

IT-Development Working Protocol

Project Name:

Version:

Table of Contents

[**1.** **INTRODUCTION** 3](#_Toc450811839)

[**2.** **WORKFLOW** 3](#_Toc450811840)

[**3.** **FOLDER STRUCTURE FOR EACH TASK** 3](#_Toc450811841)

[4. **SQL SCRIPT REQUIREMENTS** 5](#_Toc450811842)

[**4.1.** **REVISION HISTORY AND TASKCODE NUMBERS** 5](#_Toc450811843)

[**4.2.** **SCRIPT NAMING CONVENTIONS** 5](#_Toc450811844)

[**5.** **Process to work with a new BI requirement From Baktas** 6](#_Toc450811845)

1. **INTRODUCTION**

This document defines a process IT- development working protocol to effectively coordinate works between internal teamwork at Office’s Viet Nam. The detail of workflow is defined in diagram below.

This document also is to set the requirements for SQL Database release script and the way to manage script’s folder for each task

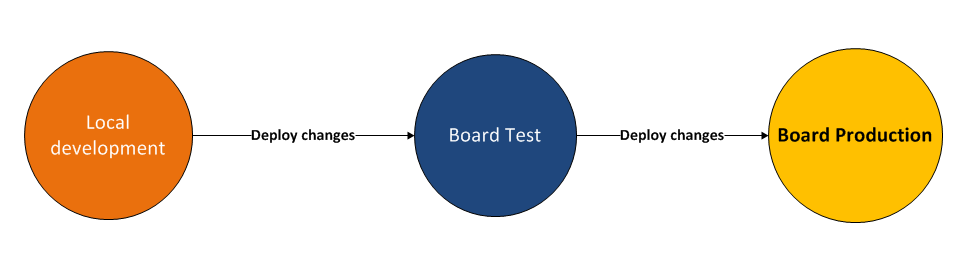
1. **WORKFLOW**

The detail of workflow is defined in this diagram and also be referred in this attachment.

>> click on attachment below

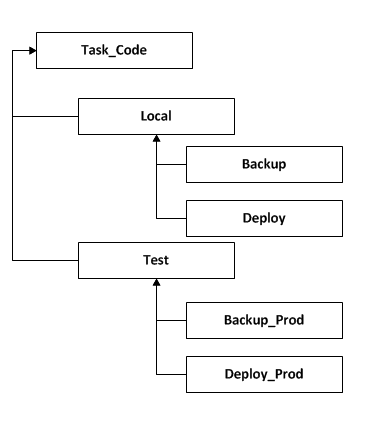


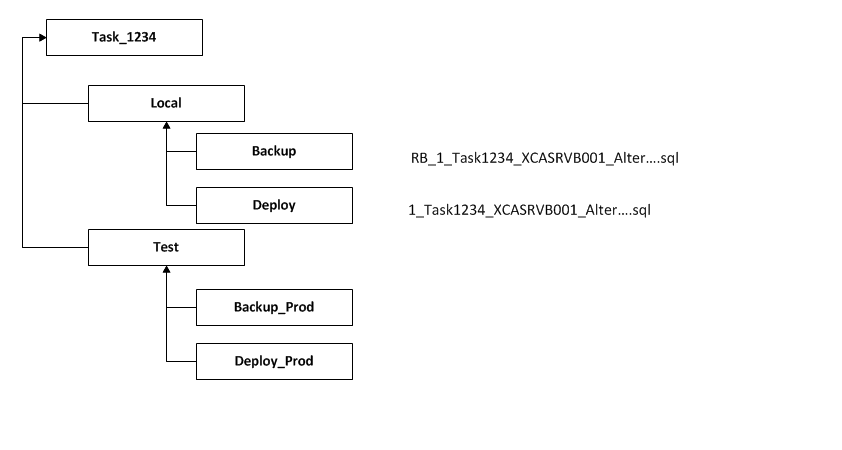
The based processing task



1. **FOLDER STRUCTURE FOR EACH TASK**

To conveniently manage result in development processes how to track script changes between environments. Developer can create folder on local to store file in development process



Example : we have task with taskcode is 1234

1. **SQL SCRIPT REQUIREMENTS**

As part of the release management to track history development system and also support developers to understand the changes in stages System.

It’s is important to control deployment of the changes on system by reviewing and checking deployment without any issue.

* 1. **REVISION HISTORY AND TASKCODE NUMBERS**

All Tasks must bear a TaskCode Change number in the comment section of the script. A revision history is required with the name of the revising developer to be added in the script. Example as follows:

--V1.0 \*\*123\*\* Initial Creation ----Minh Tran   
--V1.1 \*\*457\*\* Add certain functionalities – the name of developer

* 1. **SCRIPT NAMING CONVENTIONS**

Scripts must be named according to the following naming convention to facilitate smooth Release process.

**The naming convention is as follows:**

**n\_ TaskCodeNumber \_TargetServername\_ScriptFunction.sql**

n: release sequence, order of which the script needed to be run when

Multiple scripts are involved in one change request, and all scripts must be combined into one in the order of execution

**TaskCodeNumber**: **The TaskCode**

**Target servername**: **the target server the script is required to be run**

**ScriptFunction**: **the function of the script**

**For Example, the following is name**

**Deploy script :**

1\_Task18969\_XCASRVB001 \_CreateProcForProject1.sql

**Rollback script:**

Add “RB” in the front of the file name, for example

RB\_1\_ Task18969\_XCASRVB001 \_CreateProcForProject1.sql

1. **Process to work with a new BI requirement From Baktas**

**Step 1:**

Make sure you can understand the requirement enough to identify whether it is a big or small change (should have a rough estimate). If not, keep communicating until the requirement is clear.

**Step 2:**

If the requirement is considered to implement, then should generate a draft plan which including:

  What needs to be done in BOARD (creating new cubes, entities, data readers)

  What needs to be done in DWH (Creating/modifying the existing fact tables, ETL flows to handle the new logic)

  If the requirement requires new or extended information then have to know what exactly that information come from (i.e. which source system, which source tables or which fields)

  Estimate for each steps above, total estimate (how much time to apply the update to the live server)

**Step 3:**

Always make sure that the requirement is clear and the stakeholders are on the same page. Remember to work **on the test server first**, if checked and confirmed then apply to the live server

**Step 4:**

If there are many tasks which need to be solved at the same time, ask for priority, and try to work on only one task at one time (this is important. Otherwise, it would be very messy and more effort to handle in test environment)

**Step 5:**

In case you have the feeling that resource needs to be extended - not enough time or need more communication regarding to the requirement understanding – then inform to the task manager ASAP.

**Step 6:**

Move the task to live server: Ensure that you have already checked data both **backend** and **frontend side** using your common knowledge of interactions between the existing data on the live server and your task result on the test server then inform to the task manager for final check to get his/her confirmation.