



K2 [blackpearl] Roles and Advanced Destination Rules

USING [BLACKPEARL] TO PLAN ACTIVITIES AND DYNAMICALLY RESOLVE USERS

January 7, 2008

Learn about dynamic roles and advanced destination planning with K2 [blackpearl], including specific examples of activity planning based on common business scenarios.



INTRODUCTION

In K2 blackpearl, giving a user the ability to take action in a workflow is relatively simple. It is accomplished by adding that user as a **Destination** for the activity. As a destination, the user is typically sent a notification and the task is added to their worklist. Clicking on the task loads an action form, either an InfoPath form or an ASP.NET form, containing the available actions. This simple scenario, however, does not reflect the real world or the power of K2 blackpearl. With the introduction of advanced destination planning and roles, blackpearl provides the flexibility needed for complex workflow processes to be built for dynamic business environments.

CONTENTS

ROLES IN K2 [BLACKPEARL]..... 3

- > What makes up a Role?..... 3
- > Using a SmartObject Method in a Role 4
- > Using XML as a Destination Set..... 4
- > Synchronized vs. Cached Roles 4
- > Dynamic Resolution of Role Membership..... 4
- > Using Roles in a Process..... 5

TASK ALLOCATION, RIGHTS AND SECURITY 6

- > The Context Grid 6
- > The Difference Between Delegation and Redirection of Tasks 7

ADVANCED DESTINATION RULES..... 7

- > Plan per destination - All at once (Parallel planning)..... 8
- > Plan per destination - One at a time (Serial planning)..... 8
- > Plan per slot (no destinations)..... 9

SCENARIO EXAMPLES..... 9

- > Plan just once Activities 9
- > Plan Per Destination - All at Once (Parallel) Activities 11
- > Plan Per Destination - One at a time (Serial) Activities 13

CONCLUSION..... 14

ROLES IN K2 [BLACKPEARL]

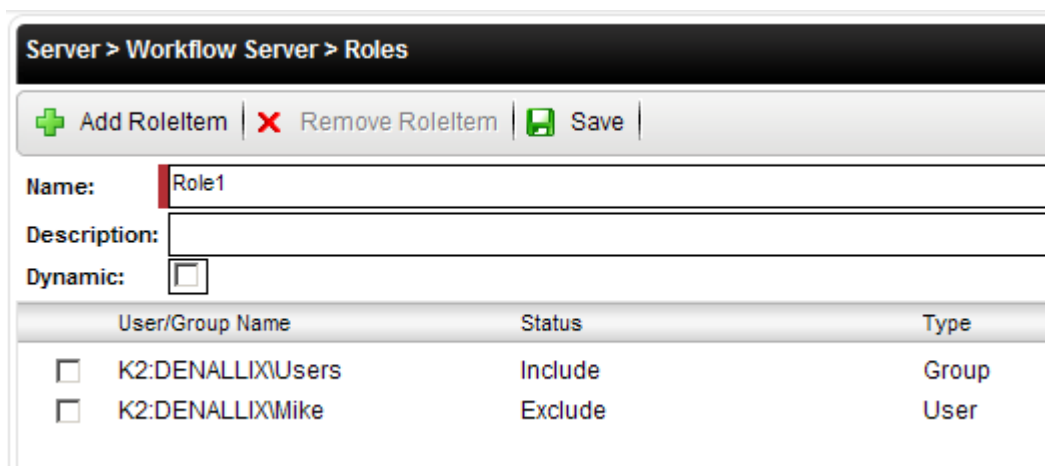
To understand advanced destination planning, it is necessary to first understand how roles are built and used in blackpearl. Much like groups or roles in other systems, such as Active Directory, roles in blackpearl are groups of users. Unlike other systems, roles in blackpearl can be dynamic if based on SmartObject queries.

WHAT MAKES UP A ROLE?

K2 roles are created in the Management Console of K2 Workspace. A new role is given a name and one or more role items. These items can be based on the following entities:


- > Users or Groups from Active Directory or SQL
- > Other K2 Roles
- > Result from SmartObject Methods

In addition to specifying one the above items for each role item, each role item can be either included in or excluded from the role. As illustrated in Figure 1, all users are included in the Role1 role except Mike. SmartObject methods can also be included or excluded from the role.



[FIGURE 1: INCLUDING AND EXCLUDING USERS AND GROUPS.]

In addition to being able to specify users from Active Directory or SQL, K2 user providers written for other systems can be used as the basis of a role item.

 **Notes:** Though beyond the scope of this article, the user providers currently written for K2 blackpearl include Active Directory and SQL User Manager, but role providers for other systems may be developed by third parties or released as enhancements to K2 blackpearl.

Each individual role item is limited to one of the sources above, but a role may contain role items that are based on any valid user entity. This means that a role can include or exclude users and groups from Active Directory, SQL, a SmartObject query and another K2 role. The ability to span user providers and define roles with a SmartObject method, in addition to the ability to include or exclude role items from a role, makes K2 roles very powerful. It is recommended, however, that the establishment of roles for use in business processes be carefully planned and executed, as a role's resultant membership may be difficult to determine if many role items are present.

USING A SMARTOBJECT METHOD IN A ROLE

When a SmartObject contains properties that directly maps to users in one or more of the user providers configured for the K2 server, a role can be configured for a method on that SmartObject. The method is typically the GetList method which returns multiple records from the SmartObject. Each record should include a username that corresponds to a username in the default user provider. The results of the SmartObject method are stored like other role membership information and the refresh interval can be configured on a per-process basis once the process using that role is deployed to the server.

USING XML AS A DESTINATION SET

User information contained in an XML structure, either as a process field or XML document outside of the process, can be used as a basis of a destination set. It is important to remember, however, that users will only be resolved by the K2 server if they are present in the default label, such as "K2." It is not possible to use an XML structure as the basis for a role.

SYNCHRONIZED VS. CACHED ROLES

Default behavior: When an instance of an activity gets created and tasks are assigned, a single task will be assigned to the role and any member of the role can action the task. Additionally, the role membership will be resolved and cached at a configurable 10 minute interval. The default destination setting is **Create a slot for each role**, which is available on one of the pages of the Destination Rules wizard in advanced mode.

Advanced behavior: The advanced Destination Rule options (see Advanced Destination Rules section in this document) can be adjusted to assign a single tasks item to every individual member of a role (**Resolve all role to users**), this happens only once and changes to roles will not be reflected. Optionally the role members and their individual tasks can be kept in sync (**Keep roles synchronized**) and tasks will be removed or added to each user's task list based on their membership. Membership is resolved and cached at a configurable interval, the default of which is 10 minutes.

Execution modes: How the role resolution behaves will be affected when using **Plan just once** or **Plan per destination - All at once** or **One at a time**. Please see the Advanced Destination Rules section for more details.

DYNAMIC RESOLUTION OF ROLE MEMBERSHIP

Depending on the solution requirement, it is sometimes necessary to use on-demand resolution of roles instead of using an interval-based role resolution. Dynamic roles are on-demand and are refreshed when dynamic roles are defined and every time worklists are opened which contain items that use the role. For example, if a solution requires tasks to be assigned to users signing on at a certain location, a role can be created to on-demand and dynamically resolve a user's location and make the tasks visible to them. By default when multiple users open worklists containing items that use the role, the dynamic roles are only resolved within a configurable three second window. When creating the role there is a **Dynamic** checkbox as shown in Figure 1. Checking this box makes every call to the role a dynamic call, regardless of the process it is used in. If the role is not dynamic, checking the **Dynamic** checkbox next to the **Interval (minutes)** text box

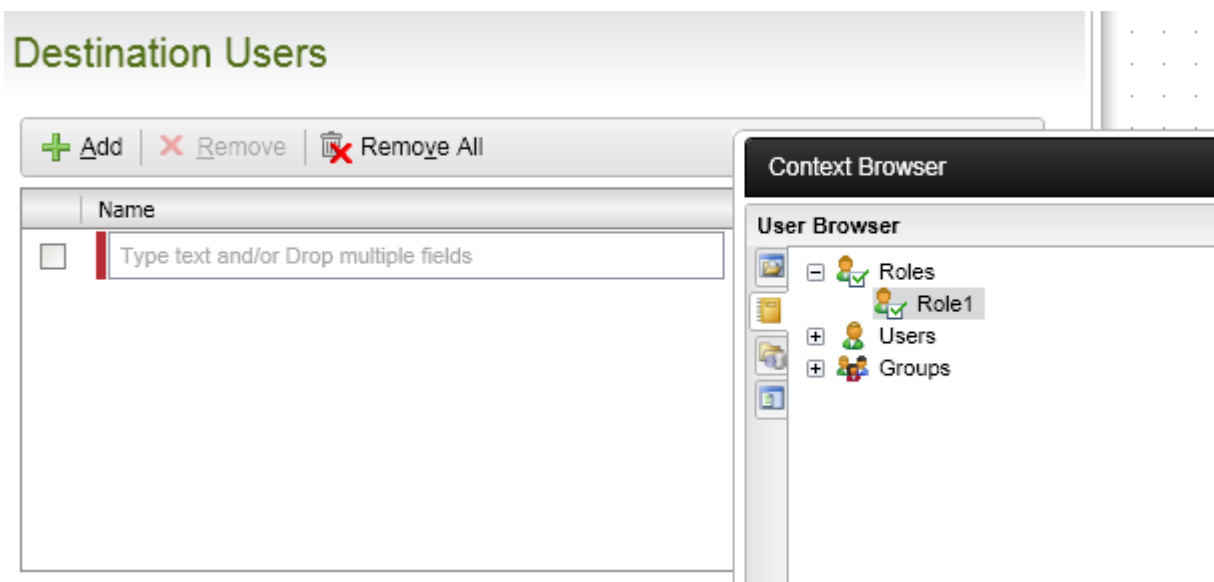
on the **Role** page of the process tree in the Management Console will force the role to resolve dynamically instead of at the set interval, but only for the process in which it is used.

**Notes:**

- > A Dynamic Role can only be used with the **Create a slot for each role** option and the **Plan just once** or **Plan Per Destination - All at Once** execution modes.
- > Dynamic roles were first introduced with K2 [blackpearl] Service Pack 1.

USING ROLES IN A PROCESS

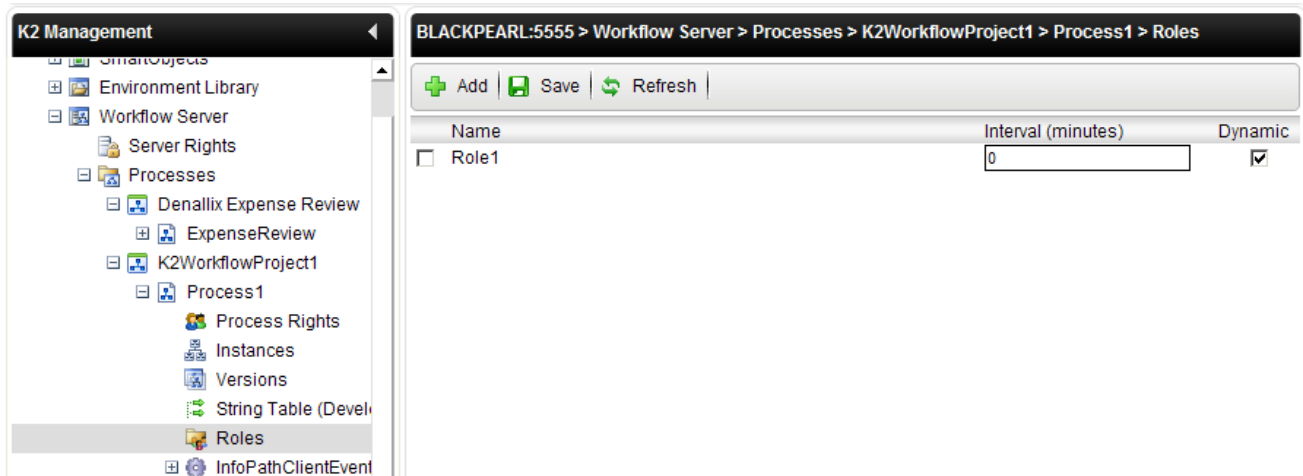
As stated before, roles are created and managed in the Management Console of K2 Workspace. Once created, roles are displayed in the Context Browser and the K2 Object Browser in the K2 Designer for Visual Studio and the K2 Designer for Visio. Roles cannot be used in the K2 Web Designer. On the User Browser tab, under Roles, each role available on the K2 server is listed, as shown in Figure 2.



[FIGURE 2: USING ROLES IN A PROCESS.]

When a role is added to a destination, the role is assigned rights to the slot that is created for the activity. Each member of the role therefore has the same rights to the activity and the task associated with the event is displayed on each member's worklist. Tasks will automatically appear or disappear from a user's worklist based on their role membership. This is the default behavior of blackpearl when using roles and also applies if using multiple individual destination users instead of a role. This is what happens for all default activities. The plan for default activities is **Plan just once**, or simply Plan once. Other planning options will be covered in the Advanced Destination Rules section. In a plan once process, when a member of the role actions the task, it is removed from everyone's worklist.

When a process that uses one or more roles is deployed, each role is displayed in the Management Console under the process tree, as shown in Figure 3.



[FIGURE 3: THE ROLES PAGE IN THE PROCESS TREE IN THE MANAGEMENT CONSOLE.]

Alternatives for planning the activity, controlling the number of slots, and creating destination sets are discussed in the next section.

TASK ALLOCATION, RIGHTS AND SECURITY

At runtime the K2 Server creates a single task item for every activity instance containing a client event, and assigns default access rights to users or roles for that single task item. This allows the server to be able to add or remove tasks to/from users and roles dynamically at runtime. The concept of slots are used to determine when a task item (client event) is completed and can move on to the next activity based on the outcome of a succeeding rule

The default actions and rights are defined at design time within the process definition. As a destination user, all actions configured for the process are available by default. Once process instances are created, these rights can be tailored in the Management Console. In addition, when a task is delegated, rights for each action must be configured by the person delegating the task. This is useful, for example, when one of the possible actions for an activity should not be completed by a delegated user. Actions that are not specifically granted are not present when the delegated user opens the form.

THE CONTEXT GRID

To determine what rights a user has at runtime, K2 blackpearl uses a matrix of users, roles, processes and activities, collectively called the Context Grid. This is an internal representation of what users can do at any given time in a process. The Context Grid is not a graphical representation in any blackpearl user interface (UI), but rather a concept that is used by blackpearl for rights and security. Many aspects of the Context Grid are exposed through the Management Console in the K2 Workspace, such as assigning a user the right to start a particular process.

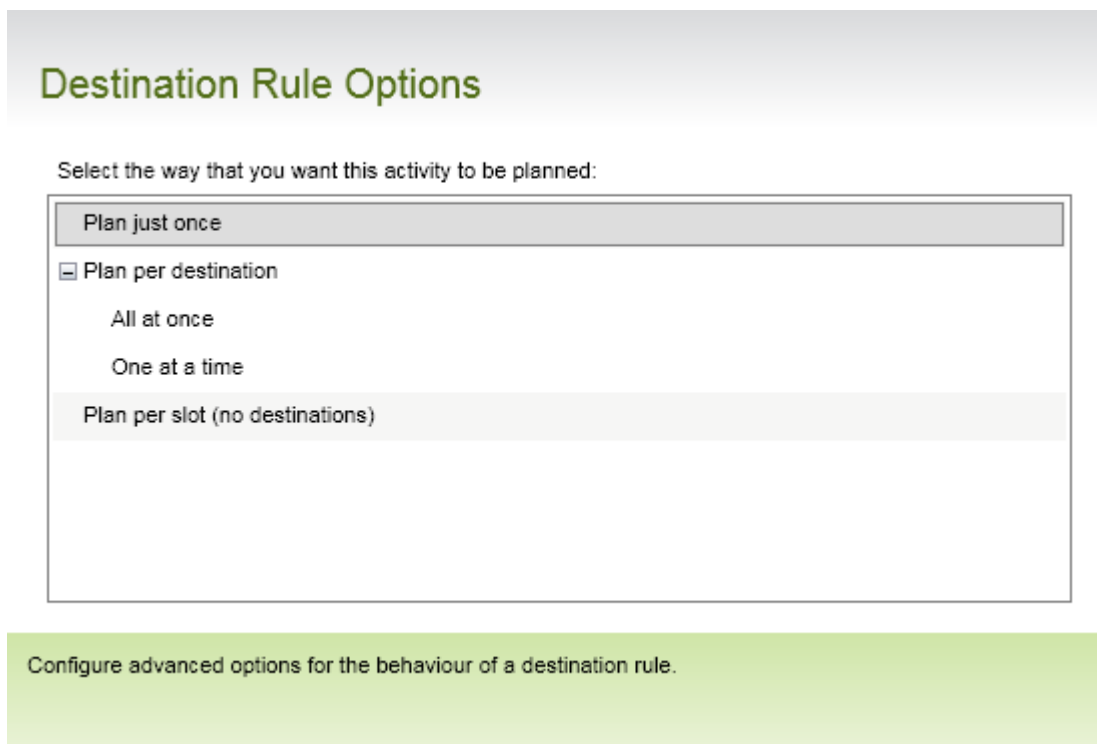
By using the Context Grid concept, user rights and the delegation, releasing and reassignment of tasks is always up to date and dynamically controlled by the server.

THE DIFFERENCE BETWEEN DELEGATION AND REDIRECTION OF TASKS

When a task is delegated or redirected (reassigned), the user or role to which the delegation or redirected occurs will receive the same rights as the person delegating or redirecting. The difference between delegation and redirection is that the task will display on both user's worklist if it is delegated, while redirection will remove the task from the user's worklist who performs the redirection. If the user is not available, the K2 administrator can redirect tasks using the **Worklists** page in Management Console.

ADVANCED DESTINATION RULES

The **Plan once** method of planning a destination rule was described as the default method of planning a blackpearl activity. Figure 4 illustrates the first page of the advanced mode **Destination Rule** wizard, containing the default and extra options for planning an activity.



The screenshot shows a web interface titled "Destination Rule Options". Below the title, it says "Select the way that you want this activity to be planned:". There is a list of options: "Plan just once" (highlighted), "Plan per destination" (with a checkbox), "All at once", "One at a time", and "Plan per slot (no destinations)". Below the list, there is a green bar with the text "Configure advanced options for the behaviour of a destination rule."

[FIGURE 4: ADVANCED DESTINATION RULE PLANNING.]

When using the default plan once option, only one activity instance is created. Events are executed once. Only one task item is created and rights assigned to the single task item. The activity instance will complete when all slots are completed or when the succeeding rule is true. The activity instance will expire when the succeeding rule is false. When using roles instead of individual users as the destination, task items are assigned to the role name rather than the individuals in the roles. This option and the number of slots can be configured on the second page of the advanced **Destination Rules** wizard as shown in Figure 5 below.

PLAN PER DESTINATION - ALL AT ONCE (PARALLEL PLANNING)

This option works in the same manner as it did in K2.net 2003. The server creates one activity instance per destination, and creates one task item per destination with the rights assigned to each destination for each task item.

The options on the following page, as illustrated in Figure 5 below, are as follows:

- > **Slot for each destination:** Required slots auto adjust depending on the number of destinations resolved when activity instance is executed at runtime
- > **Specify number of slots:** A fixed amount of slots are created. This overrides the activity setting.
- > **Resolve roles:** The role is resolved to its members during activity execution and an instance for each member of the role is created. All events execute once for each member of the role.
- > **Create slot for each role:** Roles are not resolved and a single activity instance is created. The role receives rights and all members of the role can access the task.
- > **Keep roles synchronized:** This option has no effect for parallel or serial plans and is discussed in the section *Synchronized vs. Cached Roles*.

Destination Rule Options

The option that you have selected might include destinations. How do you want to configure the slots?

- Create a slot for each destination.
- Specify the number of slots to be created.

How do you want to treat roles when they are used as destinations?

- Resolve all roles to users.
- Create a slot for each role.
- Keep roles synchronized.

Configure advanced options for the behaviour of a destination rule.

[FIGURE 5: PLANNING THE SLOTS AND ROLE RESOLUTION.]

PLAN PER DESTINATION - ONE AT A TIME (SERIAL PLANNING)

For the **One at a time** planning method, the server creates one activity instance per destination, but does so in a serial fashion, one instance at a time, and in the order the destination sets were designed. When a role is used as a destination, instances are created in the order in which the role membership is returned from the server query and there is no way to specify an order. If an order is required, do not use roles but rather individual users instead, and specify the users in the desired order.

The server creates one task item per destination at a time, and assigns rights to each task item. The next instance will be created only after the previous event is completed by the previous destination user. The server will not evaluate a succeeding rule on the activity until all activity instances are completed.



Important: For both parallel and serial planning methods, if more slots than destinations are specified, the activity cannot be completed without a custom succeeding rule, which requires code in the rule. To avoid this, ensure that there are always at least as many destinations as there are slots.

PLAN PER SLOT (NO DESTINATIONS)

The Plan per slot (no destination) method is reserved for server events based activities. As an example, this could be used for a parent process to start a variable number of child processes using IPC events. In this case, all configured destinations are ignored and the server creates one activity instance per slot. Activity data can be initialized and used for each slot. This allows each slot to have its own instance of the activity data fields.

In this method, the number of slots can be configured using activity or list data, which means that the number of slots is equal to the number of nodes in the XML field that is mapped to this field. When the runtime activity starts, the server decides how many instances of the IPC process, for example, need to be started.

Initialization data may contain any unique data that the IPC needs per child call. If a SmartObject GetList method or repeating XML node is specified for the initialization data, the server retrieves the next value for each slot, but only until it has reached the number of specified slots. Using the list field, the server creates a slot for each item returned while at the same time using the property specified as the initialization data for the new child process.

SCENARIO EXAMPLES

In the following section, ten different plans and options are used to illustrate the outcomes that can be achieved with K2 blackpearl plans and task assignments.

The following users and roles are used in the examples below:

- > Role1: Based on an Active Directory group called Team1, which contains three people.
- > Role2: Based on the Active Directory group called Team2, which contains three people.

PLAN JUST ONCE ACTIVITIES

The following three examples illustrate various combinations of single activities. Keep in mind that dynamic roles can only be used with **Plan just once** activities. If membership in a role changes frequently, the role should be marked **Dynamic** and the **Keep roles synchronized** option in the advanced Destination Rule wizard should be checked.

Example #1:

DESTINATION SET

ROLE1, BOB



DESTINATION SET	ROLE1, BOB
Create a slot for each destination	FALSE
Specified number of slots	2
Resolve all roles to users	FALSE
Create a slot for each role	TRUE
Slot(s) assigned to	SingleUser and Role1
Total number of slots	2
Outcome	This activity is actioned once by a single user and once by a member of Role1

Example #2:

DESTINATION SET	ROLE1, ROLE2
Create a slot for each destination	TRUE
Specified number of slots	N/A
Resolve all roles to users	FALSE
Create a slot for each role	TRUE
Slot(s) assigned to	Roles
Total number of slots	2
Outcome	Worklist items remain for members of a role when someone in that role has already actioned the task. If this is not desired, plan a parallel (All at once) activity with the "Create a slot for each destination" and "Create a slot for each role" options enabled.

Example #3:

DESTINATION SET	ROLE1, ROLE2
Create a slot for each destination	TRUE
Specified number of slots	N/A

Resolve all roles to users	TRUE
Create a slot for each role	FALSE
Slot(s) assigned to	Users
Total number of slots	6
Outcome	There will be as many slots created as there are users in the roles.

PLAN PER DESTINATION - ALL AT ONCE (PARALLEL) ACTIVITIES

Parallel planning enables all users to get notifications at the same time and, when the number of slots is filled, the availability of the task will disappear from the user's worklists who have not yet opened the task.

Example #4

DESTINATION SET	ROLE1
Create a slot for each destination	FALSE
Specified number of slots	1
Resolve all roles to users	TRUE
Create a slot for each role	FALSE
Slot(s) assigned to	Users
Total number of slots	1
Outcome	This works the same as Plan just once with the option Resolve all roles to users .

Example #5

DESTINATION SET	ROLE1, ROLE2
Create a slot for each destination	FALSE
Specified number of slots	1
Resolve all roles to users	FALSE
Create a slot for each role	TRUE



Slot(s) assigned to	Users
Total number of slots	1
Outcome	Once a person actions the task, all worklist items are removed.

Example #6

DESTINATION SET	ROLE1, ROLE2
Create a slot for each destination	TRUE
Specified number of slots	N/A
Resolve all roles to users	TRUE
Create a slot for each role	FALSE
Slot(s) assigned to	Users
Total number of slots	6
Outcome	The combination of Create a slot for each destination and Resolve all roles to users creates a slot for each user.

Example #7

DESTINATION SET	ROLE1, ROLE2
Create a slot for each destination	TRUE
Specified number of slots	N/A
Resolve all roles to users	FALSE
Create a slot for each role	TRUE
Slot(s) assigned to	Roles
Total number of slots	2
Outcome	This will create a slot for each role. Once a user from a role actions the task, the worklist item for the other role members is removed.



PLAN PER DESTINATION - ONE AT A TIME (SERIAL) ACTIVITIES

Serial activities are useful when multiple destination users need to review or make comments to an item in a sequential manner.

Example #8

DESTINATION SET	ROLE1, ROLE2
Create a slot for each destination	TRUE
Specified number of slots	N/A
Resolve all roles to users	TRUE
Create a slot for each role	FALSE
Slot(s) assigned to	Users
Total number of slots	6
Outcome	Each user is assigned a task after the previous user completes it. Worklist items appear for one user at a time. Changes to the role will not remove assigned tasks from a users worklist.

Example #9

DESTINATION SET	ROLE1, ROLE2
Create a slot for each destination	TRUE
Specified number of slots	N/A
Resolve all roles to users	FALSE
Create a slot for each role	TRUE
Slot(s) assigned to	Roles
Total number of slots	2
Outcome	Worklist items appear for each user in the first role. Once one of those users actions the task, the users from the second role see worklist items. When one person actions the task from the second role, the activity is completed.

Example #10

DESTINATION SET	TEAM1, TEAM2 (ACTIVE DIRECTORY GROUPS)
Create a slot for each destination	FALSE
Specified number of slots	2
Resolve all roles to users	FALSE
Create a slot for each role	TRUE
Slot(s) assigned to	Users
Total number of slots	2
Outcome	All users are assigned tasks. As soon as two tasks are actioned, the others disappear (does not matter which member of which team actioned it).

CONCLUSION

K2 blackpearl offers many powerful features to assign tasks to users and manage activities, including dynamic role resolution based on Active Directory, SQL, SmartObjects and custom user providers. By using these features of blackpearl, a wide range of scenarios for business process planning can be achieved.