Activities

Release 15.3.1.4

CONTACT Software

Juni 04, 2018

Inhaltsverzeichnis

KAPITEL 1

Einleitung

CONTACT Activities zeigt in einem fortlaufenden Aktivitätenstrom aktuelle Diskussionen und Systemmeldungen, die entweder für den persönlichen Arbeitskontext relevant sind oder sich an alle Anwender richten. Welche Informationen angezeigt werden, wird über abonnierte Kanäle gesteuert. Zudem ermöglicht *CONTACT Activities* das Teilen von Objekten im Team.

KAPITEL 2

Aktivitäten

2.1 Themen abonnieren

Ist Aktivitäten-Kanal

Jedes Objekt mit einer `cdb_object_id` kann ein Thema sein, dem Beiträge zugeordnet werden können. Alternativ zu *Thema* wird auch der Begriff *Kanal* verwendet. Im globalen Stream sehen Anwender Beiträge derzeit nur, wenn sie das entsprechende Thema abonniert haben. Ist diese Checkbox gesetzt, werden die Operationen `CDB_SubscribeToChannel` bzw. `CDB_UnsubscribeFromChannel` automatisch der Klasse zugeordnet bzw. entfernt.

2.2 Systembeitrag

Das System ist in der Lage automatisch bei Anlage, Änderung oder Statusänderung einen Systembeitrag zu erzeugen, die dann in *Aktivitäten* auftaucht. Systembeiträge werden dabei in allen aktiven Sprachen der Installation erzeugt (siehe multilanguage-activate).

Bemerkung: Aus Performancegründen werden Teile des Codes zum Erstellen von Systembeiträgen durch einen separaten Dienst durchgeführt. Die Aktivierierung dieses Dienstes mit dem Namen `System Posting Queue` ist im Installations- und Betriebshandbuch beschrieben. Damit Aufträge für die Queue erzeugt werden, muss der `cs.activitystream.activitylistener` aktiv sein. Dies ist z.B. bei reinen Powerscript-Anwendungen nicht automatisch der Fall.

Die Basiskonfiguration wird dabei in den Klasseneigenschaften vorgenommen (siehe datadict-class-properties).

Attribute, die alle Aktionen der Aktivitäten betreffen haben folgende Bedeutung:

2.2.1 Allgemeine Aktivitätenkonfiguration in der Klassendefinition

Systembeiträge erzeugen

Damit für Objekte einer Klasse überhaupt Systembeiträge erzeugt werden, muss diese Checkbox aktiviert werden. Ist sie aktiv, gilt die Einstellung auch für abgeleitete Klassen. Dies ist unabhängig davon, ob die Checkbox dort an- oder abgeschaltet ist. Ist sie deaktiviert haben alle anderen Konfigurationen zu den Aktivitäten keine Auswirkung.

Systembeiträge können nur für Klassen erzeugt werden, die das Attribut `cdb_object_id` besitzen. Das System überprüft beim Anlegen und Ändern von Klassen, ob dieses Attribut vorhanden ist. Ist dies nicht der Fall, wird es automatisch ergänzt. Die Klasse muss dann in der Regel noch manuell über die Funktion *Compile* erzeugt werden.

Regel

Hier kann eine Objektregel referenziert werden, welche die Menge der Objekte, zu denen ein Systembeitrag erzeugt werden soll, einschränkt. Beispielsweise könnten Sie festlegen, dass nur für Aufgaben einer bestimmten Kategorie Beiträge erzeugt werden sollen.

Ist eine Regel angegeben, muss dieses Regel in dieser Klasse und in allen abgeleiteten Klassen erfüllt sein, damit ein Beitrag erzeugt wird. Wird in einer abgeleiteten Klasse eine Regel definiert, muss diese zusätzlich zu allen in Basisklassen dieser Klasse definierten Regeln gelten.

Die weiteren Attribute sind in den folgenden Kapiteln beschrieben.

2.2.2 Generierung von Beiträgen bei der Anlage von Objekten

Sind die Grundvoraussetzungen für die Generierung von Systembeiträgen gegeben (siehe *Allgemeine Aktivitätenkonfiguration in der Klassendefinition* (Seite 2)), kann durch weitere Einstellungen festgelegt werden, ob bei der Anlage eines Objekts ein Beitrag generiert werden soll. Die Basiskonfiguration wird dabei in den Klasseneigenschaften vorgenommen (siehe datadict-class-properties).

Aktivitätenkonfiguration für Anlage eines neuen Objekts

Die relevanten Attribute haben folgende Bedeutung:

Systembeitrag bei Anlage

Ist diese Checkbox gesetzt, wird bei der Anlage eines neuen Objektes der Klasse ein Beitrag erzeugt. Ist sie aktiv, gilt die Einstellung auch für abgeleitete Klassen. Dies ist unabhängig davon, ob die Checkbox dort an- oder abgeschaltet ist.

Text

Hier kann über ein Label eine (Fehler-)Meldung referenziert werden. In dieser Meldung wird der Text des Systembeitrags vorgegeben. Die Auswertung erfolgt analog zur Konfiguration von Objektbeschreibung im Data Dictionary, d.h. es können feste Textbestandteile und Attributwerte konkateniert werden. Wird kein Text konfiguriert, wird die Meldung `activity_obj_created` verwendet, wobei die im Data Dictionary konfigurierte Objektbeschreibung als Replacement verwendet wird.

Kanal für Anlage-Beiträge

Es gibt den Fall, dass sich Anwender mittels der Aktivitäten über die Anlage von neuen Objekten informieren lassen wollen. Für Objekte, die keinem Thema zugeordnet sind, kann in diesem Fall ein Kanal angelegt werden, z.B. der Kanal *Neue Ideen* für das Ideen-Management (siehe *Zugriffsrechte für Beiträge* (Seite 4)). Auf diesen Kanal kann sich dann ein Anwender, der neue Ideen im globalen Aktivitätenfenster sehen will, abonnieren. Wird dieser Kanal in dem hier beschriebenen Feld referenziert, wird ein Anlage-Systembeitrag automatisch dem Kanal zugeordnet - unabhängig von der Implementierung der PowerScript- Methode `GetActivityStreamTopics`.

2.2.3 Generierung von Beiträgen bei Änderungen

Sind die Grundvoraussetzungen für die Generierung von Systembeiträgen gegeben (siehe *Allgemeine Aktivitätenkonfiguration in der Klassendefinition* (Seite 2)), kann durch weitere Einstellungen festgelegt werden, ob bei der Änderung von Objekten ein Beitrag generiert werden soll. Die Konfiguration wird dabei in der Registerkarte *Aktivitäten* der Attributeigenschaften im Data-Dictionary vorgenommen (siehe cdb-classdef-attributes).

Aktivitätenkonfiguration für Änderung

Die Attribute haben folgende Bedeutung:

Systembeitrag bei Änderung

Ist diese Checkbox gesetzt, wird bei der Änderung eines Objektes überprüft, ob dieses Attribut geändert wurde. Ist dies der Fall, wird ein Beitrag erzeugt.

Text

Hier kann über ein Label eine (Fehler-)Meldung referenziert werden. In dieser Meldung wird der Text des Systembeitrags vorgegeben. Die Auswertung erfolgt analog zur Konfiguration von Objektbeschreibung im Data Dictionary, d.h. es können feste Textbestandteile und Attributwerte konkateniert werden. Ist ein Text konfiguriert, wird ein separater Beitrag mit dem Text erzeugt, d.h. wenn mehrere Attribute geändert werden, für die jeweils ein Text konfiguriert ist, werden auch mehrere Beiträge erzeugt.

Wird kein Text konfiguriert, wird die Meldung `activity_obj_modified` verwendet. Dabei werden alle geänderten Attribute, deren Änderung einen Beitrag verursacht und für die keine spezielle Meldung konfiguriert ist, aufgesammelt. Als Replacements für die Meldung werden die Objektbeschreibung und die Liste der geänderten Attribute mit ihren Werten verwendet.

2.2.4 Generierung von Beiträgen bei Statusänderung

Sind die Grundvoraussetzungen für die Generierung von Systembeiträgen gegeben (siehe *Allgemeine Aktivitätenkonfiguration in der Klassendefinition* (Seite 2)), kann durch weitere Einstellungen festgelegt werden, ob im Rahmen einer Statusänderung ein Beitrag generiert werden soll. Die Einstellungen werden bei der Konfiguration der Statusdefinition vorgenommen.

Aktivitätenkonfiguration für Statusänderung

Die Attribute haben folgende Bedeutung:

Beitrag bei interaktiver Statusänderung generieren

Ist diese Checkbox gesetzt, wird bei einer interaktiven Statusänderung in diesen Status ein Beitrag generiert.

Beitrag bei automatischer Statusänderung generieren

Ist diese Checkbox gesetzt, wird bei einer automatischen Statusänderung in diesen Status ein Beitrag generiert. Dies sind die Statusänderungen, die das System z.B. durch den Aufruf von Statusänderungen innerhalb von User Exits durchführt.

Text

Hier kann über ein Label eine Meldung aus der (Fehler)-meldungsverwaltung referenziert werden. In dieser Meldung wird der Text des Systembeitrags vorgegeben. Die Auswertung erfolgt analog zur Konfiguration von Objektbeschreibungen im Data Dictionary, d.h. es können feste Textbestandteile und Attributwerte konkateniert werden.

Wird kein Text konfiguriert, wird die Meldung `activity_obj_wfstep` verwendet. Als Replacements für die Meldung werden die Objektbeschreibung und der Zielstatus verwendet.

2.2.5 Zuordnung der Themen

Ein Beitrag kann mehreren Themen zugeordnet sein - beispielsweise kann der Beitrag, dass eine neue Aufgabe angelegt wurde, der Aktivität des zugeordneten Projekts und der Aktivität der Aufgabe zugeordnet werden. Zum Zweck der Zuordnung wird die Methode `GetActivityStreamTopics` der PowerScript-Klasse `cdb.objects.Object` verwendet. Details entnehmen Sie bitte dem PowerScript-Handbuch.

2.2.6 Zugriffsrechte für Beiträge

Das read-Recht für Beiträge hängt im Standard am Recht `read` des Objeks, auf dass sich der Beitrag bezieht. Im Kern ist dies ähnlich wie ein Beziehungsrechteprofil abgebildet, d.h. bei einer Suche über `CDB_Search` wird das Recht nicht ausgewertet. An den Stellen, an denen Beiträge üblicherweise angezeigt werden, also in der Enterprise-Search oder in den Aktivitäten wird das Recht ausgewertet.

2.2.7 Automatisches Löschen von Beiträgen

Wird ein Objekt gelöscht, werden automatisch auch die Beiträge, die diesem Objekt zugeordnet sind, mitgelöscht.

2.3 Anzeige der Aktivitäten als Registerkarte

Sie können ein cdb-gui-maskconfig-dlgitemtypes-sect-cdbelinkcontrol einsetzen, um die Aktivitäten zu einem bestimmten Thema/Kanal anzeigen zu lassen. Das System besitzt bereits eine vordefinierte Maske, die nur dieses Control enthält. Diese Maske hat den Namen `cdb_elink_activitystream`. Sie kann als Registerkarte in einen Maskenverbund eingepflegt werden. Bei der Anlage neuer Objekte und beim Suchen bleibt die eLink-Anwendung inaktiv, daher ist es in der Regel sinnvoll, entweder unterschiedliche Maskenverbünde für die unterschiedlichen Aktionen zu definieren oder die Registerkarte mit Hilfe der PowerScript-Funktion `disable_registers()` auszublenden. Dies passiert beispielsweise in der Klasse `Channel`:

```
class Channel(Object):
    __maps_to__ = "cdbblog_channel"
    __classname__ = "cdbblog_channel"

    def _skip_activity_register(cls, ctx):
        if ctx.action != "info" and ctx.action != "modify":
            try:
                ctx.disable_registers(["cdb_elink_activitystream"])
            except:
                # Some context adaptors does not support disable_registers
                pass

    event_map = {("*", "pre_mask"): "_skip_activity_register"}
```

2.4 Kanäle

Damit ein Beitrag für den Anwender sichtbar wird, muss es ein Thema geben, dem der Beitrag zugeordnet ist. In vielen Fällen ist dieses Thema ein Objekt, z.B. ein Projekt. In manchen Fällen existiert aber kein geeignetes Objekt. Für diesen Fall können Sie einen Kanal anlegen, dem Beiträge zugeordnet werden. Ein solcher Kanal könnte beispielsweise der Kanal *Neue Ideen* sein. Wie in *Aktivitätenkonfiguration für Anlage eines neuen Objekts* (Seite 3) beschrieben, können Sie konfigurativ erreichen, dass die Systembeiträge zur Anlage neuer Objekte einem Kanal zugeordnet werden. Den Konfigurationszugang zu Kanälen erreichen sie im Menübaum über *Administration/Konfiguration → Aktivitäten → Kanäle*. Im Standard dürfen nur Administratoren neue Kanälen anlegen.

KAPITEL 3

Tägliche Zusammenfassung als E-Mail

Damit diese optional von jedem Benutzer einstellbare Funktion verwendet werden kann, muss der Dienst `cs.activitystream.daily_mail_service.DailyMailService` laufen. Dieser wiederum setzt voraus, dass ein Mailserver konfiguriert ist (üblicherweise in `$CADDOK_BASE/etc/site.conf`).

Der Dienst verschickt, wenn er aktiv ist, einmal täglich alle Benachrichtigungen. Im Auslieferungszustand passiert das jeweils um 0 Uhr eines Tages. Sie können die Startzeit ändern, indem Sie dem Dienst den Parameter "–start" mit einem Wert "20:15" (im Format "%H:%M") hinzufügen und anschließend neu starten. Bitte beachten Sie, dass Sie so eine Umstellung ggf. am Wochenende durchführen sollten, da der Dienst immer die innerhalb der letzten 24 Stunden geänderten Objekte als neu betrachtet. Es kann also bei einer Verschiebung der Startzeit zu vereinzelten Doppelmeldungen oder fehlenden Aktivitäten in der Zusammenfassung kommen.

Die Zusammenfassung enthält den vollständigen Text aller Beiträge und Kommentare des letzten Tages. Falls Sie stattdessen nur Textauszüge anzeigen möchten, kann die Vorlage für die E-Mails angepasst werden.

Die Vorlage finden Sie im Modulverzeichnis unter `chrome/activity_digest.html`. Wenn Sie diese anpassen möchten, empfiehlt es sich, die Vorlage in Ihr Kundenmodul zu kopieren und die Klasse `cs.activitystream.daily_mails.DailyMailer` so zu überschreiben, dass die neue Vorage verwendet wird. Dazu müssen Sie in Ihrer Klasse den neuen Pfad zur Vorlage angeben, z.B. so:

```
import os
from cdb import CADDOK
from cs.activitystream.daily_mails import DailyMailer

class CustomMailer(DailyMailer):
    __notification_template_folder__ = os.path.join(CADDOK.BASE, "email_templates")
    __notification_template__ = "activity_digest.html"
```

KAPITEL 4

Objekte teilen

4.1 Operationsablauf

Die Operation `Teilen (cdb_share_objects)` kann mit oder ohne Kontext aufgerufen werden. Im Wesentlichen ruft sie die URL des Dialogs zum Teilen auf, ggf. ergänzt um die cdb_object_ids der Kontextobjekte.

- URL ohne Kontext: `$CADDOK_WWWSERVICE_URL/share_objects`
- URL mit Kontext: `$CADDOK_WWWSERVICE_URL/share_objects?attachments=eb5af880-4be0-11e0-a01`

Im Dialog muss der Nutzer jeweils mindestens ein geteiltes Objekt und einen Empfänger auswählen. Da auch allgemeine Rollen und Empfängerlisten als Empfänger nutzbar sind, kann die Operation erst nach Abschluss sicher bestimmen, ob die Auswahl mindestens einen gültigen Empfänger enthält.

Sobald der Dialog erfolgreich abgeschlossen wird, wird im Hintergrund ein Objekt der Klasse `cdb_sharing (cs.sharing.Sharing)` angelegt. Dieses dient als Nachrichtenkanal, in dem sofort ein Beitrag mit dem eingegebenen Nachrichtentext und den geteilten Objekten als Anhang erstellt wird. Das Objekt selbst enthält ansonsten nur Daten der Änderungsverfolgung, wobei nur Anlagedatum und Ersteller (`cdb_cdate` respektive `cdb_cpersno`) relevant sind.

Nachrichten abonnierter Kanäle für das `cdb_sharing` Objekt sowie eventuelle E-Mail Benachrichtigungen (einstellbar in den persönlichen Einstellungen) werden asynchron durch eine Message Queue erzeugt. Wartende und fehlgeschlagene Aufträge können Sie unter *Administration/Konfiguration → Teilen → Teilen-Aufträge* einsehen und ggf. neu starten.

4.2 Empfängerlisten

Nutzer haben die Möglichkeit, sich die aktuelle Auswahl an Empfängern als eigene Empfängerliste zu speichern. Dazu vergeben sie einen Namen und es wird ein Objekt der Klasse `cdb_personal_sharing_group (cs.sharing.groups.PersonalSharingGroup)` und die Empfänger als Einträge der Klasse `cdb_sharing_group_member (cs.sharing.groups.SharingGroupMember)` angelegt.

Als zugeordnete Empfänger zulässig sind zur Zeit nur Benutzer (`angestellter`), allgemeine Rollen (`cdb_global_role`) und Empfängerlisten selbst, wobei diese Einschränkung bei der Anlage nicht geprüft wird.

Empfängerlisten können auch administrativ für bestimmte Anwenderkreise (z.B. “public”) eingerichtet werden. Ist eine Empfängerliste für eine Rolle eingerichtet, sehen alle Inhaber dieser Rolle die Empfängerliste und können sie verwenden. Änderungsrechte sind in diesem Fall aber Administratoren vorbehalten.

4.2.1 Objektspezifische Empfängerlisten

Geteilte Objekte können dynamische Empfängerlisten (Klasse `cdb_object_sharing_group (cs.sharing.groups.ObjectSharingGroup)`) besitzen, die der Nutzer als Empfänger auswählen

kann. Diese Listen sind vordefiniert und werden anhand einer Objektregel allen geteilten Objekten zugeordnet, die diese Regel erfüllen.

Aufgelöst werden diese Listen entweder anhand einer Objektmethode der Powerscriptklasse des geteilten Objekts oder eines ihrer Attribute. Im ersten Fall muss die Methode eine Liste von Tupeln aus `subject_id` und `subject_type` zurück liefern, im zweiten muss das Attribut genau eine Personalnummer (`angestellter.personalnummer`) enthalten.

4.3 Eingebettete Enterprise-Search

Der "Teilen" Dialog nutzt die Enterprise-Search für die Funktionalität des Feldes "Nach Objekten suchen". Bitte stellen Sie sicher, dass alle hierfür benötigten Dienste laufen und erreichbar sind.

4.4 Einrichtung für eigene Klassen

Damit Objekte einer Klasse geteilt werden können, muss die Klasse zunächst für die REST API freigeschaltet werden. Dazu muss in der Klassenkonfiguration die Option "REST API aktiv" an und ein eindeutiger "REST Name" vergeben sein.

4.4.1 Operation verfügbar machen

Um im Kontextmenü ihrer eigenen Klasse die Operation `Teilen` nutzbar zu machen, müssen Sie zunächst die Operation `cdb_share_objects` in der Operationskonfiguration (*Administration/Konfiguration* → *Administration* → *Operationen*) für die Klasse aktivieren. Anschließend müssen Sie in der Powerscriptklasse Code analog zu diesem Beispiel einbinden:

```
from cs.sharing.share_objects import WithSharing
from cs.documents import Document

class MyClass(Document, WithSharing):
    pass
```

4.4.2 Objektspezifische Empfängerlisten einrichten

Objektspezifische Empfängerlisten für die Klasse können wie folgt eingerichtet werden:

1. Anlage einer neuen Empfängerliste (oder Auswahl einer bestehenden) unter *Administration/Konfiguration* → *Teilen* → *Empfängerlisten*
2. Die Objektregel der Empfängerliste muss ein Prädikat mit mindestens einem Term für die Klasse enthalten. Nur Objekte, die diese Regel erfüllen, bieten die Empfängerliste im Dialog an.
3. Wenn die Empfängerliste nicht mit einem einfachen Attribut auskommt, in dem eine Personalnummer steht, muss eine Methode an der Powerscriptklasse eingerichtet werden, z.B. so:

```
from cdb.objects import Object

class myObjectsClass(Object):
    def getCustomRecipients(self, sharingGroup):
        "Empfängerliste auflösen/resolve recipient list"
        return [(self.cdb_cpersno, "Person"),
                (self.subject_id, self.subject_type),
                ("Administrator", "Common Role")]
```