

Development Server Data Cleaning

- 1. Check existing count of XPDLs
- 2. Check the count of unused XPDLs
- 3. Delete completed process instances data to clean up more XPDLs from cache
- 4. Delete the unused XPDLs

In development server, due to the fact that process design will always get updated and developed, there will be more and more XPDLs cached in the memory. In the long run, this may cause "java.lang.OutOfMemoryError" exception during runtime or server startup. To solve this issue, we will need to clean up the unused XPDLs with following steps. The following example queries are written in MySQL and MSSQL syntax.



Note

1. Please shutdown your application server.
2. Please do a full database backup before proceeding to the following steps.

1. Check existing count of XPDLs

On a common production server, normally there are only few app versions and each version only tied to 1 process version. This is because we won't do any development in production server, so the process version is only increased during the app imported. But in the development server, it will be a different case. The process version will be going up to few hundred process version which also means there are few hundred of XPDL data are cached in memory. To check how many XPDLs are cached, please run the following query in your database server.

MySQL, MSSQL, Oracle

```
select count(*) from SHKXPDLs
```

2. Check the count of unused XPDLs

In all the cached process version XPDLs, there is a case that some process version is not used by any process instances. These unused XPDLs can be safely deleted. To check the number of unused XPDLs, we can use the following query. Then, we can know how many are unused in the total number of XPDLs.

MySQL, MSSQL

```
select count(*) from SHKXPDLs x
where concat(x.XPDLId, x.XPDLVersion) not in (
    select concat(packageId,packageVersion) as id from (
        select def.PackageId as packageId, def.ProcessDefinitionVersion as packageVersion
        from SHKActivities act
        join SHKProcesses pro on act.Process = pro.oid
        left join SHKProcessDefinitions def on pro.ProcessDefinition = def.oid
        group by def.PackageId, def.ProcessDefinitionVersion
        union
        select packageId, packageVersion from app_package
    ) as used_processes group by packageId, packageVersion
);
```

Oracle

```
select count(*) from SHKXPDLs x
where concat(x.XPDLId, x.XPDLVersion) not in (
    select concat(packageId,packageVersion) as id from (
        select def.PackageId as packageId, TO_CHAR(def.ProcessDefinitionVersion) as packageVersion
        from SHKActivities act
        join SHKProcesses pro on act.Process = pro.oid
        left join SHKProcessDefinitions def on pro.ProcessDefinition = def.oid
        group by def.PackageId, def.ProcessDefinitionVersion
        union
        select packageId, TO_CHAR(packageVersion) from app_package
    ) group by packageId, packageVersion
);
```

If you would like to list the process versions, you can use the following query.

MySQL, MSSQL

```
select * from SHKXPDLs x
where concat(x.XPDLId, x.XPDLVersion) not in (
    select concat(packageId,packageVersion) as id from (
        select def.PackageId as packageId, def.ProcessDefinitionVersion as packageVersion
        from SHKActivities act
        join SHKProcesses pro on act.Process = pro.oid
        left join SHKProcessDefinitions def on pro.ProcessDefinition = def.oid
        group by def.PackageId, def.ProcessDefinitionVersion
        union
        select packageId, packageVersion from app_package
    ) as used_processes group by packageId, packageVersion
);
```

Oracle

```
select * from SHKXPDLs x
where concat(x.XPDLId, x.XPDLVersion) not in (
    select concat(packageId,packageVersion) as id from (
        select def.PackageId as packageId, TO_CHAR(def.ProcessDefinitionVersion) as packageVersion
        from SHKActivities act
        join SHKProcesses pro on act.Process = pro.oid
        left join SHKProcessDefinitions def on pro.ProcessDefinition = def.oid
        group by def.PackageId, def.ProcessDefinitionVersion
        union
        select packageId, TO_CHAR(packageVersion) from app_package
    ) group by packageId, packageVersion
);
```

3. Delete completed process instances data to clean up more XPDLs from cache

If the process instances data are not important to you or you have your process instance data captured a copy by the [Process Data Collector](#) plugin, you can delete your process instances data to increase the number of unused XPDLs. In this case, only the process version for running process instances and the latest process version for each app version will be kept.

MySQL

```
SET FOREIGN_KEY_CHECKS=0;

delete sa from SHKAssignmentsTable as sa
left join SHKProcesses as sp on sa.ActivityProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);

delete saj from SHKAndJoinTable as saj
left join SHKActivities as sac on saj.Activity = sac.oid
left join SHKProcesses as sp on sac.ProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);

delete sd from SHKDeadlines as sd
left join SHKActivities as sac on sd.Activity = sac.oid
left join SHKProcesses as sp on sac.ProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);

delete sad from SHKActivityData as sad
left join SHKActivities as sac on sad.Activity = sac.oid
left join SHKProcesses as sp on sac.ProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);
delete sac from SHKActivities as sac
left join SHKProcesses as sp on sac.ProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);

delete spd from SHKProcessData as spd
left join SHKProcesses as sp on spd.Process = sp.oid
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);

delete spr from SHKProcessRequesters as spr
left join SHKProcesses as sp on spr.Id = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);

delete from SHKProcesses where (State = 1000006 or State = 1000008 or State = 1000010);

SET FOREIGN_KEY_CHECKS=1;
```

MSSQL

```
EXEC sp_MSforeachtable "ALTER TABLE ? NOCHECK CONSTRAINT all"
```

```
delete sa from SHKAssignmentsTable as sa
left join SHKProcesses as sp on sa.ActivityProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);
```

```
delete saj from SHKAndJoinTable as saj
left join SHKActivities as sac on saj.Activity = sac.oid
left join SHKProcesses as sp on sac.ProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);
```

```
delete sd from SHKDeadlines as sd
left join SHKActivities as sac on sd.Activity = sac.oid
left join SHKProcesses as sp on sac.ProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);
```

```
delete sad from SHKActivityData as sad
left join SHKActivities as sac on sad.Activity = sac.oid
left join SHKProcesses as sp on sac.ProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);
```

```
delete sac from SHKActivities as sac
left join SHKProcesses as sp on sac.ProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);
```

```
delete spd from SHKProcessData as spd
left join SHKProcesses as sp on spd.Process = sp.oid
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);
```

```
delete spr from SHKProcessRequesters as spr
left join SHKProcesses as sp on spr.Id = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);
```

```
delete from SHKProcesses where (State = 1000006 or State = 1000008 or State = 1000010);
```

```
EXEC sp_MSforeachtable "ALTER TABLE ? WITH CHECK CHECK CONSTRAINT all"
```

Oracle

```
delete (select sa.* from SHKAssignmentsTable sa
left join SHKProcesses sp on sa.ActivityProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010));

delete saj from SHKAndJoinTable as saj
left join SHKActivities as sac on saj.Activity = sac.oid
left join SHKProcesses as sp on sac.ProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);

delete sd from SHKDeadlines as sd
left join SHKActivities as sac on sd.Activity = sac.oid
left join SHKProcesses as sp on sac.ProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010);

delete (select sad.* from SHKActivityData sad
left join SHKActivities sac on sad.Activity = sac.oid
left join SHKProcesses sp on sac.ProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010));

delete (select sac.* from SHKActivities sac
left join SHKProcesses sp on sac.ProcessId = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010));

delete (select spd.* from SHKProcessData spd
left join SHKProcesses sp on spd.Process = sp.oid
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010));

delete (select spr.* from SHKProcessRequesters spr
left join SHKProcesses sp on spr.Id = sp.ID
where (sp.State = 1000006 or sp.State = 1000008 or sp.State = 1000010));

delete from SHKProcesses where (State = 1000006 or State = 1000008 or State = 1000010);
```

If you would like to clean the all the process instances data including the running process instances, you can use the following query.

WARNING This deletes all your process records. Perform a backup before running the query below and is not recommended to be executed on a production server!

MySQL

```
SET FOREIGN_KEY_CHECKS=0;

delete from SHKAssignmentsTable;
delete from SHKAndJoinTable;
delete from SHKDeadlines;
delete from SHKActivityData;
delete from SHKActivities;
delete from SHKProcessData;
delete from SHKProcessRequesters;
delete from SHKProcesses;

SET FOREIGN_KEY_CHECKS=1;
```

MSSQL

```
EXEC sp_MSforeachtable "ALTER TABLE ? NOCHECK CONSTRAINT all"

delete from SHKAssignmentsTable;
delete from SHKAndJoinTable;
delete from SHKDeadlines;
delete from SHKActivityData;
delete from SHKActivities;
delete from SHKProcessData;
delete from SHKProcessRequesters;
delete from SHKProcesses;

EXEC sp_MSforeachtable "ALTER TABLE ? WITH CHECK CHECK CONSTRAINT all"
```

Oracle

```
delete from SHKAssignmentsTable;
delete from SHKAndJoinTable;
delete from SHKDeadlines;
delete from SHKActivityData;
delete from SHKActivities;
delete from SHKProcessData;
delete from SHKProcessRequesters;
delete from SHKProcesses;
```

4. Delete the unused XDPLs

Now, you can delete the unused XDPLs and it will be free from your memory cache when next server startup.

MySQL

```
SET FOREIGN_KEY_CHECKS=0;
delete x, xd from SHKXPDLs x join SHKXPDLData xd on x.oid= xd.XPDL
where concat(x.XPDLId, x.XPDLVersion) not in (
    select concat(packageId,packageVersion) as id from (
        select def.PackageId as packageId, def.ProcessDefinitionVersion as packageVersion
        from SHKActivities act
        join SHKProcesses pro on act.Process = pro.oid
        left join SHKProcessDefinitions def on pro.ProcessDefinition = def.oid
        group by def.PackageId, def.ProcessDefinitionVersion
        union
        select packageId, packageVersion from app_package
    ) as used_processes group by packageId, packageVersion
);
SET FOREIGN_KEY_CHECKS=1;
```

MSSQL

```
EXEC sp_MSforeachtable "ALTER TABLE ? NOCHECK CONSTRAINT all"
```

```
delete xd from SHKXPDLData as xd
left join SHKXPDLs as x on x.oid= xd.XPDL
where concat(x.XPDLId, x.XPDLVersion) not in (
    select concat(packageId,packageVersion) as id from (
        select def.PackageId as packageId, def.ProcessDefinitionVersion as packageVersion
        from SHKActivities act
        join SHKProcesses pro on act.Process = pro.oid
        left join SHKProcessDefinitions def on pro.ProcessDefinition = def.oid
        group by def.PackageId, def.ProcessDefinitionVersion
        union
        select packageId, packageVersion from app_package
    ) as used_processes group by packageId, packageVersion
);
```

```
delete x from SHKXPDLs as x
where concat(x.XPDLId, x.XPDLVersion) not in (
    select concat(packageId,packageVersion) as id from (
        select def.PackageId as packageId, def.ProcessDefinitionVersion as packageVersion
        from SHKActivities act
        join SHKProcesses pro on act.Process = pro.oid
        left join SHKProcessDefinitions def on pro.ProcessDefinition = def.oid
        group by def.PackageId, def.ProcessDefinitionVersion
        union
        select packageId, packageVersion from app_package
    ) as used_processes group by packageId, packageVersion
);
```

```
EXEC sp_MSforeachtable "ALTER TABLE ? WITH CHECK CHECK CONSTRAINT all"
```

Oracle

```
delete (select xd.* from SHKXPDLData xd
left join SHKXPDLs x on x.oid= xd.XPDL
where concat(x.XPDLId, x.XPDLVersion) not in (
    select concat(packageId,packageVersion) as id from (
        select def.PackageId as packageId, TO_CHAR(def.ProcessDefinitionVersion) as packageVersion
        from SHKActivities act
        join SHKProcesses pro on act.Process = pro.oid
        left join SHKProcessDefinitions def on pro.ProcessDefinition = def.oid
        group by def.PackageId, def.ProcessDefinitionVersion
        union
        select packageId, TO_CHAR(packageVersion) from app_package
    ) group by packageId, packageVersion
));
```

```
delete (select x.* from SHKXPDLs x
where concat(x.XPDLId, x.XPDLVersion) not in (
    select concat(packageId,packageVersion) as id from (
        select def.PackageId as packageId, TO_CHAR(def.ProcessDefinitionVersion) as packageVersion
        from SHKActivities act
        join SHKProcesses pro on act.Process = pro.oid
        left join SHKProcessDefinitions def on pro.ProcessDefinition = def.oid
        group by def.PackageId, def.ProcessDefinitionVersion
        union
        select packageId, TO_CHAR(packageVersion) from app_package
    ) group by packageId, packageVersion
));
```