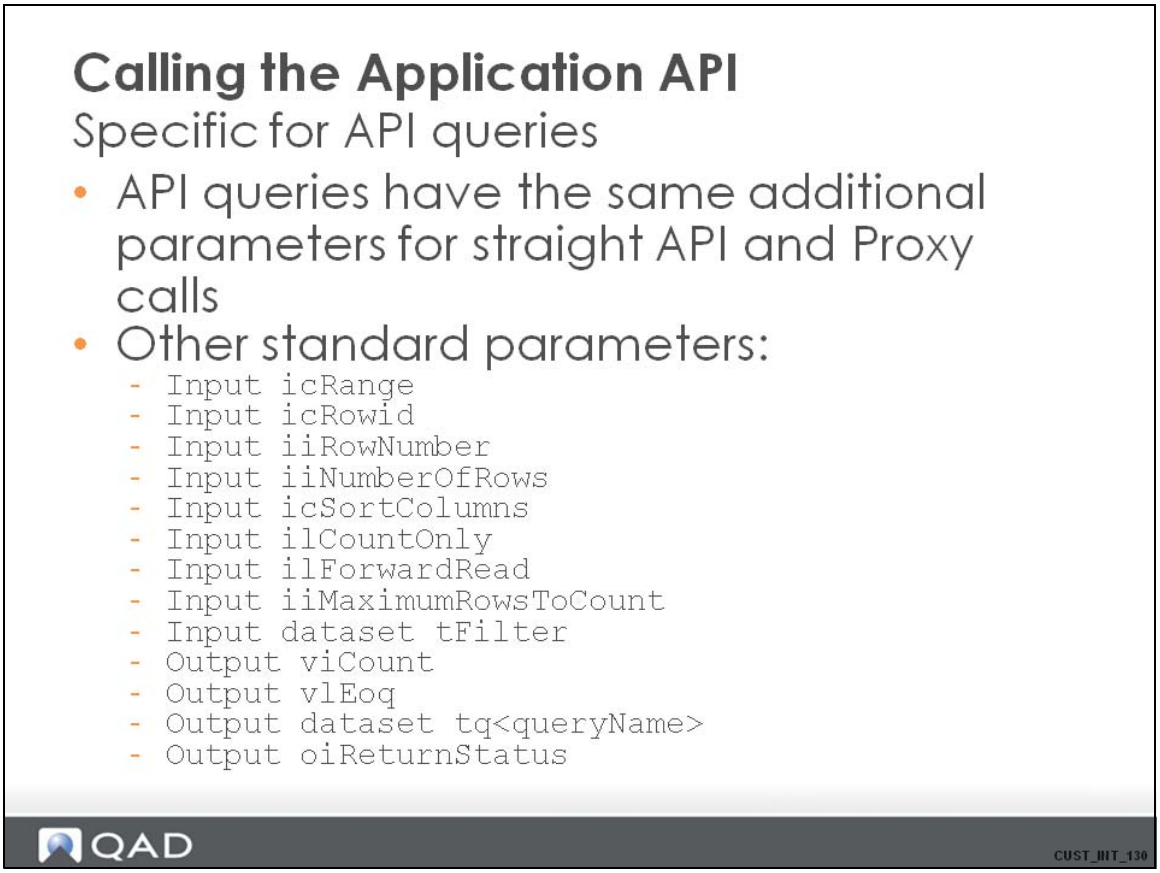
Calling the Application API 8



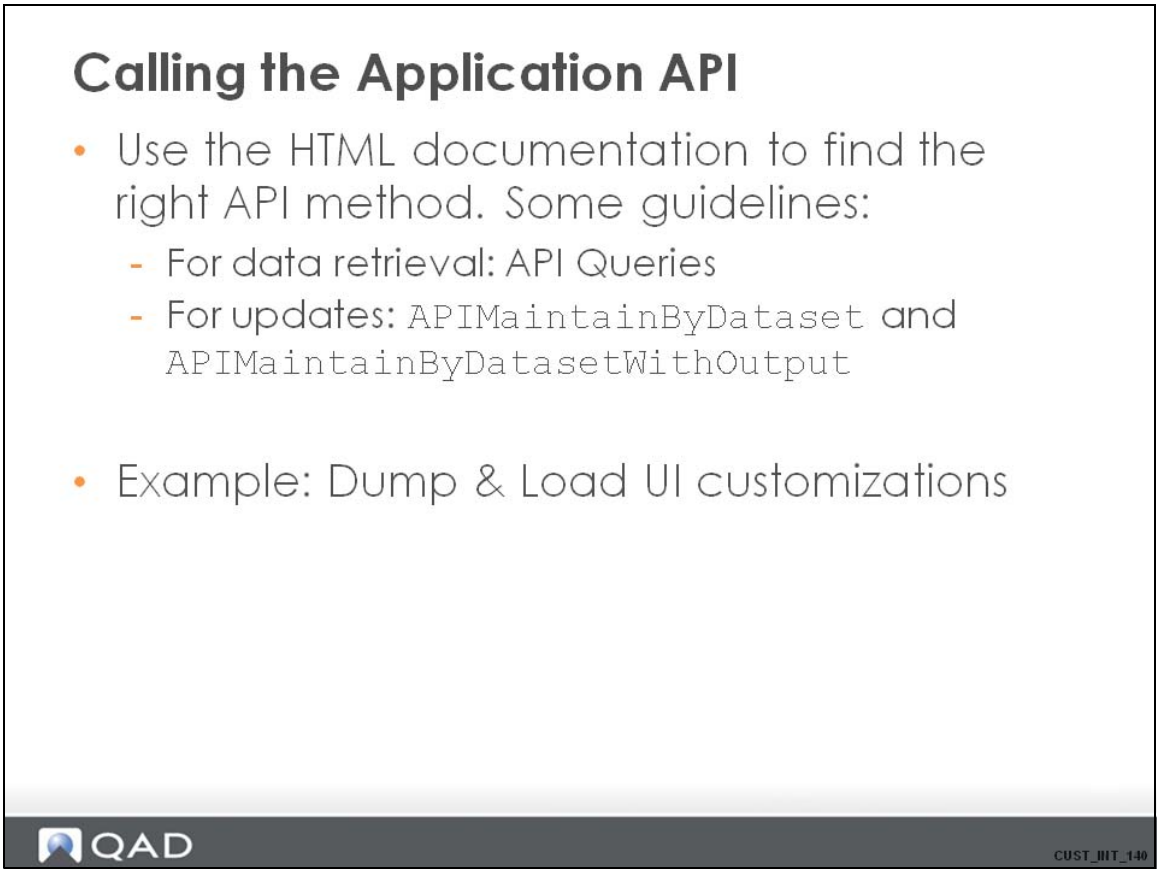
Calling the Application API 9



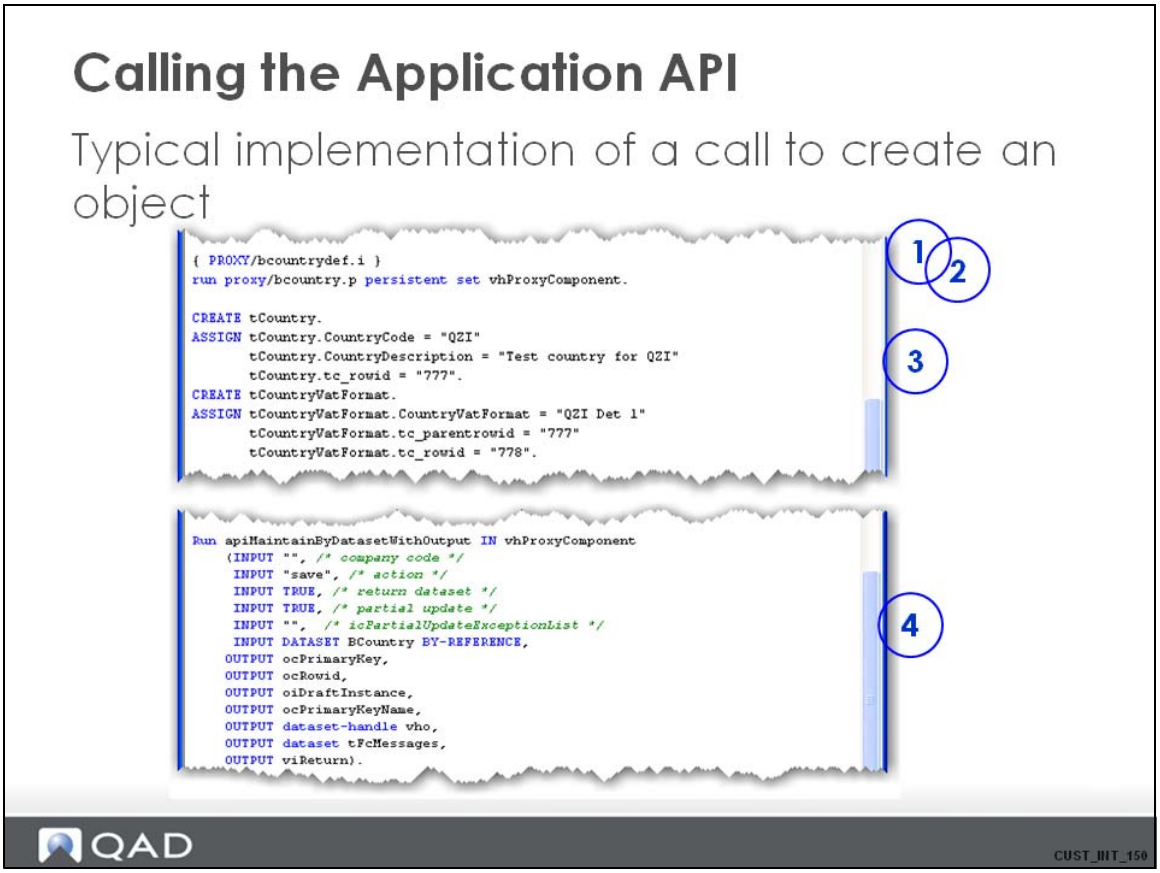
Calling the Application API 10



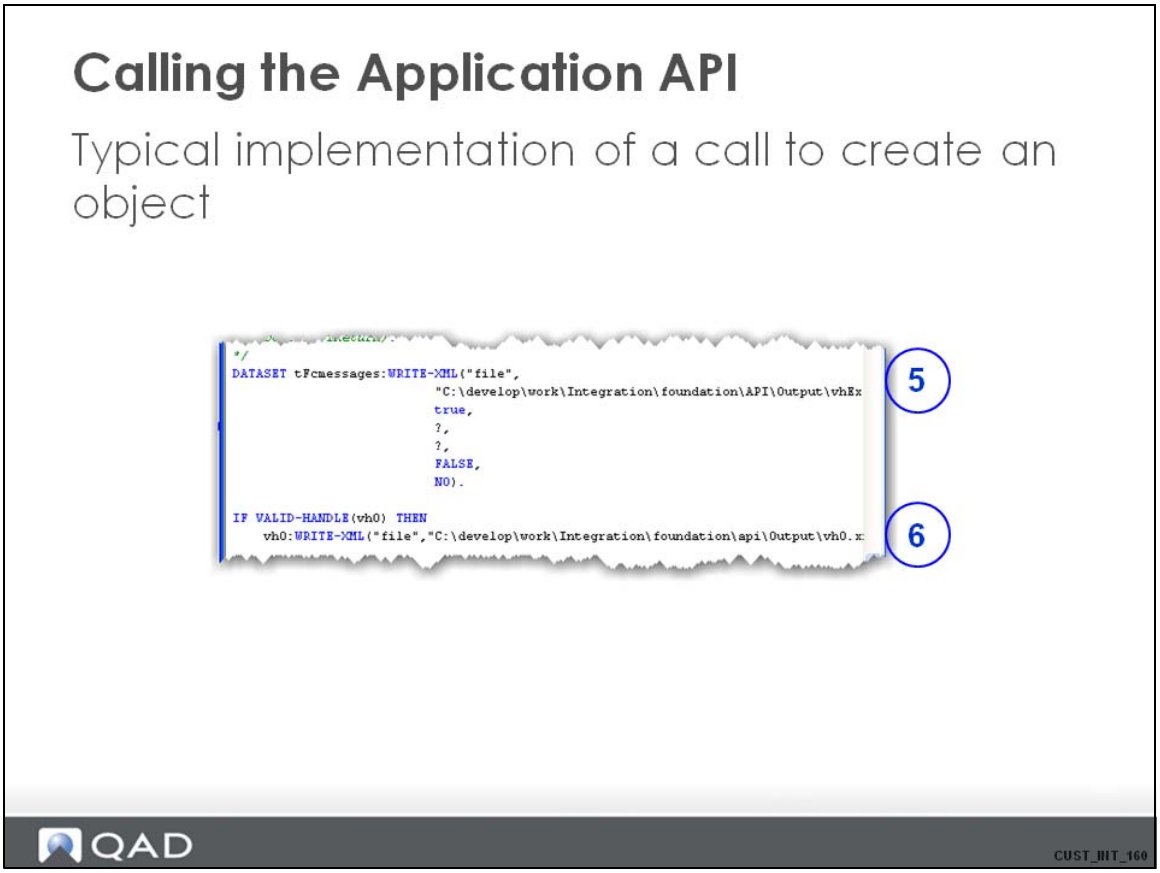
Calling the Application API 11



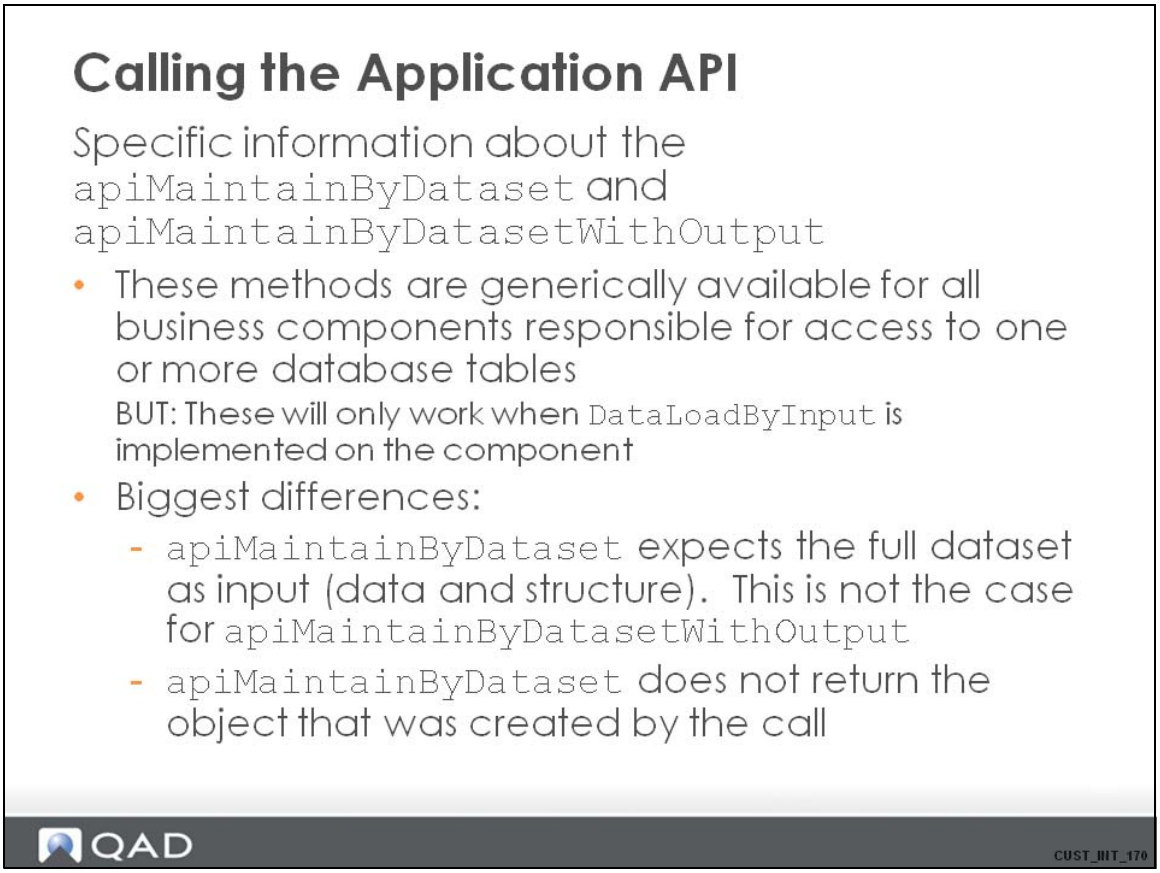
Calling the Application API 12



Calling the Application API 13



Calling the Application API 14



Calling the Application API 15