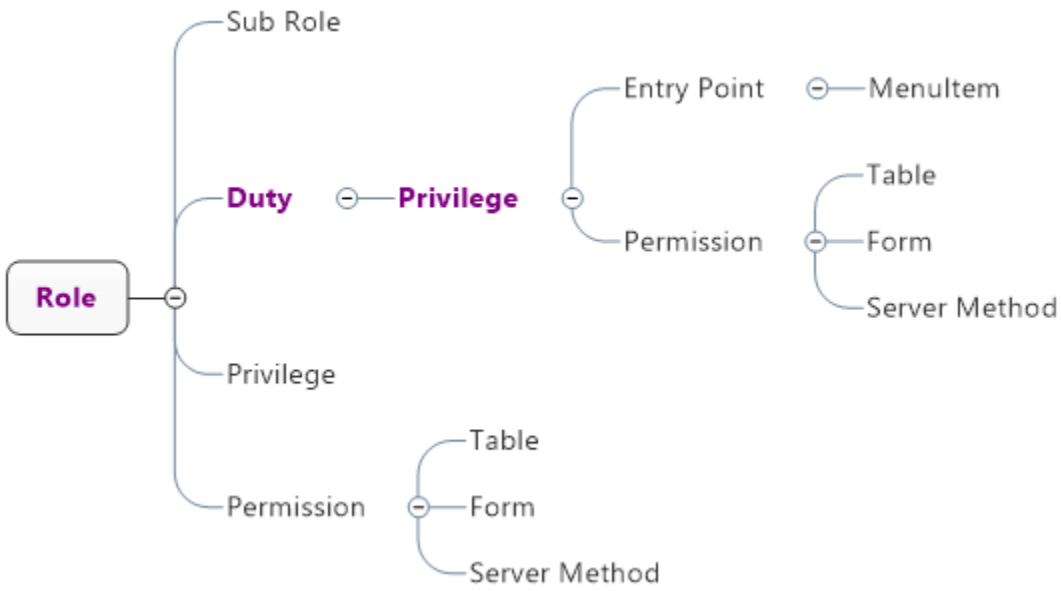
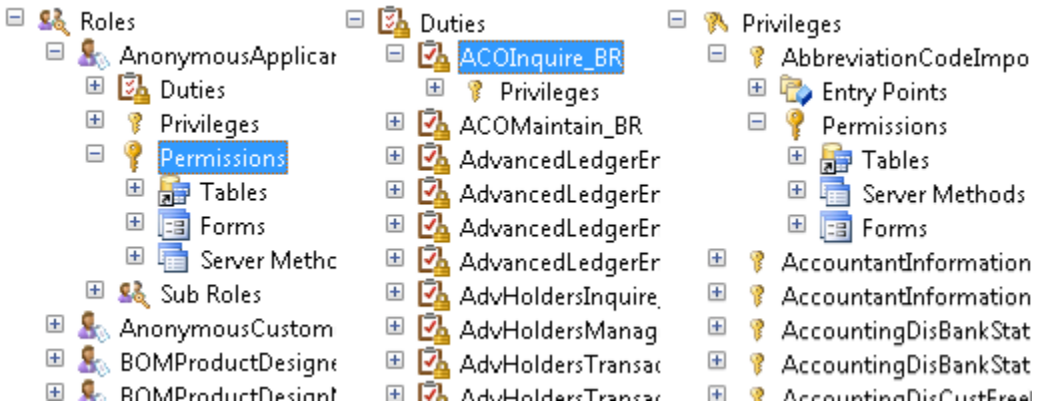
AX 2012 基于角色的权限设计 2——权限结构

本文重点结合 Microsoft Dynamics AX 2012 的权限体系和标准角色，分析如何对 Role, Duty 和 Privilege 进行合理地利用，从而设计出一种既便于初始权限收集，又利于后续维护的权限结构。初始权限收集的主要难点在于如何整理一个表述清晰的权限模板，并通过其与用户取得良好的沟通；后续权限维护的主要难点在于如何快速响应现有岗位的职责范围变动、如何将开发的新功能准确地赋予相应用户。一个合理的权限结构，应该对这两个方面有很好的支撑。

一、 基本权限结构

在 AX 系统中，权限的设定是基于角色的，下图展示了 Role, Duty 和 Privilege 各自包含的内容，及三者的相互关系。





1. **Role（角色）**

   包含 Sub Role, Duty, Privilege 和对 Table, Form, Server Method 的 Permission 设置。

2. **Duty（职责）**

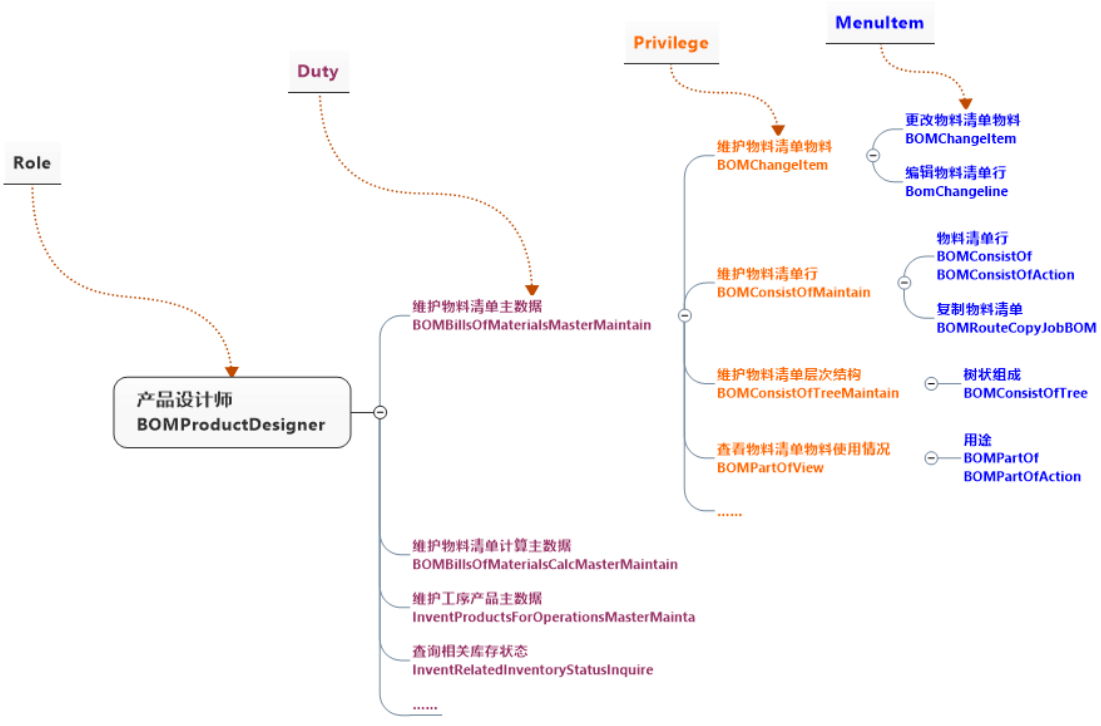
   只包含 Privilege，相当于 Privilege 的分组。

3. **Privilege（特权）**

   包含 MenuItem 和对 Table, Form, Server Method 的 Permission 设置。

二、 权限对象定义

根据上述 AX 基本权限结构，就可以完成一项有效的权限设置，但考虑到权限管理是一项持续性的工作，我们还应该对 Role, Duty 和 Privilege 这 3 个权限对象进行合理地定义，明确其各自包含的内容，才能搭建出一个清晰的、具有较强适应性的，能够帮助系统管理员快速、准确地为用户分配所需权限的权限结构。实际上，AX 系统标准角色对这 3 个权限对象的定义就十分准确、合理，以标准角色“产品设计师”为例，示意图如下：



这是一个典型的 Role→Duty→Privilege→MenuItem 形式的权限设计，对 Role, Duty 和 Privilege 的定义十分准确。

1. Role 按照岗位定义，避免了归属于同一角色的不同用户有不同权限需求的问题。在实际使用中，考虑到同一岗位在不同公司可能具有不同的职责范围，建议按照“公司+账套+岗位”的方式定义。

2. Duty 按照最小的 ERP 标准业务操作定义，保证了通用性，即同一职责可以引用到多个角色中，使权限设置更加便捷。

3. Privilege 按照最小的系统功能定义，保证了稳定性，一般不需要对已有特权做任何变动。

总结下来，对 Role, Duty, Privilege 的最佳定义方式如下：

权限对象	Role - 角色	Duty - 职责	Privilege - 特权
定义	公司+账套+岗位	最小的 ERP 标准业务操作	最小的系统功能单元
举例	MS-Corp1_产品设计师 MS_Corp1_BOMProductDesigner	维护物料清单主数据 BOMBillsOfMaterialsMasterMaintain	维护物料清单物料 BOMChangeItem
	MS-Corp2_产品设计师 MS_Corp2_BOMProductDesigner	物料清单审核 BOMApprove	维护物料清单行 BOMConsistOfMaintain
	MS-Corp1_会计经理 MS_Corp1_LedgerAccountingManager	维护主银行账户 BankBankAccountsMaintain	维护物料清单层次结构 BOMConsistOfTreeMaintain
	MS-Corp2_会计经理 MS_Corp2_LedgerAccountingManager	维护银行交易记录 BankBankTransactionsMaintain	查看物料清单物料使用情况 BOMPartOfView

注意：由于 Role→Duty→Privilege 是逐层包含的关系，Privilege 和 Duty 本身的变动一般会影响很多角色，所以，原则上 Privilege 一经创建就不应再做改动，Duty 除了放入开发的新功能外，也不应再做其他改动。用户所在岗位的变动，通过更改用户所属的 Role 实现；用户岗位的职责范围变动，通过更改 Role 包含的 Duty，或者对 Role 进行特殊设置实现。（Role 的特殊设置在下节介绍）

以上是在理论层面做的权限结构分析与总结，在实际权限结构设计过程中，由于 AX 系统功能繁多，客户的需求各有不同，这要求顾问既要了解系统的全部功能（Privilege），又要了解 ERP 的标准操作（Duty），还要了解企业的业务流程设计和岗位职责划分（Role），这对顾问的能力和耐心都是一种挑战，因此，在实际项目实施过程中，往往很难搭建出一套最佳的权限结构。

三、 角色特殊设置

在 AX 系统中，通过对 Role 的一些特殊设置，可以在不改变 Duty 和 Privilege 的前提下，实现权限的调整，这些设置的影响范围小，便于控制，因此非常实用。

1. Role 与 Privilege 直接关联

例如，为一项新开发的功能建立特权 Privilege1 后，发现其无法归属到任何职责中，即无法划分到任何一项 ERP 标准业务操作中，只有角色 RoleA 和 RoleB 应该具有该项新功能的使用权。

这种情况下，可以不用 Role→Duty→Privilege 的设置方式，而采用 Role→Privilege 的方式，跳过 Duty，直接将 Privilege1 赋予 RoleA 和 RoleB。

2. Role 的覆盖权限

例如公司 Corp1 和公司 Corp2 中分别建立了“Corp1 采购员”和“Corp2 采购员”角色，两个角色都要包含“采购订单维护”职责，但是 Corp1 要求采购员必须基于采购申请创建采购订单，即采购订单中的“物料编码”不允许编辑，而 Corp2 没有要求。

这种情况下，职责“采购订单维护”是合理的，无需调整，只需对角色“Corp1 采购员”进行“覆盖权限”操作，不允许其编辑采购订单中的“物料编码”即可。

说明：建议在 AOT 中进行覆盖权限的操作，而不要在 AX 前台，这更便于 AX 各环境（开发、测试、生产环境）的代码同步。

By Lixin Liu 2016-7-15