Deprecation of IALs

Contents

Deprecation of Information Access Layer (IAL)

INTRODUCTION

The purpose of the ***Information Access Layer*** (IAL) was to make the information in the application easier to find by defining several IAL objects that expose ***information*** in the application. ***IAL Object Developer*** (a window in IFS enterprise explorer) was used to create new IAL objects, where you could define a SQL select statement as the definition of the IAL object.

However, with the new architecture and the introduction of IFS Aurena, there have been changes to how client tier applications access entities in the database. Native OData-based RESTful APIs have been introduced across the entire suite of IFS Applications to access information. Moreover, accessing database objects directly from the client tier is deprecated in the new architecture.

Solutions like IAL objects which define SQL statements directly from the client application and deploy them as new objects into the database are not recommended in the new architecture. The backend database schema is encapsulated within OData RESTful APIs and knowledge about the database entities are not exposed to the client layer.

Due to these reasons ***IAL Object Developer*** *and* ***IAL*** **Configuration** windows will be deprecated from IFS Cloud 21R1. It will still be available in the Aurena client released with 21R1 but will be removed in future releases. In addition to IAL Objects, other tools and functionalities which uses direct SQL queries in client layer will also gradually adapt to different mechanisms to access data; examples are SQL data sources in Lobby, SQL statement type quick reports etc.

MOVING FORWARD

IAL objects are mainly used in quick reports, integrations to 3rd party systems, Lobbies, as a source for quick information sources and a few other purposes which are not in scope of this document. The above usages can be broadly classified into three main use cases described below. Each of these main use cases will be addressed and solved differently in the future.

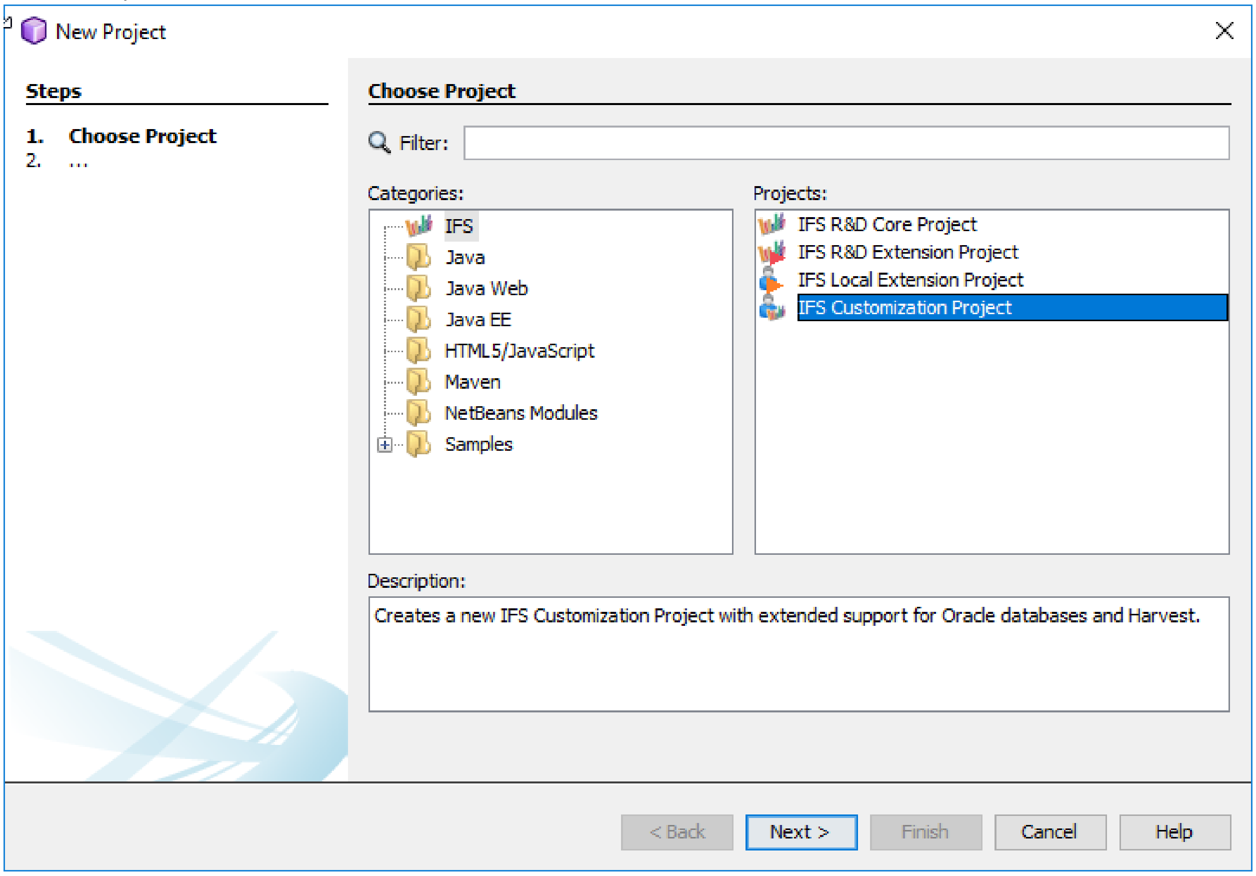
1. The use of IAL objects to create a 'named query' that can be used later by a customer as a basis for quick reports, lobbies etc.
    - This will be addressed in the future with the introduction of a 'Query Designer' tool targeting non-technical users to self-service a 'named query' using our entity model as a base.
2. The use of IAL objects to present the underlying transactional data in a way that is optimized for BI, data warehousing and reporting purposes.
    - The existing Information Source framework can be used for this purpose. Defining information sources is a developer task and will need developer tooling.
3. The use of IAL objects as one of several ways to provide self-service customization for customers instead of ordering a modification from IFS.
    - In the long term, there will be new tooling introduced to support self-service capabilities for lightweight customization. In the short term, this use case should be considered as a new customization requirement and therefore needs to be handled in IFS Developer Studio. It is recommended to create a new view in a customization project in IFS Developer Studio in order to extract information from IFS Applications. The steps to create a view using IFS Developer Studio are described here >>.

With IFS Cloud 21R1 every customer is expected to have a 'Customer Solution Repository' that stores all their customizations and configurations that they are responsible for. The Customer Solution Repository also enables a clear separation of the “as supported by IFS” and “as delivered to customers” aspects of a solution. Configuration and customization are the customers’ responsibility, regardless of whether they self-serve or buy a service from IFS consulting or from a partner to do the work. Most customers will achieve this by using the IFS Lifecycle Experience ‘Build Place’ as a service from IFS.

CREATING A NEW VIEW IN A CUSTOMIZATION PROJECT USING IFS DEVELOPER STUDIO

The following steps describe how to create a new view in a customization project. Please refer the Base Server Development for Application Core course by IFS Academy for any pre-requisite knowledge on how to work in IFS Developer Studio.

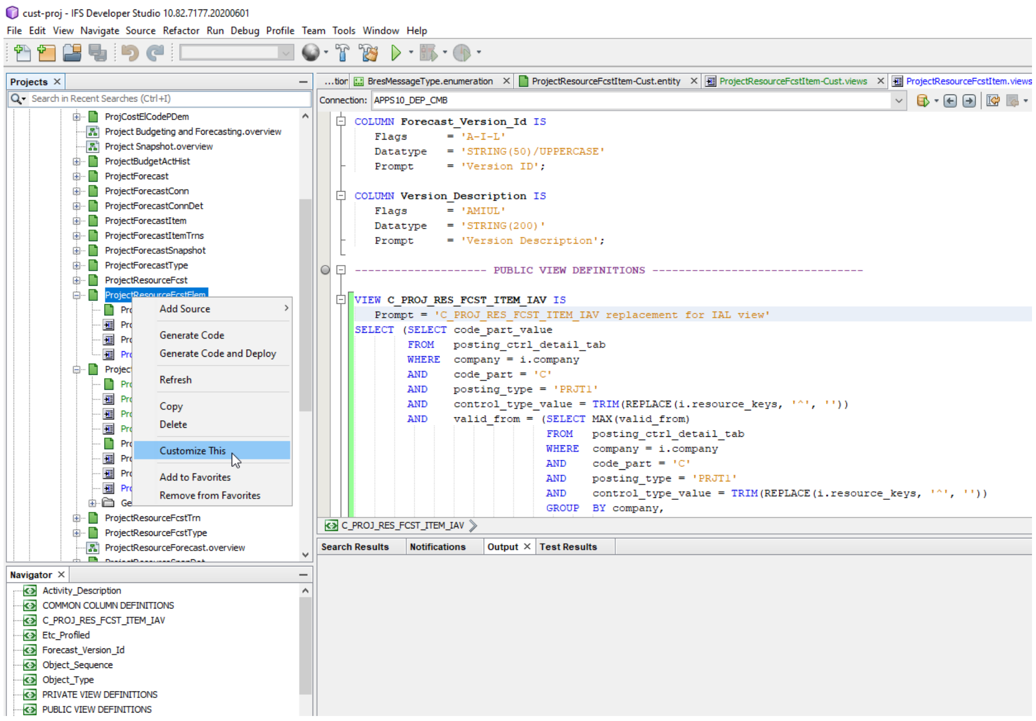
1. Locate the existing *.IAL file and extract the select statement from the IAL view definition or prepare and test your SQL statement using IFS developer studio.
2. Load the customization project in IFS Developer Studio or create a new customization project in IFS Developer Studio.



Creates a new IFS Customization Project with extended support for Oracle databases and Harvest.

3. Choose the Module and the Entity that best relate to the IAL view definition
4. Customize the entity by using the **Customize this** option available in the entity level. It will generate _cust layer files which will enable you to customize selected entity.

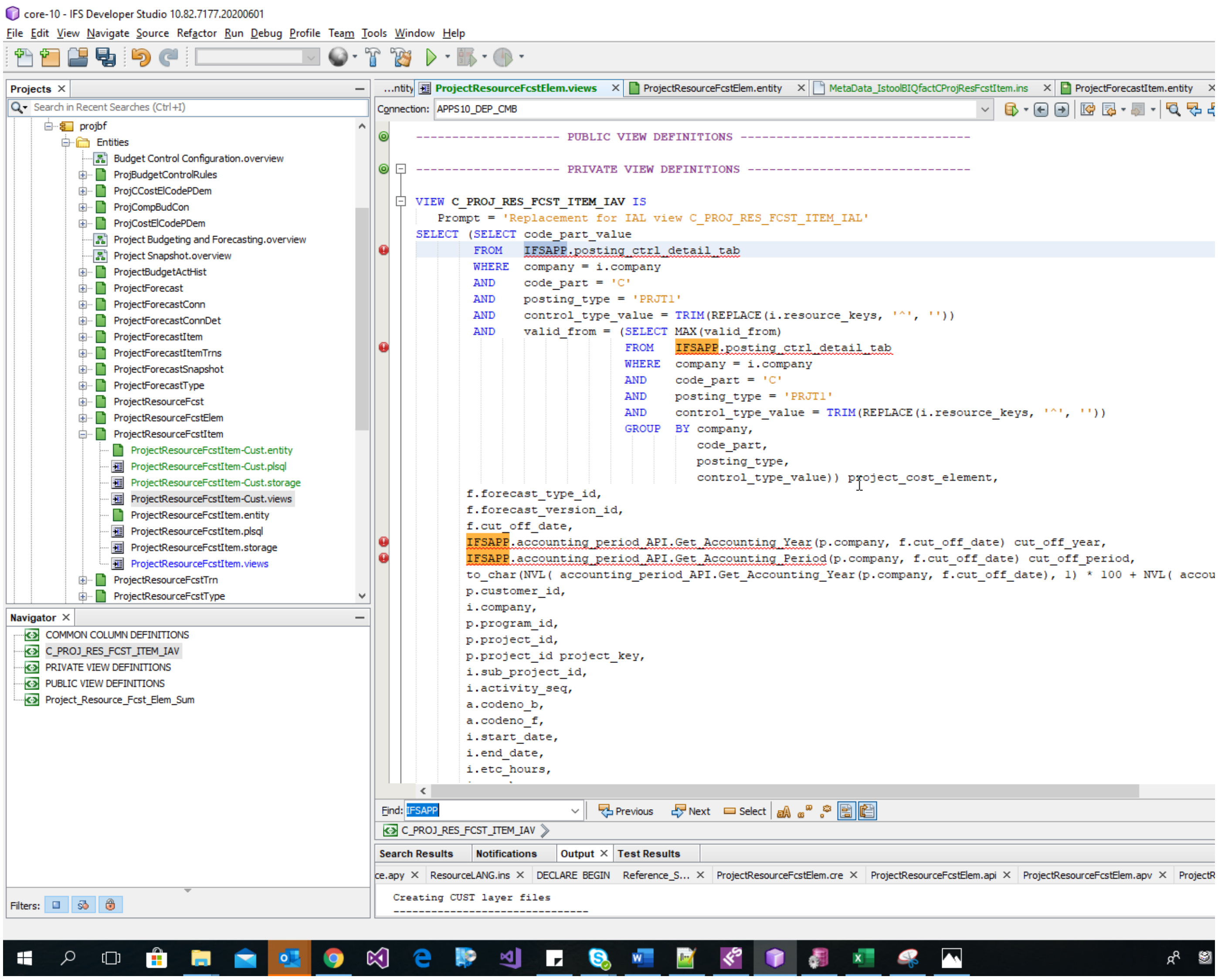
Deprecation of Information Access Layer (IAL)



```
COLUMN Forecast_Version_Id IS
   Flags       = 'A-I-L'
   Datatype    = 'STRING(50)/UPPERCASE'
   Prompt      = 'Version ID';

COLUMN Version_Description IS
   Flags       = 'AMIUL'
   Datatype    = 'STRING(200)'
   Prompt      = 'Version Description';

-------------------- PUBLIC VIEW DEFINITIONS --------------------------------

VIEW C_PROJ_RES_FCST_ITEM_IAV IS
   Prompt = 'C_PROJ_RES_FCST_ITEM_IAV replacement for IAL view'
SELECT (SELECT code_part_value
        FROM   posting_ctrl_detail_tab
        WHERE  company = i.company
        AND    code_part = 'C'
        AND    posting_type = 'PRJT1'
        AND    control_type_value = TRIM(REPLACE(i.resource_keys, '^', ''))
        AND    valid_from = (SELECT MAX(valid_from)
                             FROM   posting_ctrl_detail_tab
                             WHERE  company = i.company
                             AND    code_part = 'C'
                             AND    posting_type = 'PRJT1'
                             AND    control_type_value = TRIM(REPLACE(i.resource_keys, '^', ''))
                             GROUP  BY company,
```

5. Create a new view by providing the select statement of the IAL view (extracted in step1) as the query in the new view. Use the *.views file in Cust layer for this step.

Deprecation of Information Access Layer (IAL)


core-10 - IFS Developer Studio 10.82.7177.20200601
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Search in Recent Searches (Ctrl+I)
projbf
Entities
Budget Control Configuration.overview
ProjBudgetControlRules
ProjCCostElCodePDem
ProjCompBudCon
ProjCostElCodePDem
Project Budgeting and Forecasting.overview
Project Snapshot.overview
ProjectBudgetActHist
ProjectForecast
ProjectForecastConn
ProjectForecastConnDet
ProjectForecastItem
ProjectForecastItemTrns
ProjectForecastSnapshot
ProjectForecastType
ProjectResourceFcst
ProjectResourceFcstElem
ProjectResourceFcstItem
ProjectResourceFcstItem-Cust.entity
ProjectResourceFcstItem-Cust.plsql
ProjectResourceFcstItem-Cust.storage
ProjectResourceFcstItem-Cust.views
ProjectResourceFcstItem.entity
ProjectResourceFcstItem.plsql
ProjectResourceFcstItem.storage
ProjectResourceFcstItem.views
ProjectResourceFcstTrn
ProjectResourceFcstType
Navigator
COMMON COLUMN DEFINITIONS
C_PROJ_RES_FCST_ITEM_IAV
PRIVATE VIEW DEFINITIONS
PUBLIC VIEW DEFINITIONS
Project_Resource_Fcst_Elem_Sum
Filters:
...ntity
ProjectResourceFcstElem.views
ProjectResourceFcstElem.entity
MetaData_IstoolBIQfactCProjResFcstItem.ins
ProjectForecastItem.entity
Connection: APPS10_DEP_CMB
Find: IFSAPP
Previous
Next
Select
C_PROJ_RES_FCST_ITEM_IAV
Search Results
Notifications
Output
Test Results
ce.apy
ResourceLANG.ins
DECLARE BEGIN
Reference_S...
ProjectResourceFcstElem.cre
ProjectResourceFcstElem.api
ProjectResourceFcstElem.apv
ProjectR

```
-------------------- PUBLIC VIEW DEFINITIONS --------------------------------

-------------------- PRIVATE VIEW DEFINITIONS -------------------------------

VIEW C_PROJ_RES_FCST_ITEM_IAV IS
   Prompt = 'Replacement for IAL view C_PROJ_RES_FCST_ITEM_IAL'
SELECT (SELECT code_part_value
        FROM   IFSAPP.posting_ctrl_detail_tab
        WHERE  company = i.company
        AND    code_part = 'C'
        AND    posting_type = 'PRJT1'
        AND    control_type_value = TRIM(REPLACE(i.resource_keys, '^', ''))
        AND    valid_from = (SELECT MAX(valid_from)
                             FROM   IFSAPP.posting_ctrl_detail_tab
                             WHERE  company = i.company
                             AND    code_part = 'C'
                             AND    posting_type = 'PRJT1'
                             AND    control_type_value = TRIM(REPLACE(i.resource_keys, '^', ''))
                             GROUP  BY company,
                                       code_part,
                                       posting_type,
                                       control_type_value)) project_cost_element,
       f.forecast_type_id,
       f.forecast_version_id,
       f.cut_off_date,
       IFSAPP.accounting_period_API.Get_Accounting_Year(p.company, f.cut_off_date) cut_off_year,
       IFSAPP.accounting_period_API.Get_Accounting_Period(p.company, f.cut_off_date) cut_off_period,
       to_char(NVL( accounting_period_API.Get_Accounting_Year(p.company, f.cut_off_date), 1) * 100 + NVL( accou
       p.customer_id,
       i.company,
       p.program_id,
       p.project_id,
       p.project_id project_key,
       i.sub_project_id,
       i.activity_seq,
       a.codeno_b,
       a.codeno_f,
       i.start_date,
       i.end_date,
       i.etc_hours,
```

```
  Creating CUST layer files
  ------------------------------
```

Since IAL view is deployed into a different database schema, there could be references to the application owner's schema in the IAL view definition (see the parts highlighted in **IFSAPP.** in above image). Remove these references from the select statement (remove **IFSAPP.** sections in this case)

6. Deploy the views into the environment using developer studio options.

WORKING WITH EXISTING IAL OBJECTS

All existing IAL objects and their underlying views will remain without any change when you upgrade to 21R1 and upwards. Different types of reports and integrations which access those existing IAL views will also work without any issues in most cases.

If an existing IAL view gets invalidated due to changes to the underlying database schema, it can be updated it using ***IAL Object Developer*** or replace the IAL with a new view definitionIn the case of the latter, it is required to extract the definition of the IAL object into a new view definition. Next, it is required to modify the objects that consume the IAL object to refer the new view instead of the invalidated IAL object. A few example cases are described below.

Quick Information Sources based on IAL views

It is possible to create a quick information source based on an IAL view. The existing quick information sources that are based on an IAL view will remain after the upgrade. In most cases, the Business Reporter reports created using such quick information sources will also work without any issues.

However, if an IAL gets invalidated due to changes in the underlying table definitions referred by the IAL definition, and there is a desire to move away from IALs, it is possible to migrate the IAL view definition into an ordinary view and to change the quick information source view to refer to that new view. It could be done in following steps.

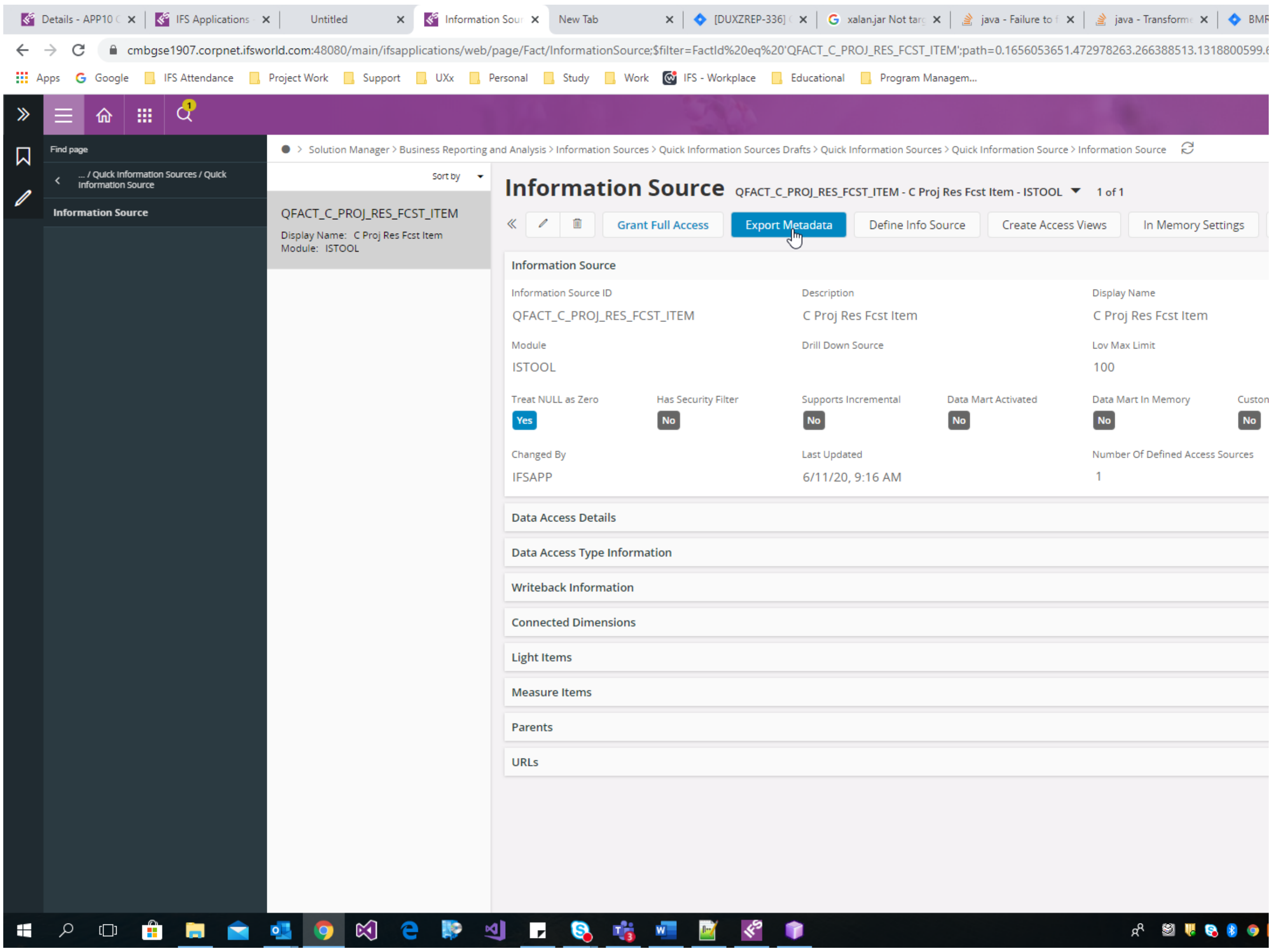
Creating a new view in customization project
Refer this section for creating a new view definition >>

This will create a new view that contains a similar definition to the IAL object. Next it is required to change the quick information source (QIS) to refer to the newly created view.

Changing QIS to refer new view instead of IAL

1. If you already have a QIS definition exported to a Metadata_XXX.ins file and if it is available in the customer's solution repository you can proceed to step 4. Otherwise it is highly recommended to have information sources metadata available in the customers solution repository using the below steps
2. Navigate to the Information Sources page in Aurena client.
3. Select the information source and click Export Metadata button. It will generate the Metadata file which should be checked in to the customers solution repository.

Deprecation of Information Access Layer (IAL)


Solution Manager > Business Reporting and Analysis > Information Sources > Quick Information Sources Drafts > Quick Information Sources > Quick Information Source > Information Source
Grant Full Access
Export Metadata
Define Info Source
Create Access Views
In Memory Settings
Data Access Details
Data Access Type Information
Writeback Information
Connected Dimensions
Light Items
Measure Items
Parents
URLs

Information Source QFACT_C_PROJ_RES_FCST_ITEM - C Proj Res Fcst Item - ISTOOL 1 of 1

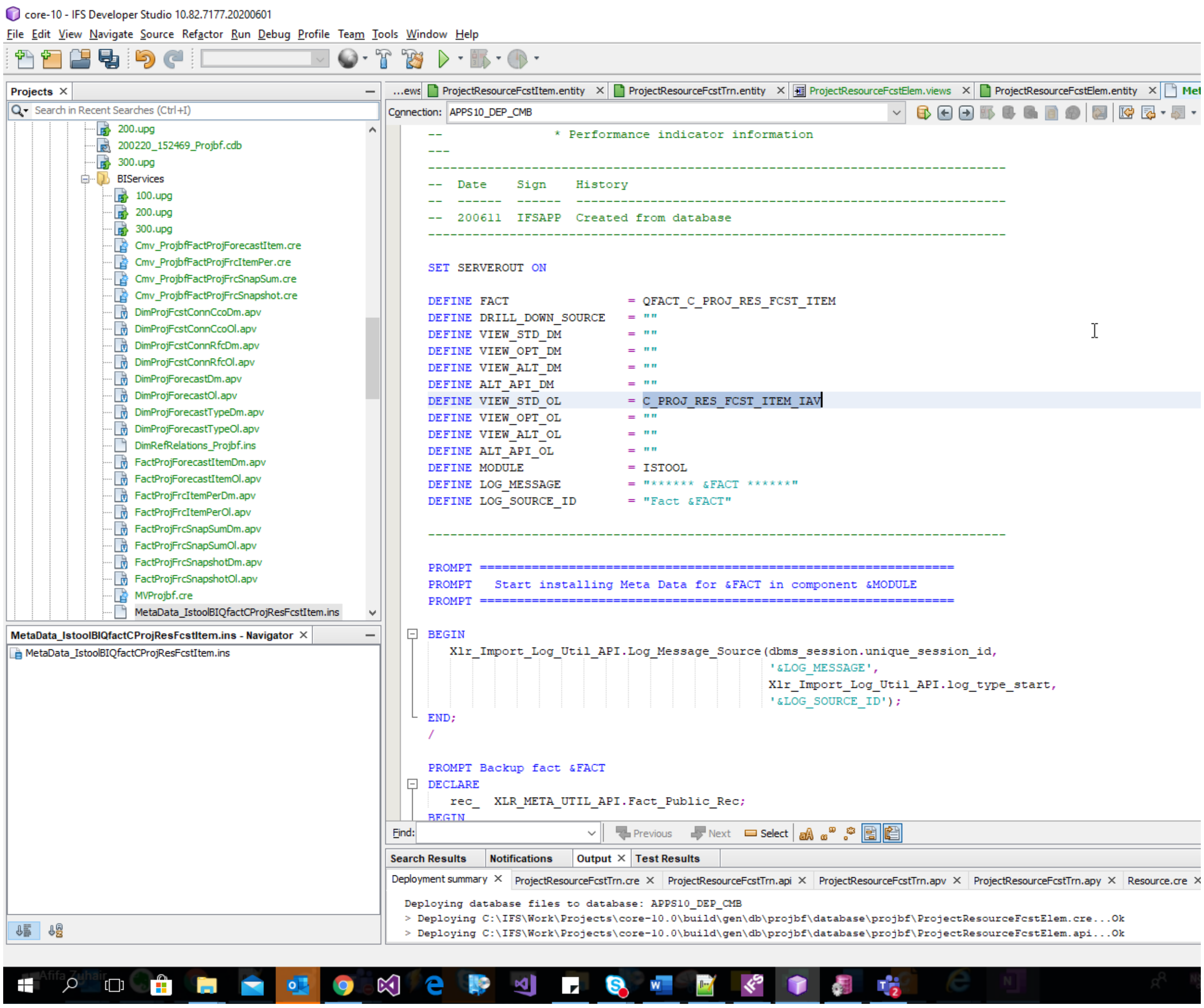
QFACT_C_PROJ_RES_FCST_ITEM
Display Name: C Proj Res Fcst Item
Module: ISTOOL

Information Source ID: QFACT_C_PROJ_RES_FCST_ITEM	Description: C Proj Res Fcst Item	Display Name: C Proj Res Fcst Item
Module: ISTOOL	Drill Down Source:	Lov Max Limit: 100
Changed By: IFSAPP	Last Updated: 6/11/20, 9:16 AM	Number Of Defined Access Sources: 1

Treat NULL as Zero	Has Security Filter	Supports Incremental	Data Mart Activated	Data Mart In Memory
Yes	No	No	No	No

4. Open the Metadata file for the information source.
5. Change value of the VIEW_STD_OL variable to the newly created view name.
6. Deploy the file.

Deprecation of Information Access Layer (IAL)


core-10 - IFS Developer Studio 10.82.7177.20200601
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
MetaData_IstoolBIQfactCProjResFcstItem.ins - Navigator
ProjectResourceFcstItem.entity
ProjectResourceFcstTrn.entity
ProjectResourceFcstElem.views
ProjectResourceFcstElem.entity
Connection: APPS10_DEP_CMB
Find:
Previous
Next
Select
Search Results
Notifications
Output
Test Results
Deployment summary
ProjectResourceFcstTrn.cre
ProjectResourceFcstTrn.api
ProjectResourceFcstTrn.apv
ProjectResourceFcstTrn.apy
Resource.cre

```
--              * Performance indicator information
---
-------------------------------------------------------------------------------
--  Date    Sign    History
--  ------  ------  -----------------------------------------------------------
--  200611  IFSAPP  Created from database
-------------------------------------------------------------------------------

SET SERVEROUT ON

DEFINE FACT                 = QFACT_C_PROJ_RES_FCST_ITEM
DEFINE DRILL_DOWN_SOURCE    = ""
DEFINE VIEW_STD_DM          = ""
DEFINE VIEW_OPT_DM          = ""
DEFINE VIEW_ALT_DM          = ""
DEFINE ALT_API_DM           = ""
DEFINE VIEW_STD_OL          = C_PROJ_RES_FCST_ITEM_IAV
DEFINE VIEW_OPT_OL          = ""
DEFINE VIEW_ALT_OL          = ""
DEFINE ALT_API_OL           = ""
DEFINE MODULE               = ISTOOL
DEFINE LOG_MESSAGE          = "****** &FACT ******"
DEFINE LOG_SOURCE_ID        = "Fact &FACT"

-------------------------------------------------------------------------------

PROMPT ================================================================
PROMPT   Start installing Meta Data for &FACT in component &MODULE
PROMPT ================================================================

BEGIN
   Xlr_Import_Log_Util_API.Log_Message_Source(dbms_session.unique_session_id,
                                              '&LOG_MESSAGE',
                                              Xlr_Import_Log_Util_API.log_type_start,
                                              '&LOG_SOURCE_ID');
END;
/

PROMPT Backup fact &FACT
DECLARE
   rec_   XLR_META_UTIL_API.Fact_Public_Rec;
BEGIN
```

```
Deploying database files to database: APPS10_DEP_CMB
> Deploying C:\IFS\Work\Projects\core-10.0\build\gen\db\projbf\database\projbf\ProjectResourceFcstElem.cre...Ok
> Deploying C:\IFS\Work\Projects\core-10.0\build\gen\db\projbf\database\projbf\ProjectResourceFcstElem.api...Ok
```

Now you have changed the view reference of the QIS to refer to the newly created view. All Business Reporter reports based on this QIS will not require any further modifications to them.

Quick Reports based on IAL Views

The existing reports created using IAL Views will not require any modifications.
However, if an IAL gets invalidated due to changes to the underline table structure and there is a desire to move away from the usage of IALs, it is possible to migrate IAL view definition into ordinary view and to change the quick report definition to refer the new view. It could be done in following steps.

Deprecation of Information Access Layer (IAL)

Creating new view in customization project

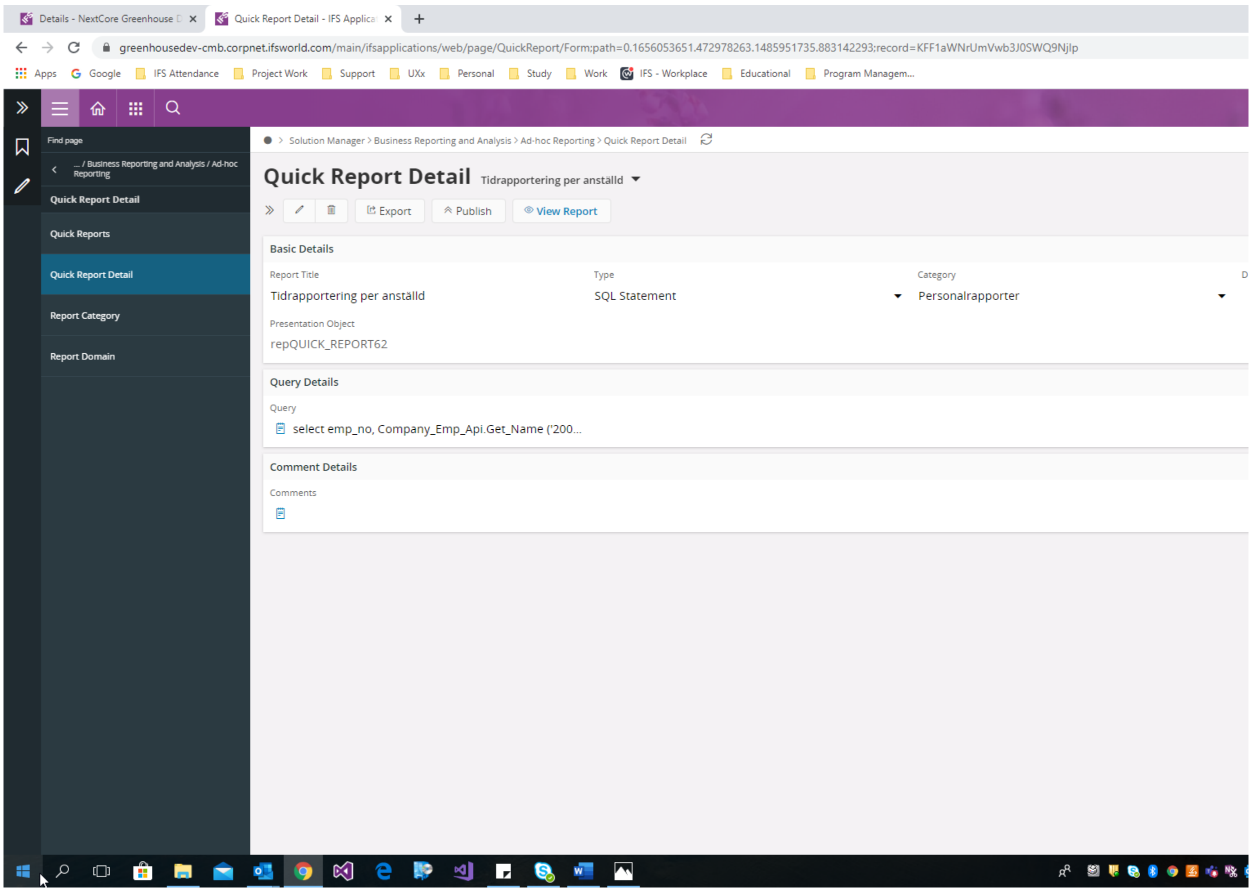
Refer this section for creating a new view definition >>

This will create a new view that contains a similar definition to the IAL object. Next it is required to change the quick report to refer the newly created view.

Changing Quick Report to refer new view instead of IAL

SQL Statement and Query Builder Quick report types

1. Navigate to Quick Report Details page in Aurena client and query for the quick report to change.


Details - NextCore Greenhouse
Quick Report Detail - IFS Applica
greenhousedev-cmb.corpnet.ifsworld.com/main/ifsapplications/web/page/QuickReport/Form;path=0.1656053651.472978263.1485951735.883142293;record=KFF1aWNrUmVwb3J0SWQ9NjIp
Apps
Google
IFS Attendance
Project Work
Support
UXx
Personal
Study
Work
IFS - Workplace
Educational
Program Managem...
Find page
.../ Business Reporting and Analysis / Ad-hoc Reporting
Quick Report Detail
Quick Reports
Quick Report Detail
Report Category
Report Domain
Solution Manager > Business Reporting and Analysis > Ad-hoc Reporting > Quick Report Detail
Export
Publish
View Report

Quick Report Detail Tidrapportering per anställd

Basic Details

Report Title: Tidrapportering per anställd
Type: SQL Statement
Category: Personalrapporter
Presentation Object: repQUICK_REPORT62

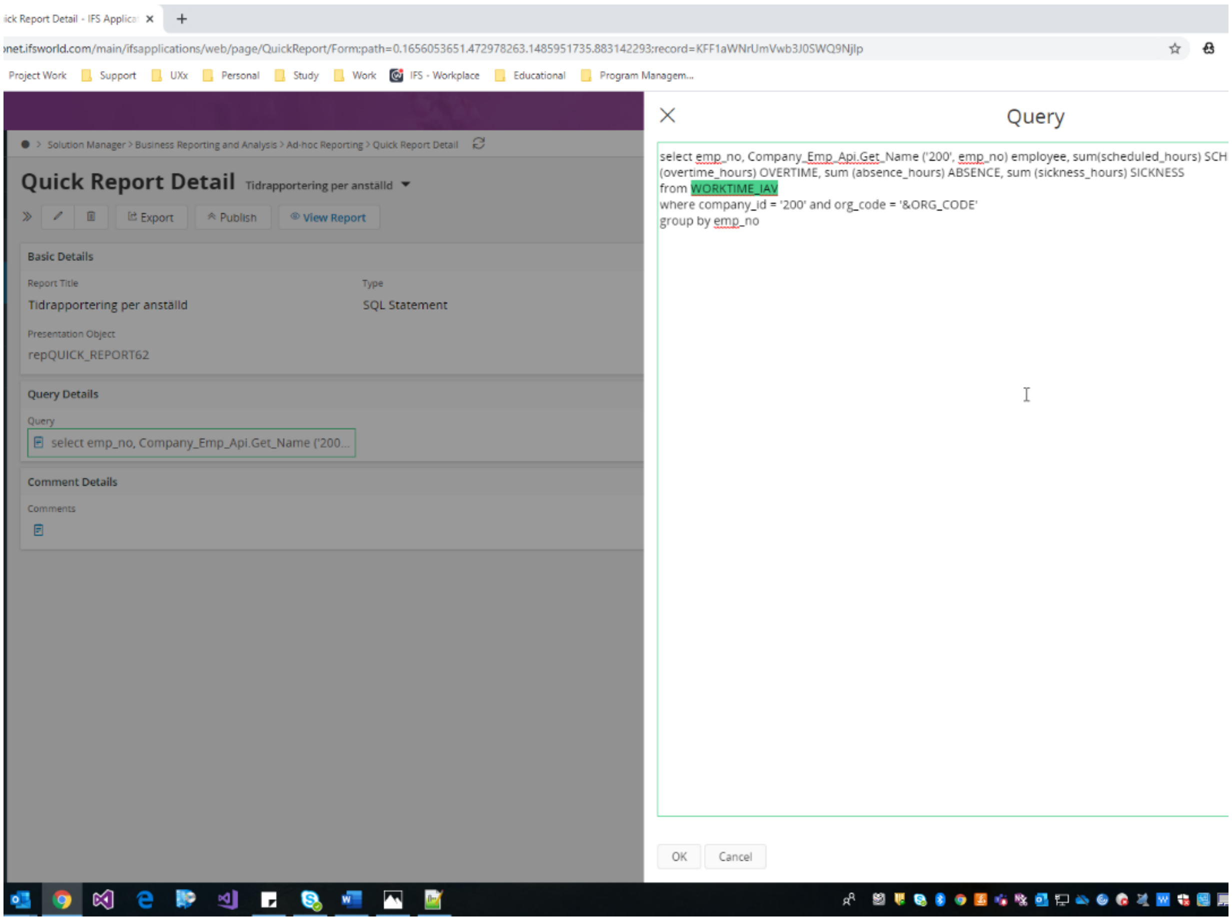
Query Details

Query: select emp_no, Company_Emp_Api.Get_Name ('200...

Comment Details

Comments:

2. Edit the Query in the Query details section providing a reference to newly created view instead of IAL view.



ick Report Detail - IFS Applica

net.ifsworld.com/main/ifsapplications/web/page/QuickReport/Form:path=0.1656053651.472978263.1485951735.883142293:record=KFF1aWNrUmVwb3J0SWQ9NjIp

Solution Manager > Business Reporting and Analysis > Ad-hoc Reporting > Quick Report Detail

Quick Report Detail Tidrapportering per anställd

Basic Details

Report Title: Tidrapportering per anställd

Type: SQL Statement

Presentation Object: repQUICK_REPORT62

Query Details

Query: select emp_no, Company_Emp_Api.Get_Name ('200...

Comment Details

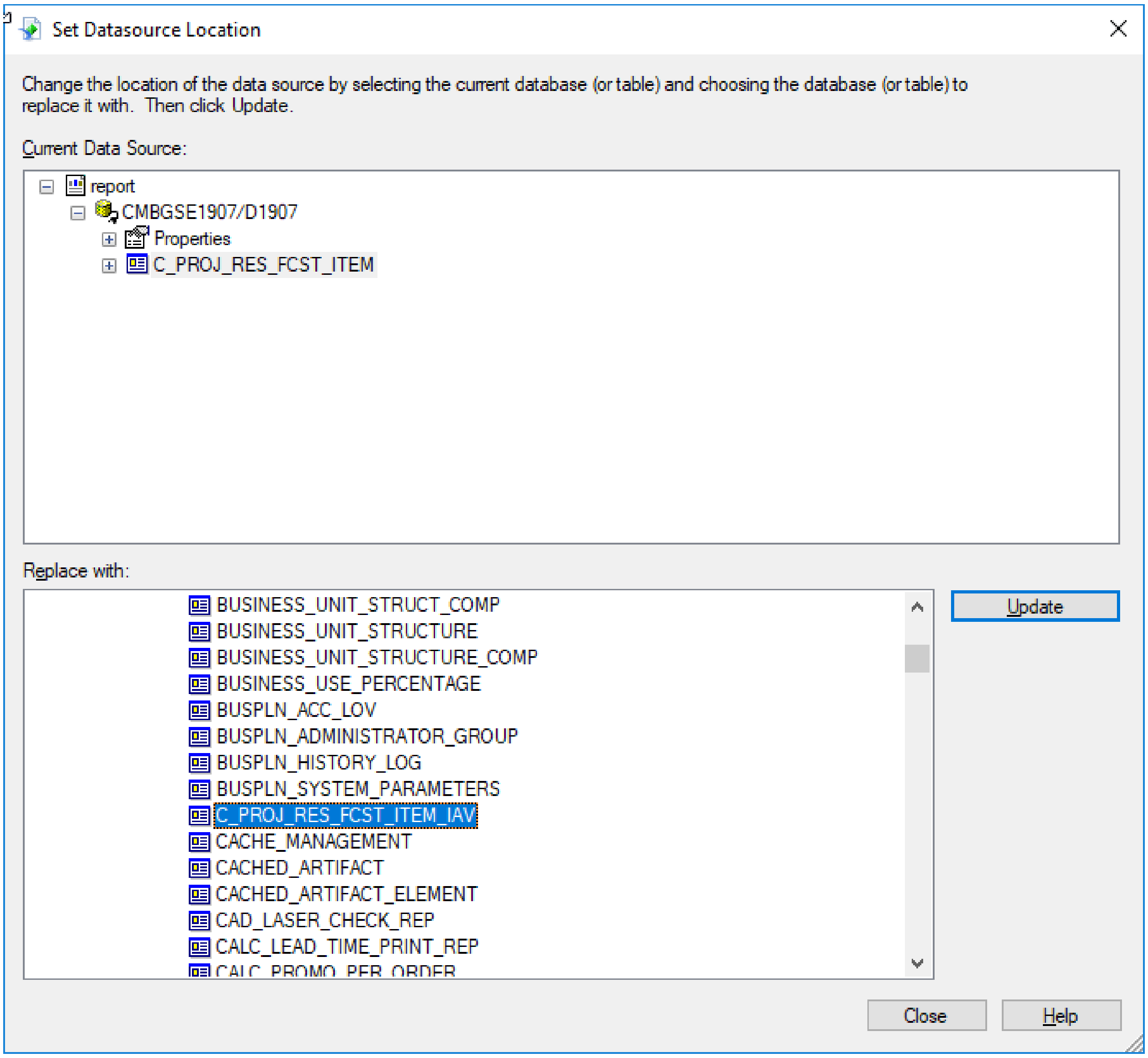
Comments

Query

```
select emp_no, Company_Emp_Api.Get_Name ('200', emp_no) employee, sum(scheduled_hours) SCH
(overtime_hours) OVERTIME, sum (absence_hours) ABSENCE, sum (sickness_hours) SICKNESS
from WORKTIME_IAV
where company_id = '200' and org_code = '&ORG_CODE'
group by emp_no
```

Crystal Reports quick report type

1. Open report in Crystal Report Designer
2. In Crystal Report Designer menu options go to Database -> Set Database Location option
3. Change the data source to newly created view by selecting the current view (IAL view) and choosing the new view to replace it with. Then click Update.



Set Datasource Location

Change the location of the data source by selecting the current database (or table) and choosing the database (or table) to replace it with. Then click Update.

Current Data Source:

```
⊟ report
    ⊟ CMBGSE1907/D1907
        ⊞ Properties
        ⊞ C_PROJ_RES_FCST_ITEM
```

Replace with:

```
BUSINESS_UNIT_STRUCT_COMP
BUSINESS_UNIT_STRUCTURE
BUSINESS_UNIT_STRUCTURE_COMP
BUSINESS_USE_PERCENTAGE
BUSPLN_ACC_LOV
BUSPLN_ADMINISTRATOR_GROUP
BUSPLN_HISTORY_LOG
BUSPLN_SYSTEM_PARAMETERS
C_PROJ_RES_FCST_ITEM_IAV
CACHE_MANAGEMENT
CACHED_ARTIFACT
CACHED_ARTIFACT_ELEMENT
CAD_LASER_CHECK_REP
CALC_LEAD_TIME_PRINT_REP
CALC_PROMO_PER_ORDER
```

PROS AND CONS OF USING CUSTOMIZATIONS OVER IAL

IALs	View as Customizations
Easy and quick deployment of object into customer database (+)	Deployment takes time. Needs to come through delivery (-)
No need to have knowledge about IFS Developer Studio (+)	IFS Developer Studio knowledge required (-)
Need to use different tools for develop the select query and for deployment (-)	Could use IFS Developer Studio for development and deployment (+)
IALs are not part of the build process (-)	Customizations are included in the build process (+)
IALs are not part of customers solution repository. Therefore, it's not visible for IFS consultant or partners (-)	Customizations are included in customer solutions repository (+)
Additional steps needed in application upgrade to deploy IALs (-)	No additional steps needed. Upgrades and related QA activities can be included in the build process with the possibility of future automation (+)

CONCLUSION

The IAL concept has been used for a very long time in IFS Applications. Therefore, it has been used for many different purposes in each customer installation. With IALs it was possible to deploy objects dynamically into the database (including the production database). However, these objects are not part of the content delivery pipeline or the IFS Applications build process. The knowledge about these objects are hidden from IFS Support and partners because these objects are not part of the customer's solution repository. Therefore, it is hard to predict the impact on these objects during upgrades and additional steps need to be taken to deploy these objects during application upgrades.

Going forward we want to move all configurations and integrations to be on the 'application model and new APIs' level rather than database / PLSQL API / view / table level. That move is necessary in order to significantly move the needle on enabling customers to both tailor/configure their solution and be more evergreen and for us to run efficiently in the cloud and open the door to data not always being in Oracle tables and logic not always being in PLSQL. These changes are necessary for our product to be long term viable and fit for the world we live in. All these changes together with their replacement solutions will not materialize in a single release, but the direction in which we must move is clear.