



IFSworld.com



IFS CUSTOM OBJECTS™

IFS ASSOCIATE COURSE – IFS APPLICATIONS™ 10

IFS ACADEMY



WELCOME

- Timings
- Safety
- Ground rules
- Objectives

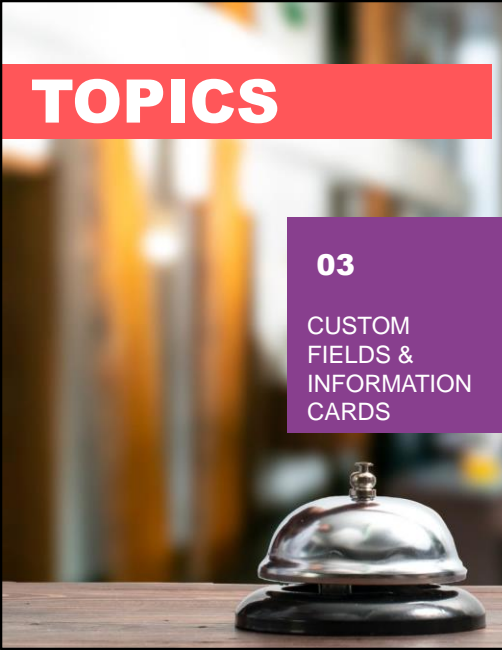



YOUR TRAINER

Name:

Contact number: -

Email:

 <h1>TOPICS</h1>	01 INTRODUCTION TO CUSTOM FIELDS	02 CUSTOM ENUMERATIONS	
	03 CUSTOM FIELDS & INFORMATION CARDS	04 CUSTOM LOGICAL UNITS, PAGES, TABS, REFERENCES	05 CUSTOM MENUS
	06 CUSTOM EVENTS	07 APPLICATION CONFIGURATI ON OACKAGES	08 CUSTOM FIELD CONFIGURATI ON CLEAN UP

SAFETY AND GROUND RULES

- Phones off or on silent
- Be social, switch of Social Media
- Respect
- Access



GROUND RULES & HOUSEKEEPING WELCOME TO THE COURSE!

Please take note of the fire, health & safety procedures at this venue.

During the course, the only rules we all need to observe are based around respect for each person, no matter what their social background, culture, sex or abilities are. This will manifest itself in allowing everyone to speak and be heard, not using any stereotypical remarks about any social group etc.

Please respect everyone's point of view, except if it is liable to cause offense. Use of inappropriate language is not acceptable. Thank you for your consideration.

As for communication and mobile devices, at least put your mobile phones to silent and only answer in case of emergency, otherwise use breaks to call back. When answering during the course, please leave the class room without interrupting the course (e.g. by loud speaking)..

For tablets, we cannot see what you are doing but expect that if they are used during the course they are exclusively used for the course, not to read, respond and update Facebook, mail etc.

We would like to have your undivided attention.



Link to Application

User Name(s)

User Password(s)

CUSTOM FIELDS

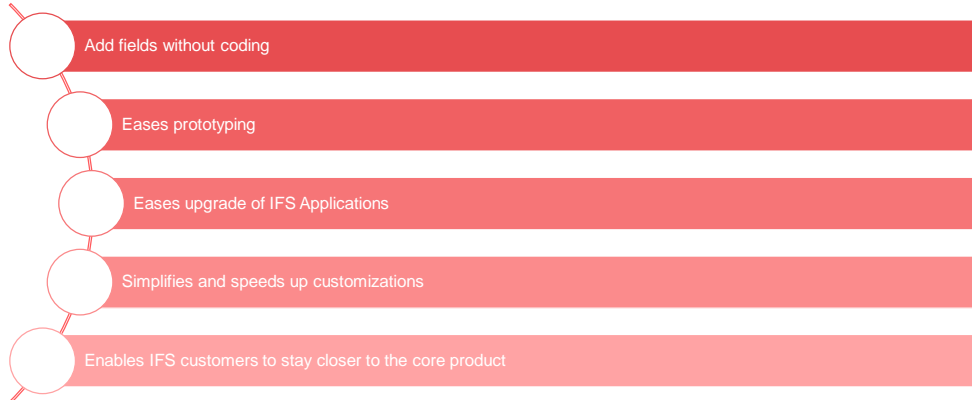
LESSON OBJECTIVES

- The overview of Custom Objects in IFS Applications and its underlying architecture.
- Different types of Custom Objects available for extending IFS Applications without customizations.
- Configuration and Deployment of Custom Fields.
- How to work with different types (persistent, read only, local) of Custom Fields for an existing logical unit.
- How to manage security for Custom Objects.
- How to export and import Custom Objects from different IFS Installations.
- How to cleanup exiting Custom Objects in an IFS Installation.



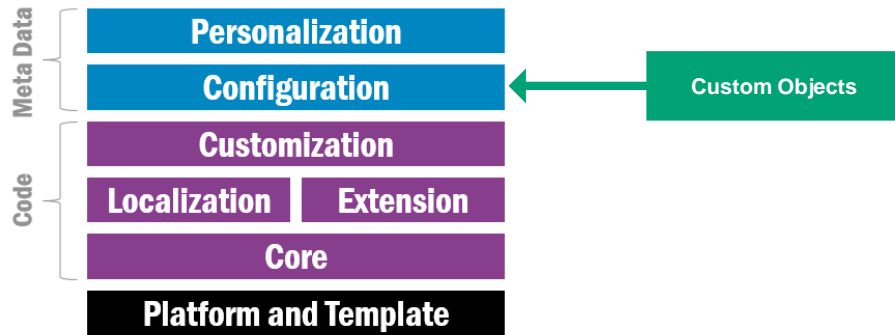
CUSTOM FIELDS

INTRODUCTION



CUSTOM FIELDS

INTRODUCTION





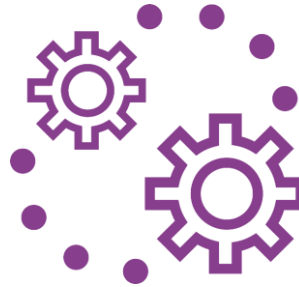
CUSTOM FIELDS

ARCHITECTURE

- Associated with a Logical Unit
- Database objects
 - One table, called xxx_CFT

CF ATTRIBUTES ARE PREFIXED CFS_

- Zero or one PL/SQL package called xxx_CFP
- One or more views called xxx_CFV



Custom Fields are associated with a Logical Unit – not just a form like Custom Menus – and are therefore available for use on any screen that utilizes that Logical Unit. This means you only need to create the Custom Field once for use on detail windows, overview windows, etc. in the same Logical Unit.

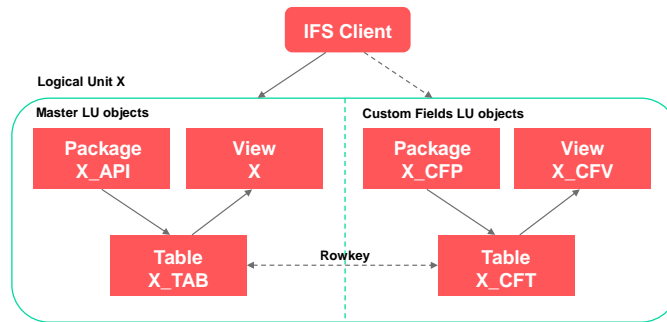
CUSTOM FIELDS

ROWKEY IS THE KEY

- Is a new system column added to all LU's.
- Is not limited to Custom Fields (Eg: Streams functionality).
- Is a persistent alternative key (treated similar as ROWID and ROWVERSION).
- Is an Oracle guid (default value `sys_guid()`).
 - "000B7C3969724B07B5D48823B1AFFF2C"
- Is used to join between master table and Custom Fields table.
- Is a unique identifier across all systems.
- Is stable for export/import, unlike `ROWID`.
- Is not generated during install(prior to Applications 9)/upgrade for performance reasons.
 - Must be activated offline later by administrator.
- For fresh installations from Applications 9 and above, ROWKEY will be generated using `SYS_GUID()`.



CUSTOM FIELDS ARCHITECTURE



select * from all_objects where object_name like '%X_CFT_'

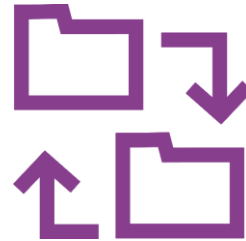
	OWNER	OBJECT_NAME	SUBOBJECT_NAME	OBJECT_ID	DATA_OBJECT_ID	OBJECT_TYPE
1	IFSAPP	SALES_PART_CFT		185872	185872	TABLE
2	IFSAPP	SALES_PART_CFP		185874		PACKAGE
3	IFSAPP	SALES_PART_CFV		185875		VIEW
4	IFSAPP	SALES_PART_CFP		185876		PACKAGE BODY



CUSTOM FIELDS

INTERACTION WITH OTHER CONCEPTS

- Custom Menu
- Custom Events (some limitations)
- Custom Field Columns in existing "List Of Value" dialogs
- Search Dialog and Saved Searches
- Conditional Formatting
- Output
- Quick Reports
- IFS Reporting
- In-Memory Database



NOTE - No business logic, apart from Custom Events, interacts with Custom Fields



- The limitation with Custom Events together with Custom Fields is due to that Custom Events are triggered from changes in database tables and Custom Fields is implemented with a table that is separate from the one it connects to.

As a consequence, if you put a Custom Event on the “original” table you will not get the event fired if you only change a custom field value, and the other way around. This is not a problem if you e.g. want to write a simple validation of the Custom Fields value.

You would need to write two custom events in order to ensure that you capture any change, one for the “original” table and one for the connected table for Custom Fields.

Custom Fields cannot be used in integration.

In-Memory:

Custom fields tables are supported with In-Memory Acceleration Packages.

When persistent custom fields are enabled, all relevant custom fields tables are automatically added to the In-Memory Acceleration Packages that contain the base tables for that particular custom fields.

CUSTOM FIELDS

CONFIGURATION

▪ Type of Custom Fields attributes

Persistent (possible to change and store)

- String, Number, Date
- Enumeration (Custom Enumeration and any existing Enumeration)

Referenced (read only)

- Expression
- Reference field
- Select statement

Local

- Special case of a read-only attribute. It is used to make an existing LU attribute visible as a Custom Attribute.



CUSTOM FIELDS

CUSTOM ATTRIBUTE TYPE EXAMPLES

Persistent field

- Delivery instructions ("Knock on the door")
- Shoe size ("38")
- Favorite artist ("ABBA")

Persistent field with enumeration

- Favorite color (Red, Blue, Yellow)
- Part type (X part, Y part)
- T-shirt size (S, M, L)

Reference

- Phone number (from Order -> Customer -> Customer contact)
- Total number of this part manufactured today (SELECT count ...)



CUSTOM FIELDS

SUPPORTED SERVER DATA TYPES

Custom Fields - Persistent

- String, Date, Number

Referenced Fields (read-only)

- Follow IFS Applications references to fields in other related LU's

Derived Fields (read-only)

- Use any predefined method (PL/SQL function) to fetch a value
- Context Substitution Variables

Expressions (read-only)

- Any SQL expression

Enumerations

- New Custom Enumerations
- All existing IFS Applications enumerations



CUSTOM FIELDS CONFIGURATION

System Parameters

- Custom Fields tablespaces (Data, Index and LOB)
- Framework ON/OFF

System Parameters			
Category	Parameter	Description	Value
Contact Capture	Contact Capture API key	No API key defined	No key
Contact Capture	Contact Capture API		IFS
Custom Objects	Tablespace for custom LOB's	Default	IFSAPP_LOB
Custom Objects	Conditional Fields framework	Conditional Fields framework enabled	ON
Custom Objects	Info Cards framework	Info Cards framework enabled	ON
Custom Objects	Custom Fields framework	Custom Fields framework enabled	ON
Custom Objects	Custom Pages framework	Custom Pages framework enabled	ON
Custom Objects	Tablespace for custom tables	Default	IFSAPP_DATA
Custom Objects	Tablespace for custom indexes	Default	IFSAPP_INDEX
Data Archive	Data archive process startup	Data archive background process di...	OFF



Translations are managed directly in the Custom Fields configuration screen

CUSTOM FIELDS

DEPLOYMENT

- Publishing a Logical Unit's Custom Fields executes the following steps
- Database objects created and deployed

TABLE, PACKAGE, VIEWS

- Presentation Object created

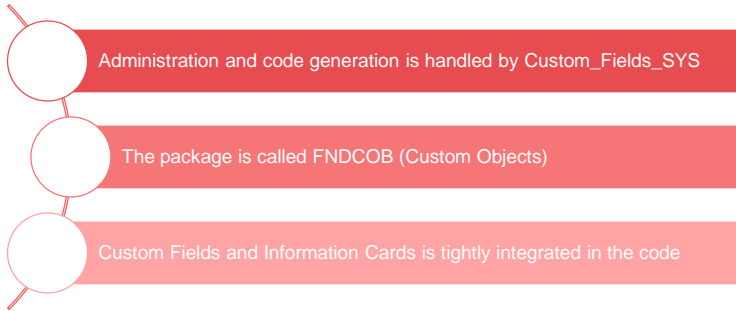
GRANTS ALL NECESSARY DATABASE OBJECTS TO THOSE ROLES WITH EXISTING ACCESS TO THE CORRESPONDING SOURCE VIEW

- Grants to necessary system users (IFSSYS, IFSWEBCONFIG)
- Refresh of Dictionary cache and Custom Field cache



CUSTOM FIELDS

BEHIND THE SCENES



CUSTOM FIELDS

DEPLOYMENT OF A CUSTOM FIELD

- The client calls Custom_Fields_API.Deploy__
- Calls to Custom_Fields_SYS package to delegate most work
 - Check if everything is correct
 - Call Generate_CF_XXX_Msg__ (XXX:cre, spec, view, body)



CUSTOM FIELDS

EXAMPLE – CUSTOM ENUMERATION

The image displays two side-by-side screenshots of the IFS Sales Part form for '20202 Blue 2 L'. The left screenshot shows the 'General' tab, and the right screenshot shows the 'Maintenance' tab. In the 'Maintenance' tab, a 'Reason Code' dropdown menu is highlighted with a red box, showing 'Tune Up' as the selected option.

Sales Part - 20202 Blue 2 L v 1 (100) *

Sales Part Number: 20202 Part Description in Use: Blue 2 L Site: 80

General Misc Part Info Descriptions Characteristics Charges Condition Code Pricing

Inventory Part

Part Number: 20202
Description: Blue 2 L
☐ Configurable ☐ DOP

Units of Measure

Inventory U/M: pcs Inv Conv Fact: 1
Price U/M: pcs Price Conv Fact: 1
Sales U/M: pcs Invert Conv Fact: 1
Catch U/M:

Created: 9/27/2001

Price: 11.00 USD
Price Ind. Tax: 11.00 USD
Exp Avg Price: USD
Rental Price: 0.00 USD
Rental Pr Ind. Tax: 0.00 USD
Min Sales Qty:

Sourcing

Sourcing Options: Inven
Sourcing Rule:

Grouping

Sales Price Grp: G1
Sales Group: 10
Discount Group:
Rebate Group:

Tax Code: 0

Tax Class: ☐ Primary Sales Part ☐ Document Text ☐ Taxable ☐ Use Price 2nd Tax

Sales Part - 20202 Blue 2 L v 1 (100) *

Sales Part Number: 20202 Part Description in Use: Blue 2 L Site: 80 Reason Code: Tune Up

General Misc Part Info Descriptions Characteristics Charges Condition Code Pricing Maintenance Part Specific Over

Inventory Part

Part Number: 20202
Description: Blue 2 L
☐ Configurable ☐ DOP

Units of Measure

Inventory U/M: pcs Inv Conv Fact: 1
Price U/M: pcs Price Conv Fact: 1
Sales U/M: pcs Invert Conv Fact: 1
Catch U/M:

Created: 9/27/2001

Price: 11.00 USD
Price Ind. Tax: 11.00 USD
Exp Avg Price: USD
Rental Price: 0.00 USD
Rental Pr Ind. Tax: 0.00 USD
Min Sales Qty:

Sourcing

Sourcing Options: Inventory Order
Sourcing Rule:

Grouping

Sales Price Grp: G1 Basis
Sales Group: 10 General Sales Group 10
Discount Group:
Rebate Group:

Tax Code: 1 Tax with no reduction 25%

Tax Class: ☐ Primary Sales Part ☒ Active ☐ Notes ☐ Quick Registered Part ☐ Export To External Application

After adding a Custom Field, it will be visible at the bottom of the list of fields on the Properties/Layout tab

CUSTOM FIELDS

EXAMPLE – CUSTOM ENUMERATION

Sales Parts					Sales Parts										
Sales Part Number	Part Description in Use	Sales Part Description	Size	Type of Sale	Sales Part Number	Part Description in Use	Sales Part Description	Reason Code	Size	Type of Sales Part	Sales Group	Sales Price Group	Rebate Group	Part No	Description
20302	Blue 2 L	Blue 2 L	80	Inventory part	20302	Blue 2 L	Blue 2 L		80	Inventory part	10	G1		20302	Blue 2 L
20301	White 10 L	White 10 L	80	Inventory part	20301	White 10 L	White 10 L		80	Inventory part	10	G1		20301	White 10 L
20401	White (new) 10 L	White (new) 10 L	80	Inventory part	20401	White (new) 10 L	White (new) 10 L		80	Inventory part	10	G1		20401	White (new) 10 L
EASYFLEX109	Garage door remote keypad	Garage door remote keypad	30	Inventory part	EASYFLEX109	Garage door remote keypad	Garage door remote keypad		30	Inventory part	10	EF		109	Remote Key Pad
EASYFLEX15	Garage Door Replacement Engine	Garage Door Replacement Engine	31	Inventory part	EASYFLEX15	Garage Door Replacement Engine	Garage Door Replacement Engine		31	Inventory part	10	EF		15	Spare Garage Door Engine Kit
P03-100	Standard Wheel P03	Standard Wheel P03	42	Inventory part	P03-100	Standard Wheel P03	Standard Wheel P03		42	Inventory part	50	G1		P03-100	Wheel
P01-100	Wheel	Wheel	42	Inventory part	P01-100	Wheel	Wheel		42	Inventory part	60	40		P01-100	Wheel
FP	Fixed price	Fixed price	60	Non inventory	FP	Fixed price	Fixed price		60	Non inventory part	10	40			
1500	Service Agreement	Service Agreement	60	Non inventory	1500	Service Agreement	Service Agreement		60	Non inventory part	10	G1			
1600	Scheduled Maintenance	Scheduled Maintenance	60	Non inventory	1600	Scheduled Maintenance	Scheduled Maintenance		60	Non inventory part	10	G1			
P11-100	15" Wheel	15" Wheel	40	Inventory part	P11-100	15" Wheel	15" Wheel		40	Inventory part	60	G1		P11-100	15" Wheel
P11-102	17" Wheel	17" Wheel	40	Inventory part	P11-102	17" Wheel	17" Wheel		40	Inventory part	60	G1		P11-102	17" Wheel
P11-103	17" All Sport Wheel	17" All Sport Wheel	40	Inventory part	P11-103	17" All Sport Wheel	17" All Sport Wheel		40	Inventory part	63	G1		P11-103	17" All Sport Wheel
BP11	BP 15 Alloy Wheel	BP 15 Alloy Wheel	1	Inventory part	BP11	BP 15 Alloy Wheel	BP 15 Alloy Wheel		1	Inventory part	63	G1		BP11	BP Wheel 15 Alloy
HOTEL	Hotel Expense	Hotel Expense	70	Non inventory	HOTEL	Hotel Expense	Hotel Expense		70	Non inventory part	SM-E	SM-E			
7000	Emergency Service	Emergency Service	70	Non inventory	7000	Emergency Service	Emergency Service		70	Non inventory part	10	50			
11000-1-1-01	Module 1, EBU 1, Part 1	Module 1, EBU 1, Part 1	90	Inventory part	11000-1-1-01	Module 1, EBU 1, Part 1	Module 1, EBU 1, Part 1		90	Inventory part	11000SG	11000		11000-1-1-01	Module 1, EBU 1, Part 1
11000-1-1-01-1	Module 1, EBU 1, Part 1	Module 1, EBU 1, Part 1	90	Inventory part	11000-1-1-01-1	Module 1, EBU 1, Part 1	Module 1, EBU 1, Part 1		90	Inventory part	11000SG	11000		11000-1-1-01-1	Module 1, EBU 1, Part 1
11000-1-1-02	Module 1, EBU 1, Part 2	Module 1, EBU 1, Part 2	90	Inventory part	11000-1-1-02	Module 1, EBU 1, Part 2	Module 1, EBU 1, Part 2		90	Inventory part	11000SG	11000		11000-1-1-02	Module 1, EBU 1, Part 2
11000-3-2	Module 3, EBU 2	Module 3, EBU 2	90	Inventory part	11000-3-2	Module 3, EBU 2	Module 3, EBU 2		90	Inventory part	11000SG	11000		11000-3-2	Module 3, EBU 2
11000-3-2-01	Module 3, EBU 2, Part 1	Module 3, EBU 2, Part 1	90	Inventory part	11000-3-2-01	Module 3, EBU 2, Part 1	Module 3, EBU 2, Part 1		90	Inventory part	11000SG	11000		11000-3-2-01	Module 3, EBU 2, Part 1
11000-3-2-03	Module 3, EBU 2, Part 3	Module 3, EBU 2, Part 3	90	Inventory part	11000-3-2-03	Module 3, EBU 2, Part 3	Module 3, EBU 2, Part 3		90	Inventory part	11000SG	11000		11000-3-2-03	Module 3, EBU 2, Part 3
7501	INTERNAL supply Direct	INTERNAL supply Direct	501	Inventory part	7501	INTERNAL supply Direct	INTERNAL supply Direct		501	Inventory part	DSG3	DPG1		7501	com part 7501
7602	Stock then internal source	Stock then internal source	501	Inventory part	7602	Stock then internal source	Stock then internal source		501	Inventory part	DSG3	DPG1		7602	com part 7602

After adding a Custom Field on a table window, to make it visible, you have to right click on the column header and select Choose Column. And from the dialog that opens up, select the custom field(s) you added.

CUSTOM ENUMERATION - DEMO

CREATE CUSTOM ENUMERATION

Custom Enumeration

Name: Prompt: Comment:

☒ Approved Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.

☒ Published Publish will generate and deploy all server objects, thus making it available to be used and referenced from the client and other objects. When publishing, only the Approved items will be included.

Published Date: 7/23/2019 at 7:04:59 PM

☒ Synchronized Synchronized means that no changes has been made to the configuration since it was last Published.

DB Value	Client Value	Seq No	Approved	Published	Translations
HIGH	High	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0
MEDIUM	Medium	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0
LOW	Low	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0

1. Navigate to Solution Manager -> Security -> User Interface -> Custom Objects -> Custom Enumeration
 2. Create SupplierImportance enumeration with values - High, Medium, Low
 3. Right click and select Approve.
 4. Right click and select publish.
- This will generate the SUPPLIER_IMPORTANCE_CFP package.

CUSTOM FIELD - DEMO

CREATE CUSTOM FIELD

Custom Fields - SupplierInfoGeneral

Logical Unit: Prompt: Module:

☒ RowKey Activated
Activating the RowKey for a Logical Unit (LU) is the first step in order to make an LU possible to be extended with Custom Fields. RowKey is the functionality that connects standard LUs with the extended Custom Field LUs.

☒ Approved
Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.

Approved Detail Views: [None \(of totally 2\)](#)

☒ Published
Publish will generate and deploy all server objects, thus making it available to be used and referenced from the client and other objects. When publishing, only the Approved items will be included.
Published Date: 7/23/2019 at 7:26:46 PM

Presentation Object: [cfSupplierInfoGeneral](#)

☒ Synchronized
Synchronized means that no changes has been made to the configuration since it was last Published.

Attribute Name	Prompt	Approved	Published	Field Type	Comment	Translations	UI Object	LU Reference
IMPORTANCE	Importance	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Persistent Field	Custom Field Importance	0	Combo...	Supplierimp...

1. Navigate to Application Base Setup > Enterprise > Supplier > Supplier page.
2. Right click on header area Custom Objects > Fields > Create New
3. Create an Enumeration Type field by using SupplierImportance enumeration created in the previous step.
4. Right click and select Approve.
5. Right click and select publish.
6. Navigate to Application Base Setup > Enterprise > Supplier > Supplier page.
7. Right click on the header area > Custom Objects > Reload Configuration.
8. Right click on the header area > Properties > Layout Tab > Check Use customized screen layout > Enable Importance enumeration.
9. Use the Object Arrow keys to position it correctly.



FEEDBACK ON THE OBJECTIVES OF THIS LESSON



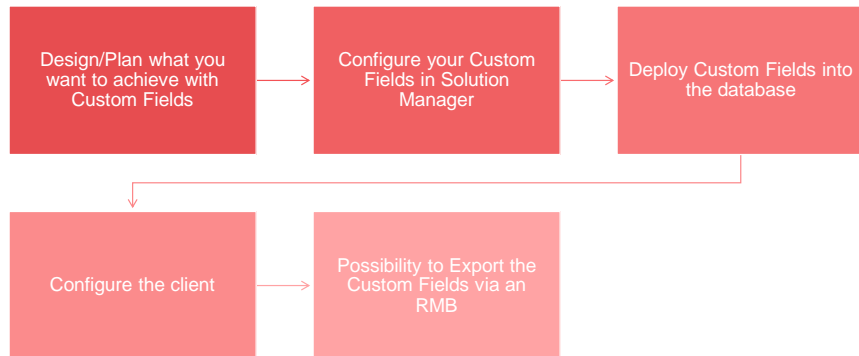
Lesson objectives:

- ...
- ...



CUSTOM FIELDS

WORKFLOW



CUSTOM FIELDS

ADD NEW FIELD – CUSTOM ENUMERATION

■ Name your Custom Logical Unit

Custom Enumeration

Name: Prompt: Comment:

☐ Approved Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.

☐ Published Publish will generate and deploy all server objects, thus making it available to be used and referenced from the client and other objects. When publishing, only the Approved items will be included.

Published Date: None available

☐ Synchronized Synchronized means that no changes has been made to the configuration since it was last Published.

+	DB Value	Client Value	Seq No	Approved	Published	Translations
---	----------	--------------	--------	----------	-----------	--------------



CUSTOM FIELDS

ADD NEW FIELD – CUSTOM ENUMERATION

■ Add Attributes for your Enumeration

Custom Enumeration

Name:	Prompt:	Comment:
<input type="text" value="NewReasonCode"/>	<input type="text" value="Reason Code"/>	<input type="text" value="Enumeration for the Reason Code"/>

- ☐ Approved Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.
- ☐ Published Publish will generate and deploy all server objects, thus making it available to be used and referenced from the client and other objects. When publishing, only the Approved items will be included.
- Published Date: None available
- ☐ Synchronized Synchronized means that no changes has been made to the configuration since it was last Published.

+	DB Value	Client Value	Seq No	Approved	Published	Translations
+	TUNE_UP	Tune Up	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0
	RETURN	Return	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0
	PRODU...	Production	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0



CUSTOM FIELDS

ADD NEW FIELD – CUSTOM ENUMERATION

- Approve: Most custom objects have an **Approved** flag to control if an object is **ready for publishing** and to temporarily disable parts of a configuration.

Custom Enumeration

Name: Prompt: Comment:

☐ Approved Approve to make available for Publishing. While not approved it will be excluded from all publishing.

☐ Published Publish will generate objects. When published, objects will be available to be used and referenced.

☐ Synchronized Synchronized means that the configuration is up to date with the current state of the system.

Published Date:

Synchronized Date:

DB Value	Client Value	Seq No	Approved
TUNE_UP	Tune Up	1	<input checked="" type="checkbox"/>
RETURN	Return	2	<input checked="" type="checkbox"/>
PRODUCE	Production	3	<input checked="" type="checkbox"/>

Approve Enumeration - IFS Applications

Information Message

Approving NewReasonCode will make it available for publishing.

OK Cancel



CUSTOM FIELDS

ADD NEW FIELD – CUSTOM ENUMERATION

■ Approved

Custom Enumeration

Name:	Prompt:	Comment:
<input type="text" value="NewReasonCode"/>	<input type="text" value="Reason Code"/>	<input type="text" value="Stores a Reason Code"/>
<input checked="" type="checkbox"/> Approved	Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.	
<input type="checkbox"/> Published	Publish will generate and deploy all server objects, thus making it available to be used and referenced from the client and other objects. When publishing, only the Approved items will be included.	
	Published Date: None available	
<input type="checkbox"/> Synchronized	Synchronized means that no changes has been made to the configuration since it was last Published.	

+	DB Value	Client Value	Seq No	Approved	Published	Translations
+	TUNE_UP	TuneUp	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0
	RETURN	Return	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0
	PRODU...	Production	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0



CUSTOM FIELDS

ADD NEW FIELD – CUSTOM ENUMERATION

- Publish: When publishing a Custom Object, Database Objects are generated from the configuration meta data and deployed to the database.

Custom Enumeration

Name: Prompt: Comment:

☒ Approved Approve to make available for Publishing. While not approved it will be excluded from the list of available objects.
☐ Published Publish will generate objects. When published, the objects will be available to be used.
☐ Synchronized Synchronized with the database since it was last published.

☒ TUNE_UP Tune Up 1 ☒
☒ RETURN Return 2 ☒
☒ PRODUCE Production 3 ☒

Actions: Approve, Disapprove, Publish, Unpublish, Translate..., Export, View, Populate, Search in Context..., Edit, Subscription

Information Message

Publishing NewReasonCode will deploy into the database making the defined attributes available for Custom Objects that reference it.
This operation may take several minutes to complete!

OK Cancel



CUSTOM FIELDS

ADD NEW FIELD – CUSTOM ENUMERATION

Published and Synchronized

Custom Enumeration ✕

Name:	Prompt:	Comment:
<input type="text" value="NewReasonCode"/>	<input type="text" value="Reason Code"/>	<input type="text" value="Enumeration for the Reason Code"/>

☒ Approved Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.

☒ Published Publish will generate and deploy all server objects, thus making it available to be used and referenced from the client and other objects. When publishing, only the Approved items will be included.

Published Date: 9/4/2018 at 12:11:38 PM

☒ Synchronized Synchronized means that no changes has been made to the configuration since it was last Published.

+	DB Value	Client Value	Seq No	Approved	Published	Translations
▶	TUNE_UP	Tune Up	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0
	RETURN	Return	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0
	PRODU...	Production	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0



CUSTOM FIELDS

ADD NEW FIELD – CUSTOM ENUMERATION

- Once published it can be viewed in Custom Objects page

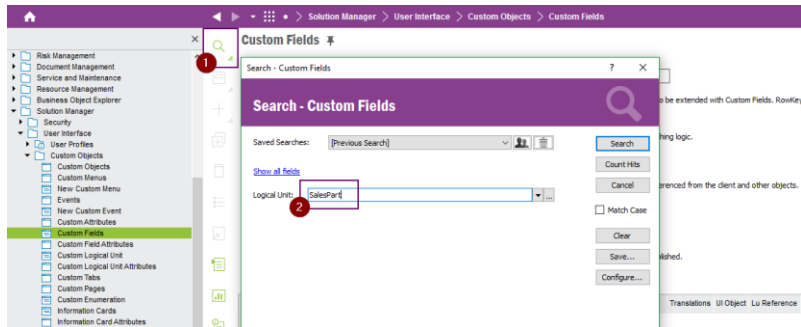
Custom Objects

Name	Informati...	Application Object	Type	Title	Approved	Published	Po id	Application Configuration Packa...
NewReasonCode	Enumera...	NewReasonCode	Custom Enumeration	Reason Code	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<div>Show details... Manage Grants... Populate</div>								



CUSTOM FIELDS

ADD CUSTOM FIELD TO LOGICAL UNIT



CUSTOM FIELDS

CHOOSE DATA TYPE

Create Custom Fields Attribute ? X

Data Type

☐ Text The Data Type decides what kind of data that can be displayed in the Custom Field.

☐ Number * Text, Number, Date and Reference will result in a data field representation.

☐ Date * Enumeration will be represented by combo box or check box, depending on its Format.

☒ Enumeration

☐ Reference

Format:

☒ Normal Enumeration with a fixed number of static values, represented as a combo box object.

☐ Boolean Enumeration with exactly two static values, represented as a check box object.

☐ Multi Enumeration represented as a multiple value selection control.

Cancel < Previous Next > Finish



CUSTOM FIELDS

CONNECT ENUMERATION TO ATTRIBUTE

Create Custom Fields Attribute

Enumeration Reference

The Logical Unit representing the Enumeration which will be displayed as a combo box object.

☒ Use Existing Enumeration

Enumeration:

☐ Create New Enumeration

Name:

Enumeration Browser

Enumeration	Component	Custom Enumeration
NetChangeBudgetControl	PROJBF	<input type="checkbox"/>
NetInv	DOP	<input type="checkbox"/>
NewReasonCode	CONFIG	<input checked="" type="checkbox"/>
NetCalendar	FINDBAS	<input type="checkbox"/>
NetInventoryPartType	ORDER	<input type="checkbox"/>
NetSerialLocation	WO	<input type="checkbox"/>
NetMailClouding	WRKSCH	<input type="checkbox"/>
NotificationType	OSHA	<input type="checkbox"/>



CUSTOM FIELDS

ADD NEW FIELD ENUMERATION

Create Custom Fields Attribute ? X

Properties

<input checked="" type="checkbox"/> Insertable	Makes it possible to enter a new value. Default Value: <input type="text" value="TUNE_UP"/>
<input checked="" type="checkbox"/> Updatable	Makes it possible to change an existing value.
<input type="checkbox"/> List of Values	Includes the field in the 'List of Values', when referenced.
<input checked="" type="checkbox"/> Searchable	Makes the field searchable, including it in the 'Search Dialog'.
<input type="checkbox"/> Indexed	Creates an index for the field, improving the search performance.

Cancel < Previous Next > Finish



CUSTOM FIELDS

ADD THE CUSTOM ATTRIBUTE TO NON-BASE VIEWS

Custom Fields - SalesPart

Logical Unit: Prompt: Module:

☒ RowKey Activated Activating the RowKey for a Logical Unit (LU) is the first step in order to make an LU possible to be extended with Custom Fields. RowKey is the functionality that connects standard LUs with the extended Custom Field LUs.

☐ Approved Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.

☐ Published Approved Detail Views: [None \(of totally 28\)](#)
Publish will generate and deploy all server objects, thus making it available to be used and referenced from the client and other objects. When publishing, only the Approved Items will be included.
Published Date: None available
Presentation Object: None available

☐ Synchronized Synchronized means that no changes has been made to the configuration since it was last Published.

Attribute Name	Prompt	Approved	Published	Field Type	Comment	Translations	UI Object	Lu Reference	Low View	Custom Field Impl Type	Expression	Metadata	Data Type	Format	Data
REASON_CD_	Reason Code	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Persistent Field	Reason Code for Sales Part - Available ...	0	Combo	NewReaso...		Not Applicable			Enumer...	Unformatted	20

Your new Custom Enumeration is now visible in the Attribute List.

Important:

By default only the main view for a Logical Unit will be extended with the new custom attribute. In some cases the logical unit may have multiple views(As shown in the slide) used by different application pages. In these cases you have to approve the relevant detail views manually.

Will be investigating this in a later slide.

CUSTOM FIELDS

APPROVE CUSTOM FIELDS FOR LOGICAL UNIT

Custom Fields - SalesPart

Logical Unit: Prompt: Module:

☒ RowKey Activated Activating the RowKey for a Logical Unit (LU) is the first step in order to make an LU possible to be extended with Custom Fields. RowKey is the functionality that connects standard LUs with the extended Custom Field LUs.

☐ Approved None (of totally 28)

☐ Published

☐ Synchronized

☐ Attribute Name

☒ REASON_CO...

Information Message

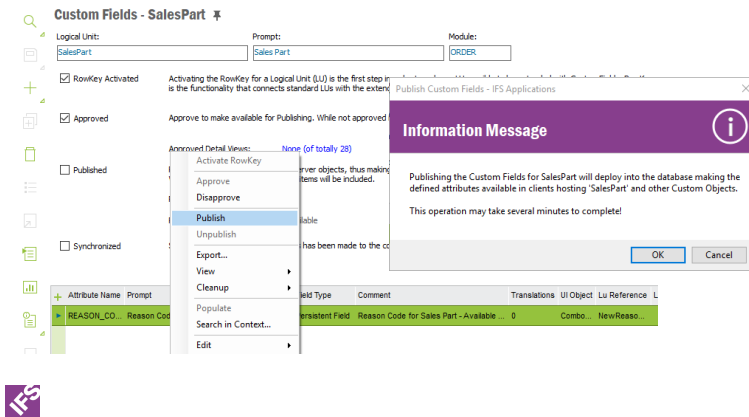
Approving Custom Fields for SalesPart will make them available for publishing.

OK Cancel

Published	Field Type	Comment	Translations	UI Object	Lu Reference
<input type="checkbox"/>	Persistent Field	Reason Code for Sales Part - Available	0	Combo...	NewReaso...

CUSTOM FIELDS

PUBLISH CUSTOM FIELDS FOR LOGICAL UNIT



Your new Custom Enumeration is now visible in the Attribute List.

Important:

By default only the main view for a Logical Unit will be extended with the new custom attribute. In some cases the logical unit may have multiple views(As shown in the slide) used by different application pages. In these cases you have to approve the relevant detail views manually.

Will be investigating this in a later slide.

CUSTOM FIELDS

PRESENTATION OBJECT FOR CUSTOM FIELD

The screenshot displays the IFS application interface for configuring custom fields. The main window is titled 'Custom Fields - SalesPart'. It features a left sidebar with icons for various actions like 'Logical Unit', 'RowKey', 'Approved', 'Published', and 'Synchronized'. The main area is divided into two panes. The top pane, titled 'Presentation Object', shows the configuration for a presentation object named 'cfSalesPart'. It includes fields for 'Entry Type' (Manual), 'Object Type' (ORDER), 'Module' (Customer Orders), and 'Layer' (Core). The bottom pane, titled 'Database Objects', shows a table of database objects and their dependencies.

Database Object	Object Type	Sub Type	Entry Type	Force Read Only	Read Only
NEW_REASON_CODE_CFPEnumerate	Method	Method	Manual	<input type="checkbox"/>	<input checked="" type="checkbox"/>
SALES_PART_CFPcf_Modify__	Method	Method	Manual	<input type="checkbox"/>	<input type="checkbox"/>
SALES_PART_CFPcf_New__	Method	Method	Manual	<input type="checkbox"/>	<input type="checkbox"/>
SALES_PART_CFV	View	Base View	Manual	<input type="checkbox"/>	<input type="checkbox"/>

Below the table, there is a list of attributes for the 'REASON_CD' field. The list includes columns for 'Attribute Name', 'Prompt', 'Approved', 'Published', 'Field Type', 'Comment', 'Translations', 'UI Object', 'Lu Reference', and 'L'.

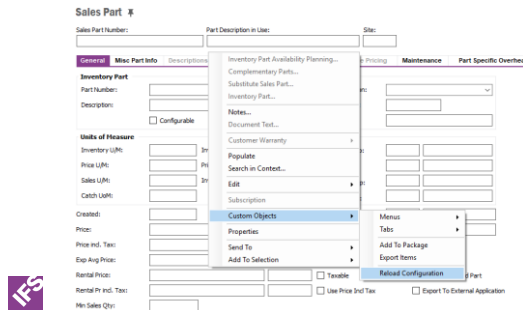
Attribute Name	Prompt	Approved	Published	Field Type	Comment	Translations	UI Object	Lu Reference	L
REASON_CD	Reason Code	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Persistent Field	Reason Code for Sales Part - Available ...	0	Combo...	NewReaso...	

A presentation object will be generated for the custom objects related to the logical Unit. You can use this generated PO to control who can access the custom objects you created.

CUSTOM FIELDS

CLIENT SIDE - RELOAD CONFIGURATION

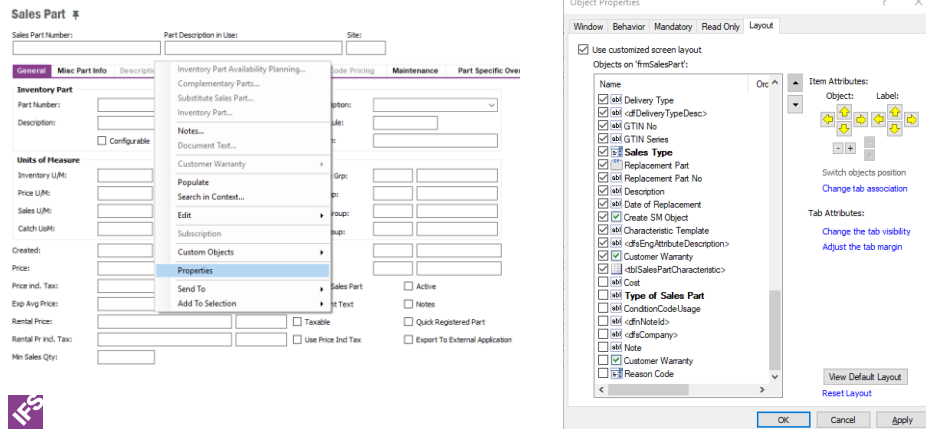
- In column header, right mouse and select Column Chooser
- New custom field will not be listed in the column list.
- Need to use Reload Configuration RMB option.



After the Custom Field is created, in order to enable it on the client side, you have to use the **Reload Configuration** RMB option.

CUSTOM FIELDS

CLIENT SIDE - CHANGE LAYOUT



Then by selecting Properties/Layout you have to add the created Custom field.

CUSTOM FIELDS

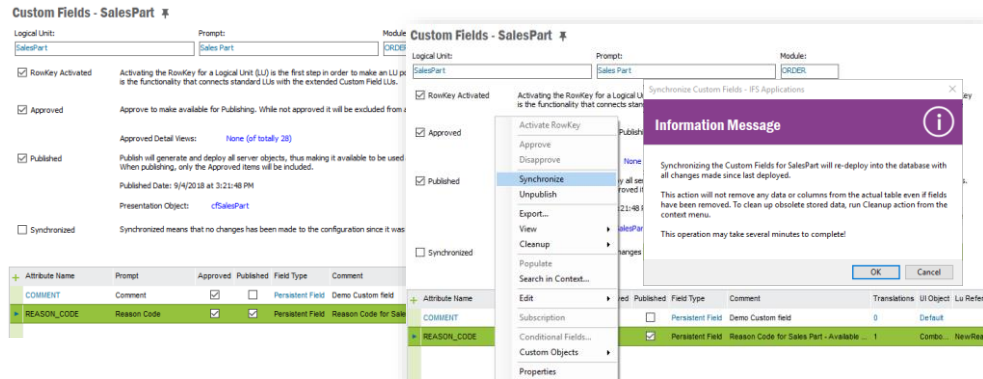
CLIENT SIDE – ADD CUSTOM FIELD TO THE LAYOUT

The screenshot illustrates the steps to add a custom field to a table layout in IFS. The 'Object Properties' dialog is open, showing the 'Layout' tab. A red box highlights the 'Custom Fields' section, and a red arrow points to the 'Add Custom Field' button. Another red box highlights the 'Season Code' field. The 'Column Chooser' dialog is also open, showing the 'Season Code' field being added to the table columns.

On a Table Window, you have to use Choose Column to select the custom field.

CUSTOM OBJECTS

SYNCHRONIZE CUSTOM FIELDS



When you make a change after publishing custom fields, the currently deployed version in the database and the modified custom field are not synced. Use the RMB option Synchronize to synchronize both version to be the same.

CUSTOM FIELDS

APPROVED VIEWS

Custom Fields - SalesPart *

Logical Unit: Prompt: Module:

☒ RowKey Activated Activating the RowKey for a Logical Unit (LU) is the first step in order to make an LU possible to be extended with Custom Fields. RowKey is the functionality that connects standard LUs with the extended Custom Field LUs.

☒ Approved Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.

☒ Published Approved Detail Views: [None \(of totally 28\)](#)
Publish will generate and deploy all server objects, thus making it available to be used as When publishing, only the Approved items will be included.

Published Date: 9/4/2018 at 3:21:48 PM

Presentation Object: [cSalesPart](#)

☐ Synchronized Synchronized means that no changes has been made to the configuration since it was last published.

Attribute Name	Prompt	Approved	Published	Field Type	Comment
COMMENT	Comment	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Persistent Field	Demo Custom field
REASON_CODE	Reason Code	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Persistent Field	Reason Code for Sales

Approved Detail Views for 'SalesPart'

The defined Custom Fields are initially hosted only by the base view. They can also be hosted by one or more detail views owned by the Logical Unit. The detail views are listed below and must first be Approved before the actual Publish operation adds Custom Fields to them.

View Name	Approved	Published	OBJKEY Available	Custom Fields View Name
SALES_PART_LOV2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SALES_PART_LOV2_CFV
SALES_PART_PACKAGE_LOV	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SALES_PART_PACKAGE_LOV_CFV
SALES_PART_PKG_COMP_LOV	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SALES_PART_PKG_COMP_LOV_CFV
SALES_PART_PL_LOV	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SALES_PART_PL_LOV_CFV
SALES_PART_PRICE_TAX_LOV	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SALES_PART_PRICE_TAX_LOV_CFV
SALES_PART_PUB	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SALES_PART_PUB_CFV
SALES_PART_SERVICE_LOV	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	SALES_PART_SERVICE_LOV_CFV
SALES_PART_UOM_LOV	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SALES_PART_UOM_LOV_CFV





FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



CUSTOM INFORMATION CARDS

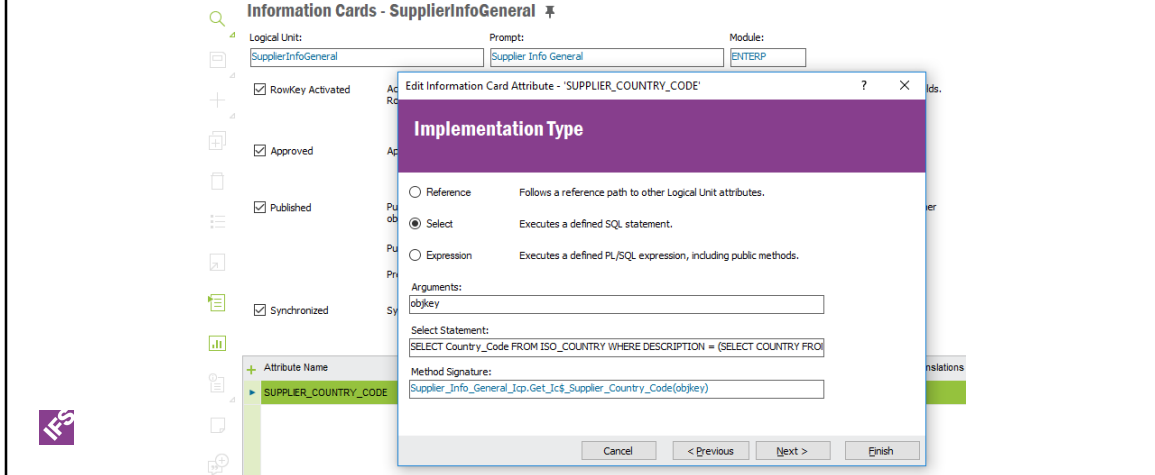
OVERVIEW

- Similar Administration as Custom Fields
- Improved end-user experience by displaying important information on an Information Card
- End-users choose which information to add on an Information Card
- Different Information Cards on each form



CUSTOM INFORMATION CARDS

SUPPLIER COUNTRY CODE - DEMO



1. Navigate to Application Base Setup > Enterprise > Supplier > Supplier page.
2. Right click on header area Custom Objects > Information Cards > Create New.
3. Create SUPPLIER_COUNTRY_CODE information card by providing the following SQL select query. (Specify objkey as an argument)

SELECT Country_Code FROM ISO_COUNTRY WHERE DESCRIPTION = (SELECT COUNTRY FROM SUPPLIER_INFO_GENERAL WHERE objkey = :objkey)

4. Right click and select Approve.
5. Right click and select publish.

This will create Supplier_Info_General_Icp package containing a function (Get_Ic\$Supplier_Country_Code) which returns a value by executing the select statement you provided.

Also it will create a new view named SUPPLIER_INFO_GENERAL_ICV which returns everything from SUPPLIER_INFO_GENERAL view and calls the above function to get a value to IC\$SUPPLIER_COUNTRY_CODE information card.

6. Navigate to Application Base Setup > Enterprise > Supplier > Supplier page.
7. Right click on the header area > Custom Objects > Reload Configuration.
8. Click on the Information Cards button and display the Supplier Country Code.

CUSTOM INFORMATION CARDS

SUPPLIER COUNTRY CODE - DEMO

The screenshot shows the IFS Supplier form for 'Supplier 1 (100)'. The form has tabs for General, Address, Contact, Comm. Method, Message Setup, Invoice, and Paym. The General tab is active, showing fields for Supplier (095210), Name (Icemell Goods), Association No., Importance (Medium), Default Language (English), Country (UNITED KINGDOM), Category (Supplier), Supplier's Own ID, Creation Date (11/16/2010), Form of Business (99), Form of Business Description (general business), Identifier Reference, and ID Reference Validation (None). A pop-up window titled 'Supplier Country Code: GB' is displayed over the form, showing the value 'GB'.

1. Navigate to Application Base Setup > Enterprise > Supplier > Supplier page.
2. Right click on header area Custom Objects > Information Cards > Create New.
3. Create SUPPLIER_COUNTRY_CODE information card by providing the following SQL select query. (Specify objkey as an argument)

```
SELECT Country_Code FROM ISO_COUNTRY WHERE DESCRIPTION = (SELECT  
COUNTRY FROM SUPPLIER_INFO_GENERAL WHERE objkey = :objkey)
```

4. Right click and select Approve.
5. Right click and select publish.

This will create Supplier_Info_General_Icp package containing a function (Get_Ic\$_Supplier_Country_Code) which returns a value by executing the select statement you provided.

Also it will create a new view named SUPPLIER_INFO_GENERAL_ICV which returns everything from SUPPLIER_INFO_GENERAL view and calls the above function to get a value to IC\$_SUPPLIER_COUNTRY_CODE information card.

6. Navigate to Application Base Setup > Enterprise > Supplier > Supplier page.
7. Right click on the header area > Custom Objects > Reload Configuration.
8. Click on the Information Cards button and display the Supplier Country Code.

CUSTOM INFORMATION CARDS

CUSTOM FIELDS VS INFORMATION CARDS

CUSTOM FIELDS

- Support business process needs
- Additional data capture
- Validation
- Event Workflow

INFORMATION CARDS

- Improved usability of reference information
- Reduced Screen "hopping"
- Individual user configuration





FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



CUSTOM LOGICAL UNITS

OVERVIEW

- How Custom Logical Units are configured to provide storage of business objects to IFS Applications.
- How Custom Pages are configured to provide table window pages to IFS Applications to show data stored in Custom Logical Units.
- How Custom References are configured to provide references between Logical Units in IFS Applications.



CUSTOM LUS, PAGES, TABS & REFERENCES

OVERVIEW

Custom Logical Units

- Configure and add storage of business objects to IFS Applications

Custom Pages

- Configure and add table window pages to IFS Applications to show data stored in Custom Logical Units.

Custom Tabs

- Custom Tabs lets you attach pages as additional tabs to any tab form window

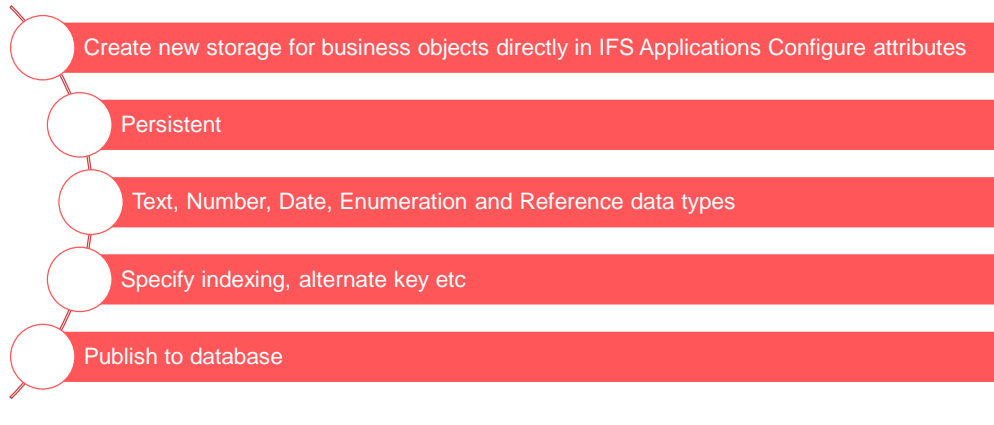
Custom References

- Configure and add references between Logical Units in IFS Applications. This is typically available to end users as List of Value (LOV) dialogs.



CUSTOM LOGICAL UNITS

OVERVIEW



Purpose of slide: Introduce Custom Logical Units

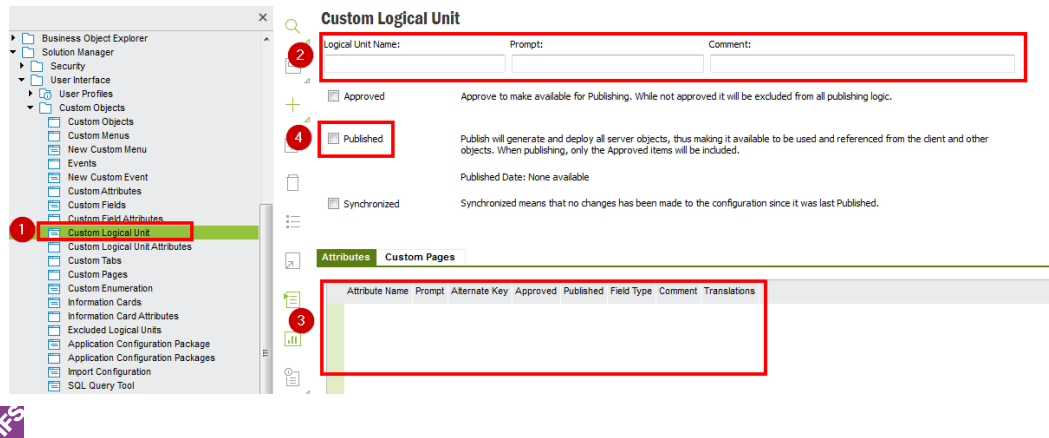
Things to say: Logical Units store business info in IFS Applications. They're typically packages with tables, views and methods. Custom Logical Units are created dynamically, and used to extend instances of IFS Applications with new business objects.

To a Custom Logical Unit, attributes are associated. These build on Custom Fields, so many of the available types and settings are available. There are some restrictions for technical reasons.

Custom Logical Units always use a generated key attribute, called ObjKey. It's possible to set an alternate key, though, by selecting the attribute(s) that should be part of it.

CUSTOM LOGICAL UNITS

CREATE A CUSTOM LOGICAL UNIT



Custom Logical Unit

Logical Unit Name: _____ Prompt: _____ Comment: _____

☐ Approved Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.

☐ Published Publish will generate and deploy all server objects, thus making it available to be used and referenced from the client and other objects. When publishing, only the Approved items will be included.

Published Date: None available

☐ Synchronized Synchronized means that no changes have been made to the configuration since it was last Published.

Attributes

Attribute Name	Prompt	Alternate Key	Approved	Published	Field Type	Comment	Translations
----------------	--------	---------------	----------	-----------	------------	---------	--------------

Purpose of slide: Introduce Custom Logical Units

Things to say: Logical Units store business info in IFS Applications. They're typically packages with tables, views and methods. Custom Logical Units are created dynamically, and used to extend instances of IFS Applications with new business objects.

To a Custom Logical Unit, attributes are associated. These build on Custom Fields, so many of the available types and settings are available. There are some restrictions for technical reasons.

Custom Logical Units always use a generated key attribute, called ObjKey. It's possible to set an alternate key, though, by selecting the attribute(s) that should be part of it.

CUSTOM LOGICAL UNITS

CREATE CUSTOM LOGICAL UNITS

Custom Logical Unit - TrnCompanyCars

Logical Unit Name: TrnCompanyCars Prompt: Company Cars Comment: Custom LU to store Company Cars

☒ Approved

☒ Published

☒ Synchronized

1 SELECT * FROM trn_company_cars_clt

SELECT * FROM trn_company...

Page Size: 100 Total Rows: 0 Page: 1 of 0 Matching Rows:

#	ROWKEY	CF\$_BRAND	CF\$_MODEL	CF\$_YEAR	ROWVERSION

Attributes Custom Pages

Attribute Name	Prompt	Alternate Key	Approved	Published	Field Type	Comment	Translations
BRAND	Brand	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Persistent Field	Brand column - Custom LU to store Company ...	0
MODEL	Model	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Persistent Field	Model column - Custom LU to store Company ...	0
YEAR	Year	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Persistent Field	Custom LU to store Company Cars	0

Purpose of slide: Introduce Custom Logical Units

Things to say: Logical Units store business info in IFS Applications. They're typically packages with tables, views and methods. Custom Logical Units are created dynamically, and used to extend instances of IFS Applications with new business objects.

To a Custom Logical Unit, attributes are associated. These build on Custom Fields, so many of the available types and settings are available. There are some restrictions for technical reasons.

Custom Logical Units always use a generated key attribute, called ObjKey. It's possible to set an alternate key, though, by selecting the attribute(s) that should be part of it.

1. Navigate to Solution Manager > User Interface > Custom Objects > Custom Logical Unit page.
2. Create TrnCompanyCar Logical Unit with following fields.
REGISTRATION_NO - Alternate Key
MAKE

MODEL

YEAR

3. Right click and select Approve.

4. Right click and select publish.

This will create the following table view and packages.

Table - TRN_COMPANY_CAR_CLT

View - TRN_COMPANY_CAR_CLV

Package - TRN_COMPANY_CAR_CLP

CUSTOM LOGICAL UNITS AND PAGES

CREATE CUSTOM PAGES

- Create Custom Pages from Custom Logical Unit

Custom Logical Unit - TrnCompanyCars

Logical Unit Name: TrnCompanyCars Prompt: Company Cars Comment: Custom LU to store Company Cars

☒ Approved Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.

☒ Published Publish will generate and deploy all server objects, thus making it available to be used and referenced from the client and other objects. When publishing, only the Approved items will be included.

Published Date: 4/26/2016 at 11:46:48 AM

☒ Synchronized Synchronized means that no changes has been made to the Logical Unit.

Attributes	Custom Pages
Page Name	Page Title Page Type
cpgCompanyCarsDetail	Company Cars Form Window
cpgCompanyCars	Company Cars Table Window

Page Name	Page Title	Page Type	Logical...	Comment	Application Configuration Package
cpgComp...	Compa...	Table Window	TrnCo...	Custom...	
cpgComp...	Compa...	Form Window	TrnCo...	Custom...	

Purpose of slide: Introduce Custom Pages

Things to say: From a published Custom Logical Unit, it's possible to create dynamically a Custom Page as a table window. It's simply a name and title.

Notes:

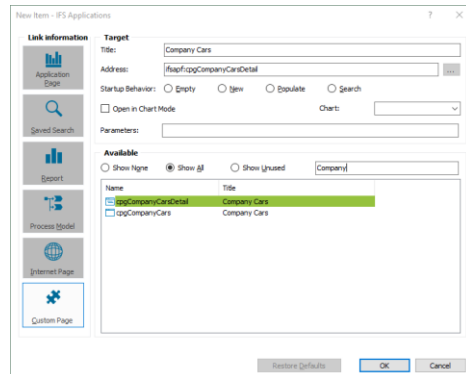
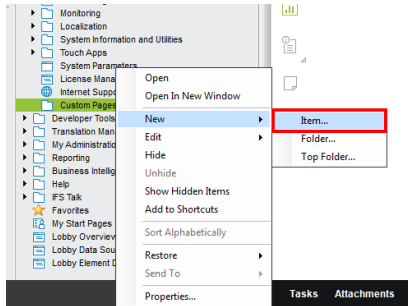
Use RMB option **Manage Grants** and grant the page to the permission sets that should include this page. All user granted to permission set *CUSTOM_OBJECTS_ADMIN* will be granted the page automatically.

1. Navigate to the Custom Pages tab of the TrnCompanyCar logical unit.
2. Create following two pages.
 - cpgCompanyCarDetails - Company Car - Form Window
 - cpgCompanyCars - Company Cars - Table Window

CUSTOM LOGICAL UNITS AND PAGES

ADD CUSTOM PAGE TO NAVIGATOR

- Add Custom Page to Navigator or Shortcuts



Purpose of slide: Show how Custom Pages can be added for access from IFS Applications

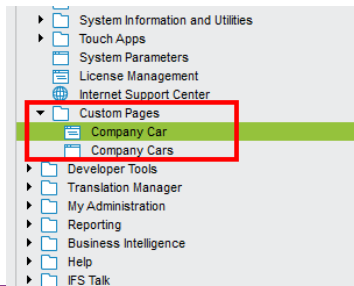
Things to say: Custom Pages can be added to the Navigator or Shortcuts, to be used by normal users, using ordinary features in IFS Applications

1. Right click on navigator area > New > Item.
2. Click on Custom Page.
3. Click on Show All radio button to populate available custom pages.
4. Click on a page you want to add and create it in the navigator.

CUSTOM LOGICAL UNITS AND PAGES

ACCESSING CUSTOM PAGES

- Use like ordinary form window or table window page



Two screenshots of the 'Company Cars' form. The top screenshot shows the search criteria: Brand (Volvo), Model (C30), and Year (2006). The bottom screenshot shows the resulting table of cars.

Brand	Model	Year
Volvo	C30	2006
BMW	M5	2015

Purpose of slide: Show a Custom Page

Things to say: Custom Pages are shown as table windows in IFS Applications. They work like ordinary table windows, with searching, sorting, conditional formatting, send to, selections and all.

CUSTOM REFERENCES

DEFINE CUSTOM REFERENCES

- Create Custom Reference from page where it should be shown using Custom Fields

The screenshot shows the '10 - Employee File' form in the IFS system. The form has several tabs: 'Summary', 'Personal', 'Employment', and 'Disciplinary and Grievance'. The 'Summary' tab is active. A context menu is open over the 'Summary' tab, showing various actions like 'Transfer Employee Data...', 'Generate Document...', 'Set Employee Status...', etc. The 'Custom Objects' option is selected, which opens a sub-menu. In this sub-menu, 'Fields' is selected, which opens another sub-menu. In this final sub-menu, 'Create New...' is highlighted. The IFS logo is visible in the bottom left corner.

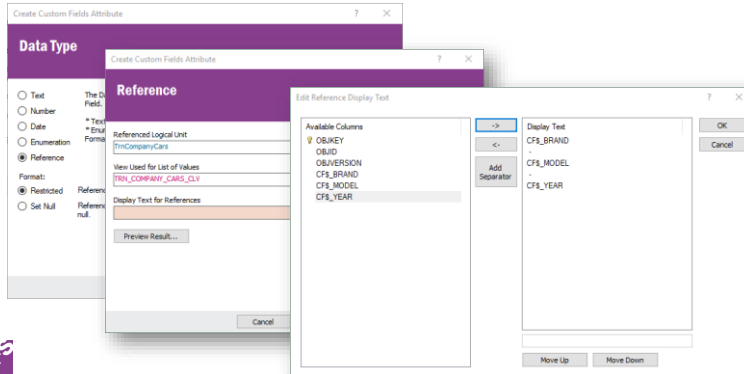
Purpose of slide: Introduce Custom References

Things to say: In order to use Custom Logical Units and Custom Pages properly, it must be possible to reference the data. A Custom Reference is added as a Custom Field. The easiest way is just to do it from the form where you need the List of Value (LOV) dialog to chose the reference.

CUSTOM REFERENCES

DEFINE CUSTOM REFERENCES

- Use Custom Fields wizard to configure the Custom Reference



Purpose of slide: Brief view of how it works

Things to say: When adding Custom Fields, a wizard dialog is used. This has been augmented with functionality to add use Custom References. Both ordinary Logical Units and Custom Logical Units can be targets for Custom References. The display text can be configured easily using an edit dialog. In this example it's configured to show data from three of the attributes on the referenced Logical Unit.

CUSTOM REFERENCES

ADD CUSTOM REFERENCES

- Then add List of Value (LOV) dialog field to the page using the Custom Screen Layout functionality

10 - Employee File #

Emp No: Person ID: Employee Name: Employee Status: ☐ Valid Data Processing Purpose

Insurance ID: Organization Code: Org Code Name: Position ID: Position Title:

Summary Personal Employment Contract

Object Properties ? X

Window: Behavior: Mandatory: Read Only: Layout

☒ Use custom screen layout (Object is in the TabContainer)

Name	Order	Item Attributes
allEmployeeName	1	Object Label
allSpec	1	
allName	1	
allCompany	1	
allOffice	1	
allPosition	1	
allOrganization	1	
allJob	2	
allSupervisor	2	
allWorkPhone	2	
allWorkFax	2	
allWorkMobile	2	
allWorkEmail	2	
allAddress	2	
allSignature	2	
allEmployeeCategory	2	
allEmploymentType	2	
allEmploymentDate	2	
allTermination	2	
allDataProcessingPurpose	2	
allCompanyCap	2	

Search object position:

Change tab association:

Tab Attributes:

Work Phone:
Work Fax:
Work Mobile:
Work E-Mail:

IFS

Purpose of slide: Putting a Custom Reference field on a page

Things to say: When the Custom Reference attribute has been published, it can be put on the form it should be shown on using Custom Screen Layout.

CUSTOM REFERENCES

USAGE OF CUSTOM REFERENCES

- Use the List of Value (LOV) dialog to select value

10 - Employee File - 1004 Alexander Danielsson ✓ 3 (100) ✕

Emp No:	Person ID:	Employee Name:	Employee Status:	<input type="checkbox"/> Valid Data Processing Purpose
1004	ALEX	Alexander Danielsson	Active	
Insurance ID:	Organization Code:	Org Code Name:	Position ID:	Position Title:
	CORFIN	Cooperative Finance	SUP1	Supervisor
				Company Car:

Summary Personal Employment Compensation Qualifications Incidents Dis

Basics Address Contact Persons Documents Dependents

Name:

First Name:	Middle Name:	Prefix:	Last Name:
Alexander			Danielsson
Birth Name:	Alias:	Initials:	Title:
Internal Display Name:	External Display Name:		
Alexander Danielsson	Alexander Danielsson		

Misc Physical Personality

Date of Birth:	Place of Birth:	Citizen of:
Gender:	Blood Type:	Marital Status:

Company Car - IFS Applications

Company Car

Saved Searches:

Display Text

- Volvo - C30 - 2006
- BMW - M5 - 2015
- Tesla - Model X - 2016
- Ferrari - Italia - 2016

OK Cancel Populate Search...

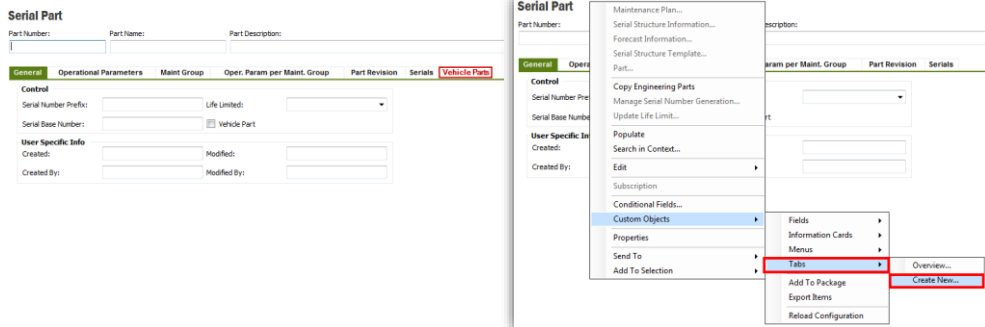
Purpose of slide: Show LOV dialog with Custom Reference

Things to say: Custom References uses standard LOV dialogs. Note that the display text is shown both in the form and in the LOV dialog

CUSTOM TABS

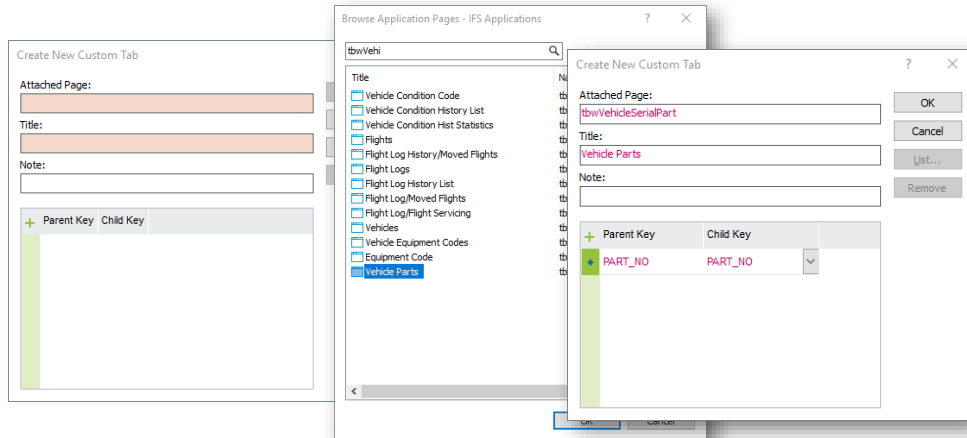
OVERVIEW

- Custom Tabs lets you attach pages as additional tabs to any tab form window.
- The attached pages can be both existing application pages or Custom Pages.



CUSTOM TABS

CREATE CUSTOM TABS



Things to say: If the host page has key columns that match columns in the attached page, you will get the standard master detail behavior automatically. If the column names differ you need to specify the column in the table manually.

CUSTOM TABS

CUSTOM TAB ATTACHED TO THE PAGE

Serial Part - AIR-2000 ∨ 61 (100)

Part Number: AIR-2000 Part Name: Aircraft Part Description: AIR-2000

General	Operational Parameters			Maint Group	Oper. Param per Maint. Group		Part Revision	Serials	Vehicle Parts	
Part Number	Part Name	Part Description	Primary Oper Param	Primary Oper Param Desc	Vehicle ID Mandatory	Vehicle ID as Object ID	Created	Created By	Modified	Modified By
▶ AIR-2000	Aircraft	AIR-2000	DEV-FHR	Flying Hours	<input checked="" type="checkbox"/>	<input type="checkbox"/>	11/10/2015 9:29:58 AM	VIM	4/5/2016 10:32:33 AM	TRJOLK



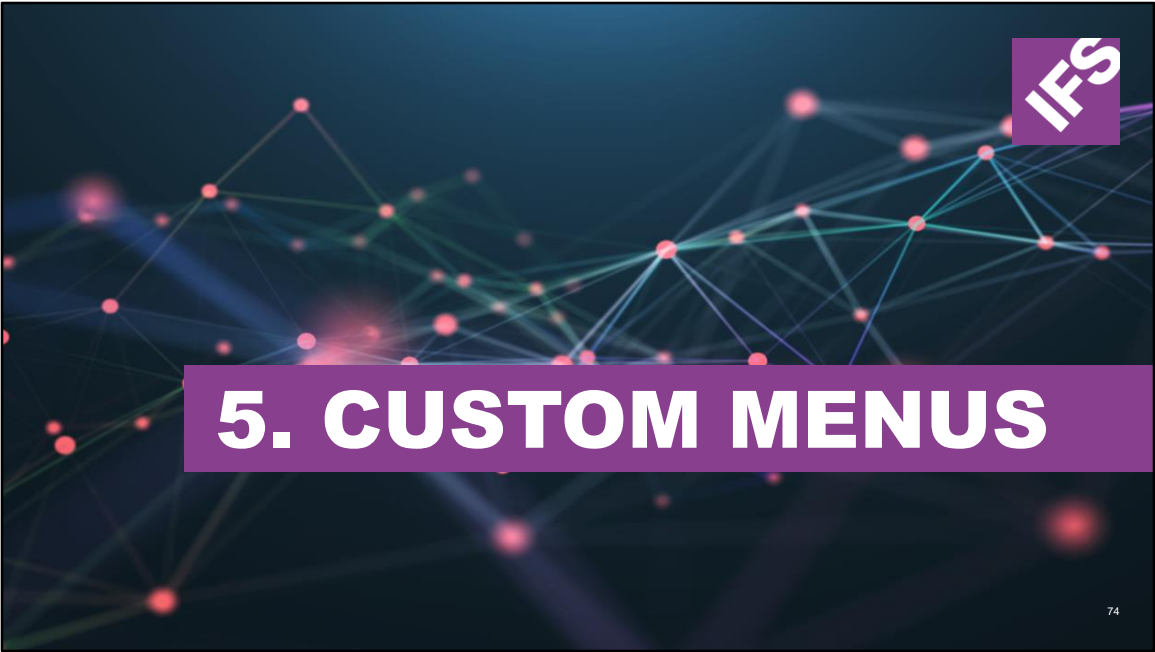


FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



CUSTOM MENUS

LESSON OBJECTIVES

- How Custom Menus work in IFS Applications.
- How to configure a custom menu item for an IFS Application Window.
- The different types of custom menu items available.
- How to import and export custom menu configurations to different installations.



CUSTOM MENUS

OVERVIEW

Allows customization of right-click menus without rewriting the application

Items that can be added to any window's popup menu

- Create Window
- Create Window with transfer
- Report
- Quick Report
- URL
- External Program
- PL/SQL Block
- User Method
- State Machine
- Separator - (a horizontal dividing line across the popup menu)

User can choose to see original menu items, custom items, or both

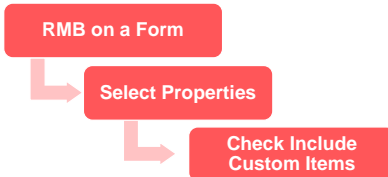
Export/Import functionality



CUSTOM MENUS

OVERVIEW

- To see Custom Menu items for a form,



The screenshot shows the 'Customer Order' form with the 'Object Properties' dialog box open. The dialog box has tabs for 'Window', 'Behavior', 'Mandatory', 'Read Only', 'Layout', and 'Customer Order'. The 'Customer Order' tab is selected, showing the 'Menu Configuration' section. In this section, the 'Include Custom Items' checkbox is checked, and the 'Include Original Items' checkbox is also checked. The 'Advanced...' button is visible next to the 'Include Custom Items' checkbox. The 'Default' button is at the bottom right of the dialog box. The background form shows fields for 'Order No.', 'Customer Name', 'Site', 'Coordinator', 'Order Type', 'Currency', 'Priority', 'Reference', 'Refer', 'Delivery Address', 'Delivery Address Name', and 'Document'.

CUSTOM MENUS

CUSTOM MENU ITEMS

Create window

- Opens a new window with no transfer of data

Create window with transfer

- Opens a new window and transfers the specified rows
- The receiving window MUST accept data transfer
- Overview windows are usually NOT suitable
- Detail windows are usually suitable

Report

- The Order Report dialog is opened and populated with the report specified in the Parameter field

Quick Report

- The Quick Report specified in the Parameter field will be executed



CUSTOM MENUS

CUSTOM MENU ITEMS

URL - Performs a navigate to the URL specified in the Parameter field

User method

- Invokes the method UserMethod
- Knowledge of the internal structure is needed

External Program - Starts an external program

PL/SQL Block - Executes a PL/SQL Block

State machine

- Specifies an event, E.g. releasing an order
- Note: internal columns (hidden) objevents, objstate must be present in window



Separator - Used for grouping menu items

CUSTOM MENUS

CREATE WINDOW WITH TRANSFER

custom me

- Sales
 - Customer Schedule
 - Customer Schedule Merged Line
 - Customer Schedule Merged Lines
- Solution Manager
- User Interface
 - Custom Objects
 - Custom Menus
 - New Custom Menu**

New Custom Menu

Menu ID: 770921 Description:

Window:

Action: Create Window with Transfer Menu Text:

Action Parameters

Destination Window:

Key Translation:

Source Key Name	Dest Key Name
<input type="text"/>	

Presentation Object:

Process: NONE

Menu Item Order:

Condition:

Create Window With Transfer example (1/5)

Solution Manager browse to *Configuration /Custom Objects/Menus/New Custom Menu*

CUSTOM MENUS

CREATE WINDOW WITH TRANSFER

The screenshot shows the 'New Custom Menu' configuration form. Red boxes and lines highlight specific fields with corresponding text annotations:

- Menu ID:** 770921
- Description:** Custom Menu Item on Customer Order context Customer Order Lines
- Object:** (empty)
- Window:** frmCustomerOrder (Annotated: "Name of the window we are attaching the custom menu item")
- Action:** Create Window with Transfer (Annotated: "Text that will appear as the menu option")
- Menu Text:** Summary - Overview Order Lines (Annotated: "Text that will appear as the menu option")
- Action Parameters:**
 - Destination Window:** tbwOverviewCustOrdLine (Annotated: "Name of the destination form window")
- Key Translation:** (Empty table with columns Source Key Name and Dest Key Name)
- Condition:** (Empty field)
- Translations:**

Language	Title
en - English	Summary - Overview Order Lines

(Annotated: "Translations of the Menu Text")

Create Window With Transfer example

The **Window** field is the form from which this Custom Menu items is available. *frmCustomerOrder* is the Customer Order detail screen.

The **Action Type** is set to *Create window with transfer* in this example.

The **Menu Item Order** section controls the order in which the menu item is listed in the RMB popup window. Arrange the menu items as desired.

The **Parameter** field is set to the name of the window we want to open and transfer the parameter value to. *tbwOverviewCustOrdLine* is the Overview Customer Order Line screen.

The **Menu Text** field allows you to specify the display that is shown on the RMB popup window. The en translation will default in to the Translations section. This same Custom Menu item can be defined with a different display text for each language in use.

CUSTOM MENUS

CREATE WINDOW WITH TRANSFER

In the Key Translations section, add the fields to pass as parameters

Custom Menu

Menu ID: 210306 Description: Custom menu item on Customer Order context menu to show the Customer Order Lines

Window: frmCustomerOrder

Action Type: Create Window with Transfer Menu Text: Summary - Overview Order Lines

Action Parameters: Destination Window: tbwOverviewCustOrdLine

Key Translation:

Source Key Name	Destination Key Name
CUSTOMER_NO	CUSTOMER_NO

Map keys on the Source window with Destination window

Menu Item Order:

- Summary - Overview Order ...
- Customer 360 View
- Reserve All Order Lines
- Order Flow Handling
- Contribution Margins
- All Customer Finance Details
- Print Customer Invoice
- External Vendor Invoice

Rearrange the order of the Custom Menu items

Condition:

Translations:

Language	Title
en - English	Summary - Overview Order Lines
sv - Swedish	Sammanfattning - Översikt Orderrader

Add translations

Create Window With Transfer example (4/5)

In the **Key Translations** section, enter details of the parameter value which will be transferred to the target window.

In this example, we are sending the *CUSTOMER_NO* value from frmCustomerOrder and matching it to the *CUSTOMER_NO* value in the receiving window (tbwOverviewCustOrdLine). If the keys of the source and the keys of the destinations are the same, then no need to specify key translations. (But in this example we will demonstrate how to add key translations)

CUSTOM MENUS

EXAMPLE – HTTP URL

Custom Menu

Menu ID: 508207	Description: Custom menu to search customer	Presentation Object: frmCustomerOrder	Process: NONE
Window: frmCustomerInfo	Browse...		
Action Type: URL	Menu Text: Search for Customer on Google	Menu Item Order: Customer 360 View View in Map Customer Order Lines Delivery Statistics AR Payment Details Invoiced Sales Credit Management Search for Customer on Google	
Action Parameters Url: http://www.google.com/search?q=&NAME			
Open...		Move Up Move Down Add Separator Add New Item	



Adding a HTTP URL example (1/2)

A web browser is opened for the defined URL address.

Parameters can be used in the URL call, using **&Parameter** to represent context sensitive values from the window (e.g. &NAME means the customer name in this example), and **#Parameter#** to represent general environment parameters (e.g. #USER_COMPANY# means the current user's Company value).

NOTE: “&&” must be used to represent the AND operator in any URL, rather than the typical single “&”.

CUSTOM MENUS

EXAMPLE – HTTP URL

Customer 2 (100)

Customer: FR_AIRBUS Name: Airbus France Association No: 4521053230009

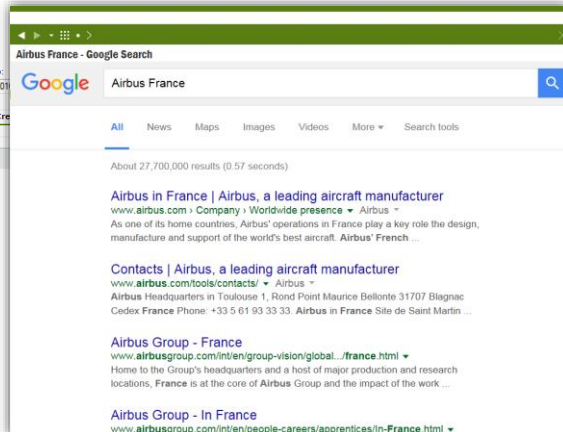
General Address Contact Comm. M

Default Language: French

Country: SWEDEN

Category: Customer

Customer 360 View
View in Map
Customer Order Lines
Delivery Statistics
Invoiced Sales
Credit Management
Search for Customer on Google
Copy Customer...



Adding an HTTP URL example (2/2)

A web browser is opened for the defined URL address.



FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



CUSTOM EVENTS

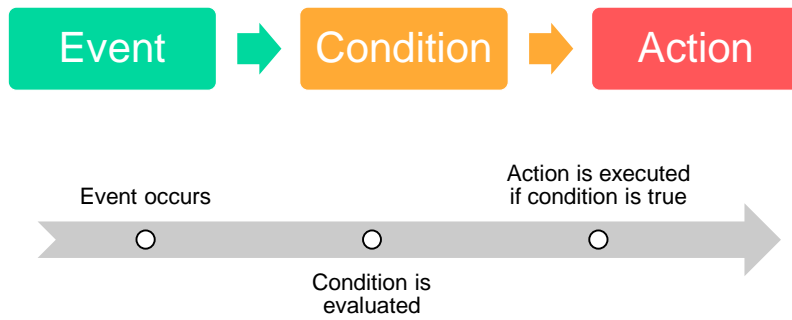
OVERVIEW

- The overview of Events in IFS Applications and How they work.
- How to create and configure Events and their actions using IFS Solution Manager.
- How to work with custom defined events and their configuration.
- Troubleshooting events in IFS Applications.



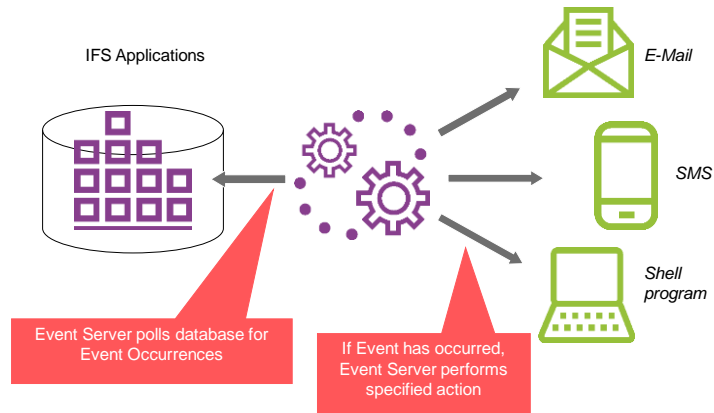
CUSTOM EVENTS

EVENT OVERVIEW



CUSTOM EVENTS

EVENT OVERVIEW



CUSTOM EVENTS

EVENT OVERVIEW

Each event has a number of defined parameters

Use in action

- e.g. values to include in message

Use as condition

- e.g. action upon Invoice Printing, not Pick List

Use to find receiver

- e.g. recipient email address

Multiple actions may be connected to one event



Email



SMS



Shell script

+ others...



For a given event, you can choose multiple actions – for example, you may send one person an email, and one a message to their cell phone!

CUSTOM EVENTS

EVENT CONFIGURATION

- IFS Application Server required services must be running
- IFS Connect MAIL_SENDER1 must be configured with the SMTP Mail Host

Setup IFS Connect

MAIL_SENDER1 Configuration

Description: Mail sender sends messages using the SMTP protocol

Instance Type: Mail

Parameters

SMTP_HOST	YourMailHost
SMTP_PORT	25
SMTP_TIMEOUT	60
MAIL_RETRIES	60
RETRY_INTERVAL	60
PERFORM_SMTP_AUTH	<input checked="" type="checkbox"/>
USER_NAME	MailUser
PASSWORD	
DEFAULT_INSTANCE	<input checked="" type="checkbox"/>
DEFAULT_MAIL_SENDER	connect@localhost
CONTENT_TYPE	text/plain
USE_SSL	<input type="checkbox"/>
USE_TLS	<input type="checkbox"/>

In Solution Manager/Integration/IFS Connect/ Setup IFS Connect – Connector Senders/Mail_Sender1 you must configure the SMTP_HOST to point to your mail server

Apache, Jboss(Weblogic in Apps 9), Batch Processor and Connect Server services must be running.

CUSTOM EVENTS

TROUBLESHOOTING

- *Solution Manager / Administration / Logs and Queues / Application Messages*, click on the Application Message ID to display the detailed **Error Text** of a Failed message.

The screenshot shows the SAP Solution Manager interface. On the left, the 'Business Object Explorer' sidebar has 'Explore Application Messages' selected. The main window is titled 'Application Messages' and shows a list of messages. The message with ID 169382 is highlighted, and its details are shown on the right. The 'Status' is 'Failed' and the 'Error Text' is 'Mail address 'To' field is not specified'. The 'Created From' section shows 'Initiated By: PFCONNECT' and 'Initiated On: 4/26/2018 8:44 AM'. The 'Message Input Data' section is empty.

Address	Connector	Status	Text	Message Data
	Mail	Failed	Mail address 'To' field is not specified	

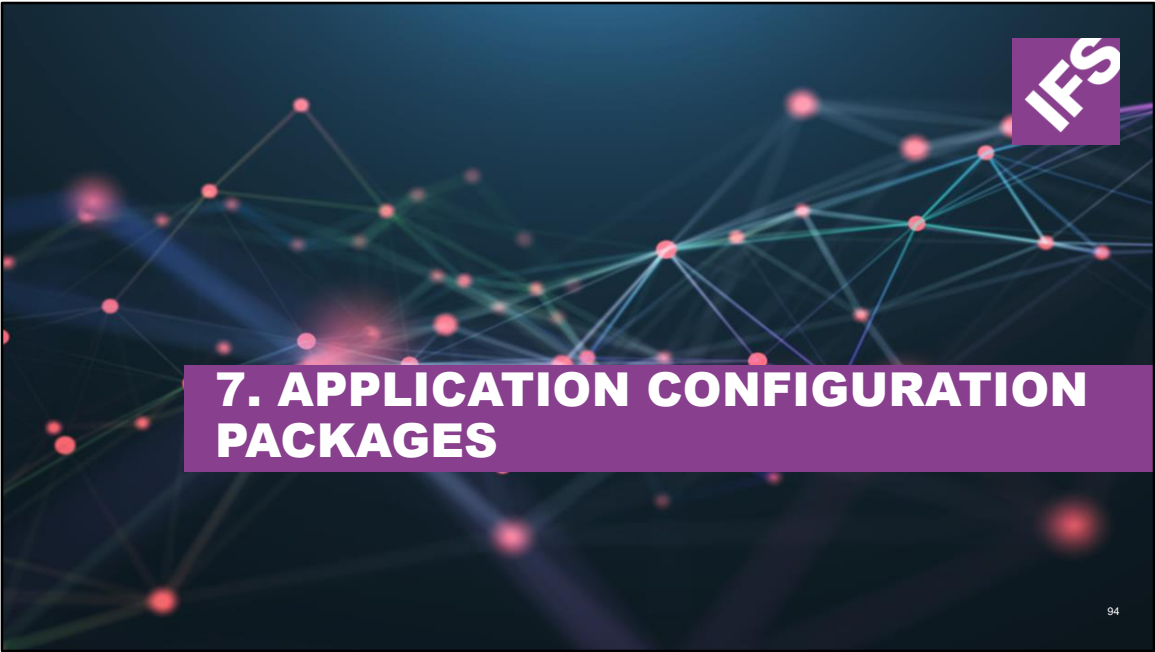


FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



7. APPLICATION CONFIGURATION PACKAGES

94

CUSTOM FIELDS

APPLICATION CONFIGURATION PACKAGE

- Application Configuration Packages functionality provides means to package, export and install certain application-wide configurations such as Custom Objects.
- Using Application Configuration Packages, it is now possible to add a configuration or modify an existing configuration package in one environment and then export and install it in another environment.
- Application Configuration Packages can be *developed*, *exported* and *installed*.



CUSTOM FIELDS

APPLICATION CONFIGURATION PACKAGE

- Packages are created in the *Application Configuration Package* page in Solution Manager. Click new and enter a package name, a description and an author.

Application Configuration Package - Demo Config package ⓘ

Package Name: Description:
Author: [Import](#)
[Publish](#)
[Export](#)
[Validate](#)
Last Modified:

Import Details

Version: Imported Date:
Origin: Version Time Stamp:

Translations

Name	Type	Associated Object	Presentation Object	Definition Changed	Approved	Published	In Sync



CUSTOM FIELDS

APPLICATION CONFIGURATION PACKAGE

■ Add items to the package

The screenshot displays the 'Application Configuration Package - Demo Config package' interface. It includes a sidebar with navigation options like 'Package Name', 'Author', 'Last Modified', 'Import Details', and 'Items'. The main area shows the 'Add New Configuration Item' dialog with fields for 'Item Type', 'Item Name', 'Version', 'Imported Date', 'Origin', and 'Version Time Stamp'. Below the dialog, the 'Items' table lists configuration items.

Name	Type	Associated Object	Presentation Object	Definition Changed	Approved	Published	In Sync
Reason	Custom Field Persistent	SalesPart	cfSalesPart	9/6/2016 11:46:35 AM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NewReason	Custom Enumeration	NewReasonCode		9/6/2016 11:47:59 AM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Comments	Custom Field Persistent	SalesPart	cfSalesPart	9/6/2016 11:45:02 AM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

CUSTOM FIELDS

APPLICATION CONFIGURATION PACKAGE

■ Export Application Configuration Package

Application Configuration Package - Demo Config package

Package Name: Demo Config package
Author: J. Scandura
Last Modified: 8/6/2016 11:57:34 AM

Description: This demo configuration package contains Comments and Reason Code Custom fields related configurations.

Import
Publish
Export
Validate

Import Details

Version:
Origin:

Items Translations

Name	Type	Assoc
Reason	Custom Field Persistent	Sales
NewReason	Custom Enumeration	NewReason
Comments	Custom Field Persistent	Sales

Export Package

Export Path: C:\SystemTest\DemoConfigPackage Browse

☐ Export as a Zip file

Version: 1.0.1

Comments: Demo Application Configuration Package

Export Cancel



CUSTOM FIELDS

EXPORT CUSTOM FIELDS

Custom Fields - SalesPart

Logical Unit: Prompt: Module:

☒ RowKey Activated Activating the RowKey for a Logical Unit (LU) is the first step in order to make an LU possible to be extended with Custom Fields. RowKey is the functionality that connects standard LUs with the extended Custom Field LUs.

☒ Approved Approve to make available for Publishing. While not approved it will be

☒ Published Approved Detail Views: 2 (of totaly 22)
Publish will generate and deploy all server objects, thus making it available. When publishing, only the Approved items will be included.
Published Date: 4/22/2016 at 11:44:02 AM
Presentation Objects: cSalesPart

☒ Synchronized Synchronized means that no changes has been made to the configuration

Attribute Name Prompt Approved Published Field Type Comment

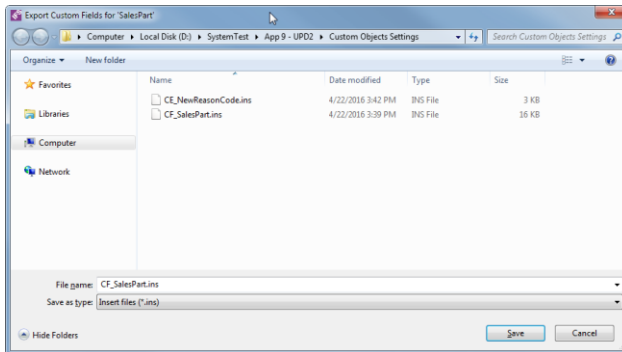
PART_DAMAGED	Part Damaged	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Persistent Field	Yes/No indication whe
REASON_CODE	Reason Code	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Persistent Field	Reason Code for Sales

Export...

Individual enumerations or overall Custom Field settings can be exported From the header of the Custom Fields screen, RMB, Export Browse to location and export file. File will be named with CE for Custom Enumeration or CF for Custom Field and will then include the LU Name Exporting a Custom Field will export all settings and attributes, exporting an Enumeration will export Enumeration settings

CUSTOM FIELDS

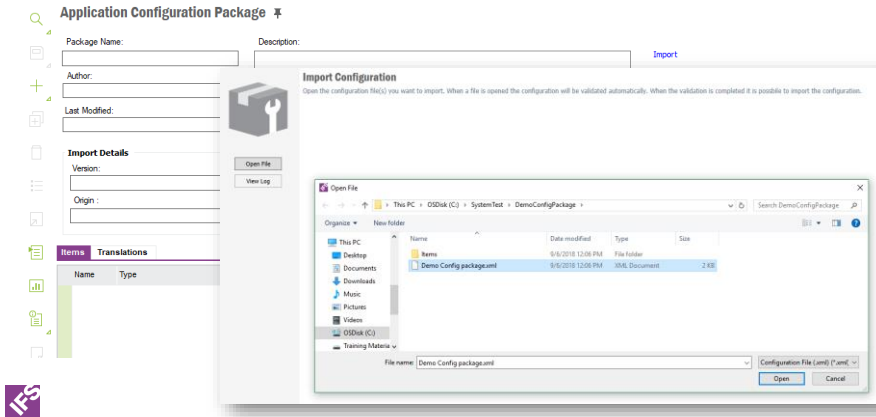
EXPORT CUSTOM FIELDS



Individual enumerations or overall Custom Field settings can be exported
From the header of the Custom Fields screen, RMB, Export
Browse to location and export file. File will be named with CE for Custom
Enumeration or CF for Custom Field and will then include the LU Name – see above
example

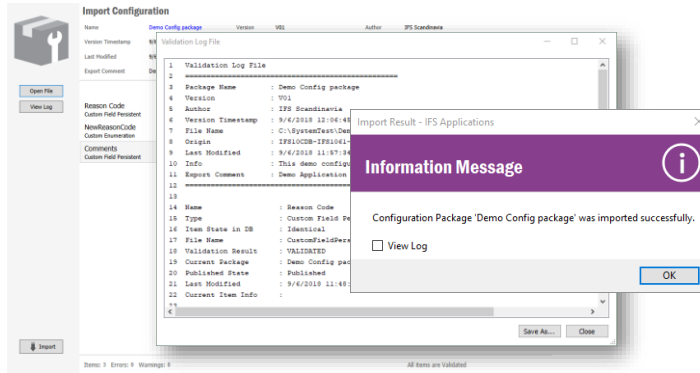
CUSTOM FIELDS

IMPORT APPLICATION CONFIGURATION PACKAGE



CUSTOM FIELDS

IMPORT APPLICATION CONFIGURATION PACKAGE



CUSTOM FIELDS

VALIDATION ON IMPORT

The screenshot displays the 'Import Configuration' window in the IFS application. The window is divided into a left sidebar with navigation icons and a main content area. The main content area shows a table of configurations and their validation status.

Import Configuration Details:

- Name:** [Click](#)
- Version:** v2
- Author:** R&D Project Environment
- Version Timestamp:** 2019-08-24 14:07:03
- File Name:** CarLent
- Origin:** 58-PSID061-PeerValidationUIP
- Last Modified:** 2019-08-24 14:07:03
- Info:** This package contains configurations for Company Vehicle functionality.
- Export Comment:** Added a Custom Ref to the ImportPackage Configuration, and changed the location of the GLU to be an alternate key.

Configuration Table:

Configuration	Import Type	Used For	Status
Vehicle Type Custom Enumeration	Identical	Used for Company vehicles	Validated OK
Company Vehicles Custom Page	New		Validated OK
Company Vehicle Custom Page	New		Validated OK
Engine Type Custom Enumeration	Identical	Engines used in vehicles	Validated OK
Default Vehicle Custom Field Persistent	Modified	Used for default vehicle on most organisations.	Validated with errors
Company Vehicles Custom UI	New	Used to handle company vehicles	Error Error: Another Custom Field Attribute with the same name "DEFAULT_ATTRIBUTE" exists for "Organisation". Published

At the bottom of the window, a status bar indicates 'All items are validated'.

CUSTOM FIELDS

APPLICATION CONFIGURATION PACKAGE

Application Configuration Package - Demo Config Package

Name: Demo Config Package
Description: This demo package contains the configuration for Part Damaged and Reason Code custom objects.
Author: IFS Scandinavia
Last Modified Date: 4/25/2016 12:32:27 PM

Import Details
Version: V01
Origin: S14604-IFS1061-Strike_BnT_CMB
Imported Date: 4/25/2016 12:32:27 PM
Version Time Stamp: 4/25/2016 11:55:19 AM

Items **Translations**

Name	Item Type	Associated Object	Presentation Object
Reason Code	Custom Field Persistent	SalesPart	
Reason Code	Custom Enumeration	NewReasonCode	
Part Damaged	Custom Field Persistent	SalesPart	

Items **Translations**

Name	Item Type	Associated Object	Presentation Object	Definition Changed	Approved	Published	In Sync
Reason Code	Custom Field Persistent	SalesPart	cSalesPart	4/25/2016 9:38:52 AM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Reason Code	Custom Enumeration	NewReasonCode		4/25/2016 9:33:59 AM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Part Damaged	Custom Field Persistent	SalesPart	cSalesPart	4/25/2016 9:38:55 AM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>



CUSTOM FIELDS

IMPORT CUSTOM FIELDS/ENUMERATIONS

- To import custom fields or enumerations, you must use a tool such as SQL Plus or PL/SQL Developer to execute the exported .ins script.
- IFS Enterprise Explorer cannot be used to deploy scripts.



CUSTOM FIELDS

VERIFY CUSTOM FIELD AFTER IMPORT

Custom Fields - SalesPart

Logical Unit: Prompt: Module:

☒ **RowKey Activated** Activating the RowKey for a Logical Unit (LU) is the first step in order to make an LU possible to be extended with Custom Fields. RowKey is the functionality that connects standard LUs with the extended Custom Field LUs.

☒ **Approved** Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.

Approved Detail Views: [None \(of totally 22\)](#)

☒ **Published** Publish will generate and deploy all server objects, thus making it available to be used and referenced from the client and other objects. When publishing, only the Approved items will be included.

Published Date: 4/25/2016 at 2:58:07 PM

Presentation Object:

☒ **Synchronized** Synchronized means that no changes has been made to the configuration since it was last Published.

Attribute Name	Prompt	Approved	Published	Field Type	Comment	Translations	Enabled On Reports	Application Configuration Package
PART_DAMAGED	Part Damaged	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Persistent Field		0	<input checked="" type="checkbox"/>	Demo Config Package
REASON_CODE	Reason Code	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Persistent Field	Reason Code which is connected to NEW_RE...	0	<input checked="" type="checkbox"/>	Demo Config Package





FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...

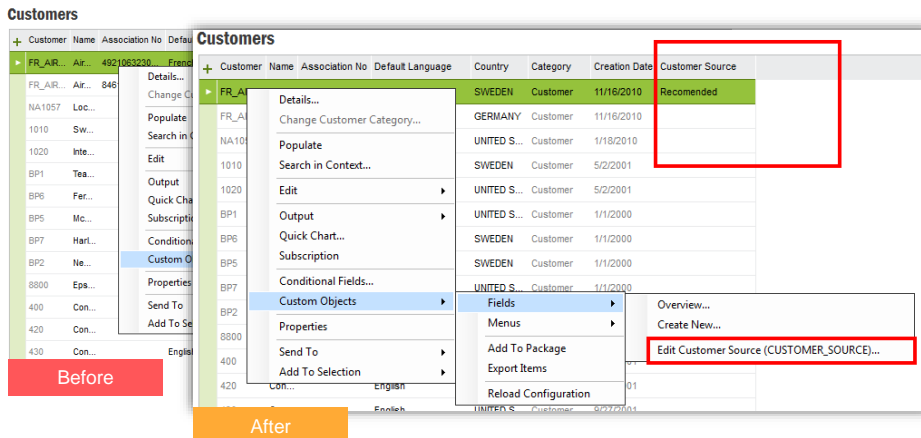


8. CUSTOM FIELDS CONFIGURATION CLEAN UP

108

CUSTOM FIELDS

CLEANUP DATA FROM CUSTOM FIELDS



On a screen containing your custom field, RMB , select Custom Objects\Fields\Reload Configuration

Check Custom Object\Fields and you will see that the field is now gone and the field is also no longer visible on the overview window

Data will still be visible in the table (CFT object) but not in the view (CFV)

CUSTOM FIELDS

APPLICATION CONFIGURATION PACKAGE

- Application Configuration Packages functionality provides means to package, export and install certain application-wide configurations such as Custom Objects.
- Using Application Configuration Packages, it is now possible to add a configuration or modify an existing configuration package in one environment and then export and install it in another environment.
- Application Configuration Packages can be *developed, exported and installed*.



CUSTOM FIELDS

CLEANUP DATA FROM CUSTOM FIELDS

Custom Fields - CustomerInfo

Logical Unit: CustomerInfo Prompt: Customer Info Module: ENTERP

☒ RowKey Activated Activating the RowKey for a Logical Unit (LU) is the first step in order to make an LU possible to be extended with Custom Fields. RowKey is the functionality that connects standard LUs with the extended Custom Field LUs.

☒ Approved Approve to make available for Publishing. While not approved it will be excluded from all publishing logic.

☒ Published Approved Detail Views: [None \(of totally 4\)](#)

☒ Synchronized Publish will generate and deploy all server objects, thus making it available to be used and referenced from the client and other objects. When publishing, only the Approved items will be included.

Published Date: 4/26/2016 at 9:08:50 AM

Presentation Object: [cfCustomerInfo](#)

Synchronized means that no changes has been made to the configuration since it was last Published.

Attribute Name	Prompt	Approved	Published	Field Type	Comment	Translations	UI Object	LU Reference	LOV View	Custom Field Impl Ty
CUSTOMER_SOURCE	Customer Source	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Persistent Field		1	Combobox	CustEnum		Not Applicable
SALES_TAX	Sales Tax	<input type="checkbox"/>	<input type="checkbox"/>	Persistent Field		1	Default			Not Applicable

To disconnect a Custom Field, you can highlight the attribute line, RMB and select Mark as Disapproved

Then in the Header, RMB and select Publish again

Once complete, the attribute row Approved and Published checkboxes should be unchecked

CUSTOM FIELDS

CLEANUP DATA FROM CUSTOM FIELDS

- **Remove unused Custom data** - Finds columns that exists in a Custom Fields table but is not published and removes the data in the columns simply by dropping the columns from the table.

Custom Fields - CustomerInfo

Logical Unit: Prompt: Module:

☒ RowKey Activated Activating the RowKey for a Logical Unit (LU) is the first step in order to make an LU possible to be extended with Custom Fields. RowKey is the functionality that connects standard LU with the extended Custom Field LU.

☒ Approved Approve to make available for Publication

☒ Published Publish will generate and deploy all Custom Fields. When publishing, only the Approved Custom Fields will be used and referenced from the client and other objects.

☒ Synchronized Synchronized means that no changes are allowed to the Custom Fields.

Published Date: 4/26/2016 at 9:00 AM

Presentation Object:

Actions:

- Activate RowKey
- Deactivate RowKey
- Approve
- Disapprove
- Synchronize
- Unpublish
- Export...
- View
- Cleanup**
 - Remove Unused Custom Data...**
 - Remove All Custom Data...
- Populate
- Search in Context...
- Edit

Relationships	UI Object	LU Reference	LOV View	Custom Field Impl
Subscription	Combobox	CustEnum		Not Applicable
Conditional Fields...	Default			Not Applicable
Custom Objects				

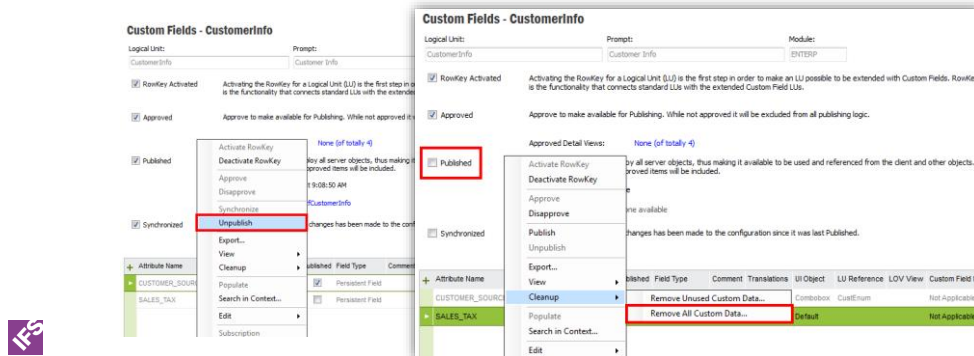
Attribute Name	Prompt	Approved	Published
CUSTOMER_SOURCE	Customer Source	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SALES_TAX	Sales Tax	<input type="checkbox"/>	<input type="checkbox"/>

RMB in the header and select Cleanup\Remove Unused Custom Data

CUSTOM FIELDS

CLEANUP DATA FROM CUSTOM FIELDS

- Remove all Custom data - Drops the complete Custom Fields table (even if columns are published).



Unpublish all attributes and now RMB in the header and select Cleanup\Remove All Custom Data to clear all data for the LU



FEEDBACK ON THE OBJECTIVES OF THIS LESSON



Lesson objectives:

- ...
- ...



WRAP-UP





QUESTIONS?



PLEASE PROVIDE US FEEDBACK



THANK YOU



IFSworld.com

© COPYRIGHT© 2020 BY INDUSTRIAL AND FINANCIAL SYSTEMS, IFS AB (PUBL). ALL RIGHTS RESERVED. THIS MATERIAL AND ITS CONTENT IS PRODUCED BY THE IFS ACADEMY FOR AUTHORIZED TRAINING PURPOSES ONLY AND REMAINS THE INTELLECTUAL PROPERTY OF IFS. NEITHER THE MATERIAL OR ITS CONTENT MAY BE COPIED, REPRODUCED, OR DISTRIBUTED WITHOUT IFS' EXPRESS WRITTEN PERMISSION.

IFS DOES NOT WARRANT, EITHER EXPRESSLY OR IMPLIED, THE ACCURACY, TIMELINESS, OR APPROPRIATENESS OF THE INFORMATION CONTAINED IN THIS TRAINING MATERIAL AND DISCLAIMS ANY RESPONSIBILITY FOR CONTENT ERRORS, OMISSIONS, OR INFRINGING MATERIAL. IFS ALSO DISCLAIMS ANY RESPONSIBILITY ASSOCIATED WITH RELYING ON THE INFORMATION PROVIDED IN THIS DOCUMENT AND ANY AND ALL LIABILITY FOR ANY MATERIAL CONTAINED ON OTHER CHANNELS THAT MAY BE LINKED TO THE IFS TRAINING MATERIAL.