

Personnalisation de l'application **Arcade** **GED** à l'aide de scripts Groovy

Digitech

Version 1.1, 19/05/2025 17:48:28

Sommaire

1. Introduction	7
2. Quel script pour quelle action ?	8
3. Données communes à chaque script	9
4. AttachmentDisplaySorting	10
4.1. Description	10
4.2. Paramètres	10
4.3. Sample	10
5. AttachmentInitializer	11
5.1. Description	11
5.2. Paramètres	11
5.3. Sample	11
6. AttachmentTableSorting	12
6.1. Description	12
6.2. Paramètres	12
6.3. Exemple	12
7. AttachmentUploaded	13
7.1. Description	13
7.2. Paramètres	13
7.3. Exemple	13
8. BasketInitializer	15
8.1. Description	15
8.2. Paramètres	15
8.3. Exemple	15
9. ConnectorPropertiesInitializer	17
9.1. Description	17
9.2. Paramètres	17
9.3. Sample	17
10. CustomHomeBlock	18
10.1. Description	18
10.2. Paramètres	18
10.3. Exemple	18
11. DisplayRule	19
11.1. Description	19
11.2. Paramètres	19
11.3. Exemple	19
12. DocCreation	21
12.1. Description	21
12.2. Paramètres	21

12.3. Exemple	21
13. DocumentActions	23
13.1. Description	23
13.2. Paramètres	23
13.3. Exemple	23
14. DocumentChecker	25
14.1. Description	25
14.2. Paramètres	25
14.3. Exemples	25
15. DocumentInitializer	26
15.1. Description	26
15.2. Paramètres	26
15.3. Exemple	26
16. ElementFilter	27
16.1. Description	27
16.2. Paramètres	27
16.3. Exemple	27
17. ExportInitializer	29
17.1. Description	29
17.2. Paramètres	29
17.3. Exemple	29
18. ExternalScript	30
18.1. Description	30
18.2. Exemple	30
19. FieldDependency	31
19.1. Description	31
19.2. Paramètres	31
19.3. Exemple	31
20. FieldInitializer	34
20.1. Description	34
20.2. Paramètres	34
20.3. Exemple	34
21. GlobalLibrary	36
21.1. Description	36
21.2. Exemple	36
22. GlobalSearch	37
22.1. Description	37
22.2. Paramètres	37
22.3. Exemple	37
23. FullTextSearch	39
23.1. Description	39

23.2. Paramètres	39
23.3. Exemples	39
24. HierarchicalTreeNodeInitializer	41
24.1. Description	41
24.2. Paramètres	41
24.3. Exemple	41
25. Task	43
25.1. Description	43
25.2. Paramètres	43
25.3. Exemple	43
26. WSAttachmentInitializer	44
26.1. Description	44
26.2. Paramètres	44
26.3. Exemple	44
27. WSDocumentInitializer	45
27.1. Description	45
27.2. Paramètres	45
27.3. Exemple	45
28. WSExtendDocumentInitializer	47
28.1. Description	47
28.2. Paramètres	47
29. Annexes	48
29.1. Groovy Bonnes pratiques, astuces,	48
29.1.1. Arcade GED/Groovy conventions.	48
29.1.2. Opérateur elvis	48
29.1.3. Opérateur de navigation "sécurisé"	48
29.1.4. Portée des variables	48
29.1.5. Substitution dans les chaines de caractères	50
29.2. Méthodes Arcade GED utiles	51
29.2.1. Répertoires de configuration	51
29.2.2. Initialisation/Finalisation de script Groovy	51
29.2.3. Cache	52
29.2.4. Lecture d'un fichier properties	52
29.2.5. Lecture d'un flux JSON	53
29.2.6. Pool de connections à une base de données	54
29.2.7. Récupération de données en base de données	54
29.2.8. Affichage d'un message en fin de traitement	55
29.2.9. Construction d'un recherche / récupération de documents.	55
29.2.10. document	56
29.2.11. Accès aux services communs d'Arcade GED	56
29.3. Quelques exemples	57

29.3.1. DisplayRule Rendre visible une action custom si une pièce jointe est disponible.	57
29.3.2. DocumentInitializer script sur déclenchement d'une action custom	57
29.3.3. Autres exemples/fichiers divers	57

**Digitech SA**Copyright (c) 2025 **Digitech SA**, tous droits réservés.*ZAC Saumaty Séon - 21 av. Fernand Sardou - CS 40173
13322 Marseille Cedex 16**Table 1. Notes de suivi*

Date	Version	État	Objet	Rédacteur	Valideur
02/08/2024	1.0	Validé	Création du document	MRA	NFE
14/02/2025	1.1	Validé	Ajout d'exemples	NFE	

1. Introduction

Ce document décrit les différents scripts [Groovy](#) permettant la personnalisation d'Arcade GED.

2. Quel script pour quelle action ?



Le nombre de types de scripts étant important, il peut paraître difficile de détecter quel type de script s'applique dans quel cas.
Ce chapitre permet d'identifier rapidement les cas d'utilisation.

Action souhaitée	Type de Script	Observation
------------------	----------------	-------------

3. Données communes à chaque script

L'ensemble des scripts décrit dans les chapitres suivants contiennent les 2 paramètres suivants :

Nom	Type	Description
Entrées		
_scriptLogger	org.log4j.Logger	Logger par défaut pour inscrire des informations dans les fichiers de logs .
_userContext	com.digitech.dossier.common.model.IUserCoreContext	Informations relatives à l'utilisateur exécutant le script.
Entrées/sorties		
_output	com.digitech.dossier.script.model.IScriptResultModel	Généralement non utilisé directement dans les scripts. La variable suivante _result semble convenir à la plupart des cas.
_result	com.digitech.dossier.script.model.IScriptResultValueModel	Structure de retour. Voir pour chaque script la liste des attributs disponibles. Les attributs communs sont : <ul style="list-style-type: none"> • status : de type com.digitech.common.script.model.EnumScriptStatus • messageSeverity : de type com.digitech.dossier.script.model.IScriptResultValueModel#Severity • messageSummary : message si nécessaire • messageDetail : message détaillé si nécessaire

4. AttachmentDisplaySorting

4.1. Description

Ce script permet de définir quelle pièce jointe est affichée *par défaut* en vue hiérarchique.

4.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_docId</code>	Integer	ID du document
<code>_attachmentList</code>	List<IAttachment>	Liste des pièces jointes
Entrées/sorties		
<code>_result</code>	<code>com.digitech.dossier.script.model.impl.result.ScriptResultValueAttachmentToDisplay</code>	contient les variables document et attachment correspondant au document et sa pièce jointe à afficher

4.3. Sample

Tri de la liste des pièces jointes par ordre alphabétique.

```
import com.digitech.dossier.common.model.backend.airs.impl.comparator.AttachmentComparator
import org.apache.commons.collections4.CollectionUtils

if(CollectionUtils.isNotEmpty(_attachmentList)) {
    _result.attachment = _attachmentList.get(0)
}
```

5. AttachmentInitializer

5.1. Description

à documenter

5.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
_docId	Integer	ID du document
_attachment	com.digitech.dossier.commun.model.backend.airs.IAttachment	Pièce jointe
Entrées/sorties		
_result	com.digitech.dossier.script.valid.model.impl.result.ScriptResultValueChecker	

5.3. Sample

6. AttachmentTableSorting

6.1. Description

Ce script est utilisé lors de la récupération des pièces jointes d'un document, afin de les trier avant de les afficher.

6.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
_docId	Integer	ID du document
_attachmentList	List<IAttachment>	Liste des pièces jointes
Entrées/sorties		
_result	com.digitech.dossier.script.model.impl.result.ScriptResultValueAttachmentToDisplay	contient les variables document et attachment correspondant au document et sa pièce jointe à afficher


























6.3. Exemple

Tri de la liste des pièces jointes par ordre alphabétique.

```
import com.digitech.dossier.common.model.backend.airs.impl.comparator.AttachmentComparator
import org.apache.commons.collections4.CollectionUtils

if(CollectionUtils.isNotEmpty(_attachmentList)) {
  _attachmentList.sort(AttachmentComparator.ATT_LABEL_COMP)
}
```

Exemple d'affichage d'une liste de pièces jointes triées par ordre alphabétique

Contenu (Document)		Pièce jointe	Lien	Réconciliation	Historique	Commentaire
<div> <div></div> <div></div> <div></div> </div>						
<input type="checkbox"/>	Id	Actions	Intitulé	Type	Nom du fichier	Versions
<input type="checkbox"/>	10869	    	 cell.xlsx	ORIGINAL	cell.xlsx	
<input type="checkbox"/>	10406	    	 document.docx	ORIGINAL	document.docx	
<input type="checkbox"/>	10868	    	 new.docx	ORIGINAL	new.docx	
<input type="checkbox"/>	10630	    	 pomme.docx	ORIGINAL	pomme.docx	Version courante [Billon Benjamin] 

7. AttachmentUploaded

7.1. Description

Ce type de script est utilisé lorsque l'on dépose ou supprime une pièce jointe.

7.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_document</code>	<code>com.digitech.dossier.common.model.backend.airs.IDocument</code>	Document
<code>_attachment</code>	<code>com.digitech.dossier.common.model.backend.airs.IAttachment</code>	Pièce jointe
<code>_mode</code>	<code>AttachmentUploadEventMode</code>	Mode (ajout/suppression)
Entrées/sorties		
<code>_result</code>	<code>com.digitech.dossier.script.model.impl.result.SimpleScriptResultModel</code>	--

7.3. Exemple

Remplissage du champ *Désignation* lors de la création d'un compte-rendu via le dépôt de pièce jointe depuis la page d'accueil.

```
import com.digitech.dossier.common.utils.FieldUtils
import com.digitech.dossier.script.model.impl.typedef.AttachmentUploadEventMode

des = FieldUtils.getValue(_document, "CR_DES")

if(_mode == AttachmentUploadEventMode.REMOVE) {
    if(_attachment?.fileName == des) {
        FieldUtils.setValue(_document, "CR_DES", null)
    }
}
else {
    if(des == null) {
        FieldUtils.setValue(_document, "CR_DES", _attachment?.fileName)
    }
}
```

}

Exemple de création de compte-rendu avec pièce jointe

Type de Contenu	Compte rendu
Thématique *	
Rédacteur *	Billon Benjamin
Désignation *	document.docx
Résumé	
Date événement *	20/06/2024

Ajout de pièces jointes

Intitulé

Déposer un nouveau fichier ici (ou cliquer pour en sélectionner un)

document.docx

8. BasketInitializer

8.1. Description

Ce type de script est utilisé pour initialiser les bannettes.

8.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
_basket	IBasketModel	Bannette
Entrées/sorties		
_result	com.digitech.dossier.script.model.impl.result.ScriptResultValueBasketInitializer	contient la variable properties qui contient les propriétés de la bannette

8.3. Exemple

Masquer les bannettes ne contenant aucun document.

```
import com.digitech.dossier.script.model.impl.result.ScriptResultValueBasketInitializer

if(_basket.getResultNumber(_userContext) == 0) {
    _result.getProperties().put(ScriptResultValueBasketInitializer.BasketProperty.DISPLAYED,
    "false")
}
```

Exemple de liste des bannettes (sans script)

Référentiel documentaire	▼
DOCUMENT(S) À TRAITER	 48
 Documents charte officielle	45
 Documents globaux Digitech	0
 Documents du service intégration	3

Exemple de liste des bannettes (avec script)

Référentiel documentaire		▼
DOCUMENT(S) À TRAITER		 48
	Documents charte officielle	45
	Documents du service intégration	3

9. ConnectorPropertiesInitializer

9.1. Description

à documenter

9.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
_document	com.digitech.dossier.common.model.backend.airs.IDocument	document
_attachment	com.digitech.dossier.common.model.backend.airs.IAttachment	Pièce jointe
_flowModel	String	Flow code
Entrées/sorties		
_result	com.digitech.dossier.script.model.impl.result.ScriptResultValueConnectorRule	annexeDocument : liste de IAttachment

9.3. Sample

10. CustomHomeBlock

10.1. Description

Ce script est utilisé pour l'affichage des blocs personnalisables de l'écran d'accueil

10.2. Paramètres

cf. [attributs communs](#).

<i>Nom</i>	<i>Type</i>	<i>Description</i>
Entrées		
_parameterMap	Map<String, Object>	Paramètres
Entrées/sorties		
_result	com.digitech.dossier.script.model.impl.result.CustomButtonScriptResultModel	contient les propriétés du bloc

10.3. Exemple

Personnalisation du style de l'entête avec un aspect de sous-titre ("h2").

```
_model.setHeaderStyleClass("h2")
```

11. DisplayRule

11.1. Description

Ce type de script est déclenché pour l'affichage des actions personnalisées.

11.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_document</code>	<code>com.digitech.dossier.common.model.backend.airs.IDocument</code>	Document
<code>_attachment</code>	<code>com.digitech.dossier.common.model.backend.airs.IAttachment</code>	Pièce jointe
Entrées/sorties		
<code>_result</code>	<code>com.digitech.dossier.script.model.impl.result.ScriptResultValueDisplayRule</code>	contient la variable valid qui détermine l'affichage de l'action

11.3. Exemple

Affichage de l'action "Diffuser le CR à Digitech !" si le document contient des pièces jointes et que le champ *Thématique* contient la valeur *Réunion transversale*.


```
import com.digitech.dossier.common.model.backend.airs.IDocument
import com.digitech.dossier.common.service.IAuthorityList
import com.digitech.dossier.common.service.ServiceManager
import com.digitech.jcorbairs.Term






Term updatedFieldCode = getAuthorityListService().getTerm((Integer) document.getField(
"CR_THEME").getValue())
_result.setValid(_document?.getAttachments(_userContext)?.size > 0 && updatedFieldCode?.
.getCode() == "TH_TRANS")

/**
 * @return IAuthorityList the Authority List
 */
public static IAuthorityList getAuthorityListService() {
    return (IAuthorityList) ServiceManager.getInstance().getService(com.digitech.dossier.
common.service.Constants.SERVICE_AIRS_AUTHORITYLIST_MGR
```

}

Exemple d'affichage de l'action



 Contenu (Compte rendu)	 Pièce jointe (1)	 Lien	 Réconciliation	 Historique
Identifiant	8972			
Désignation	test réunion tranverse			
Date événement	21/06/2024			
Résumé				
Thématique	Réunion Transverse			
Rédacteur	Billon Benjamin			

Exemple où l'action est masquée



 Contenu (Compte rendu)	 Pièce jointe (12)	 Lien	 Réconciliation	 Historique
Identifiant	8826			
Désignation	test			
Date événement	17/01/2024			
Résumé				
Thématique	Groupe de Travail			
Rédacteur	Adm Dossier			

12. DocCreation

12.1. Description

Ce script est déclenché lors de la création d'un document, afin de préremplir les données du formulaire.

12.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_document</code>	<code>com.digitech.dossier.common.model.backend.airs.IDocument</code>	Document
<code>_updateFields</code>	<code>Collection<UpdateField></code>	Champs du document
<code>_attachmentList</code>	<code>List<IAttachment></code>	Liste des pièces jointes
Entrées/sorties		
<code>_result</code>	<code>com.digitech.dossier.script.model.impl.result.ScriptResultDocCreationInitializer</code>	contient la variable <code>updatedFields</code> définissant les champs affichés

12.3. Exemple

Ajout de l'utilisateur courant en tant que rédacteur et de la date du jour lorsque l'on accède à la page de création d'un document.

```
import com.digitech.dossier.common.model.backend.airs.IDocument
import java.time.LocalDate
import java.time.format.DateTimeFormatter
import com.digitech.dossier.common.utils.FieldUtils
import com.digitech.dossier.common.model.backend.params.UpdateField

def today = LocalDate.now().format(DateTimeFormatter.ofPattern("dd/MM/yyyy"))
UpdateField uf = _updateFields.get('CR_DATE')
if(uf != null) {
    FieldUtils.setValue(_document as IDocument, 'CR_DATE', today)
}
uf = _updateFields.get('CR_REDACTEUR')
if(uf != null) {
    FieldUtils.setValue(_document as IDocument, 'CR_REDACTEUR', _userContext.getUserId())
}
```

Exemple de création d'un Compte rendu

Type de Contenu	Compte rendu
Thématique*	
Rédacteur*	Billon Benjamin
Désignation*	
Résumé	
Date événement*	14/06/2024

Ajout de pièces jointes

Intitulé

Déposer un nouveau fichier ici (ou cliquer pour en sélectionner un)

13. DocumentActions

13.1. Description

Ce script est utilisé lors de la récupération des actions pour un document.

13.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_document</code>	<code>com.digitech.dossier.common.model.backend.airs.IDocument</code>	Document courant
<code>_fieldValueMap</code>	<code>Map<String, IFieldValue></code>	Liste des champs
Entrées/sorties		
<code>_result</code>	<code>com.digitech.dossier.script.model.impl.result.ScriptResultValueDocumentActions</code>	contient les variables allActions (toutes les actions) et unavailableActions (actions indisponibles)

13.3. Exemple

Ajout de l'interdiction d'effectuer l'action **Verrouiller** pour un utilisateur n'ayant pas le profil `DOC_ADMIN_CLIENT`.

Exemple : l'utilisateur courant n'est pas rattaché au profil **DOS_ADMIN_CLIENT**

```
import com.digitech.dossier.common.utils.UserUtils
import com.digitech.dossier.script.model.impl.result.ScriptResultValueDocumentActions

if(!UserUtils.hasProfile(_userContext, "DOS_ADMIN_CLIENT")) {
    _result.addUnavailableAction(ScriptResultValueDocumentActions.Action.LOCK)
}
```

Exemple : l'utilisateur courant n'est pas le créateur du document

```
import com.digitech.dossier.script.model.impl.result.ScriptResultValueDocumentActions
import com.digitech.dossier.common.model.backend.Constants.AdvancedAuditType
import com.digitech.dossier.common.model.backend.airs.IDocument.IEvent

List<IEvent> events = _document.getEvents(AdvancedAuditType.ADV_EVENT_DOC_CREATE)

boolean canEdit = ((events?.size() ?: -1) > 0) && (events?.get(0)?.getUserId() ==
_userContext?.getUserId())
```

```
_scriptLogger.info("canEdit: {}, userID: {}", canEdit, _userContext?.getUserId())

if (!canEdit) {
    _result.addUnavailableAction(ScriptResultValueDocumentActions.Action.EDIT)
}
```





































Exemple : l'utilisateur courant n'est pas le REDACTEUR du document

```
import com.digitel.dossier.script.model.impl.result.ScriptResultValueDocumentActions

def uEditorID = _document.getFieldValue("CR_REDACTEUR")

if (_userContext?.getUserId() != uEditorID) {
    _result.addUnavailableAction(ScriptResultValueDocumentActions.Action.EDIT)
}
```

Exemple de résultats de recherche pour un utilisateur n'ayant pas le droit de verrouiller les documents

		Nom client	Code client	Type d'entité	Avenant RGPD	Numéro INES
<input type="checkbox"/>	Actions					
<input type="checkbox"/>	   	Venelles	VENELLES	Ville	Non	2151
<input type="checkbox"/>	    	CERGY-PONTOISE (CA)	CERGY-PONTOISE	Communauté d'agglomération / urbaine	Non	440
<input type="checkbox"/>	   	CGI_AUDE	CGI_AUDE	Conseil Départemental	Attente de réponse	
<input type="checkbox"/>	   		BAS_RHIN			
<input type="checkbox"/>	   	CIRIL HERAULT ENERGIE	CIRIL_HERAULT_ENERGIE	Syndicat Informatique	Non	
	<input type="checkbox"/>	 Saint Maur Des Fossés	SaintMaurDesFossés	Ville	Non	
	<input type="checkbox"/>		ville_annecy			
<input type="checkbox"/>	   	SDIS 64	SDIS_64	SDIS		
<input type="checkbox"/>	   		EVREUX			
<input type="checkbox"/>	    	Conseil Départemental de la Corrèze (CD19)	CD 19	Conseil Départemental		567

220 éléments



14. DocumentChecker

14.1. Description

Ce script est utilisé avant une recherche pour valider les critères.

14.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
_search	ISearchModel	Recherche
-parameters	Map<String, Object>	Paramètres
Entrées/sorties		
_result	com.digitech.dossier.script.model.impl.result.ScriptResultValueChecker	contient la variable valid qui détermine si la recherche est valide

14.3. Exemples

Blocage d'une recherche si une certaine condition est rencontrée.

```
boolean condition = checkIfSearchIsValid(_search)
result.setValid(condition)

private static boolean checkIfSearchIsValid(def sm) {
    //
    return ok
}
```

Ajout d'un critère de recherche

```
if(!userCanSeeArchiveFolder(_userContext)) {
    _search.setAirsRequest(_search.getAirsRequest(_userContext)) + " ET ARCHIVE=0"
}

private static boolean userCanSeeArchiveFolder(IUserCoreContext uc) {
    // whatever...
    return uc.getProfileCodeList().contains("ARCHIVE_RIGHT")
}
```

15. DocumentInitializer

15.1. Description

Ce type de script est déclenché lorsqu'une action personnalisée est lancée ou lors d'un export.

15.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_document</code>	<code>com.digitech.dossier.common.model.backend.airs.IDocument</code>	Document
<code>_customModel</code>	CustomActionModel	CustomActionModel rattachée au script
<code>_airsDocument</code>	IDocument	déprécié , utiliser la variable <code>_document</code> svp
Entrées/sorties		
<code>_result</code>	<code>com.digitech.dossier.script.model.impl.result.ScriptResultValueDocumentInitializer</code>	contient la variable documentList

15.3. Exemple

Affichage d'un message à la fin d'une action personnalisée.

```
import com.digitech.dossier.script.model.impl.result.ScriptResultValueDocumentInitializer
import com.digitech.dossier.script.model.IScriptResultValueModel.Severity

_result.setMessageSeverity(Severity.INFO)
_result.setMessageSummary("Action personnalisée sur le document '" + _document?.getAirsRefId() + "'")
```

Exemple de message



16. ElementFilter

16.1. Description

Ce type de script est utilisé pour filtrer les listes d'éléments.

16.2. Paramètres

cf. [attributs communs](#).

<i>Nom</i>	<i>Type</i>	<i>Description</i>
Entrées		
_document	com.digitech.dossier.common.model.backend.airs.IDocument	Document
_users	List<?>	Liste des utilisateurs
_elements	List<?>	Liste des éléments
Entrées/sorties		
_result	List<?>	Variable _elements qui correspond à la liste des éléments en sortie

16.3. Exemple

Filtrage sur une liste d'utilisateurs : affichage des personnes dont le nom de famille commence par un "D".


```
import com.digitech.dossier.common.utils.UserUtils
import com.digitech.jcorbairs.User




List<User> filteredList = _users.findAll({it.getLastName().startsWith("D")})
_elements.clear()
_elements.addAll(filteredList)
```

Exemple de filtre sur une liste d'utilisateurs

SÉLECTION D'UTILISATEUR(S) ✕

3 éléments Résultats par page **10** ▼



<input type="checkbox"/>	Nom ↕	Prénom ↕	Trigramme ↕	Organisation ↕
<input type="checkbox"/>	 DEBOUILLANNE	JULIEN		
<input type="checkbox"/>	 Delaye	Olivier	ODE	• Production / Intégration (PROD)
<input type="checkbox"/>	 DOLADILLE	Denis	DDO	• Production / Intégration (PROD)

3 éléments

ANNULER **SÉLECTIONNER**

17. ExportInitializer

17.1. Description

Ce script est déclenché lorsqu'un export est lancé.

17.2. Paramètres

cf. [attributs communs](#).

<i>Nom</i>	<i>Type</i>	<i>Description</i>
Entrées		
_documents	List<IDocument>	Liste de documents à exporter
Entrées/sorties		
_result	com.digitech.dossier.script. model.impl.result.ScriptRe sultValueExportInitializer	contient les variables fileResultPath et fileResultName

17.3. Exemple

Définition des variables **fileResultPath** et **fileResultName**.

```
_result.setFileResultPath("fileResultPath")  
_result.setFileResultName("fileResultName")
```

18. ExternalScript

18.1. Description

Ce type de script n'est pas directement géré par Dossier (bibliothèques externes).

18.2. Exemple

19. FieldDependency

19.1. Description

Ce script est déclenché lorsque l'on souhaite avoir une dépendance entre les valeurs de certains champs.

19.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_fieldToUpdate</code>	<code>IField</code>	Champ à mettre à jour
<code>_fieldMap</code>	<code>Map<String, IField></code>	Liste des champs
<code>_updatedField</code>	<code>IField</code>	Champ que l'on modifie
<code>_updateFieldValues</code>	<code>Object</code>	
<code>_parameterMap</code>	<code>List<IParameter></code>	
Entrées/sorties		
<code>_result</code>	<code>com.digitech.dossier.script.model.impl.result.IScriptResultModel<T></code>	

19.3. Exemple

Modification de la liste des choix pour le champ *Plan de classement* lorsque la valeur du champ *Catégorie de document* est modifiée.

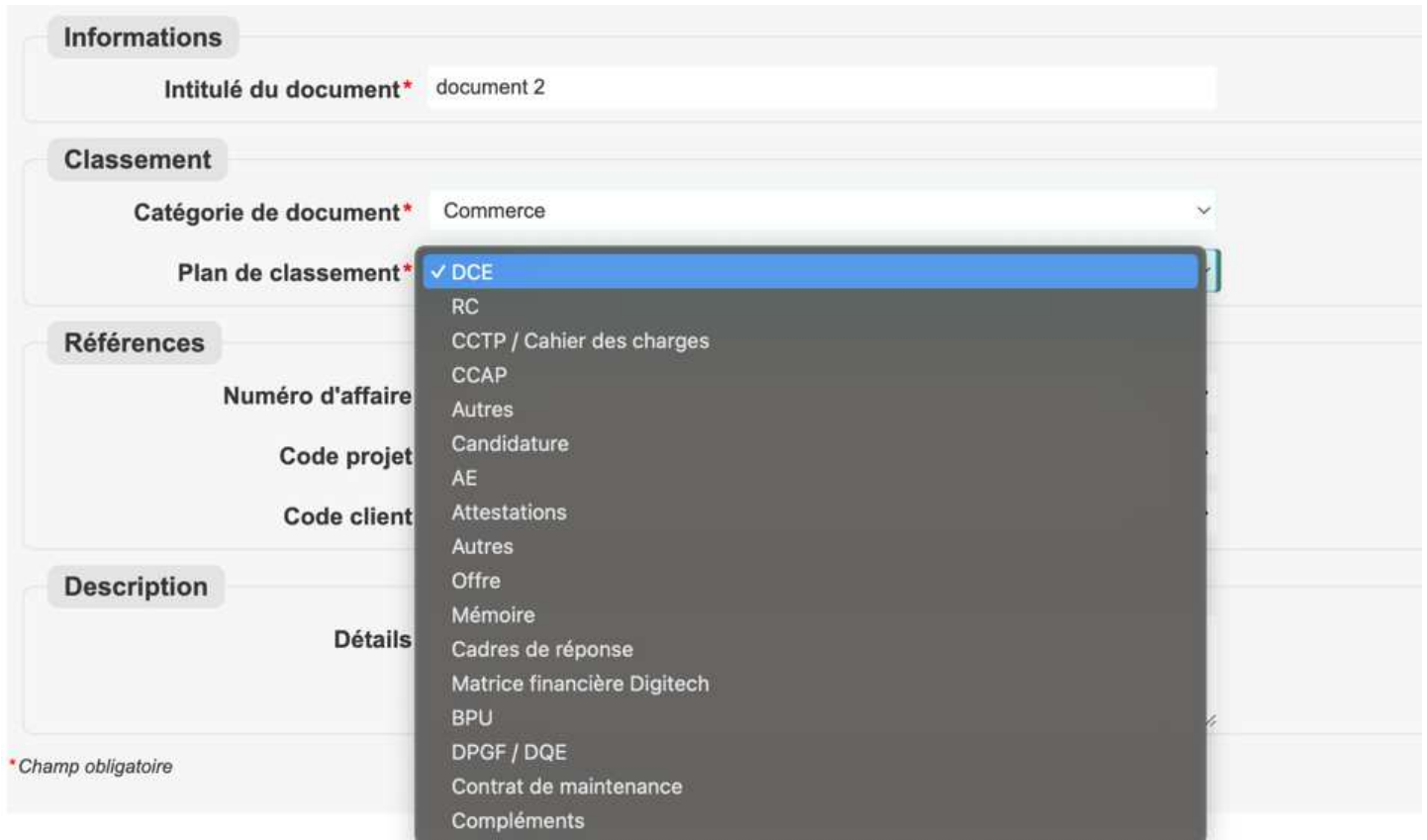
```
List<SelectItem> selectItems = new SelectItemFactory().getAuthorities(_fieldToUpdate
.getConfigField());
Term updatedFieldCode = getAuthorityListService().getTerm((Integer) _updatedField.getValue())

List<SelectItem> test = new ArrayList<>();
for(items in selectItems) {
    if(items.getValue() != null) {
        Term code = getAuthorityListService().getTerm((Integer) items.getValue());
        SelectItem item = new SelectItem()
        item.setLabel(code.getValues()[0] as String)
        item.setValue(code.getCode())
        test.add(item)
    }
}
List<SelectItem> filteredList = test.findAll({it.getValue().startsWith(updatedFieldCode
.getCode() as String)})
```

```
HtmlSelectOneMenu component = ((HtmlSelectOneMenu) ((UIComponent) _fieldToUpdate
.getComponent()));
component.getChildren().get(0).setValue(filteredList);

/**
 * @return IAuthorityList the Authority List
 */
public static IAuthorityList getAuthorityListService() {
    return (IAuthorityList) ServiceManager.getInstance().getService(com.digitech.dossier.
common.service.Constants.SERVICE_AIRS_AUTHORITYLIST_MGR);
}
```

Exemple de choix pour la catégorie Commerce



Informations

Intitulé du document* document 2

Classement

Catégorie de document* Commerce

Plan de classement* **✓ DCE**

- RC
- CCTP / Cahier des charges
- CCAP
- Autres
- Candidature
- AE
- Attestations
- Autres
- Offre
- Mémoire
- Cadres de réponse
- Matrice financière Digitech
- BPU
- DPGF / DQE
- Contrat de maintenance
- Compléments

Références

Numéro d'affaire

Code projet

Code client

Description

Détails

* Champ obligatoire

Exemple de choix pour la catégorie Comptabilité

Informations

Intitulé du document*

document 2

Classement

Catégorie de document*

Comptabilité

Plan de classement*

✓ Facture

Avoir

Commande

Références

Numéro d'affaire

Code projet

Code client

Description

Détails

* Champ obligatoire

20. FieldInitializer

20.1. Description

Ce script est déclenché lors de l'affichage des champs d'un document.

20.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_airsDocument</code>	<code>IDocument</code>	Document Airs
<code>_document</code>	<code>com.digitech.dossier.common.model.backend.air.IDocument</code>	Document
<code>parameters</code>	<code>Map<String, Object></code>	Paramètres
Entrées/sorties		
<code>_result</code>	<code>com.digitech.dossier.script.model.impl.result.ScriptResultValueFieldInitializer</code>	contient les variables results , properties et validatorAttributs

20.3. Exemple

Affichage des champs `CR_REDACTEUR` et `CR_DATE` en lecture seule.

```
import com.digitech.dossier.script.model.impl.result.ScriptResultValueFieldInitializer

if(_updateField.getFieldCode() == "CR_REDACTEUR" || _updateField.getFieldCode() == "CR_DATE")
{
    _result.getProperties().put(ScriptResultValueFieldInitializer.FieldProperty.READ_ONLY,
    "true")
}
```

Exemple de champs en lecture seule lors de la création d'un compte-rendu

Type de Contenu	Compte rendu
Thématique *	
Rédacteur *	Billon Benjamin
Désignation *	
Résumé	
Date événement *	18/06/2024

Ajout de pièces jointes

Intitulé	
	<div>Déposer un nouveau fichier ici (ou cliquer pour en sélectionner un)</div>

21. GlobalLibrary

21.1. Description

Ce type de script est utilisé pour définir des méthodes qui pourront être utilisées dans d'autres scripts.

21.2. Exemple

Nous avons le script de type GlobalLibrary **DocUtils.groovy** ci-dessous :

```
public static String getThirdFromList(List<String> list) {  
    if(list.size() >= 3) {  
        return list.get(2)  
    } else {  
        return null  
    }  
}
```

On pourra alors utiliser la méthode **getThirdFromList** dans un autre script de cette façon :

```
List<String> list = new ArrayList<>()  
list.add("a")  
list.add("b")  
list.add("c")  
list.add("d")  
String res = DocUtils.getThirdFromList(List<String> list)
```

22. GlobalSearch

22.1. Description

Ce script est déclenché lors de l'exécution d'une recherche dite "*rapide*".

22.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_search</code>	<code>ISearchModel</code>	recherche courante
<code>_searchPhrase</code>	<code>String</code>	Critères de recherche saisis
<div>  <p>Ne modifiez pas le paramètre <code>_searchPhrase</code>, préférez injecter - si votre action est celle-ci - la nouvelle requête comme il suit :</p> <pre><code>_search.setAirsRequest("<new criteria>")</code></pre> </div>		
Entrées/sorties		
<code>_result</code>	<code>com.digitech.dossier.script.model.impl.result.ScriptResultValueChecker</code>	contient la variable valid qui permet (optionnellement) d'annuler la requête

22.3. Exemple

Ajout d'un critère

```
import com.digitech.dossier.common.model.backend.Constants

if(!_searchCriteria.contains(Constants.FIELD_CODE_D_CREAT)) {
    _search.getAirsRequest() + " ET " + Constants.FIELD_CODE_D_CREAT + ">'01/01/2020'"
}
```

Invalidation de la requête

```
if(_searchCriteria.contains("données sensibles")) {
    _result.valid = false
}
```

```
}
```

23. FullTextSearch

23.1. Description

Ce script est utilisé avant une recherche fulltext pour ajouter ou modifier les filtres de recherche.

23.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_searchPhrase</code>	String	Recherche entrée par l'utilisateur
<code>_filter</code>	IFTFilter	Filtre
<code>_contentTypes</code>	List<String>	Liste des types de contenus sélectionnés pour la recherche
Entrées/sorties		
<code>_result</code>	<code>com.digitech.dossier.script.model.impl.result.ScriptResultFullTextSearch</code>	contient la variable valid qui détermine si la recherche est valide, ainsi que la variable searchPhrase qui peut être modifiée

23.3. Exemples

Blocage d'une recherche si une certaine condition est rencontrée.

```
if (_searchPhrase == "blabla") {
    _result.setValid(false)
    _result.fillKOResult("Ce mot est interdit!")
    return
}
```

Complétion d'une recherche saisie par l'utilisateur.

```
if(_searchPhrase == "digi") {
    _result.setSearchPhrase("digitech")
    _result.fillOKResult("Votre recherche a été modifiée!")
}
```

Ajout de critères de recherche

```
String contentType = "CR"

_filter.addFilterValue(contentType, "CR_RESUME", "test")
_filter.addFilterDate(contentType, "D_MODIF", "01/01/2022", ILocutionModel.Operator
```

```
.OPERATOR_VALUE_GREATER)  
_filter.addFilterValueFromList(contentType, "PRO_APP_OS", "WINDOWS")  
_filter.addFilterDates(contentType, "D_CREAT", "01/01/2019", "31/12/2019", ILocutionModel  
.Operator.OPERATOR_VALUE_BETWEEN)
```


24. HierarchicalTreeNodeInitializer

24.1. Description

Ce script est utilisé lors de l'affichage de l'arborescence dans la vue plan de classement.

24.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
_treeNode	IHierarchicalTreeNode	Arbre
Entrées/sorties		
_result	com.digitech.dossier.script.model.impl.result.ScriptResultValueHierarchicalTreeNodeInitializer	contient la variable properties qui correspondent aux propriétés de l'arbre

24.3. Exemple

Personnalisation des icônes des nœuds de l'arborescence et ajout du déploiement intégral de l'arbre.

```
import
com.digitech.dossier.script.model.impl.result.ScriptResultValueHierarchicalTreeNodeInitializer.TreeNodeProperty

_result.getProperties().put(TreeNodeProperty.EXPANDED, "true")
_result.getProperties().put(TreeNodeProperty.ICON_FOLDER, "/ui/common/img/dossier_logo.svg")
_result.getProperties().put(TreeNodeProperty.ICON_LEAF, "/ui/common/img/alert_fatal.svg")
```

Exemple d'affichage du plan de classement (sans script)



Exemple de personnalisation de l'affichage du plan de classement (avec script)

- ▼  **Venelles (INES 2151)**
 - ▼  Principale - Bascule CityWeb (2016/45831 - Mijoint Olivier - Hors marché)
 - ▼  Commerce
 - ▼  Offre
 - 🔥 Devis (25/09/2017) 
 - ▼  Mémoire
 - 🔥 Mémoire (25/09/2017) 
 - ▼  Comptabilité
 - ▼  Commande
 - 🔥 Bon de commande (25/09/2017) 
 - ▼  Service client
 - ▼  Bon d'intervention
 - 🔥 MAJ_4.2.8 CITYWEB (21/02/2018) 
 - ▼  Compte rendu d'installation
 - 🔥 CR_INSTALL_PROD (21/02/2018) 
 - 🔥 CR_INSTALL_TEST (21/02/2018) 
 - ▼  Documentations
 - 🔥 RAPPORT_MIGRATION (21/02/2018) 

25. Task

25.1. Description

Ce type de script est utilisé afin de décrire l'exécution d'une tâche.

25.2. Paramètres

cf. [attributs communs](#).

<i>Nom</i>	<i>Type</i>	<i>Description</i>
Entrées		
<code>_flowCodes</code>	List<String>	Flow codes
<code>_parameterMap</code>	Map<String, Object>	Paramètres de la tâche

25.3. Exemple

On peut passer un paramètre dans la configuration de la tâche (fichier **task.xml**) de cette façon :

```
<parameter name="param" type="STRING" value="machin"/>
```

Et ainsi récupérer ce paramètre dans le script afin de l'utiliser lorsque la tâche s'exécute :

```
import com.digitech.dossier.common.model.backend.air.impl.Parameter

Parameter param = _parameterMap?.get("param") as Parameter
String res = param?.getValue()
```

26. WSAttachmentInitializer

26.1. Description

Ce type de script est utilisé lors de l'exécution des webservices sur les pièces jointes (ajout ou mise à jour).

26.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_docId</code>	Integer	Doc ID
<code>_attData</code>	IWSAttachmentUpdateType	Attachment data
<code>_mode</code>	ScriptModeRun	Mode
Entrées/sorties		
<code>_result</code>	*com.digitel.dossier.script. contient les variables secretLevel , model.impl.result.ScriptResult codeFieldValues , ctxValues et document tWSDocument 2+	

26.3. Exemple

Arrêt du chargement d'une pièce jointe si l'attribut **label** n'a pas été renseigné dans la requête.

```
curl --location 'http://localhost:7373/dossierweb/rest/attachment/9097' \
--header 'Authorization: Basic <credentials>' \
--form 'attachmentData="{
  \"scriptCode\": \"test\"
}";type=application/json'
--form 'fileData=@"/path/document.docx"'
```

```
import com.digitel.dossier.script.model.impl.result.ScriptResultWSDocument
import com.digitel.dossier.webservices.type.WSAttachmentUpdateType

WSAttachmentUpdateType att = _attData
if(att.getLabel() == null) {
  _result.fillKOResult("Empty label")
}
```

27. WSDocumentInitializer

27.1. Description

Ce type de script est utilisé lors de l'exécution des webservices sur les documents (ajout ou mise à jour).



À la fois disponible pour le [Web Service WSDocument](#) (endpoint `/document`)

27.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_docType</code>	WSDocument: com.digitech.dossier.webse rvices.ws.document.type.W SDocumentType	
<code>_mode</code>	ScriptModeRun	Mode (CREATE ou UPDATE)
Entrées/sorties		
<code>_result</code>	com.digitech.dossier.script. model.impl.result.ScriptRe sultWSDocument	contient les variables secretLevel , codeFieldValues , ctxValues et document

27.3. Exemple

Ajout de commentaire dans le champ **CR_RESUME** si le document est mis à jour.

```
import com.digitech.dossier.common.model.backend.air.IDocument
import com.digitech.dossier.script.model.impl.result.ScriptModeRun
import com.digitech.dossier.script.model.impl.result.ScriptResultWSDocument
import java.text.SimpleDateFormat

def today = new Date()
def sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss")
if (_mode == ScriptModeRun.UPDATE) {
    IDocument doc = _document as IDocument
    String resume = doc.getField("CR_RESUME").getAirsValue()
    String newResume = resume + "\n" + "Mis à jour le " + sdf.format(today)
    _result.addFieldValue("CR_RESUME", newResume)
}
```

Exemple d'ajout de commentaire dans le champ **Résumé** lors de la mise à jour d'un document

[Compte Rendu] [Accueil](#) / [Vue Document](#)



Contenu (Compte rendu) Pièce jointe (0) Historique Commentaire

Identifiant 9099

Désignation Test

Date événement 16/07/2024

Résumé Mis à jour le 18/07/2024 16:24:32
Mis à jour le 18/07/2024 16:26:09

Thématique Réunion Transverse

Rédacteur Adm Adm

28. WSExtendDocumentInitializer

28.1. Description

Ce type de script est utilisé lors de l'exécution des webservices sur les documents (ajout ou mise à jour) en mode étendu.



À la fois disponible pour le [Web Service](#) (endpoint `/extendDocument`)

28.2. Paramètres

cf. [attributs communs](#).

Nom	Type	Description
Entrées		
<code>_docType</code>	<code>WSExtendDocument:</code> <code>com.digitech.dossier.webse</code> <code>rvices.extend.type.WSDocu</code> <code>mentType</code> ou	document structure
<code>_parentDocType</code>	<code>com.digitech.dossier.webse</code> <code>rvices.extend.type.WSDocu</code> <code>mentType</code>	parent document structure
<code>_mode</code>	<code>ScriptModeRun</code>	Mode
Entrées/sorties		
<code>_result</code>	<code>com.digitech.dossier.script.</code> <code>model.impl.result.ScriptRe</code> <code>sultWSDocument</code>	contient les variables <code>secretLevel</code> , <code>codeFieldValues</code> , <code>ctxValues</code> et <code>document</code>

29. Annexes

29.1. Groovy Bonnes pratiques, astuces, ...

29.1.1. Arcade GED/Groovy conventions.

Toutes les variables contextuelles transférées par l'application Arcade GED (et donc disponibles dans les scripts) sont préfixées du caractère '_'.

Le point virgule ';' en fin d'instruction n'est pas nécessaire.

29.1.2. Opérateur elvis

Le langage Groovy est (très) proche du langage Java. En tout cas, le compilateur Groovy interprète une syntaxe pure Java.

Cependant, le langage se trouve enrichi (*du point de vue de l'auteur de ce document [ok]*) par certaines caractéristiques, notamment l'opérateur **elvis**, un raccourci à l'opérateur ternaire Java

```
displayName = user.name ? user.name : 'Anonymous'
// devient
displayName = user.name ?: 'Anonymous'
```

29.1.3. Opérateur de navigation "sécurisé"

ou dans la langue de Shakespeare **Safe Navigation Operator**.

L'opérateur est identique à *elvis* et permet d'appel à des méthodes d'objet sans crainte de **NullPointerException**

```
// si person ou address ou home est NULL, roadNumber sera NULL
def roadNumber = person?.address?.home?.roadNumber
```



Les 2 opérateurs précédemment décrits peuvent être utilisés conjointement !

```
def roadNumber = person?.address?.home?.roadNumber ?: 666
```

29.1.4. Portée des variables

En Groovy, pour déclarer une variable

- Soit vous la déclarez directement, **sans préfixe/type**, votre variable - Quelle que soit sa position dans le fichier/la classe - sera alors **globale** (hors méthodes statiques !)


```
monText1 = "je suis une variable globale!"

def localFct() {
    monText2 = "je suis également une variable globale!"
}
```

- Soit vous la préfixez du type souhaité ou du mot clé **def**, celle-ci sera alors **locale** au contexte

```
String monText1 = "je suis une variable locale!"

def localFct() {
    def monText2 = "je suis également une variable locale!"
    println(monText1 ?: "inconnue") // print "inconnue"
}
```

exemple complet

```
logger = Logger.getLogger("scopeVariables")
y = 2

class GlobalProps {
    static def POOL_NAME = "dbName"
}

def localFunction() {
    def q = 333
    logger.info("y value: '{}'", y) // ok, print '2'
    logger.info("q value: '{}'", q) // ok, print '333'

    z = 666
}

static def staticFunction() {
    logger.info("dbName value: '{}'", GlobalProps.POOL_NAME) // ok, print 'dbName'
    logger.info("y value: '{}'", y) // this will fail!
}

localFunction()
staticFunction()

logger.info("- Local variable doesn't exist outside")
logger.info("y value: '{}'", y) // ok, print '2'
logger.info("7 value: '{}'", z) // ok, print '666'
logger.info("q value: '{}'", q) // this will fail!
```

29.1.5. Substitution dans les chaines de caractères

Comme indiqué dans la [documentation](https://docs.groovy-lang.org/latest/html/documentation/index.html#_string_interpolation), il est possible de formater facilement des messages/chaines de caractères avec des données contextuelles en utilisant le `_placeholder` `'${}'` ou `'$'`.

String interpolation

```
def who = "n.felix"
_scriptLogger.info("hello ${who}!")
_scriptLogger.info("hello $who!")

def whoPerson = new PersonData(1, "n.felix")

_scriptLogger.info("hello $whoPerson.login! (id: $whoPerson.id)")

class PersonData {
    Integer id
    String login
}
```

29.2. Méthodes **Arcade GED** utiles

Vous trouverez dans ce chapitre quelques exemples/méthodes récurrentes sur vos divers projets.

29.2.1. Répertoires de configuration

La récupération des chemins absolus vers les répertoires de configuration d'**Arcade GED** s'effectue comme il suit :

```
import com.digitel.dossier.common.utils.AppConfigurationUtilsCore

// répertoire *xml*
AppConfigurationUtilsCore.getXMLConfigurationFolderPath()
// répertoire *config*
AppConfigurationUtilsCore.getConfigFolderPath()
// répertoire *config/script*
AppConfigurationUtilsCore.getGroovyScriptFolderPath()
```

29.2.2. Initialisation/Finalisation de script **Groovy**

Chaque script **Groovy** peut automatiquement appeler une méthode :

- Au démarrage de l'application (ou à la recompilation du fichier). Cette méthode est nommée setup.

```
import com.digitel.dossier.script.utils.ScriptUtilities

/**
 * (optional) method called when script is loaded (or reloaded)
 * */
void setup() {
    // initialisation du pool de connexion (par exemple)
    def poolJdbc = dd//

    // et mise en cache (voir chapitre suivant)
    ScriptUtilities.addToCache("poolScript", poolJdbc)
}
```

- A l'arrêt de l'application (ou avant la recompilation du fichier). Cette méthode est nommée tearDown.

```
import com.digitel.dossier.script.utils.ScriptUtilities

/**
 * (optional) method called when script is unloaded (or before being reloaded)
 * */
void tearDown() {
    // libération du pool de connexion (par exemple)
    try {
```

```
def poolJdbc = ScriptUtilities.getFromCache("poolScript")
poolJdbc?.release()
}
finally{
    ScriptUtilities.removeFromCache("poolScript")
}
}
```

29.2.3. Cache

Il peut être intéressant de pouvoir conserver en mémoire des données pour les traiter, récupérer, ... entre plusieurs scripts ou entre plusieurs exécutions d'un même script.

Pour cela, les méthodes suivantes peuvent être utilisées :

```
import com.digitel.dossier.script.utils.ScriptUtilities

// ajout au cache
ScriptUtilities.addToCache("<key>", obj)

// récupération du cache
def value = ScriptUtilities.getFromCache("<key>")

// suppression du cache
ScriptUtilities.removeFromCache("<key>")
```

29.2.4. Lecture d'un fichier properties

Sur vos projets, peut-être aurez-vous besoin de lire un fichier *.properties* contenant des paramètres propres à votre environnement.

Sample properties

```
airs.db.class=<dbDriver>
airs.db.url=<dbUrl>
airs.db.user=<dbUser>
airs.db.password=<dbPwd>
```

```
import com.digitel.dossier.common.utils.AppConfigurationUtilsCore
import com.digitel.dossier.script.utils.ScriptUtilities

def propFile = Paths.get(AppConfigurationUtilsCore.getXMLConfigurationFolderPath() +
"<ws.properties>")

/**
 * Le 2ème paramètre ("wsProps") permet de mettre en cache le résultat.
 * Celui-ci sera recalculé seulement si le fichier est modifié (chargement à chaud)
```

```
*/
Properties conf = ScriptUtilities.getOrLoadProperties(propFile, "wsProps"/** <cacheKey>*/)

String driver = conf.getProperty("airs.db.class")
```

29.2.5. Lecture d'un flux JSON

Il est fréquent de devoir extraire des données d'un flux **JSON**, celui pouvant provenir d'un fichier, d'une **API REST**, ...

Groovy fournit pour cela un *parser* très simple à utiliser :

L'exemple ci-dessous utilise pour source de données le fichier **JSON** disponible en [annexe](#).

```
import groovy.json.JsonSlurper

JsonSlurper jsonParser = new JsonSlurper()

// weatherResponse a été récupéré précédemment, via un appel aux APIs *openweater*
def parsedJson = jsonParser.parseText(weatherResponse)

for(int i=0; i<parsedJson.list.size; ++i) {
    def dayData = parsedJson.list[i]
    OpenWeatherDay newDay = new OpenWeatherDay(i)
    newDay.desc = dayData.weather[0].description
    newDay.icon = dayData.weather[0].icon
    newDay.tempDay = dayData.main.temp
    newDay.tempMin = dayData.main.temp_min
    newDay.tempMax = dayData.main.temp_max
    newDay.windSpeed = dayData.wind.speed

    // ok, l'instance *newDay* peut être stockée pour un traitement ultérieur (affichage, ...)
}

/**
 * inner class permettant de stocker les données relatives à la météo d'une journée
 */
class OpenWeatherDay {
    final int dayIdx
    String desc
    String icon
    int tempMin
    int tempMax
    int tempDay
    int windSpeed

    OpenWeatherDay(int dayIdx) {
        this.dayIdx = dayIdx
    }
}
```

}

29.2.6. Pool de connections à une base de données

L'instanciation d'une connection à une base de données est une opération coûteuse, c'est pourquoi un *pool* de connections est fréquemment utilisé.

La classe **SqlUtils** permet de faciliter cette opération, comme il suit

```
import com.digitech.dossier.common.utils.sql.SqlUtils

def POOL_NAME = "dbPool"

// création ou récupération
def dbPool1 = SqlUtils.getOrCreatePool(POOL_NAME, "jdbc:oracle:thin:@//XXX", "<user>", "<pwd>", "oracle.jdbc.OracleDriver")

// récupération
def dbPool2 = SqlUtils.getPool(POOL_NAME)

// Libération
SqlUtils.releasePool(POOL_NAME)

// récupération d'une connexion à la base de données
try(def conn = SqlUtils.getPoolConnection(POOL_NAME)) {
    // vous pouvez alors travailler avec cette connexion à la base de données
}
```

29.2.7. Récupération de données en base de données

Opération fastidieuse et souvent redondante.

Utilisez donc la classe **SqlUtils** comme il suit :

```
import com.digitech.dossier.common.utils.sql.SqlUtils

def POOL_NAME = "dbPool"
List<Map<String, Object>> users = SqlUtils.runQuery(POOL_NAME, "select * from users")

// Il est possible de mapper les résultats directement dans un POJO
// soit l'attribut et le nom dans le pojo doivent correspondre
// soit l'alias et le nom dans le pojo doivent correspondre
List<PersonData> toto = SqlUtils.runQuery(POOL_NAME, "SELECT u.USR_ID, u.USR_LOGIN as \"login\" FROM users u", PersonData.class)

@groovy.transform.ToString
```

```
class PersonData {
    Integer usr_id
    String login
}
```

29.2.8. Affichage d'un message en fin de traitement

Les scripts qui sont exécutés manuellement (via une action utilisateur) peuvent afficher à l'écran un message de succès (ou échec) comme il suit.

```
import com.digitech.dossier.common.resources.BundleUtils

try {
    // opération ici
    // ...
    _result.fillOKResult("User '$_userContext.userFullName' successfully did the job!")

} catch (Exception e) {
    _scriptLogger.error("Error while blabla...: '{}'", e.getLocalizedMessage(), e)
    _result.fillKOResult("Error while blabla...", e)
}

// Si le message doit être récupérés à partir des ressources, pour une gestion multi-langues,
// le code suivant doit être utilisé

_result.fillOKResult(BundleUtils.getTranslation("msg_key_OK", new Object[]{ _userContext
.userFullName })))
```

29.2.9. Construction d'une recherche / récupération de documents.

Utilisation des classes **DocumentUtils** et **ISearch**

```
import com.digitech.dossier.common.model.backend.airs.LocutionModel
import com.digitech.dossier.common.model.backend.airs.impl.IDocument
import com.digitech.dossier.common.model.backend.airs.ILocutionModel.Operator
import com.digitech.dossier.common.service.ServiceUtils
import com.digitech.dossier.common.utils.DocumentUtils

// construction des locutions
def locutionModel = new LocutionModel()
DocumentUtils.buildLocutionModelWithDate(locutionModel, "D_ECCEANCE", Operator
.OPERATOR_VALUE_EQUAL, new Date())
DocumentUtils.buildLocutionModel(locutionModel, "FIELD_CODE_D_CLOTURE", Operator
.OPERATOR_ISNULL, null)

// lancement de la recherche et chargement direct des documents
```

```
List<IDocument> docs = DocumentUtils.search(_userContext, locutionModel, "COU_COURRIER_OUT")

// ou lancement simple pour récupération des docIDs
def searchModel = ServiceUtils.getSearchService().getSearchModel(_userContext, locutionModel,
"COU_COURRIER_OUT")
List<Integer> docIds = ServiceUtils.getSearchService().getSearch(_userContext, searchModel,
false)

for(Integer dId : docIds) {
    IDocument loadedDocument = ServiceUtils.getDocumentService().getDocument(_userContext, dId)
}
```

29.2.10. document

```
import com.digitech.dossier.common.model.backend.air.LocutionModel
import com.digitech.dossier.common.model.backend.air.impl.IDocument
import com.digitech.dossier.common.model.backend.air.ILocutionModel.Operator
import com.digitech.dossier.common.service.ServiceUtils
import com.digitech.dossier.common.utils.DocumentUtils

// IDocument _document

// mise à jour d'un index
_document.setFieldValue("FIELD_CODE_1", "nouvelle valeur 1")
_document.setFieldValue("FIELD_CODE_2", "nouvelle valeur 2")
// sauvegarde du document
_document.save()

// récupération d'un index
def summary = _document.getFieldValue("CR_RESUME")
```

29.2.11. Accès aux services communs d'**Arcade GED**

La classe **com.digitech.dossier.common.service.ServiceUtils** contient la liste des services principaux d'**Arcade GED**.

Méthode	Type	Description
getDocumentService()	com.digitech.dossier.common.service.IDocument	Service d'accès au document
getAuthorityListService()	com.digitech.dossier.common.service.IAuthorityList	Service d'accès aux listes d'autorité
getSearchService()	com.digitech.dossier.common.service.ISearch	Service de recherche

29.3. Quelques exemples

29.3.1. DisplayRule Rendre visible une *action custom* si une pièce jointe est disponible

```
/**
 * Script de type {DisplayRule}
 * Variables en entrée: _scriptLogger, _result, _userContext, _document
 */

_scriptLogger.debug(">>> sample_visibility (result: '{}', doc:'{}')", _result, _document)

_result.setValid(_document?.getAttachments(_userContext)?.size > 0)

_scriptLogger.debug("<<< sample_visibility")
```

29.3.2. DocumentInitializer script sur déclenchement d'une *action custom*

```
/**
 * Script de type {DocumentInitializer}
 * Variables disponible: _scriptLogger, _result, _userContext, _document
 */

_scriptLogger.debug(">>> sample_action (doc:'{}', ucc: '{}')", _document, _userContext)

try {
    // ajouter votre code ici...

    _result.fillOKResult("We did it!")
} catch (Exception e) {

    _scriptLogger.error("Error while ....: '{}'", e.getLocalizedMessage(), e)
    _result.fillKOResult("blabla error", e)
}

_scriptLogger.debug("<<< sample_action")
```

29.3.3. Autres exemples/fichiers divers

Sample weather

```
{
  "cod": "200",
  "message": 0,
  "cnt": 5,
  "list": [
```

```
{
  "dt": 1747386000,
  "main": {
    "temp": 19.75,
    "feels_like": 19.06,
    "temp_min": 18.9,
    "temp_max": 19.75,
    "pressure": 1011
  },
  "weather": [
    {
      "id": 500,
      "main": "Rain",
      "description": "légère pluie",
      "icon": "10d"
    }
  ],
  "wind": {
    "speed": 4.74,
    "deg": 181,
    "gust": 5.53
  },
  "visibility": 3047,
  "pop": 0.85,
  "rain": {
    "3h": 0.7
  }
},
{
  "dt": 1747396800,
  "main": {
    "temp": 19.34,
    "feels_like": 18.84,
    "temp_min": 18.53,
    "temp_max": 19.34,
    "pressure": 1012
  },
  "weather": [
    {
      "id": 500,
      "main": "Rain",
      "description": "légère pluie",
      "icon": "10d"
    }
  ],
  "wind": {
    "speed": 4.65,
    "deg": 241,
    "gust": 4.72
  },
  "visibility": 9707,
```

```

    "pop": 1,
    "rain": {
      "3h": 2.37
    }
  },
  "city": {
    "id": 2995469,
    "name": "Marseille",
    "coord": {
      "lat": 43.2965,
      "lon": 5.3698
    },
    "country": "FR",
    "population": 794811,
    "timezone": 7200,
    "sunrise": 1747368811,
    "sunset": 1747421788
  }
}

```