Interface

Table of Contents

[Helper: 2](#_Toc157776947)

[BAPI: 4](#_Toc157776948)

[Function Module for BAPI: 4](#_Toc157776949)

[Business Object For BAPI: 7](#_Toc157776950)

[IDOC: 16](#_Toc157776951)

# Helper:

Created an Employee Table with two fields Employee ID (NUMC 10) and Employee Name (Char 40)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Technical Settings:

A screenshot of a computer

Description automatically generated

Created a Structure with two fields Employee ID (NUMC 10) and Employee Name (Char 40)

A screenshot of a computer

Description automatically generated

# BAPI:

Creation of BAPI (BAPI - Function Module + Business Object):

PRE-REQUISITES FOR CREATING Function Module:

- Must create one Function Module (SE37) inside a Function Group (SE80)

- Function Module name must start with <namespace>BAPI\_<businessobject>\_<method> e.g., ZBAPI\_EMPLOYEE\_CREATE

- All parameter associated type must begin with <namespace>BAPI\_<businessobject> e.g., ZBAPI\_EMPLOYEE

- Export parameter must have one parameter with name 'RETURN' and of Type 'BAPIRET2'

- All Import/Export parameter must be pass by value

- Function Module must be RFC enabled

- Function Module must be released (Goto -> Release)

## Function Module for BAPI:

Create Function Module:

* Created a function group YZFG\_RJ\_EMP
* Created a function module YZBAPI\_EMPLOYEE\_CREATE (SE37)
* Release the Function Module

A screenshot of a computer

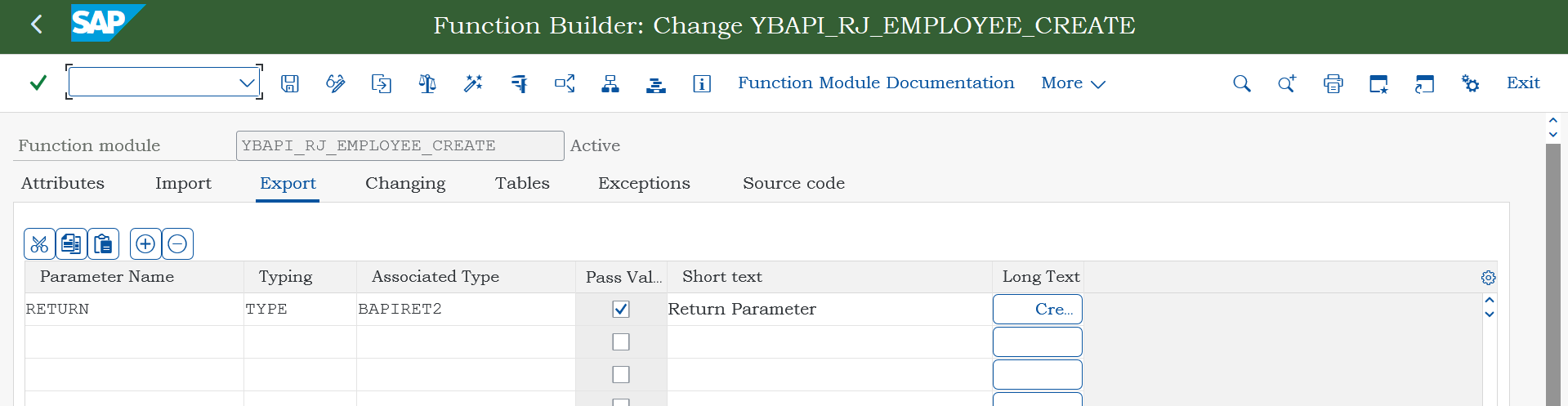
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Business Object For BAPI:

PRE-REQUISITES FOR CREATING BUSINESS OBJECT

> Create Business Object using T-code SW01

> Utilities -> API Methods -> Add Method (here give FM name)

> Object Type -> Change Release status -> Implemented

> Object Type -> Change Release status -> Released

> Click on generate button

> Finally you can check it in BAPI Explorer using T-code BAPI

Create Business Object SW01:

* Super Type : To Inherit the components of existing object type
* Object Type : Internal identification of business object
* Object Name : External identification of business object
* Name : Descriptive name of business object

Created a business object - YBO\_RJ\_EMP

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Create API method ‘Create’

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Change the Object Type status from modeled to implemented

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Test/Execute:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

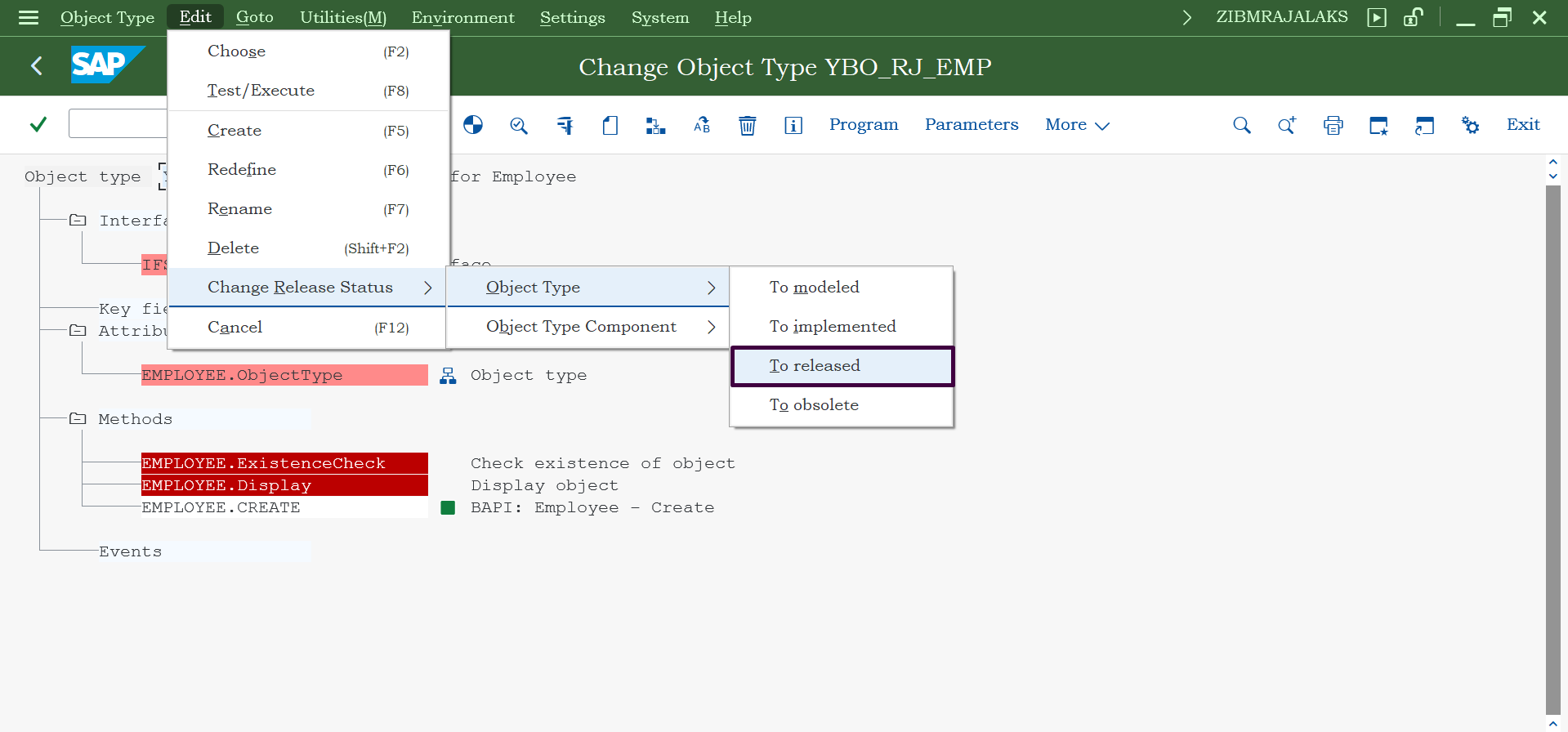
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Change object type status from Implemented to released



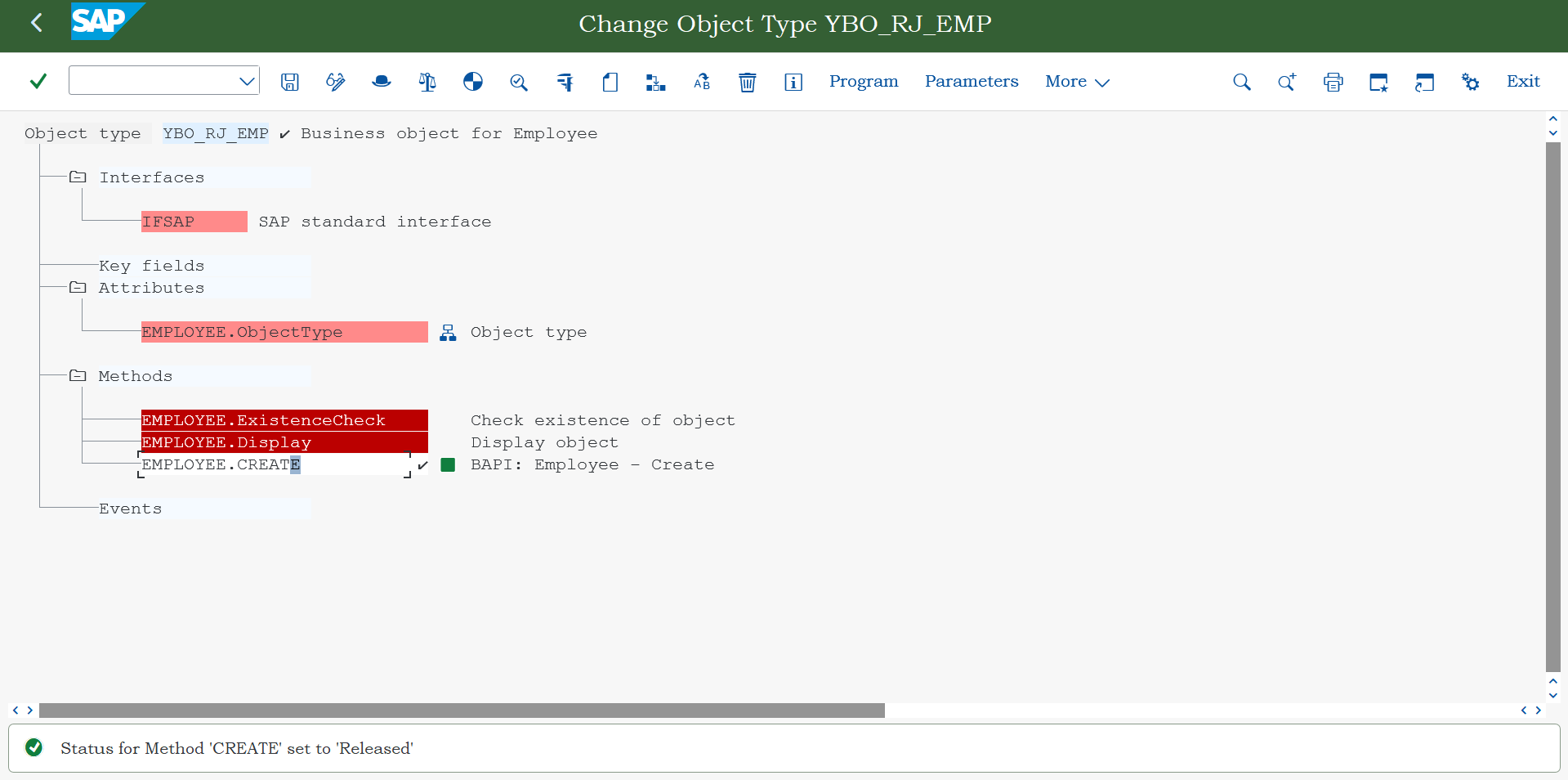
A screenshot of a computer

Description automatically generated

Change object type component from Implemented to released.

A screenshot of a computer

Description automatically generated



Generate Object type (to see this BAPI in BAPI explorer list)

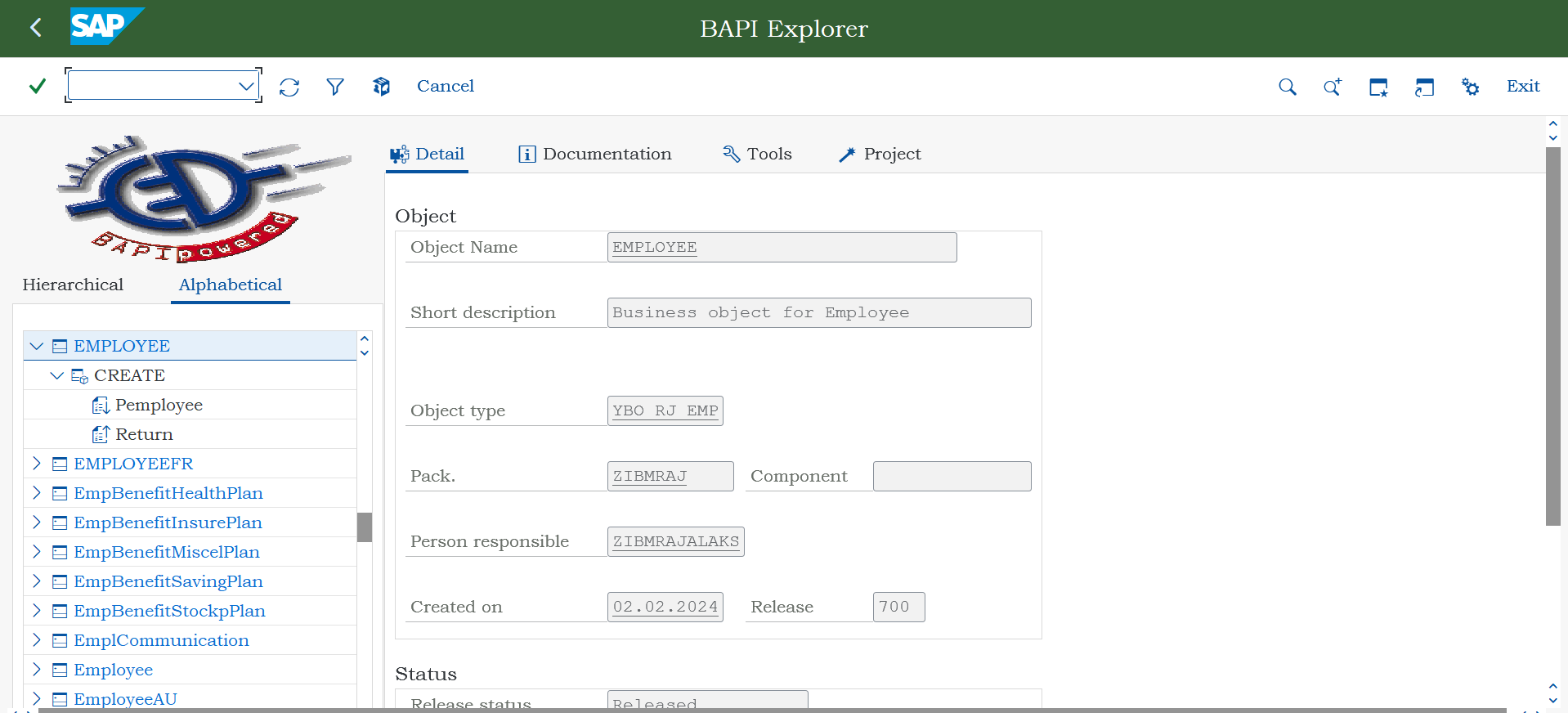
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Check in BAPI Explorer (T-code : BAPI)



# IDOC:

Generate ALE Interface for BAPI using the same object type and method

A screenshot of a computer

Description automatically generated

ALE is for Inbound Processing, so no need of FM for outbound processing.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Create Port (File Port) WE21

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Standard IDOC:

Create Segment – WE31

* E1MARAM

Create IDOC type/Basic type (Created on top of segments) – WE30

* MATMAS01

Create Message Type – WE81

* MATMAS

Assign Message Type to IDoc type – WE82

* MATMAS : MATMAS01

Create Logical system – SALE / BD54

* Sender : DEVCLNT108
* Receiver : UTSCLNT888

Assign Logical system – SALE / SCC4

* 108 : DEVCLNT108
* 888 : UTSCLNT888

Create RFC destination (Receiver details) – SM59

* UTS888RECV

Create RFC Port (Created on top of RFC Destination) – WE21

* A000000027 (generated by SAP)

Create Model View – SALE / BD64

* MATMASMV

Create Partner Profile – WE20

Test IDoc – WE19

A close-up of a list of information

Description automatically generated

An IDOC is a document consisting of the

* **Control Record** with the **Message Type, Basis Type/IDoc Type & Extension**, and
* **Status Record** which holds the history of statuses and
* **Data Record** which holds the actual data in Segments which holds the actual field values and are arranged in a predefined sequence via the Basic Type/IDoc Type which defines segment sequence, mandatory segments and/or repeating segments where the Extension is optional definition of additions to an existing Basic Type.
* **EDDID/EDDI4** stores Data record
* **EDDIC** stores the control information of IDoc.
* **EDDIS** stores the Status of IDoc
* **TBDLS** stores the logical system data.

Message Type (WE81)

IDOC Type or Basic Type (WE30)

Assign Msg Type to IDoc Type (WE82)

Segments (WE31)

Create Logical System (For both client and target) (SALE/BD54)

Assign Logical System to Client (SALE/SCC4)

Create RFC Destination (SM59)

Create Port and assign RFC Destination to it (WE21)

Create Model (BD64) and generate Partner Profile (WE20)

A diagram of a diagram

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated