

Travail de diplôme 2008

Filière Informatique de gestion

SAP Interactive Forms by Adobe



Etudiant : David Jacquin

Professeur : Dr. Werner Maier

Avant propos

L'information est aujourd'hui une donnée qui représente un enjeu de taille pour toute organisation. Il est primordial de savoir l'obtenir, la transmettre ou encore l'analyser d'une manière à la fois rapide, fiable et bien évidemment peu coûteuse. Après tout, c'est ce qui établit un lien entre une entreprise, des partenaires et des collaborateurs.

L'un des principaux supports de l'information est le formulaire. Par "formulaire" on entend un document basé sur un modèle et qui est rempli avec des données qui peuvent dépendre de celui qui l'envoie, celui qui le reçoit ou encore de ce que l'un veut communiquer à l'autre. Bien que la manière de gérer ces formulaires change d'une entreprise à une autre, il existe toujours une contrainte à laquelle il faut faire face: la flexibilité. En effet, un formulaire ne doit jamais être dépassé et il doit pouvoir prendre en compte tous les changements qui s'opèrent dans l'entreprise à partir du moment où ça le concerne. Négliger cette flexibilité, c'est prendre le risque de diffuser des informations erronées qui peuvent par la suite déboucher sur des malentendus voire des décisions peu optimales et donc des coûts supplémentaires superficiels.

Il s'agit de l'une des problématiques à laquelle répondent les SAP Interactive Forms by Adobe, le résultat de la collaboration entre ces deux géants dans leurs domaines respectifs. C'est ainsi que ce travail de diplôme va présenter cette solution, l'évaluer et enfin montrer par le biais de réalisations concrètes certaines de ses possibilités.

Table des matières

1	Introduction	20
1.1	Sujet	20
1.2	Objectifs	20
1.2.1	Installer Adobe Interactive Forms	20
1.2.2	Etablir un lien entre ECC 6.0 et Adobe Acrobat	20
1.2.3	Prototype d'interaction entre Adobe et ECC 6.0.....	21
1.2.4	Deux formulaires pour l'Etat du Valais	21
1.2.5	Un formulaire pour Campus Management	21
1.2.6	Démarche	21
1.3	Planification.....	22
1.3.1	Découpage prévu	22
1.3.2	Découpage effectif	23
1.3.3	Comparaison.....	24
1.4	Participations.....	25
1.5	Matériel utilisé.....	26
2	SAP & Adobe	27
2.1	SAP	27
2.2	Présentation Adobe	28
2.3	Opinion.....	28
3	SAP Interactive Forms by Adobe	29
3.1	Présentation générale.....	29
3.2	Situation classique	29
3.2.1	Inconvénients du support papier	29
3.3	Avantages des SAP Interactive Forms by Adobe.....	30
3.4	Exemple	31
3.4.1	Le scénario.....	31
3.4.2	Scénario sans SAP Interactive Forms by SAP.....	31
3.4.3	Scénario avec SAP Interactive Forms by Adobe	32
3.4.4	Avantages	33

3.5 Alternatives possible	34
3.5.1 SAP Smart Forms	34
3.5.2 Microsoft InfoPath	42
3.5.3 Formulaires HTML	45
4 Fichiers PDF	46
4.1 Vision classique	46
4.2 Réalité, possibilités	46
5 Architecture & Installation.....	49
5.1 Présentation	49
5.2 Architecture utilisée	50
5.2.1 Serveur D07 (Système de développement):	50
5.2.1.1 SAP NetWeaver 7.0 (aussi appelé 2004s)	50
5.2.1.2 Adobe Document Services (SP13)	51
5.2.1.3 Adobe Credentials (Reader rights)	51
5.2.1.4 SAP ECC 6.0 (IDES)	51
5.2.2 Machine de développement & client.....	51
5.2.2.1 SAP NetWeaver Developer Studio 7.0.09 (développement)	51
5.2.2.2 Avec Adobe LiveCycle Designer 8.0 installé (développement)	52
5.2.2.3 Acrobat Reader 8.1.2 (client)	52
5.2.2.4 SAP Logon 7.10 (client + développement)	53
5.2.2.5 Internet Explorer 6+ (client + développement)	54
5.2.2.6 Active Component Framework (ACF) (client + développement)	54
5.2.3 Problèmes rencontrés liés à l'installation	55
5.2.3.1 D07	55
5.2.3.2 Client & Développement.....	56
6 Configuration.....	58
6.1 Remarque	58
6.2 Pré-requis	58
6.3 Configuration sur la D07	58
6.3.1 Ajout d'une ligne dans les propriétés de démarrage Java pour l'ADS	58
6.4 Vérifier le web service AdobeDocumentServices	61

6.4.1	Remarque	61
6.4.2	Test de l'AdobeDocumentServices.....	61
6.4.2.1	Définir ADSUSER en tant que Technical User, confirmation du mot de passe ..	64
6.5	Installer le Reader Rights Credential	68
6.5.1	Remarque	68
6.5.2	Présentation	68
6.5.3	Procédure d'installation	69
6.6	Problèmes possibles	71
6.6.1	Déploiement d'un formulaire: erreur "401: Unauthorized")	71
6.6.2	Visual Administrator: la fenêtre Web Services Security ne s'affiche pas.....	72
7	Développement	73
7.1	Introduction.....	73
7.2	Bases du développement.....	73
7.2.1	Utilisation de la perspective Webdynpro	73
7.2.2	Réglages de SAP NetWeaver Developer Studio	74
7.2.3	Interface de SAP NetWeaver Developer Studio	76
7.2.3.1	Web Dynpro Explorer (1).....	77
7.2.3.2	Outline (2).....	78
7.2.3.3	Fenêtre de travail (3)	78
7.2.3.4	Fenêtre de rapports (4)	79
7.2.4	Les nœuds de données, contextes	81
7.2.4.1	Nœuds de données	81
7.2.4.2	Contexte	82
7.2.5	Incorporation d'un formulaire interactif	85
7.2.6	Présentation d'Adobe LiveCycle Designer.....	89
7.2.7	Déploiement d'une application	95
7.3	Tutoriel 1: réalisation d'une application simple.....	100
7.3.1	Description	100
7.3.2	Création du projet	100
7.3.3	Création du composant	102
7.3.4	Création d'une nouvelle vue	104
7.3.4.1	Navigation	105

7.3.4.2	Actions.....	109
7.3.4.3	Design de StartView	111
7.3.4.4	Définition du contexte	116
7.3.4.5	Mappings.....	119
7.3.4.6	Design de la 2 ^{ème} vue	125
7.3.4.7	Design du formulaire.....	126
7.3.4.8	Calcul simple.....	132
7.3.4.9	Implémentation.....	134
7.3.4.10	Création de l'application	136
7.3.4.11	Déploiement, test, résultat	139
7.4	Les BAPIs/RFC.....	142
7.4.1	Définitions	142
7.4.2	Fonctionnement	142
7.4.3	Comment trouver une BAPI?	143
7.4.3.1	Via le code transaction BAPI	143
7.4.3.2	Via le code transaction SE37 (Function Builder Initial Screen)	143
7.4.3.3	Via le SDN	147
7.4.4	Implementation d'une BAPI	147
7.4.4.1	Pré-requis: utilisateur technique	147
7.4.5	Déploiement d'une application contenant au moins une BAPI.	148
7.4.5.1	Création d'une archive dans SAP NetWeaver Developer Studio:	148
7.4.5.2	Déploiement de l'archive vers le serveur J2EE:.....	149
7.4.5.3	Création des connexions JCO dans le Web Dynpro Content Administrator.	150
7.4.5.4	Exécution de l'application	150
7.5	Tutoriel 2: Importation et utilisation d'une BAPI.....	151
7.5.1	Description	151
7.5.2	Pré-requis	151
7.5.3	Création du projet et importation d'une BAPI	151
7.5.4	Mapping Custom Controller → modèle.....	159
7.5.5	Mapping SearchView → Custom Controller	162
7.5.6	Méthode du Custom Controller	166
7.5.6.1	Action de SearchView.....	166
7.5.7	Design du formulaire.....	167

7.5.8	Implémentation.....	168
7.5.8.1	wdDoInit() (Custom Controller).....	168
7.5.8.2	executeBapi_GetEmployeeData() (Custom Controller)	169
7.5.8.3	onActionSearch() (SearchView).....	169
7.5.9	Déploiement.....	170
7.5.9.1	Connexions JCO	171
7.5.9.2	Création et lancement de l'application	177
7.5.10	Test & résultat	178
7.6	Tutoriel 3: réalisation d'une application complète	179
7.6.1	Objectifs.....	179
7.6.2	Description	179
7.6.3	Création du projet	179
7.6.4	Importation des BAPIs.....	180
7.6.5	Création du Custom Controller, déclaration d'usage du modèle.....	181
7.6.6	Mapping.....	181
7.6.6.1	Mapping du Custom Controller → Modèle.....	182
7.6.7	Nœud, attribut personnalisé.....	184
7.6.7.1	Explication	184
7.6.7.2	Création d'un type simple	185
7.6.7.3	Création du nœud personnalisé.....	187
7.6.8	Mapping StartView → Custom Controller	190
7.6.9	Création des méthodes du Custom Controller.....	201
7.6.10	Implémentation.....	204
7.6.10.1	execute_Filter(int) (Custom Controller)	204
7.6.10.2	wdDoInit() (Custom Controller).....	205
7.6.10.3	execute_Search() (Custom Controller).....	207
7.6.10.4	execute_CrateAbsence() (Custom Controller)	207
7.6.11	Actions de StartView	208
7.6.12	Implémentation des actions de StartView	209
7.6.12.1	onActionSelecteType (StartView).....	209
7.6.12.2	onActioncreateAbsence (StartView)	210
7.6.12.3	onActionfliter (StartView)	210
7.6.12.4	onActionSearch (StartView)	210

7.6.13	Modification du nom du fichier .pdf	210
7.6.14	Design du formulaire interactif	212
7.6.14.1	Exemple de résultat	212
7.6.14.2	Liaisons des éléments de saisie	213
7.6.14.3	Partie supérieure	214
7.6.14.4	Partie centrale	218
7.6.14.5	Partie inférieure	221
7.6.14.6	Ordre de tabulation	222
7.6.15	Déploiement, test, résultat	223
7.6.16	Vérification	225
7.6.17	Commentaires	228
7.7	Tutoriel 4: Formulaire hors-ligne (Offline Form)	229
7.7.1	Objectif	229
7.7.2	Description	229
7.7.3	Pré-requis	229
7.7.3.1	Création d'un type simple	230
7.7.3.2	Création du Custom Controller et de son contexte	230
7.7.3.3	Implémentation de wdDoInit() (Custom Controller)	230
7.7.3.4	Mapping vue temporaire → Custom Controller	231
7.7.3.5	Création du formulaire	231
7.7.3.6	Exemple de formulaire	233
7.7.3.7	Template du formulaire	233
7.7.4	Importation des BAPIs	235
7.7.5	Usage du modèle	236
7.7.6	Mapping Custom Controller → Modèle	236
7.7.7	Création et implémentation des méthodes du Custom Controller	237
7.7.7.1	wdDoInit() du Custom Controller	237
7.7.8	execute_GetPersData() du Custom Controller	239
7.7.8.1	execute_SaveAbs(): du Custom Controller	240
7.7.9	Création des vues	241
7.7.10	Développement de la vue DownloadView	242
7.7.10.1	Design de DownloadView	244
7.7.11	Développement de la vue UploadView	245

7.7.11.1	Design de UploadView	247
7.7.12	Développement de la vue ResultView	247
7.7.12.1	Design de ResultView	248
7.7.13	Déploiement, test, résultat	248
7.7.14	Vérification	253
7.7.15	Commentaires	253
7.8	Pour aller plus loin	253
7.8.1	Standalone Forms.....	253
7.8.2	Intégration des formulaires dans les processus	254
7.8.3	Intégration dans Campus Management.....	254
7.9	Opinion sur l'implémentation	255
7.10	Problèmes rencontrés	256
8	Conclusion.....	257
8.1	Avis Personnel	257
8.2	Remerciements.....	257
9	Attestation.....	258
10	Sources	259
10.1	Documents.....	259
10.1.1	Informations	259
10.1.2	Installation, Configuration.....	259
10.1.3	Développement	260
10.1.4	Tutoriels.....	260
10.2	Sujets Forum SDN	261
10.3	Blogs SDN.....	262
10.4	SAP Notes	262
10.5	Sites divers.....	262

Table des illustrations

Figure 1 - Découpage du travail prévu	22
Figure 2 - Découpage effectif du travail.....	23
Figure 3 - Comparaison du découpage prévu/effectif	24
Figure 4 - Scénario sans Interactive Form: Demande de congé.....	32
Figure 5 - Scénario avec Interactive Form: Demande de congé	33
Figure 6 - Architecture Smart Forms	34
Figure 7 - SAP Smart Forms: Ecran initial	35
Figure 8 - SAP Form Builder: Affichage d'un formulaire	36
Figure 9 - Smart Forms: Migration vers Interactive Form	37
Figure 10 - Smart Forms: Export, destination	37
Figure 11 - Smart Forms: Export, destination	38
Figure 12 - Smart Forms: Paramètre d'exportation	38
Figure 13 - Smart Forms: Edition de l'objet, information	39
Figure 14 - Smart Forms: création d'une entrée pour l'objet	39
Figure 15 - Smart Forms - création d'une entrée pour l'objet	40
Figure 16 - Smart Forms: création de l'entrée, avertissement	40
Figure 17 - Smart Forms: création d'une Workbench request	40
Figure 18 - Smart Forms: création d'une Workbench request	41
Figure 19 - Smart Forms: Workbench request, validation	41
Figure 20 - Smart Forms: migration effectuée	42
Figure 21 - Microsoft InfoPath: affichage dans un navigateur.....	43
Figure 22 - Microsoft InfoPath: gestion des formulaires	44
Figure 23 - PDF: vision classique	46
Figure 24 - PDF: plus un conteneur qu'un document	46
Figure 25 - PDF: exemple de calcul	47
Figure 26 - PDF: définition en XML.....	48
Figure 27 - Architecture SAP WebAS.....	49
Figure 28 - SAP NetWeaver 7.0: structure	50
Figure 29 - SAP NetWeaver Developer Studio avec Adobe LiveCycle Designer lancé	52
Figure 30 - Adobe Reader: exemple.....	53
Figure 31 - SAP Logon 7.10.....	54
Figure 32 - Active Component Framework: message IE	54
Figure 33 - ACF: s'en passer en changeant le mode d'affichage	55
Figure 34 - Mode natif: éléments pour le mode natif.....	55
Figure 35 - Version des composants SAP-JEECOR, SAP-JEE, ADSSAP.....	56
Figure 36 - Configuration Tool: paramètres Java	59
Figure 37 - Configuration Tool: ligne ajoutée	59
Figure 38 - SAP Management Console: arrêt D07.....	60
Figure 39 - SAP Management Console: démarrage D07	60
Figure 40 - AdobeDocumentServices: Page d'accueil	61
Figure 41 - AdobeDocumentServices: test de l'opération rpData	62

Figure 42 - AdobeDocumentServices: saisie des paramètres pour le test.....	62
Figure 43 - AdobeDocumentServices: identification.....	63
Figure 44 - AdobeDocumentServices: résultats du test.....	64
Figure 45 - SAP J2EE Engine Start Page	65
Figure 46 - User Management: identification	65
Figure 47 - User Management: écran principal.....	66
Figure 48 - User Management: recherche, modification	67
Figure 49 - User Management: modification de ADSUSER	68
Figure 50 - Répertoire de destination du fichier .pfx	69
Figure 51 - Reader Rights Credential: Ecran de configuration	70
Figure 52 - Reader Rights Credential: saisie des paramètres.....	71
Figure 53 - Adobe Reader Right Credential: credential ajouté	71
Figure 54 - SAP NetWeaver Developer Studio: changement de perspective	73
Figure 55 - SAP NetWeaver Developer Studio: accès aux préférences.....	74
Figure 56 - SAP NetWeaver Developer Studio: paramètres J2EE Engine.....	75
Figure 57 - SAP NetWeaver Developer Studio: interface.....	76
Figure 58 - SAP NetWeaver Developer Studio: Web Dynpro Explorer	77
Figure 59 - SAP NetWeaver Developer Studio: outline	78
Figure 60 - SAP NetWeaver Developer Studio: fenêtre de travail	78
Figure 61 - SAP NetWeaver Developer Studio: fenêtre de travail (code)	79
Figure 62 - SAP NetWeaver Developer Studio: fenêtre de rapport (Tasks)	79
Figure 63 - SAP NetWeaver Developer Studio: fenêtre de rapport (Properties)	80
Figure 64 - SAP NetWeaver Developer Studio: fenêtre de rapports (J2EE Engine)	80
Figure 65 - SAP NetWeaver Developer Studio: fenêtre de rapport (Deploy Output View)	80
Figure 66 - Nœuds de données: cardinalités.....	81
Figure 67 - Contextes: structure.....	82
Figure 68 - Contexte: exemple	83
Figure 69 - Exemple de vue	83
Figure 70 - Liaison d'un champ à un attribut	84
Figure 71 - Liaison d'un champ.....	84
Figure 72 - Interactive Form: via drag & drop.....	85
Figure 73 - Interactive Form: insertion via Outline	86
Figure 74 - Liste des éléments insérables.....	86
Figure 75 - Interactive Form: affichage dans la fenêtre de design.....	87
Figure 76 - Contexte	87
Figure 77 - Context: création d'un attribut	87
Figure 78 - Contexte: nommage de l'attribut.....	88
Figure 79 - Attribut "pdfSource": type binaire	88
Figure 80 - Interactive Form: lancement d'Adobe LiveCycle Designer	89
Figure 81 - Adobe LiveCycle Designer dans SAP NetWeaver Developer Studio	90
Figure 82 - Interactive Form: onglet dans SAP NetWeaver Developer Studio.....	90
Figure 83 - Adobe LiveCycle Designer: interface principale	91
Figure 84 - Interactive Form: événements pour les scripts.....	92

Figure 85 - Interactive Form: langages des scripts.....	93
Figure 86 - Interactive Form: importation	93
Figure 87 - Sélection du .pdf à importer	93
Figure 88 - Interactive Form: options d'importation	94
Figure 89 - Interactive Form: exemple de formulaire importé	95
Figure 90 - SAP NetWeaver Developer Studio: déploiement	96
Figure 91 - SAP NetWeaver Developer Studio: mot de passe pour le déploiement.....	96
Figure 92 - SAP NetWeaver Developer Studio: exemple d'une application déployée	97
Figure 93 - SAP J2EE Start Page.....	97
Figure 94 - Outils Web Dynpro.....	98
Figure 95 - Identification pour le Web Dynpro Content Administrator.....	98
Figure 96 - Web Dynpro Content Administrator: écran principal.....	99
Figure 97 - Web Dynpro Content Administrator: liste des applications déployées.....	99
Figure 98 - Tutoriel 1: création d'un nouveau projet.....	100
Figure 99 - Tutoriel 1: nommage du nouveau projet.....	101
Figure 100 - Tutoriel 1: Structure du projet.....	101
Figure 101 - Tutoriel 1: Création d'un composant	102
Figure 102 - Tutoriel 1: Nommage du composant, package, vue par défaut.....	103
Figure 103 - - Tutoriel 1: structure du projet	103
Figure 104 - Tutoriel 1: création d'une vue.....	104
Figure 105 - Tutoriel 1: nommage d'une nouvelle vue	104
Figure 106 - Tutoriel 1: nouvelle vue	104
Figure 107 - Tutoriel 1: interface de travail	105
Figure 108 - Tutoriel 1: ajout d'une vue au diagramme.....	106
Figure 109 - Tutoriel 1: "Result View" dans le diagramme de navigation	106
Figure 110 - - Tutoriel 1: nommage du point de sortie.....	107
Figure 111 - Tutoriel 1: point de sortie sur le diagramme de navigation	107
Figure 112 - Tutoriel 1: points d'entrée/sortie	108
Figure 113 - Tutoriel 1: liaison sortie → entrée.....	108
Figure 114 - Tutoriel 1: liaison sortie → entrée.....	109
Figure 115 - Tutoriel 1: StartView dans Web Dynpro Explorer.....	109
Figure 116 - Tutoriel 1: onglet "Action"	110
Figure 117 - Tutoriel 1: création de l'action toPDF	110
Figure 118 - Tutoriel 1: création de l'action "back"	111
Figure 119 - Tutoriel 1: onglet "Layout".....	112
Figure 120 - Tutoriel 1: éléments ajoutés dans la fenêtre d'	112
Figure 121 - Tutoriel 1: design de StartView.....	113
Figure 122 - Tutoriel 1: design de StartView.....	113
Figure 123 - Tutoriel 1: propriétés de DefaultTextView	114
Figure 124 - Tutoriel 1: design de StartView.....	114
Figure 125 - Tutoriel 1: réorganisation des éléments dans la fenêtre d'	114
Figure 126 - Tutoriel 1: design de StartView.....	115
Figure 127 - Tutoriel 1: design de StartView.....	115

Figure 128 - Tutoriel 1: design de StartView	115
Figure 129 - Tutoriel 1: Component Controller	116
Figure 130 - Tutoriel 1: Onglet Context	116
Figure 131 - Tutoriel 1: Création d'un nœud de valeurs	117
Figure 132 - Tutoriel 1: nommage du nœud	117
Figure 133 - Tutoriel 1: création d'un attribut	118
Figure 134 - Tutoriel 1: Context du component	118
Figure 135 - Tutoriel 1: Contexte du component	119
Figure 136 - Tutoriel 1: Ouverture du Data Modeler	119
Figure 137 - Tutoriel 1: Data modeler, schéma de l'application	120
Figure 138 - Tutoriel 1: étape pré-mapping	120
Figure 139 - Tutoriel 1: choix des nœuds/attributs à mapper	121
Figure 140 - Tutoriel 1: cochages des nœuds/attributs à mapper	121
Figure 141 - Tutoriel 1: mapping entre la vue et le component	122
Figure 142 - Tutoriel 1: Data Modeler	122
Figure 143 - Tutoriel 1: mapping entre la vue et le component	123
Figure 144 - Tutoriel 1: Data Modeler	123
Figure 145 - Tutoriel 1: vue StartView	124
Figure 146 - Tutoriel 1: sélection de l'attribut à lier au champ	124
Figure 147 - Tutoriel 1: vue StartView	124
Figure 148 - Tutoriel 1: valeur d'un champ (invalide)	125
Figure 149 - Tutoriel 1: valeur d'un champ (valide)	125
Figure 150 - Tutoriel 1: vue ResultView	125
Figure 151 - Tutoriel 1: vue ResultView	125
Figure 152 - Tutoriel 1: vue ResultView	126
Figure 153 - Tutoriel 1: lancement d'Adobe LiveCycle Designer	127
Figure 154 - Tutoriel 1: onglet du formulaire interactif	127
Figure 155 - Tutoriel 1: Adobe LiveCycle Designer	128
Figure 156 - Tutoriel 1: élément Text	128
Figure 157 - Tutoriel 1: modification du titre	129
Figure 158 - Tutoriel 1: dépassement d'un champ	129
Figure 159 - Tutoriel 1: Data View	129
Figure 160 - Tutoriel 1: champs de saisie	130
Figure 161 - Tutoriel 1: attributs liés dans le formulaire	130
Figure 162 - Tutoriel 1: bouton (ActiveX)	131
Figure 163 - Tutoriel 1: ajout du bouton dans le formulaire	131
Figure 164 - Tutoriel 1: code du bouton "Submit"	131
Figure 165 - Tutoriel 1: événement onSubmit du formulaire dans SAP NetWeaver Developer Studio	132
Figure 166 - Tutoriel 1: champ numérique	132
Figure 167 - Tutoriel 1: champs de saisie	132
Figure 168 - Tutoriel 1: choix de l'évènement "calculate"	133
Figure 169 - Tutoriel 1: choix du langage	133

Figure 170 - Tutoriel 1: nom du champ numérique	133
Figure 171 - Tutoriel 1: vérification du calcul.....	134
Figure 172 - Tutoriel 1: Component Controller.....	134
Figure 173 - Tutoriel 1: onglet Implementation.....	134
Figure 174 - Tutoriel 1: méthode wdDoInit() dans l'.....	135
Figure 175 - Tutoriel 1: nœud vide: conséquence	135
Figure 176 - Tutoriel 1: création de l'application	136
Figure 177 - Tutoriel 1: nommage de l'application	137
Figure 178 - Tutoriel 1: choix du component.....	137
Figure 179 - Tutoriel 1: sélection du component.....	138
Figure 180 - Tutoriel 1: Application dans Web Dynpro Explorer	138
Figure 181 - Tutoriel 1: déploiement de l'application.....	139
Figure 182 - Tutoriel 1: affichage de l'application dans Internet Explorer	139
Figure 183 - Tutoriel 1: saisie de données	140
Figure 184 - Tutoriel 1: pré-remplissage dans le formulaire.....	140
Figure 185 - Tutoriel 1: modification des données du formulaire	141
Figure 186 - Tutoriel 1: affichage des données modifiées	141
Figure 187 - Tutoriel 1: BAPI	142
Figure 188 - BAPI: BAPI Explorer	143
Figure 189 - BAPI: Function Builder	144
Figure 190 - BAPI: exemple de résultats de recherche d'une BAPI.....	145
Figure 191 - BAPI: Function Builder	145
Figure 192 - BAPI: saisie d'un paramètre pour test	146
Figure 193 - BAPI: résultat du test de la BAPI	146
Figure 194 - BAPI: informations retournées	146
Figure 195 - Utilisateur technique: détails de connexion	147
Figure 196 - Utilisateur technique: profile	148
Figure 197 - Création d'un archive	149
Figure 198 - Déploiement d'une archive.....	149
Figure 199 - Connexions JCO.....	150
Figure 200 - Lancement de l'application	150
Figure 201 - Tutoriel 2: Création d'un modèle	151
Figure 202 - Tutoriel 2: Types de modèles.....	152
Figure 203 - Tutoriel 2: nommages pour le modèle.....	152
Figure 204 - Tutoriel 2: connexion à SAP	153
Figure 205 - Tutoriel 2: liste des BAPIs.....	154
Figure 206 - Tutoriel 2: sélection de la BAPI désirée	154
Figure 207 - Tutoriel 2: journal d'importation du modèle	155
Figure 208 - Tutoriel 2: modèle dans le Web Dynpro Explorer.....	155
Figure 209 - Tutoriel 2: vue SearchView	156
Figure 210 - Tutoriel 2: Création d'un Custom Controller.....	156
Figure 211 - Tutoriel 2: nommage du Custom Controller	157
Figure 212 - Tutoriel 2: Custom Controller dans Web Dynpro Explorer	157

Figure 213 - Tutoriel 2: déclaration d'utilisation d'un modèle.....	158
Figure 214 - Tutoriel 2: modèles disponibles	158
Figure 215 - Tutoriel 2: modèle utilisé	158
Figure 216 - Tutoriel 2: Lancement du Data Modeler	159
Figure 217 - Tutoriel 2: schéma de l'application	159
Figure 218 - Tutoriel 2: mapping entre le Custom Controller et le modèle	160
Figure 219 - Tutoriel 2: choix des nœuds/attributs à mapper	160
Figure 220 - Tutoriel 2: éléments cochés pour le mapping.....	161
Figure 221 - Tutoriel 2: mapping entre le Custom Controller et la vue	161
Figure 222 - Tutoriel 2: liaison Custom Controller → modèle	161
Figure 223 - Tutoriel 2: Contexte du Custom Controller	162
Figure 224 - Tutoriel 2: attribut pdfSource	162
Figure 225 - Tutoriel 2: mapping entre la vue et le Custom Controller	163
Figure 226 - Tutoriel 2: sélection du mapping	163
Figure 227 - Tutoriel 2: mapping de l'attribut paramètre.....	164
Figure 228 - Tutoriel 2: choix des attributs retournés par la BAPI.....	164
Figure 229 - Tutoriel 2: mapping entre la vue et le Custom Controller	165
Figure 230 - Tutoriel 2: mapping entre la vue et le Custom Controller	165
Figure 231 - Tutoriel 2: liaison entre la vue et le Custom Controller.....	166
Figure 232 - Tutoriel 2: méthode executeBapi_GetEmployeeData	166
Figure 233 - Tutoriel 2: action Search	167
Figure 234 - Tutoriel 2: vue SearchView	167
Figure 235 - Tutoriel 2: design du formulaire.....	168
Figure 236 - Tutoriel 2: création de l'archive	170
Figure 237 - Tutoriel 2: déploiement de l'archive	171
Figure 238 - Tutoriel 2: Web Dynpro Content Administrator	171
Figure 239 - Tutoriel 2: Connexions JCO	172
Figure 240 - Tutoriel 2: onglet "JCO Connections".....	172
Figure 241 - Tutoriel 2: Connexions JCO (inactives).....	173
Figure 242 - Tutoriel 2: création de la connexion JCO	173
Figure 243 - Tutoriel 2: création de la connexion JCO	174
Figure 244 - Tutoriel 2: création de la connexion JCO	174
Figure 245 - Tutoriel 2: création de la connexion JCO	174
Figure 246 - Tutoriel 2: création de la connexion JCO	175
Figure 247 - Tutoriel 2: création de la connexion JCO	175
Figure 248 - Tutoriel 2: résumé des paramètres de la connexion JCO	176
Figure 249 - Tutoriel 2: connexions JCO (1/2 active)	176
Figure 250 - Tutoriel 2: connexions JCO (actives)	177
Figure 251 - Tutoriel 2: création de l'application	177
Figure 252 - Tutoriel 2: lancement de l'application	178
Figure 253 - Tutoriel 2: test de l'application	178
Figure 254 - Tutoriel 3: Web Dynpro Explorer	180
Figure 255 - Tutoriel 3: Modèle dans Web Dynpro Explorer	181

Figure 256 - Tutoriel 3: Custom Controller et modèle utilisé	181
Figure 257 - Tutoriel 3: lancement du Data Modeler	182
Figure 258 - Tutoriel 3: renommage des nœuds récurrents.....	183
Figure 259 - Tutoriel 3: renommage des nœuds récurrents.....	183
Figure 260 - Tutoriel 3: BAPIs utilisées.....	183
Figure 261 - Tutoriel 3: mapping du Custom Controller vers le modèle.....	184
Figure 262 - Tutoriel 3: création d'un type simple	185
Figure 263 - Tutoriel 3: nommage du type simple	185
Figure 264 - Tutoriel 3: Définition du type simple	186
Figure 265 - Tutoriel 3: énumération du type simple	187
Figure 266 - Tutoriel 3: création d'un Value Node.....	188
Figure 267 - Tutoriel 3: nœud personnalisé pour les types d'absence	188
Figure 268 - Tutoriel 3: type de l'attribut absType (avant)	188
Figure 269 - Tutoriel 3: type de l'attribut absType (modification).....	189
Figure 270 - Tutoriel 3: type de l'attribut absType (résultat).....	189
Figure 271 - Tutoriel 3: type de l'attribut absType (résultat).....	190
Figure 272 - Tutoriel 3: nœud personnalisé pour les catégories d'absence	190
Figure 273 - Tutoriel 3: onglet propriétés de la vue StartView	191
Figure 274 - Tutoriel 3: choix du contrôleur requis pour la vue	191
Figure 275 - Tutoriel 3: sélection du Custom Controller.....	192
Figure 276 - Tutoriel 3: type de l'attribut pdfSource	192
Figure 277 - Tutoriel 3: édition du mapping vue → Custom Controller	193
Figure 278 - Tutoriel 3: choix du nœud à mapper	193
Figure 279 - Tutoriel 3: sélection de Bapi_Absence_Create_Input	194
Figure 280 - Tutoriel 3: nœuds et attributs disponibles pour le mapping	194
Figure 281 - Tutoriel 3: éléments mapping pour Bapi_Absence_Create_Input	195
Figure 282 - Tutoriel 3: contexte de StartView	195
Figure 283 - Tutoriel 3: édition du mapping	196
Figure 284 - Tutoriel 3: sélection de Bapi_Persdata_Getdetailedlist_Input.....	196
Figure 285 - Tutoriel 3: attributs et nœuds disponibles pour mapping.....	197
Figure 286 - Tutoriel 3: sélection des attributs retournés	198
Figure 287 - Tutoriel 3: éléments sélectionnés pour le mapping	198
Figure 288 - Tutoriel 3: contexte de la vue StartView.....	199
Figure 289 - Tutoriel 3: lancement du Data Modeler	199
Figure 290 - Tutoriel 3: liaison StartView → Custom Controller.....	200
Figure 291 - Tutoriel 3: mapping StartView → Custom Controller.....	200
Figure 292 - Tutoriel 3: mapping StartView → Custom Controller.....	201
Figure 293 - Tutoriel 3: nommage de la méthode	202
Figure 294 - Tutoriel 3: paramètres de la méthode	202
Figure 295 - Tutoriel 3: création du paramètre	203
Figure 296 - Tutoriel 3: paramètres de la méthode	203
Figure 297 - Tutoriel 3: liste des méthodes du Custom Controller.....	204
Figure 298 - Tutoriel 3: liste des actions de la vue StartView	209

Figure 299 - Tutoriel 3: design de StartView	212
Figure 300 - Tutoriel 3: formulaire, aperçu du résultat.....	213
Figure 301 - Tutoriel 3: liaison des données dans le formulaire	214
Figure 302 - Tutoriel 3: formulaire: partie supérieure	214
Figure 303 - Tutoriel 3: formulaire, sélection du type de champ de saisie	215
Figure 304 - Tutoriel 3: formulaire, modification de l'affichage du champ	215
Figure 305 - Tutoriel 3: formulaire, modification du type de valeur.....	216
Figure 306 - Tutoriel 3: formulaire, calcul de la date du jour.....	216
Figure 307 - Tutoriel 3: formulaire, insertion d'une image.....	216
Figure 308 - Tutoriel 3: formulaire, message sur la connexion réseau	217
Figure 309 - Tutoriel 3: formulaire, explorateur de fichier (image)	217
Figure 310 - Tutoriel 3: formulaire, image	217
Figure 311 - Tutoriel 3: formulaire, liaison de l'image au formulaire	218
Figure 312 - Tutoriel 3: formulaire, partie centrale	219
Figure 313 - Tutoriel 3: formulaire, bouton Submit (natif)	221
Figure 314 - Tutoriel 3: formulaire, partie inférieure.....	222
Figure 315 - Tutoriel 3: formulaire, ordre de tabulation.....	223
Figure 316 - Tutoriel 3: test de l'application	224
Figure 317 - Tutoriel 3: test de l'application, pré-remplissage	225
Figure 318 - Tutoriel 3: test de l'application, connexion à SAP.....	226
Figure 319 - Tutoriel 3: test de l'application, fenêtre de gestion du temps de travail dans SAP	226
Figure 320 - Tutoriel 3: test de l'application, saisie du matricule de l'employé dans SAP	227
Figure 321 - Tutoriel 3: test de l'application: affichage des absences de l'employé dans SAP	228
Figure 322 - Tutoriel 4: aperçu du résultat.....	229
Figure 323 - Tutoriel 4: création d'un type simple	230
Figure 324 - Tutoriel 4: nœud FormNode (avec types) dans le contexte du Custom Controller	230
Figure 325 - Tutoriel 4: mapping vue temporaire → Custom Controller.....	231
Figure 326 - Tutoriel 4: enumerated drop-down list (no select)(ActiveX)	232
Figure 327 - Tutoriel 4: drag & drop du type d'absence	232
Figure 328 - Tutoriel 4: options de binding.....	232
Figure 329 - Tutoriel 4: aperçu du formulaire	233
Figure 330 - Tutoriel 4: fichier .xdp dans l'explorateur Windows (dossier source)	234
Figure 331 - Tutoriel 4: contenu du répertoire de destination	235
Figure 332 - Tutoriel 4: fichiers disponibles dans Web Dynpro Explorer	235
Figure 333 - Tutoriel 4: modèle importé	236
Figure 334 - Tutoriel 4: utilisation d'un modèle	236
Figure 335 - Tutoriel 4: mapping Custom Controller → modèle	237
Figure 336 - Tutoriel 4: lancement du diagramme de navigation.....	241
Figure 337 - Tutoriel 4: configuration de la navigation	242
Figure 338 - Tutoriel 4: type Resource	242
Figure 339 - Tutoriel 4: contexte de DownloadView.....	243
Figure 340 - Tutoriel 4: propriétés de l'élément FileDownload	245
Figure 341 - Tutoriel 4: exemple de design de la vue DownloadView	245

Figure 342 - Tutoriel 4: ajout du Custom Controller dans les propriétés de UploadView	246
Figure 343 - Tutoriel 4: exemple de design de UploadView	247
Figure 344 - Tutoriel 4: contexte de ResultView	247
Figure 345 - Tutoriel 4: exemple de design de ResultView	248
Figure 346 - Tutoriel 4: test de l'application, saisie d'un matricule	249
Figure 347 - Tutoriel 4: test de l'application, message	250
Figure 348 - Tutoriel 4: test de l'application: téléchargement du pdf	250
Figure 349 - Tutoriel 4: test de l'application, ouverture du formulaire dans Adobe Reader.....	251
Figure 350 - Tutoriel 4: test de l'application, upload du formulaire	252
Figure 351 - Tutoriel 4: test de l'application, résumé des données saisies	252
Figure 352 - Tutoriel 4: test de l'application, vérification de la saisie dans SAP	253
Figure 353 - Campus Management + Application Web Dynpro avec une Interactive Form	255

1 Introduction

1.1 Sujet

Le titre exact de ce travail de diplôme est "Adobe Interactive Forms (Authoring) & SAP ECC". Cet outil, que l'on nomme plus communément "SAP Interactive Forms by Adobe" permet d'étendre la couverture des applications SAP en touchant davantage d'utilisateurs et ce de manière plus efficace. Ces formulaires peuvent être utilisés en ligne ou hors ligne et ils peuvent être configurés de manière à pouvoir échanger des informations en provenance et à destination de SAP.

Ce travail de diplôme est réalisé en collaboration avec le Centre de compétence SAP de la HES-SO Valais.

1.2 Objectifs

- Installer Adobe Interactive Forms
- Etablir un lien entre ECC 6.0 et Adobe Acrobat
- Réaliser un prototype d'interaction entre Adobe et ECC 6.0
- Réaliser 2 Formulaires pour l'Etat du Valais
- Réaliser 1 Formulaire pour Campus Management

1.2.1 Installer Adobe Interactive Forms

Il s'agit de l'installation de tout ce qu'il faut pour pouvoir commencer à utiliser les SAP Interactive Forms. Cela veut dire que cette installation est coupée en 2 parties:

- **Sur le serveur:**
Installation et configuration de tout ce qu'il faut pour pouvoir manipuler des formulaires interactifs.
- **Sur la machine de développement:**
Installation et configuration de tout ce qu'il faut pour pouvoir créer et tester des formulaires interactifs.

1.2.2 Etablir un lien entre ECC 6.0 et Adobe Acrobat

En d'autres termes, un formulaire doit pouvoir:

- Recevoir des données provenant de SAP (ECC 6.0) et les afficher dans un formulaire .pdf
- Envoyer des données saisies dans le formulaire .pdf vers SAP (ECC 6.0)

L'objectif précise qu'il faut Adobe Acrobat, mais nous verrons qu'il suffit de l'Adobe Reader (version gratuite) pour le faire.

1.2.3 Prototype d'interaction entre Adobe et ECC 6.0

Réalisation d'un formulaire prototype qui doit pouvoir faire usage du lien entre ECC 6.0 et le formulaire.

1.2.4 Deux formulaires pour l'Etat du Valais

Réalisation de 2 formulaires pour l'Etat du Valais. Ils doivent être basés sur:

- Deux formulaires actuellement utilisés par l'Etat du Valais (formulaires d'absences et présences) pour le thème
- Le prototype, pour ce qui est de la partie technique (liaison avec ECC 6.0)

Remarque: suite à une séance avec le Dr. Maier, le nombre de formulaire à développer pour l'Etat du Valais est passé à 1.

1.2.5 Un formulaire pour Campus Management

Réalisation d'un formulaire pour Campus Management. Il doit être basé sur:

- Le formulaire de demande de déplacement du service militaire fourni par le Dr. Maier
- Le prototype, pour ce qui est de la partie technique (liaison avec ECC 6.0)

Remarque: suite à une séance avec le Dr. Maier, le développement de ce formulaire a été abandonné.

1.2.6 Démarche

Le but de ce travail de diplôme est avant tout de découvrir, évaluer et tester cette technologie qu'est SAP Interactive Forms by Adobe. L'idée est de pouvoir fournir un document de référence pour une personne qui souhaiterait en savoir plus sur cette technologie et qui aimerait pouvoir acquérir les bases nécessaires au développement des formulaires interactifs. Pour ce faire, ce rapport proposera diverses analyses et tutoriels détaillés.

1.3 Planification

Ce travail se déroule sur 12 semaines, soit 600 heures, du 18 février 2008 au 19 mai 2008. La 13^{ème} semaine correspond à une période de vacance. La planification détaillée ainsi que les heures effectuées se trouvent en annexe. La partie qui suit va en faire la synthèse.

1.3.1 Découpage prévu

La planification initiale donne le découpage qui suit:

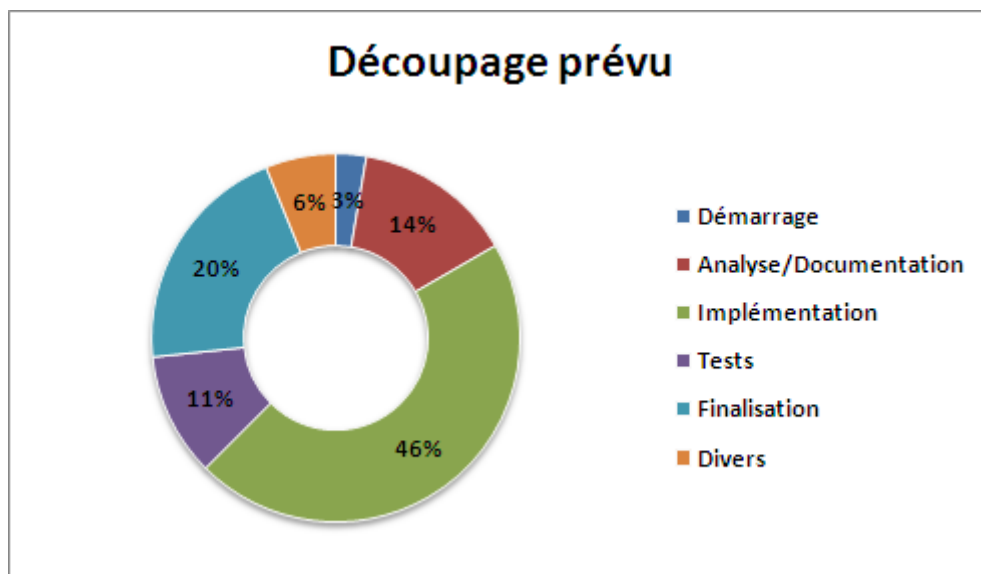


Figure 1 - Découpage du travail prévu

Les 2 parts qui peuvent choquer sont celles de l'implémentation et de la finalisation. La raison est cependant simple:

Etant donné que ce projet comporte une grande part de découverte, je n'ai pas hésité à attribuer près de 50% du temps disponible à l'implémentation. La phase de finalisation quant à elle représente 20% du travail car elle comporte la rédaction du rapport final qui contiendra plusieurs tutoriels. Or je sais d'expérience que ces derniers sont relativement laborieux à faire si on souhaite être à la fois précis et compréhensible.

Enfin, la partie "Divers" comporte les séances, la rédaction des rapports hebdomadaires et les communications.

1.3.2 Découpage effectif

Voici le découpage effectif constaté durant la dernière semaine du travail de diplôme:

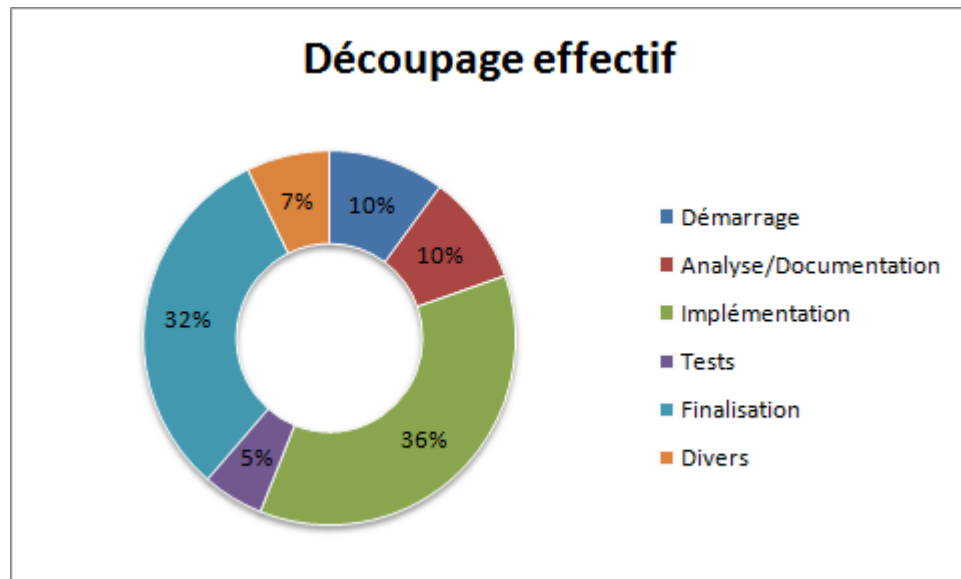


Figure 2 - Découpage effectif du travail

Ce graphique se base sur les suivis des heures hebdomadaires ainsi sur des ajustements personnels. En effet, le planning initial n'était pas aisé à reprendre car certaines tâches qui se sont révélées être importantes (notamment la configuration du serveur) n'ont pas été planifiées. De plus, le suivi est un peu moins détaillé que la planification, ce qui m'a obligé à effectuer quelques répartitions basées sur des estimations. Le résultat obtenu devient ainsi comparable.

1.3.3 Comparaison

Le graphique ci-dessous représente les différences entre la répartition prévue et effective des tâches:

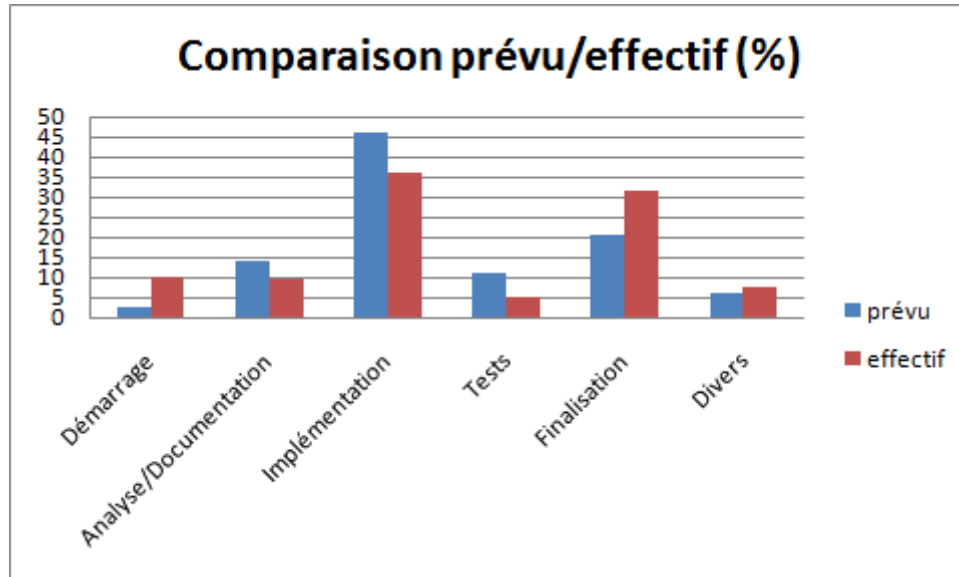


Figure 3 - Comparaison du découpage prévu/effectif

Explication des disparités par phase:

- Démarrage:**
 La principale différence vient du fait que la configuration du serveur n'a pas du tout été planifiée et qu'elle a pris un temps non négligeable. De plus, l'installation de l'Adobe LiveCycle Designer sur le poste de travail a pris un peu plus de temps que prévu car ne n'arrivais pas à trouver la bonne version à utiliser.
- Analyse/Documentation:**
 La phase d'analyse et de recherche de renseignement a simplement pris moins de temps que prévu. Après coup je constate que le nombre d'heures attribués à certaines tâches de cette phase était un peu exagéré.
- Implémentation:**
 C'est la plus grosse disparité en termes d'heures: pratiquement 100. Il y a plusieurs raisons à ça. Tout d'abord, il y a 2 formulaires dont la création a été abandonnée suites aux séances avec le Dr. Maier. Selon le planning, cela représente environ 70 à 80 heures. L'autre raison est qu'à partir du moment où on commence à comprendre comment certains mécanismes fonctionnent, on tombe sur des étapes qui se répètent et qui sont réalisées bien plus rapidement. Cela n'a pas été pris en compte initialement.
- Tests:**
 Ici aussi, la disparité entre les deux pourcentages est assez grande. Les raisons sont les même que pour l'implémentation: 2 formulaires ont été abandonnés et à partir

du moment où on trouve comment corriger un bug, on aura plus besoin de refaire des recherches et des tests la fois suivante.

- **Finalisation:**

La finalisation a pris bien plus de temps que prévu. Il y a tout d'abord la rédaction du rapport pour laquelle je savais par expérience que j'y passerais un temps conséquent. Malgré ça, j'ai quand même sous-estimé le temps consacré. Je dirais que la principal raison vient du fait que ce rapport contient plusieurs tutoriels et que l'exigence de clarté a requis un temps plus accru.

- **Divers:**

Bien que les séances ont pris globalement 2 fois moins de temps que prévu, ce qui a rallongé cette phase était la création de plusieurs tutoriels et démos. Même si le programme utilisé (Wink) est relativement facile à prendre en main, il a fallu contrôler et parfois paramétrer chaque étape de la vidéo, ce qui, en plus des tests, prend très facilement du temps.

Enfin, un nombre qui pourrait surprendre est le total des heures: 472. A première vue cela semble être peu, mais il faut savoir qu'en parallèle à ce travail de diplôme, j'ai suivi le module 626 qui avait lieu chaque mercredi de 13h15 à 16h30. Certes, ce module ne doit pas entrer en compte dans la planification du travail mais je pense l'avoir incorporé de manière inconsciente dans la charge de travail hebdomadaire, m'empêchant ainsi de réaliser que j'aurai pu faire quelques heures supplémentaires chaque semaine.

Un autre élément qui n'est pas comptabilisé dans les heures est le fait qu'il m'arrivait très régulièrement de travailler un peu chez moi, principalement quand je coïnçais sur un problème et qu'il fallait trouver une solution à tester dès le lendemain. En tenant compte de ça, le nombre d'heures réel devrait donc tourner entre 500 et 510 heures.

Enfin la principale conclusion que je tire de ce suivi des heures est que ma démarche n'a pas été assez rigoureuse.

1.4 Participations

Voici la liste des personnes qui ont participé à réalisation de ce travail de diplôme:

Nom	Rôle	Lieu	E-mail
David Jacquin	Auteur	HES-SO Sierre	david.jacquin@students.hevs.ch
Dr. Werner Maier	Professeur	HES-SO Sierre	werner.maier@hevs.ch
Ivan Schnyder	Assistant	HES-SO Sierre	ivan.schnyder@hevs.ch
Olivier Tosi	Assistant	HES-SO Sierre	olivier.tosi@hevs.ch
Michael Müller	Collègue	HES-SO Sierre	michael.mueller@students.hevs.ch

1.5 Matériel utilisé

Deux machines ont été utilisées:

Nom	CPU	RAM	Système d'exploitation
WHK37006 (client)	Intel Core 2 6300 1.86Ghz	2 Go	Windows Server 2003, Enterprise Edition (32 bits)
mshsap15 (serveur)	Intel Xeon E5320 1.86Ghz	8 Go	Windows Server 2003, Enterprise x64 Edition (64 bits)

Remarque: le serveur mshsap15 correspond au système de développement **D07**. Par mesure de simplicité, nous utiliserons cette abréviation pour le désigner dans ce rapport.

2 SAP & Adobe

2.1 SAP

Basé à Walldorf en Allemagne, SAP est actuellement le premier fournisseur de logiciels destinés aux entreprises et à leurs inter-collaborations¹. SAP compte plus de 12 millions d'utilisateurs, plus de 46'000 clients répartis dans 120 pays différents et environ 1500 partenaires². Parmi ces partenaires, on retrouve notamment Intel, IBM, Hewlett-Packard, Cisco, Oracle et bien évidemment Adobe (depuis 2002). Le principal produit de SAP est SAP Business Suite, qui regroupe 5 applications:

- **SAP Customer Relationship Management (CRM):** pour optimiser la relation entre l'entreprise et ses clients.
- **SAP ERP (Enterprise Resource Planning):** qui vise à répondre à tous les besoins en application des moyennes et grandes entreprises et ce quelque soit leur secteur d'activité. SAP ERP est composé de 5 solutions:
 - o SAP ERP enhancement packages: qui comme son nom l'indique permet d'améliorer l'ERP de manière beaucoup plus flexible et adaptée aux besoins.
 - o SAP ERP Financials: pour la gestion de la partie finance de l'entreprise.
 - o SAP ERP Human Capital Management: pour la gestion des ressources humaines.
 - o SAP ERP Operations: pour simplifier et automatiser les processus métier de l'entreprise.
 - o SAP ERP Corporate Services: pour la gestion des coûts et des risques
- **SAP Product Lifecycle Management (PLM):** qui vise à gérer complètement les produits d'une entreprise.
- **SAP Supply Chain Management (SCM):** qui vise à optimiser la collaboration et la coordination de tout ce qui est relatif à la logistique.
- **SAP Supplier Relationship Management (SRM):** pour optimiser la relation entre l'entreprise et ses fournisseurs.

Il faut aussi savoir que SAP AG s'intéresse aussi aux petites et moyennes entreprises en offrant des solutions tels que SAP Business One, SAP Business All-In-One ou, sorti plus récemment, SAP Business ByDesign, qui sont des alternatives à la fois plus adaptées et financièrement plus abordables.

SAP AG dispose aussi d'organismes de recherche et de développement ("SAP Labs") un peu partout dans le monde et qui sont chacun spécialisés dans un domaine particulier.

¹ Source: SAP: <http://www.sap.com/about/company/history.epx>

² Source: SAP: <http://www.sap.com/about/company/history.epx>

A cela s'ajoute toute la communauté SDN³ (SAP Developer Network) qui regroupe notamment des développeurs, analystes ou encore consultants et qui propose par l'intermédiaire de ses forums, blogs, articles ou téléchargements divers une véritable mine d'informations et d'aides pour les nombreuses solutions de SAP.

Enfin, il est aussi nécessaire de mentionner le SAP Service Marketplace⁴ qui correspond à un vaste portail visant à améliorer la collaboration entre SAP, ses clients et ses partenaires.

Aux dernières nouvelles, les bénéfices de SAP AG ont plus que triplés en 5 ans, passant de 509 à 1871 millions d'Euros⁵. La dernière acquisition majeure de SAP AG date d'octobre 2007, où 6.8 milliards de dollars⁶ ont été dépensés pour le rachat de Business Objects, une entreprise française spécialisée dans les logiciels de "Business Intelligence", ou plus simplement la gestion et le traitement d'informations dans le but de fournir une aide à la décision.

2.2 Présentation Adobe

Basé à San José en Californie, Adobe est actuellement l'un des leaders sur le marché des logiciels graphiques et vidéos, avec des produits tels qu'Adobe Illustrator, Adobe Photoshop ou encore Adobe Premiere. En 2005, Adobe a acquis Macromedia, mettant ainsi la main sur Flash et Dreamweaver entre autre, deux poids lourds en matière de création de pages et contenus destiné à la publication sur Internet. Adobe est aussi à l'origine du format PDF (Portable Document Format), créé en 1993 et dont l'idée de base est de pouvoir échanger et afficher des documents qui conservent leur apparence quels que soient le système ou le matériel utilisés⁷. Ce format est utilisé partout aujourd'hui.

2.3 Opinion

Nous avons ici la rencontre entre 2 géants dans leurs domaines respectifs. Ils ont chacun un poids et une expérience conséquente sur leurs marché, ce qui veut dire qu'à priori on peut s'attendre à utiliser un outil d'une certaine qualité et offrant des possibilités à la fois nombreuses et intéressantes.

³ Adresse: <http://sdn.sap.com>

⁴ Adresse: <http://service.sap.com>

⁵ Source: SpiegelOnline: <http://www.spiegel.de/spiegel/0,1518,504625,00.html>

⁶ Source: BusinessWeek:

http://www.businessweek.com/globalbiz/content/jan2008/gb20080116_457764.htm

⁷ Source: PDF: PDF Reference.pdf

3 SAP Interactive Forms by Adobe

3.1 Présentation générale

SAP Interactive Forms by Adobe est le résultat du partenariat signé en 2002 entre ces 2 poids lourds. L'un des buts de cette solution est de minimiser les pertes de temps et d'argent liées aux inconvénients du support papier lors de l'utilisation de formulaires, mais aussi de pouvoir les intégrer dans les processus métier d'une entreprise.

Comme son nom l'indique, il permet de faire appel à des formulaires interactifs. Concrètement, cette solution optimise grandement l'utilisation de formulaires au sein d'une entreprise ou organisation en proposant d'utiliser des documents à la fois personnalisés, fiables et aussi simples d'utilisation que leurs équivalents au format papier.

3.2 Situation classique

Dans toute entreprise ou organisation, l'utilisation de formulaires est une pratique inévitable. Que ce soit une fiche d'inscription, un bon de commande ou encore une demande de congé, on retrouve toujours la transmission d'un document qui contient un certain nombre d'informations, qui demande éventuellement d'en saisir quelques-unes et qui doit ensuite être envoyé à la personne ou organisme responsable de son traitement. C'est un scénario tout ce qu'il y a de plus classique et de manière générale, plus la taille d'une entreprise est grande et plus il se produit souvent.

Aujourd'hui, une grande partie de ces formulaires ont comme support le papier. En effet, il est relativement aisé d'imprimer un formulaire, le transmettre par fax, courrier ou simplement en personne à son destinataire afin qu'il le remplisse et enfin de saisir les nouvelles informations dans le système de l'entreprise une fois la fiche retournée. Cependant, ce moyen de communication présente des inconvénients à chaque étape de l'échange.

3.2.1 Inconvénients du support papier

- **Lors de l'envoi :**
 - o Dans les cas d'envois par Fax, courrier ou mains propres, il existe un risque d'égarement du document (coordonnées incorrectes, bug, etc..).
 - o L'envoi peut prendre un certain temps, allant de quelques minutes à plusieurs jours selon le mode de transmission.
- **Lors de la réception:**
 - o Le formulaire peut arriver avec du retard, être endommagé voire même ne plus être d'actualité.

- **Lors de la saisie:**
 - o Le formulaire peut-être neutre, c'est-à-dire que la personne qui saisit des informations doit préciser ses coordonnées (nom, adresse, etc..) en plus de l'information principale qu'il lui est demandé de fournir.
 - o La saisie peut comporter des erreurs, comme par exemple des fautes d'orthographe, un numéro d'identification invalide, une date qui n'existe pas, etc. A moins de prendre le temps d'effectuer une relecture minutieuse, il n'est pas possible de vérifier de manière fiable les données saisies à ce moment-là.
 - o En cas de saisie à la main, il est possible que l'écriture soit illisible ou mal interprétée dans la suite du processus.
- **Lors du renvoi:**
 - o Mêmes inconvénients que lors de l'envoi: possibilité de perte du document et transmission qui peut prendre un certain temps.
 - o Lors de la mise à jour des données du système:
 - o En cas d'informations invalide, c'est en principe à ce moment-là qu'on s'en compte ce qui signifie qu'il faut recontacter la personne qui a complété le formulaire afin d'obtenir l'information exacte, ce qui représente une certaine perte de temps.

3.3 Avantages des SAP Interactive Forms by Adobe

Cette solution offre toute une panoplie d'avantages par rapport au support papier:

- Moins d'erreurs de saisie. Pas besoin de les traquer.
- Formulaires personnalisés, pré-remplis, donc gain de temps
- Utilisable hors-ligne (utilisation délocalisée) ou en ligne.
- Possibilité d'effectuer des opérations dans les champs de saisie. Deux langages sont disponibles:
 - o FormCalc
 - o JavaScript
- Sécurité possible via des certificats et du SSL
- Apparence visuelle identique au formulaire sur papier: le formulaire garde donc une certaine familiarité avec l'utilisateur, même peu expérimenté.
- Plus facile à mettre à jour: obsolescence plus simple à gérer.

3.4 Exemple

Afin d'illustrer ce que peut apporter cette solution, nous allons voir ce que donne un scénario de demande de congé lors qu'il est appliqué sans l'intervention des SAP Interactive Forms by Adobe. Nous verrons ensuite comment ça se passe avec.

3.4.1 Le scénario

Le scénario est le suivant: pour demander un congé, un employé remplit et transmet à la secrétaire du responsable des ressources humaines le formulaire adéquat. La secrétaire dépose le formulaire sur le bureau du responsable. Le responsable accepte ou refuse la demande, ajoute s'il le souhaite un commentaire et transmet le formulaire à la secrétaire. Cette dernière en fait une copie pour les archives et le transmet ensuite à l'employé.

3.4.2 Scénario sans SAP Interactive Forms by SAP

Sans les formulaires interactifs, voici comment ce scénario pourrait se dérouler:

- L'employé récupère sur l'intranet de l'entreprise un formulaire de demande de congé, l'imprime, et le remplit entièrement à la main.
- Une fois remplit, il transmet le formulaire à la secrétaire du responsable des ressources humaines.
- La secrétaire transmet le formulaire au responsable des ressources humaines en le posant sur son bureau.
- Le responsable des ressources humaines étudie la demande, l'accepte ou la refuse et écrit un petit commentaire.
- Le responsable des ressources humaines transmet le formulaire à sa secrétaire.
- La secrétaire en fait une photocopie pour archivage.
- La secrétaire transmet la demande acceptée ou refusée à l'employé.

Voici un schéma qui décrit ce processus:

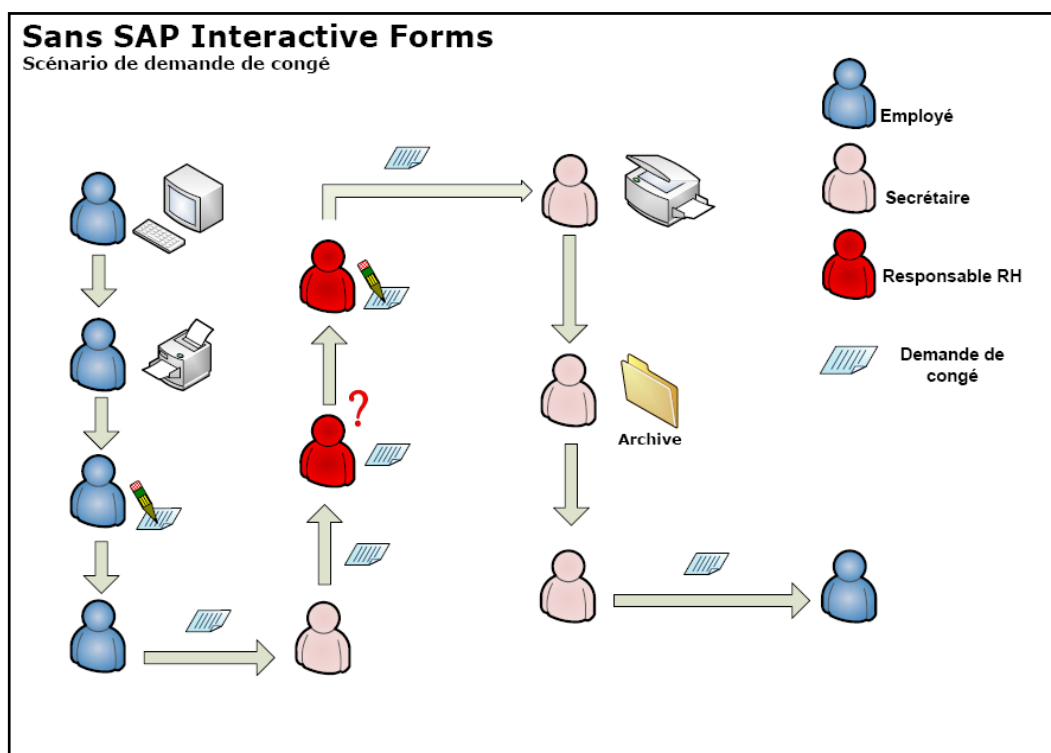


Figure 4 - Scénario sans Interactive Form: Demande de congé

3.4.3 Scénario avec SAP Interactive Forms by Adobe

L'employé récupère sur l'intranet de l'entreprise un formulaire interactif de demande de congé, l'ouvre dans Adobe Reader et saisit les informations relatives au congé. Ses informations personnelles et la date du jour sont déjà pré-remplies dans les champs correspondants.

- L'employé signe de manière digitale le formulaire afin de certifier son identité.
- L'employé clique sur le bouton Envoyer disponible sur le formulaire.
- Le formulaire est transmis au système SAP qui le transmet ensuite au responsable des ressources humaines.
- Le responsable des ressources humaines ouvre le fichier PDF, met une coche dans la partie relative à sa décision par rapport à la demande et saisit un commentaire.
- Le responsable des ressources humaines ajoute sa signature digitale pour certifier que la demande a été traitée par la bonne personne.
- Le responsable des ressources humaines clique sur le bouton Envoyer sur le formulaire.
- Le formulaire est renvoyé vers le système SAP qui transmet alors la demande signée à l'employé ainsi qu'une copie à la secrétaire qui pourra alors l'archiver.

Voici un schéma qui décrit ce processus:

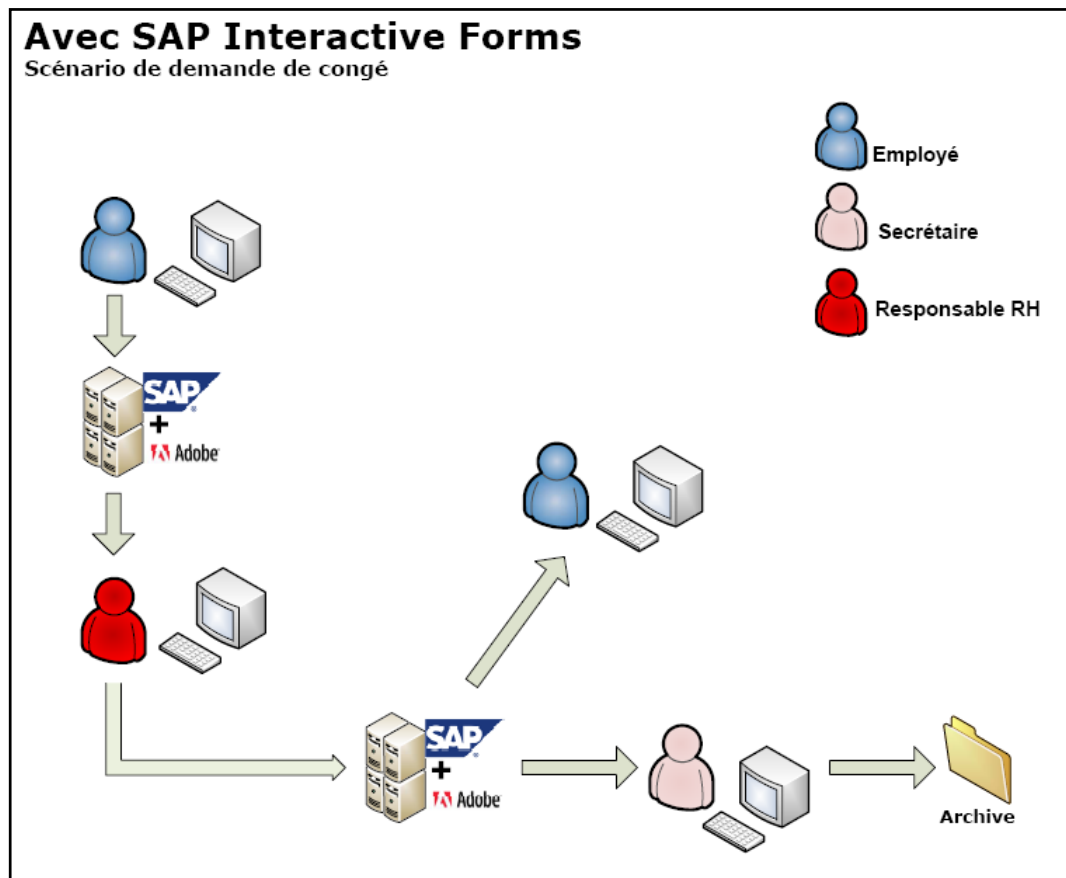


Figure 5 - Scénario avec Interactive Form: Demande de congé

3.4.4 Avantages

Nous pouvons constater plusieurs avantages:

- Le processus est moins long et plus clair.
- Tout ce passe par ordinateur. L'impression devient une étape optionnelle.
- Les acteurs du processus interviennent moins. Ils le font uniquement lorsqu'ils ont un rôle à jouer. Plus besoin d'imprimer, ni de se déplacer.
- Le formulaire est géré par le serveur. Contrairement aux formulaires au format papier, le risque d'égarement ou de détérioration est faible, voire nul.
- L'employé ne doit pas attendre que la secrétaire fasse une copie du formulaire pour les archives. Ils le reçoivent les deux en même temps, ce qui représente une optimisation du processus.

3.5 Alternatives possible

Nous allons à présent nous pencher sur les éventuelles alternatives aux formulaires interactifs: voir ce qu'elles proposent et s'il s'agit réellement de solutions concurrentes.

3.5.1 SAP Smart Forms

SAP Smart Forms est avant tout une solution destinée à l'impression en masse, par exemple quand une entreprise a besoin de s'adresser à tous ses clients (annonce, décompte, etc..). Tout comme les SAP Interactive Forms by Adobe, elles permettent d'extraire des données telles que des noms, des adresses, des quantités, etc.... depuis un système SAP et de les afficher dans les documents à publier. Leur publication peut passer par une imprimante, un fax ou encore un navigateur web, comme le montre le schéma ci-dessous:

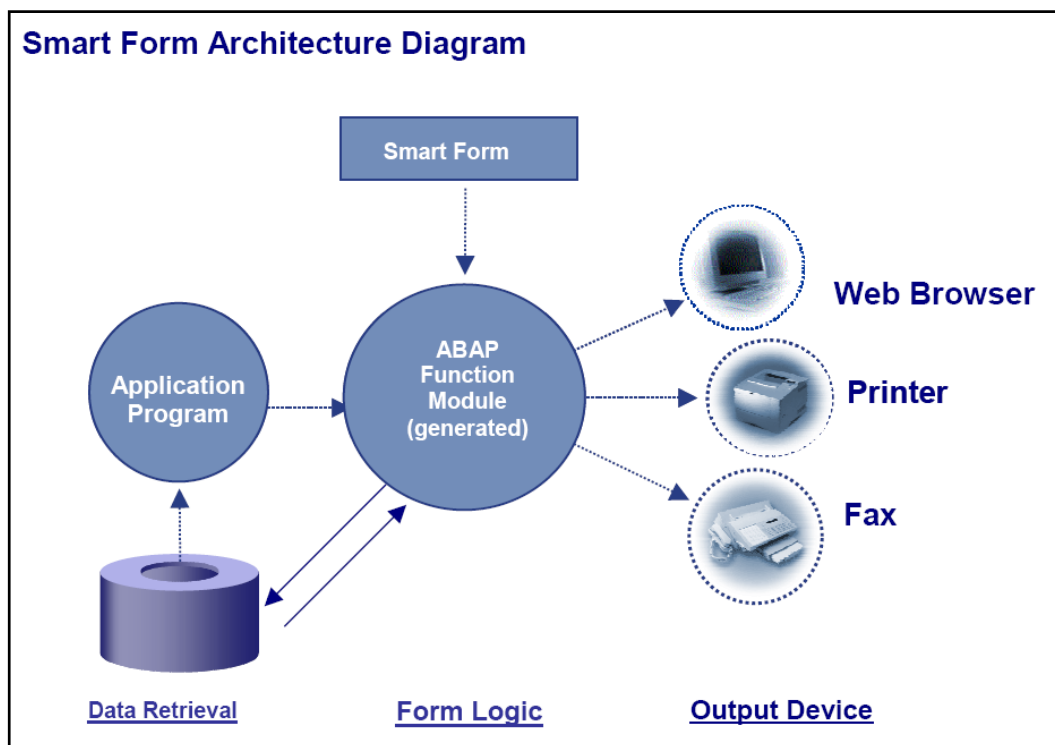


Figure 6 - Architecture Smart Forms⁸

De manière générale, les SAP Smart Forms permettent à l'utilisateur de créer un formulaire en faisant peu, voire pas de programmation. Pour utiliser l'éditeur, il suffit d'ouvrir une session avec SAP Logon et de taper le code transaction "SMARTFORMS".

⁸ Source: SAP: <http://www.sap.com/businessmaps/82087E7A8EB94E1F885C5A1F62C789FB.htm>

L'écran suivant apparaît:

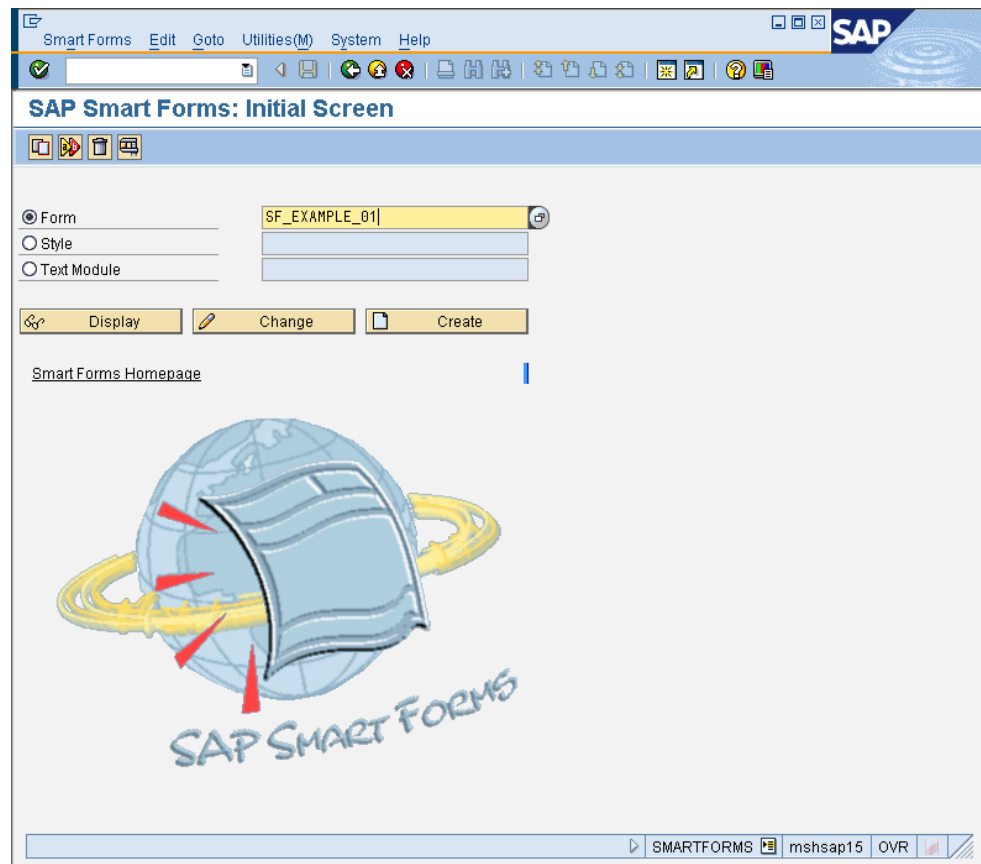


Figure 7 - SAP Smart Forms: Ecran initial

A partir de là, il est possible d'afficher, modifier ou créer une Smart Form.

Dans le cadre de cette brève présentation, nous allons simplement afficher un formulaire d'exemple déjà disponible dans le système. Pour ce faire, il suffit de saisir SF_EXAMPLE_01 dans le champ "Form" et ensuite de cliquer sur le bouton "Display".

Le SAP Form Builder apparaît alors:

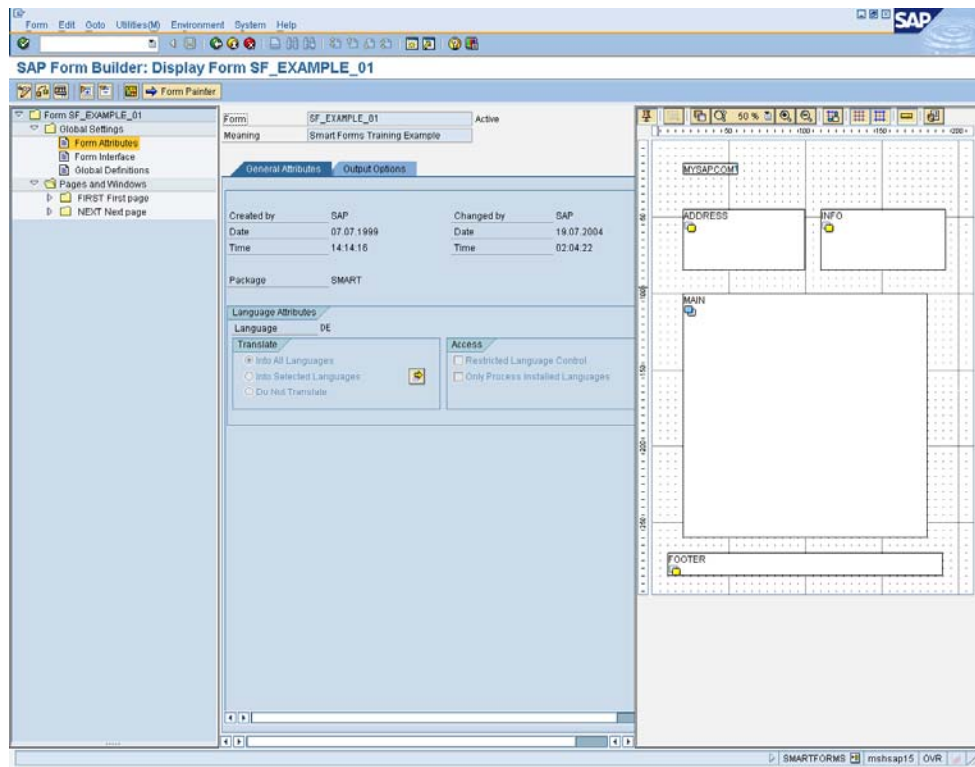


Figure 8 - SAP Form Builder: Affichage d'un formulaire

Comme le montre cette capture d'écran, la conception d'une Smart Form est grandement facilitée par cet outil. A gauche on retrouve essentiellement les différents éléments du formulaire, au centre tous les réglages associés à l'élément sélectionné et enfin à droite le designer qui permet de mettre en place les différents éléments du formulaire.

La principale différence entre SAP Smart Forms et SAP Interactive Forms by Adobe est que les Smart Forms ne permettent pas la saisie d'informations dans le but de les renvoyer vers le système SAP. On peut donc considérer que les Smart Forms et les Interactive Forms sont plus des compléments que des concurrents. En effet, l'une est destinée à l'impression en masse sur un support papier alors que l'autre révèle tout son potentiel quand on veut justement éviter d'avoir recours au papier durant les diverses étapes d'un business process.

Il est aussi intéressant de noter qu'il est possible d'effectuer une migration pour passer d'une Smart Form à une Interactive Form. Cependant, il faut savoir que cette procédure a quelques limitations et qu'avant d'entreprendre une migration, il est recommandé de se demander s'il ne serait pas plus simple de recréer en partant de rien la Smart Form que l'on veut convertir en Interactive Form.

Pour accéder à l'outil de migration, il suffit de saisir le code transaction "SMARTFORMS", de saisir le nom du formulaire que l'on souhaite convertir et enfin de cliquer dans le menu du haut "Utilities"→"Migration"→"Interactive Form"→Export, comme l'indique la capture d'écran si dessous:

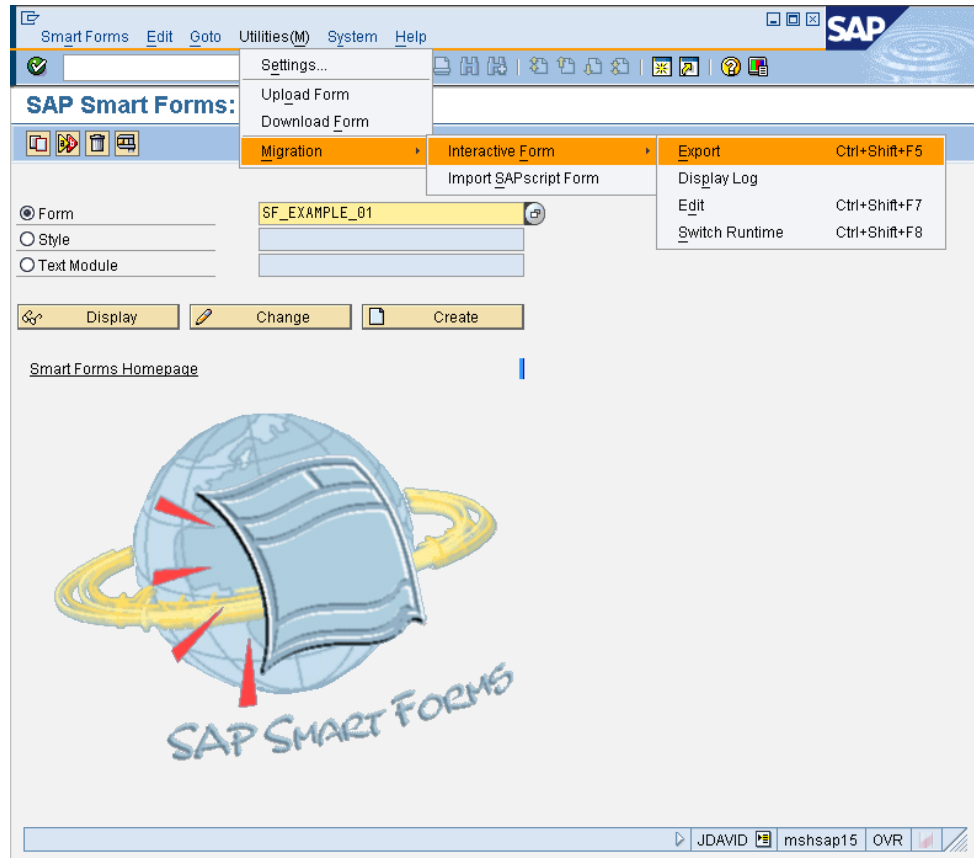


Figure 9 - Smart Forms: Migration vers Interactive Form

La fenêtre suivante apparaît alors:

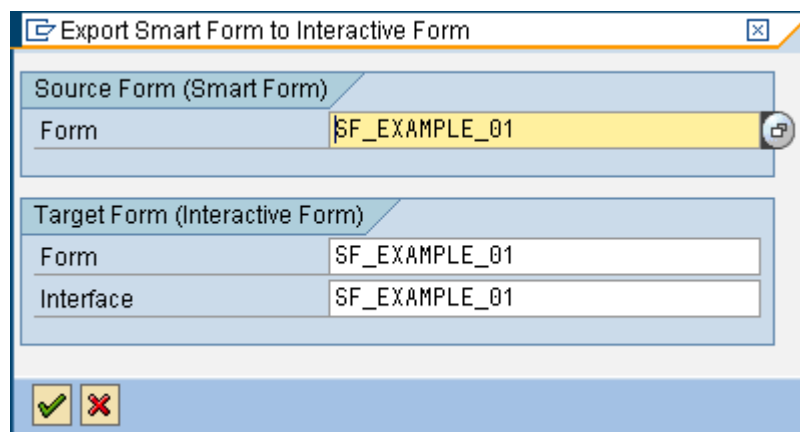


Figure 10 - Smart Forms: Export, destination

Par défaut, le formulaire source, le formulaire et l'interface d'arrivée ont le même nom. Pour plus de clarté, nous rajoutons l'extension "_IF" au nom du formulaire et de l'interface d'arrivée, comme indiqué ci-dessous:

Figure 11 - Smart Forms: Export, destination

Après avoir cliqué sur le bouton "Continue", la fenêtre suivante apparaît:

Figure 12 - Smart Forms: Paramètre d'exportation

Ici, il est possible de paramétrer la migration et de choisir ce que l'on souhaite convertir ou pas. Etant donné qu'il s'agit d'une présentation générale de l'outil, nous allons simplement laisser les réglages par défaut et cliquer sur le bouton "Continue". Peu après, la fenêtre ci-dessous, apparaît. Il suffit de cliquer sur "Continuer".

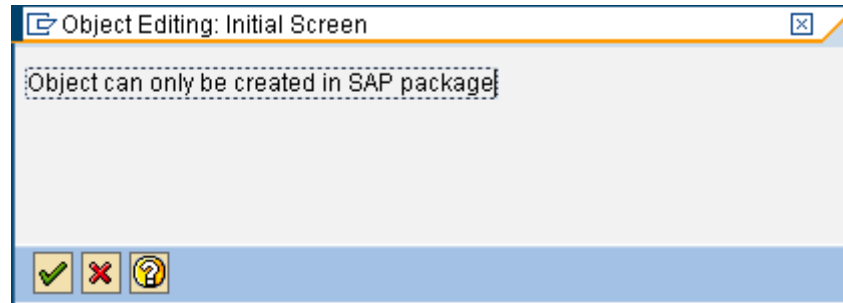


Figure 13 - Smart Forms: Edition de l'objet, information

La fenêtre ci-dessous apparaît et demande d'indiquer un nom de Package.

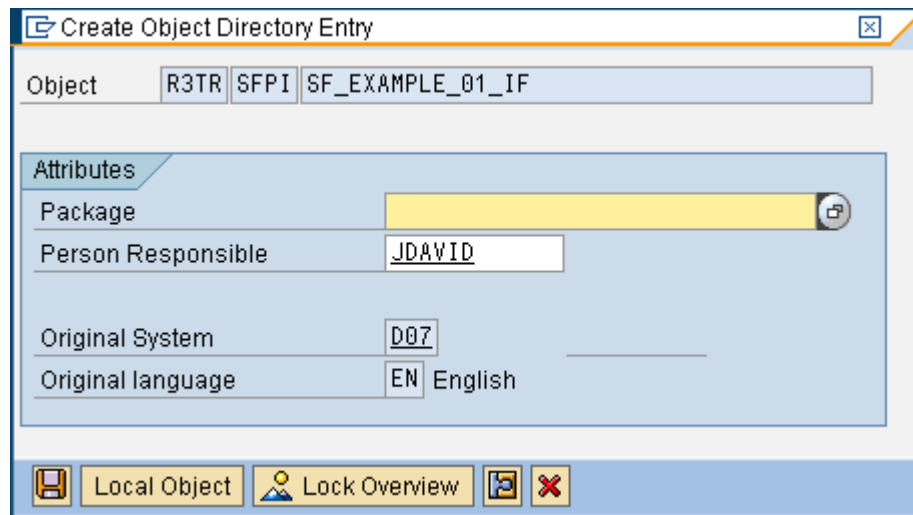
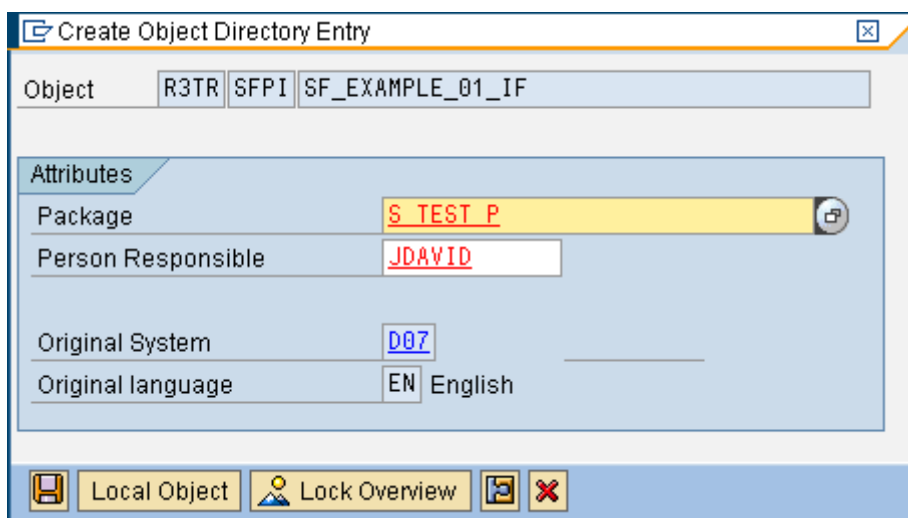


Figure 14 - Smart Forms: création d'une entrée pour l'objet

On saisit un nom de Package de test.

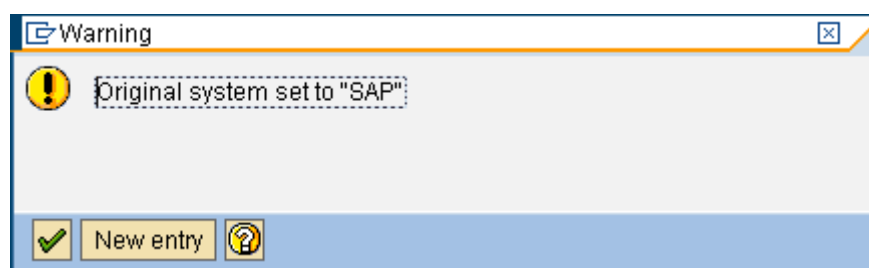


The dialog box is titled "Create Object Directory Entry". It contains the following fields:

- Object: R3TR | SFPI | SF_EXAMPLE_01_IF
- Attributes section:
 - Package: S_TEST_P (highlighted in yellow)
 - Person Responsible: JDAVID
 - Original System: D07
 - Original language: EN English
- Buttons at the bottom: Local Object, Lock Overview, and a red X button.

Figure 15 - Smart Forms - création d'une entrée pour l'objet

On clique sur "Save" puis "Continue" dans la fenêtre qui suit.

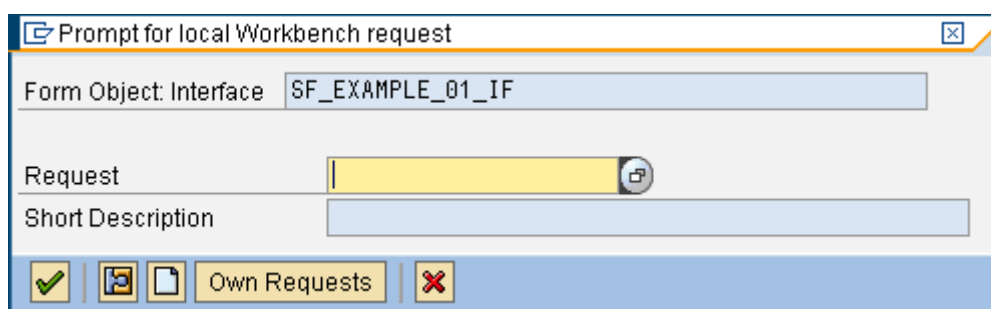


The dialog box is titled "Warning". It contains the following elements:

- A yellow warning icon and the text: Original system set to "SAP"
- Buttons at the bottom: a green checkmark icon, "New entry", and a question mark icon.

Figure 16 - Smart Forms: création de l'entrée, avertissement

Il faut alors créer une Workbench request. Pour ce faire, cliquer sur le bouton avec une feuille blanche.



The dialog box is titled "Prompt for local Workbench request". It contains the following fields:

- Form Object: Interface | SF_EXAMPLE_01_IF
- Request: (empty field with a lock icon)
- Short Description: (empty field)
- Buttons at the bottom: a green checkmark icon, a document icon, "Own Requests", and a red X button.

Figure 17 - Smart Forms: création d'une Workbench request

La fenêtre ci-dessous apparaît, il suffit de saisir une description. Puis de cliquer sur "Save".

Create Request

Request: Workbench request

Short Description: Smart Forms to Interactive Forms demo

Project:

Owner: JDAVID

Status: New

Last changed: 12.03.2008 10:03:14

Source client: 100

Target:

Tasks:

- User JDAVID

Buttons: Save, Cancel, Close

Figure 18 - Smart Forms: création d'une Workbench request

Nous obtenons alors un numéro de requête.

Prompt for local Workbench request

Form Object: Interface SF_EXAMPLE_01_IF

Request: D07K900014 Workbench request

Short Description: Smart Forms to Interactive Forms demo

Buttons: OK, Cancel, Own Requests, Close

Figure 19 - Smart Forms: Workbench request, validation

Cliquer ensuite sur les boutons "Save" ou "Continue" selon les cas jusqu'à ce que la fenêtre d'accueil du code transaction "SMARTFORMS" apparaisse.

On pourra alors lire en bas de la fenêtre que la Smart Form "SF_EXAMPLE_01" a bien été migrée, comme on peut le voir sur la capture d'écran ci-dessous:

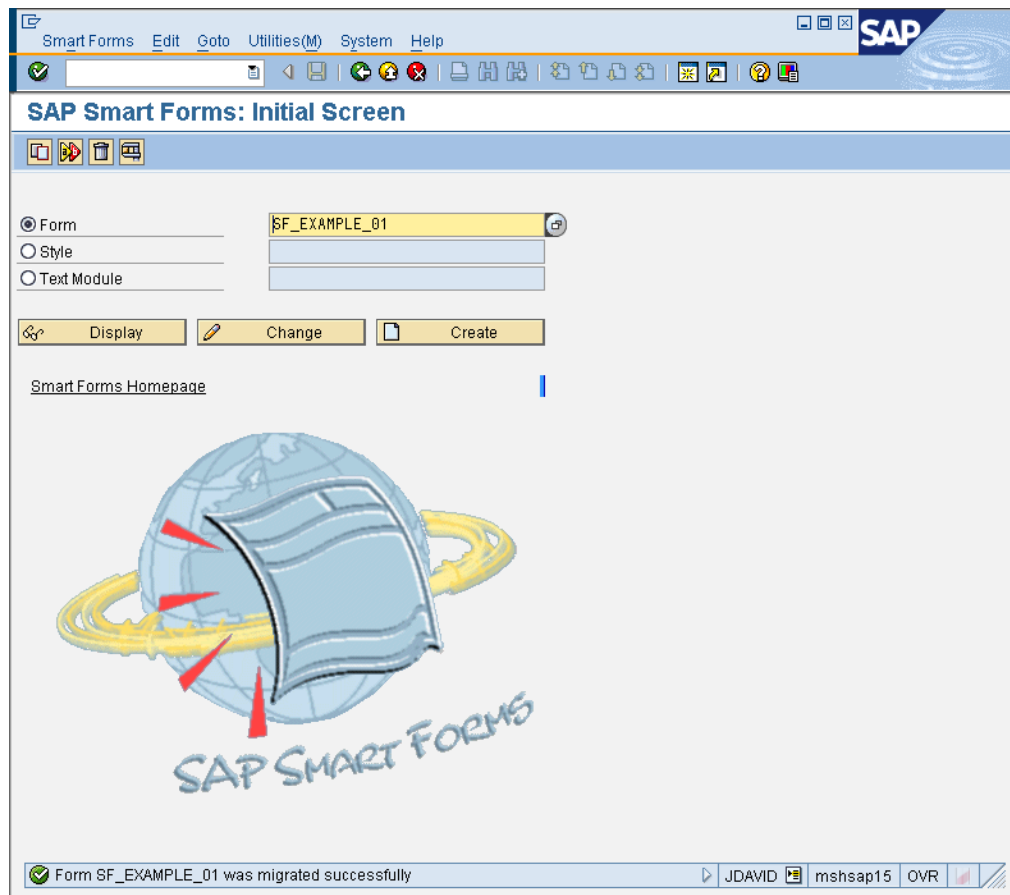


Figure 20 - Smart Forms: migration effectuée

3.5.2 Microsoft InfoPath

Microsoft InfoPath est ce qu'on pourrait considérer comme un concurrent direct aux SAP Interactive Forms by Adobe. En effet, cette solution a de nombreux points communs, entre autre:

- Elimination du support papier pour la transmission de formulaires
- Possibilité d'afficher et d'interagir avec des formulaires depuis Outlook ou une page Web.
- Possibilité de travailler hors ligne
- Possibilité de pré-remplir des champs avec des données personnalisées.
- Possibilité d'effectuer des calculs
- Utilisation du langage XML pour rendre les formulaires facilement exportables.
- Il faut aussi savoir qu'il est possible de recevoir et d'envoyer des données en provenance d'un système SAP.

Voici un exemple de formulaire créé par InfoPath et affiché dans un navigateur:

The screenshot shows a Netscape browser window displaying an 'EXPENSE REPORT' form. The form is titled 'EXPENSE REPORT' and contains several sections:

- Report Date:** 7/5/2006
- Expense Codes:** (empty field)
- Start Date:** (empty field)
- End Date:** (empty field)
- Business Purpose:** (empty text area)
- Employee Information:**
 - Name: (empty field)
 - Title: (empty field)
 - Department: (empty field)
 - ID Number: (empty field)
 - E-mail Address: (empty field)
- Manager Information:**
 - Name: (empty field)
 - E-mail Address: (empty field)
- Itemized Expenses:**

Date	Description	Category	Cost
		Other	0.00
Subtotal			0.00
Less cash advance			0.00
Total expenses			0.00
- Add Expense:** (checkbox)
- Item Details:** (empty text area)

Figure 21 - Microsoft InfoPath: affichage dans un navigateur⁹

⁹ Source: Office Online Home Page: <http://office.microsoft.com>

La gestion des formulaires est faite de manière centralisée, comme le montre l'image ci-dessous:

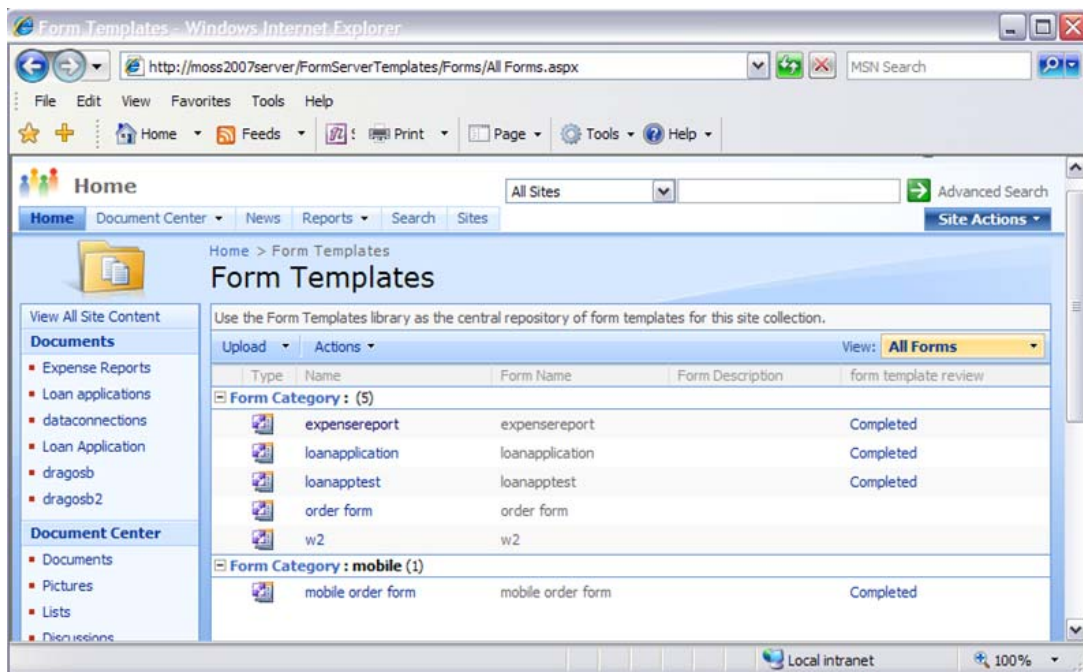


Figure 22 - Microsoft InfoPath: gestion des formulaires¹⁰

Comparé au SAP Interactive Forms by Adobe, cette solution a 2 inconvénients majeurs:

- **Plus lourde:**

Il faut un outil dédié (InfoPath) pour développer des formulaires. Il faut aussi disposer d'Office SharePoint Server pour gérer les formulaires de manière centralisée et donc pour éviter les problèmes d'obsolescence par exemple. Enfin, pour pouvoir intervenir au niveau des business processes il faut aussi avoir Microsoft BizTalk Server.

Pour les SAP Interactive Forms by Adobe, SAP NetWeaver contient déjà les Adobe Document Services qui permettent de gérer les Adobe Interactive Forms ainsi que les business processes. Pour leur création, il suffit d'installer une version d'Adobe LiveCycle Designer destinée spécialement pour SAP NetWeaver Developer Studio.

- **Moins flexible:**

Que ce soit pour la création ou l'utilisation de formulaires, il est préférable et parfois indispensable d'avoir des produits Microsoft. Bien que ce soit un poids lourd sur le marché, cela limite la portabilité de cette solution. Les SAP Interactive Forms by Adobe utilisant le format PDF, leur portabilité va au-delà des systèmes d'exploitation. Le fait que l'Adobe Reader soit gratuit et très populaire facilite grandement la communication entre une entreprise utilisant un système SAP et un client ou fournisseur travaillant sur un autre système.

¹⁰ Source: Office Online Home Page: <http://office.microsoft.com>

3.5.3 Formulaires HTML

Dès qu'on navigue sur des pages web, on peut s'attendre à utiliser des formulaires HTML à un moment ou un autre. Que ce soit pour s'inscrire sur un site, passer une commande ou encore effectuer une recherche, on saisit systématiquement une information pour la transmettre à quelqu'un ou encore pour en obtenir une en retour. Ces formulaires sont relativement simples à créer et permettent aussi, via divers scripts, d'effectuer des calculs ou encore de générer du XML. Cependant, ils présentent trois inconvénients majeurs:

- **Présentation irrégulière:**

Cette irrégularité se retrouve à deux endroits: à l'écran et à l'impression. Lorsqu'on affiche un formulaire HTML, il y a beaucoup des chances que la disposition des champs de saisie ainsi que la police utilisée change d'un navigateur à un autre, voire même d'une résolution à une autre. A partir de là on est obligé de tester de manière extensive le simple fait d'afficher un formulaire. Quant à l'impression, ce mode n'est simplement pas adapté étant donné que dans la majorité des cas, on ne retrouve pas exactement ce que l'on voit à l'écran sans compter des éléments inutiles comme des liens et des boutons.

- **Utilisation en ligne uniquement:**

De par sa nature, un formulaire HTML n'est pas exploitable hors ligne.

- **Impossibilité d'en garder une copie locale:**

En effet, lorsque vous sauvegardez une page web contenant un formulaire rempli, celle-ci ne conserve tout simplement pas les informations saisies.

4 Fichiers PDF

4.1 Vision classique

De manière générale, un fichier PDF est considéré comme un document sous forme électronique, facile à échanger et auquel on associe 3 actions basiques: affichage, impression et sauvegarde. Il s'agit en réalité d'une vision très réductrice de ce format et comme nous le verrons tout au long de ce travail de diplôme sur les formulaires interactifs, il est possible de gérer énormément de choses liées à la transmission d'informations rien qu'avec un PDF.

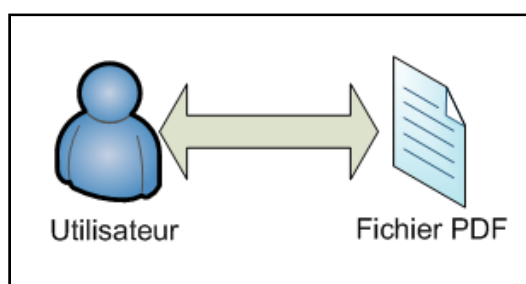


Figure 23 - PDF: vision classique

4.2 Réalité, possibilités

Avant de continuer, il convient donc de définir plus clairement comment un fichier PDF doit être vu dans le cadre de ce travail de diplôme. Le schéma ci-dessous l'explique:

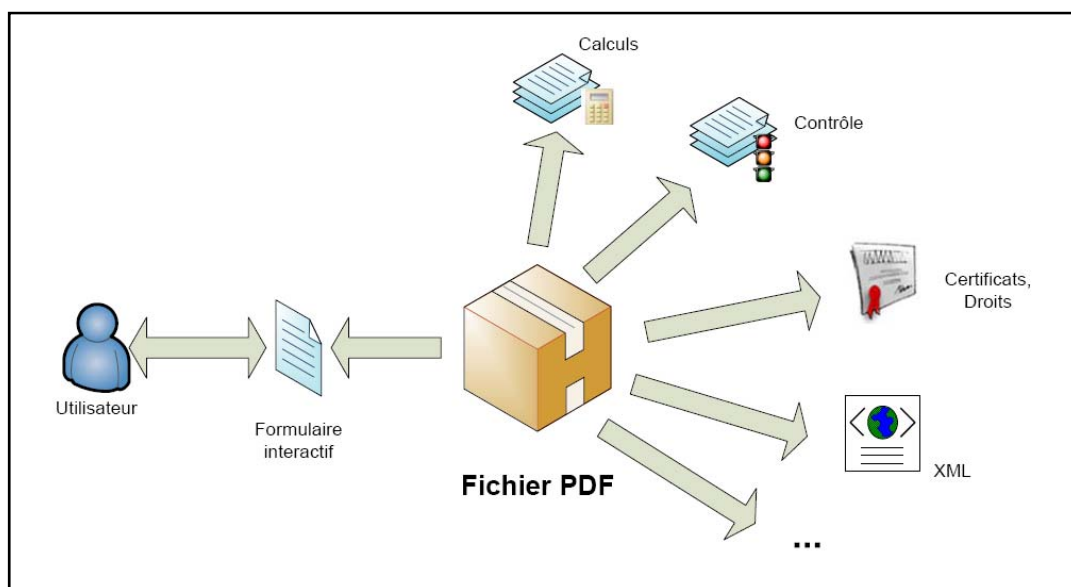


Figure 24 - PDF: plus un conteneur qu'un document

L'information la plus importante à retenir est qu'un fichier PDF est avant tout un conteneur. A l'intérieur, on y retrouve notamment les éléments suivants:

- **Le formulaire interactif:**

Ou plus généralement le document. L'utilisateur peut l'afficher, l'imprimer, le sauvegarder et, dans le cas qui nous intéresse, le modifier en saisissant des données par exemple.

- **La possibilité d'effectuer des calculs:**

Pour ce faire, deux langages sont disponibles: JavaScript et FormCalc, ce dernier étant plutôt simple d'utilisation et généralement employé pour effectuer des opérations courantes, telles que le calcul de sommes par exemple. L'exécution des calculs peut être liée à un événement, comme par exemple la mise à jour d'un autre champ, le clic d'un bouton, le passage de la souris sur une zone, etc...

Voici un exemple de code utilisant FormCalc: l'initialisation d'un champ date auquel on donne la date du jour par défaut, dans le format désiré:

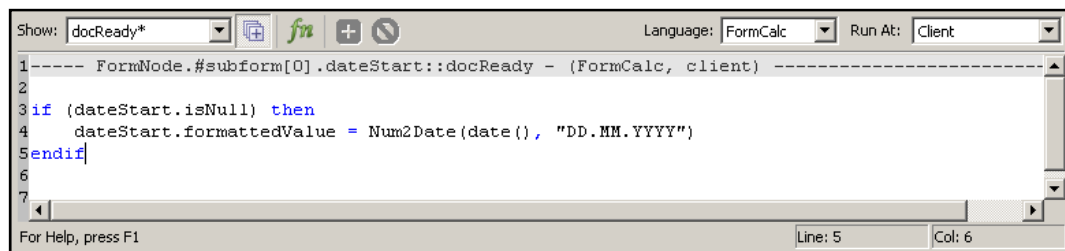


Figure 25 - PDF: exemple de calcul

- **Contrôle des données:**

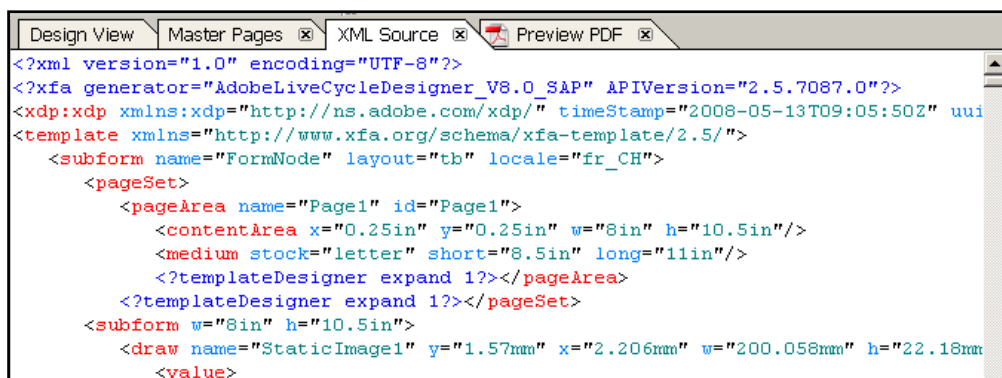
En effet, il est possible de spécifier comment une donnée peut-être entrée et de simplement refuser la saisie d'informations erronées.

- **Sécurité:**

Il est possible de définir certaines permissions sur un fichier, comme par exemple celles de le modifier, l'imprimer, ou encore d'en faire une sauvegarde locale. Dans le cadre des formulaires interactifs, un fichier PDF peut offrir la capacité de signer de manière électronique un document, permettant ainsi d'identifier de manière beaucoup plus fiable la personne qui a saisi des données et qui souhaite les renvoyer vers le système SAP. Nous verrons plus tard qu'il sera possible via les Adobe Credentials d'attacher plusieurs droits à un fichier PDF et de déverrouiller certaines fonctionnalités autrement bloquées dans la version gratuite de l'Adobe Reader.

- **Langage XML:**

Dans le cas des formulaires interactifs, les informations sont aussi disponibles en langage XML, permettant ainsi d'offrir une portabilité maximale ainsi qu'un langage commun entre le fichier PDF et le système SAP. Lorsqu'un formulaire est ajouté à une application, un template, c'est-à-dire un modèle, de ce dernier est créé en même temps. Ce modèle est alors sauvegardé dans la structure du projet, dans un fichier dont l'extension se termine par .xdp. Voici un petit aperçu du contenu d'un fichier .xdp:



```
<?xml version="1.0" encoding="UTF-8"?>
<?xfa generator="AdobeLiveCycleDesigner_V8.0_SAP" APIVersion="2.5.7087.0"?>
<xdp:xdp xmlns:xdp="http://ns.adobe.com/xdp/" timeStamp="2008-05-13T09:05:50Z" uui
<template xmlns="http://www.xfa.org/schema/xfa-template/2.5/">
  <subform name="FormNode" layout="tbl" locale="fr_CH">
    <pageSet>
      <pageArea name="Page1" id="Page1">
        <contentArea x="0.25in" y="0.25in" w="8in" h="10.5in"/>
        <medium stock="letter" short="8.5in" long="11in"/>
        <?templateDesigner expand 1?></pageArea>
      <?templateDesigner expand 1?></pageSet>
    <subform w="8in" h="10.5in">
      <draw name="StaticImage1" y="1.57mm" x="2.206mm" w="200.058mm" h="22.18mm
      <value>
```

Figure 26 - PDF: définition en XML

A partir du moment où on dispose de ce modèle, le formulaire et son contenu deviennent très facilement manipulables par n'importe quelle application.

5 Architecture & Installation

5.1 Présentation

Le schéma ci-dessous indique les différents éléments qui permettent de gérer et d'utiliser les SAP Interactive Forms by Adobe.

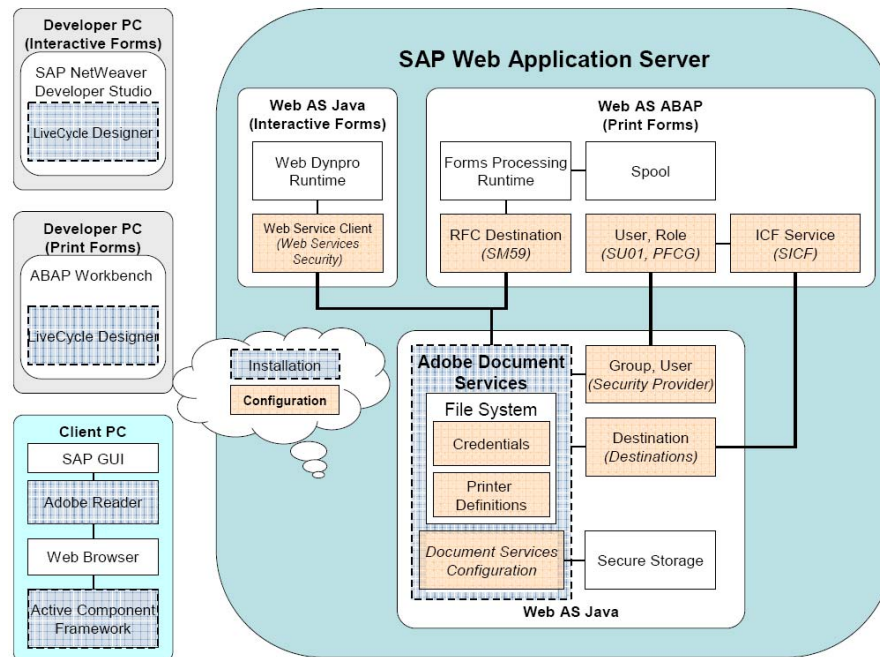


Figure 27 - Architecture SAP WebAS¹¹

On distingue 3 parties:

- Le SAP Web Application Server (aussi appelé WebAS):**
C'est lui qui permet de gérer et de générer les formulaires interactifs via l'AdobeDocumentServices. Pour ce travail de diplôme, nous nous intéresserons uniquement à la partie Java du WebAS de la D07.
- Le PC du développeur:**
Permet de créer les formulaires interactifs avec le SAP NetWeaver Developer Studio. A noter qu'il est aussi possible de développer des formulaires via le SAP GUI. Cependant, cette manière de faire ne sera pas abordée dans ce rapport.
- Le PC du client:**
Peut récupérer les formulaires interactifs, les afficher, imprimer, travailler dessus et renvoyer les données vers le système SAP.

¹¹ Source: PDF: ADS Configuration Guide

Remarque: pour ce travail de diplôme, le PC du développeur et celui du client correspondent à une seule et même machine (WHK370006).

5.2 Architecture utilisée

5.2.1 Serveur D07 (Système de développement):

5.2.1.1 SAP NetWeaver 7.0 (aussi appelé 2004s)

SAP NetWeaver constitue la base applicative de SAP. Il a 2 buts principaux: étendre les possibilités actuelles de SAP et intégrer les applications non SAP. L'arrivée de ce dernier est une petite révolution en soi étant donné qu'il marque une volonté d'ouverture.

SAP NetWeaver est composé de 4 couches:

- **People integration**
- **Information Integration**
- **Process Integration**
- **Application Server**

Le schéma ci-dessous décrit la composition de ces différentes couches:

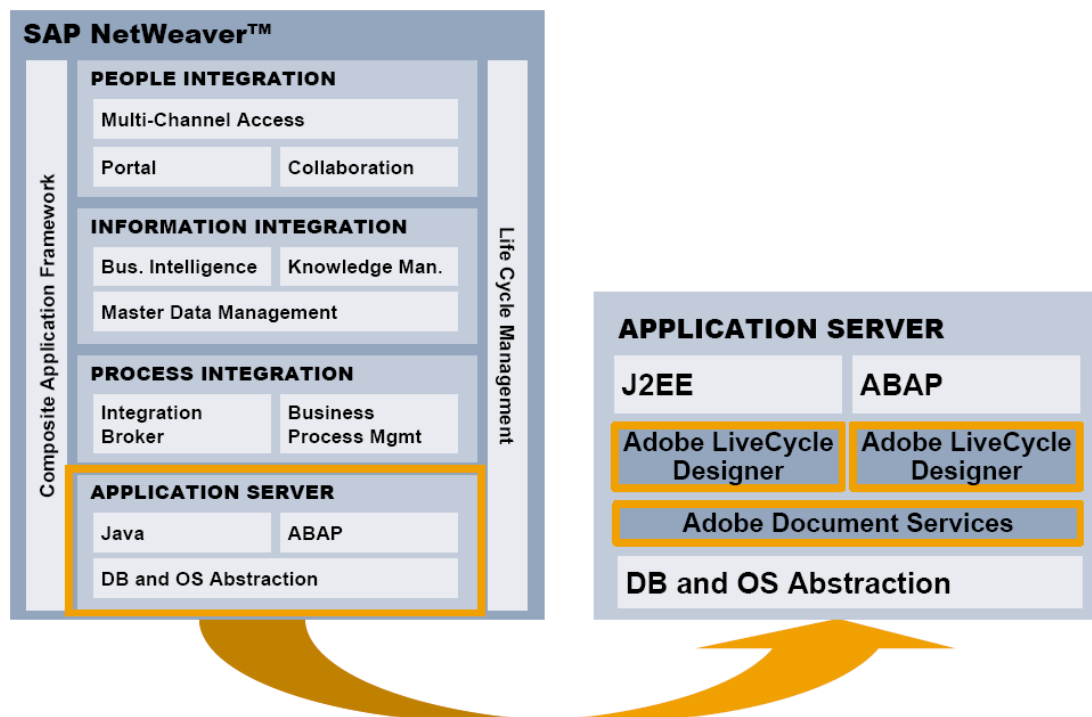


Figure 28 - SAP NetWeaver 7.0: structure¹²

Pour ce travail de diplôme, nous feront uniquement appel à la partie J2EE, c'est-à-dire Java, de l'Application Server.

¹² Source: PDF: SAP NetWeaver ABAP and Adobe Forms

5.2.1.2 Adobe Document Services (SP13)

Les Adobe Document Services, ou plus simplement ADS, correspondent à la partie du Web AS Java qui gère tout ce qui touche aux formulaires interactifs.

5.2.1.3 Adobe Credentials (Reader rights)

En disposant des Adobe Credentials (un simple fichier), les Adobe Document Services peuvent attacher en quelque sorte aux fichiers PDF qu'ils génèrent des droits supplémentaires qui autorisent l'utilisateur d'Acrobat Reader d'effectuer des actions autrement désactivées, telles que la saisie de données par exemple. Pour ce travail de diplôme, les Adobe Credentials ont été obtenu par Olivier Tosi, qui a effectué une demande selon la SAP Note 736902 et qui me les a ensuite envoyées par email.

5.2.1.4 SAP ECC 6.0 (IDES)

SAP ECC (ERP Central Component) correspond au cœur de l'architecture SAP. La grande différence par rapport à la version R/3 4.6c est qu'elle repose à présent sur la couche NetWeaver, permettant ainsi d'obtenir une ouverture plus grande sur le système.

De plus il s'agit ici d'une version IDES (Internet Demo and Evaluation System) ce qui signifie que le système est pré-rempli avec des données fictives. Cela permet d'effectuer des tests et des démonstrations.

Dans ce travail de diplôme, nous accéderons à ces données via le SAP Logon et les BAPIs.

5.2.2 Machine de développement & client

5.2.2.1 SAP NetWeaver Developer Studio 7.0.09 (développement)

C'est l'outil qui permet de développer des applications en Java et qui pourront ensuite être déployées sur la D07. Il s'agit d'Eclipse (www.eclipse.org) configuré pour être utilisé avec SAP NetWeaver.

Le SDK (Software Development Kit) java utilisé est la version 1.4.2_16 (disponible sur le site de Sun: <http://java.sun.com/j2se/1.4.2/download.html>)

5.2.2.2 Avec Adobe LiveCycle Designer 8.0 installé (développement)

Indispensable pour créer et manipuler des formulaires interactifs, Adobe LiveCycle Designer se télécharge depuis la SAP Marketplace. En effet, la version proposée sur ce site est prévue pour s'intégrer à l'environnement de développement SAP NetWeaver Developer Studio, comme le montre la capture d'écran ci-dessous:

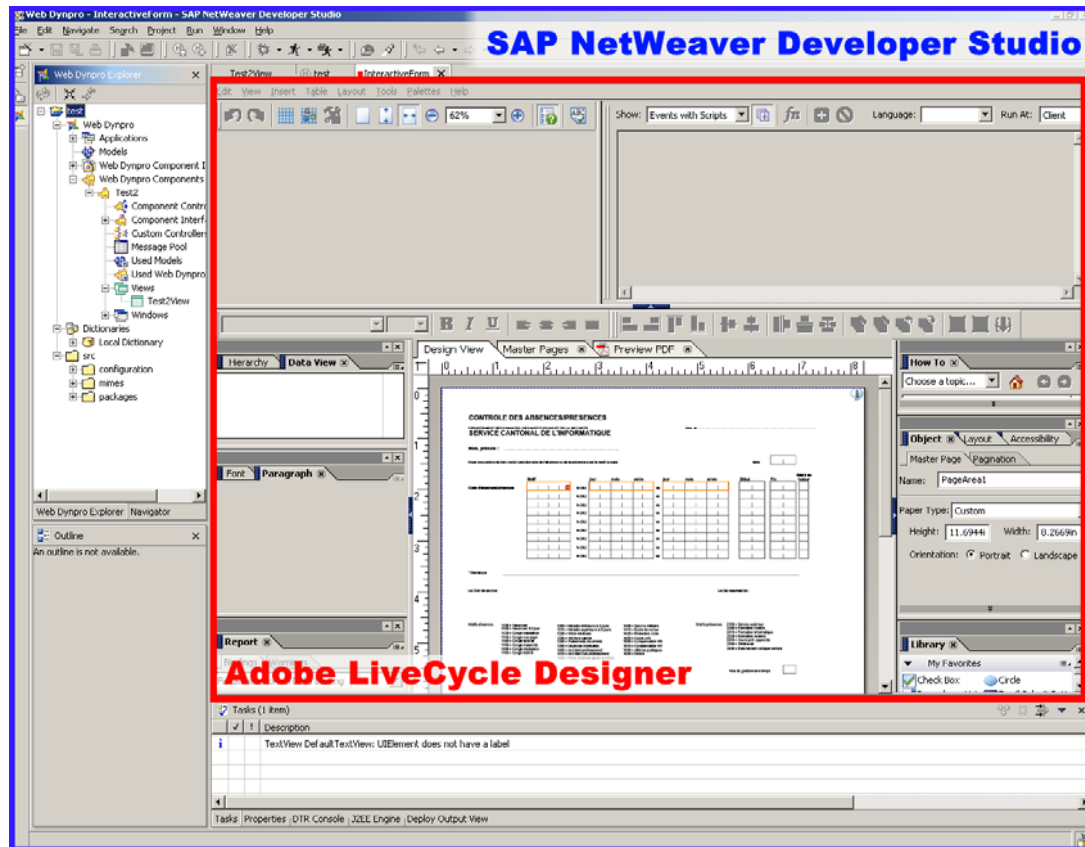


Figure 29 - SAP NetWeaver Developer Studio avec Adobe LiveCycle Designer lancé

5.2.2.3 Acrobat Reader 8.1.2 (client)

Il s'agit du lecteur de fichiers PDF d'Adobe. Il se télécharge depuis leur site (www.adobe.com) et son utilisation est entièrement gratuite. Dans le cadre de ce travail de diplôme, c'est lui qui permettra d'effectuer diverses actions sur les formulaires interactifs créés, tels que la sauvegarde, l'impression, la saisie de données ou tout simplement l'affichage.

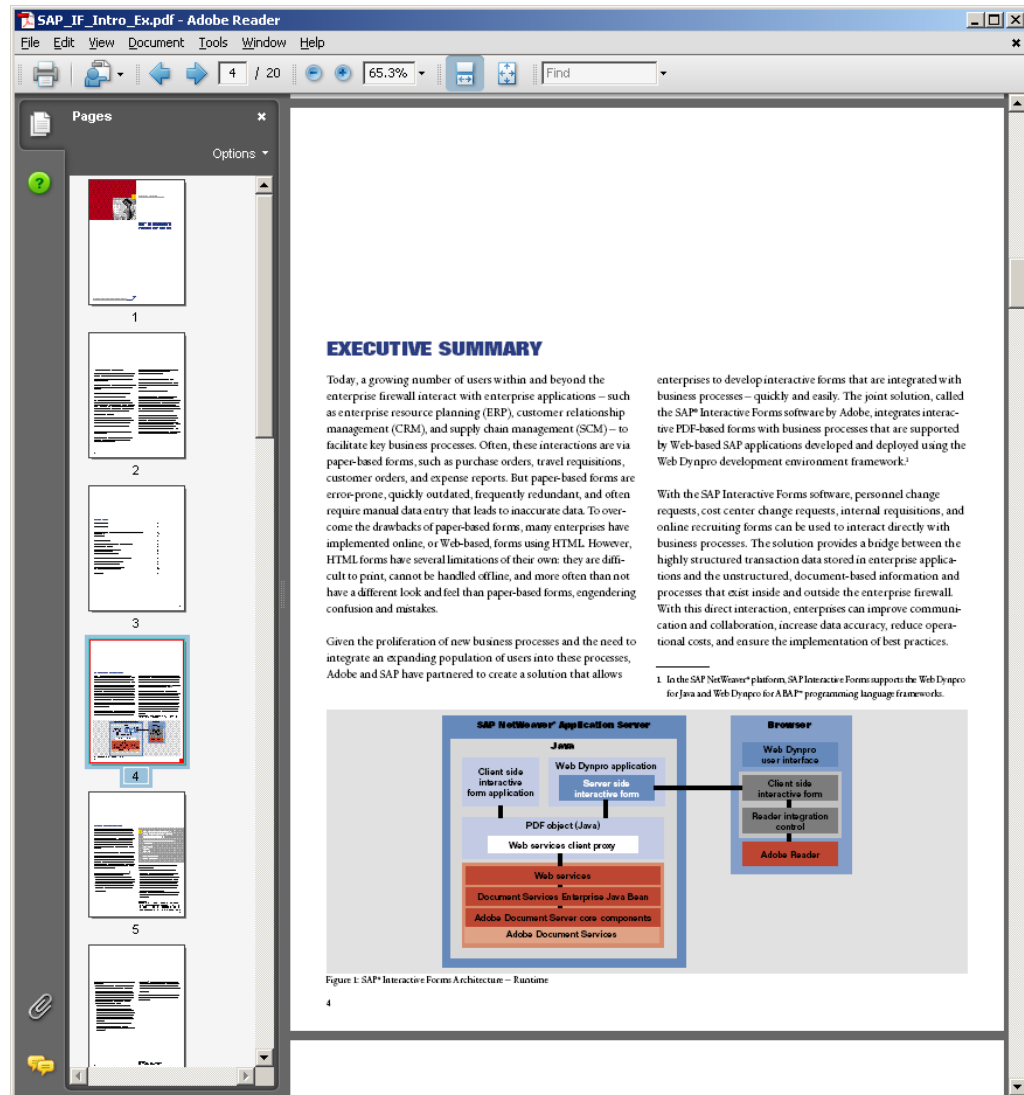


Figure 30 - Adobe Reader: exemple

5.2.2.4 SAP Logon 7.10 (client + développement)

C'est le logiciel qui permet de se connecter à un système SAP. La version utilisée est la 7.10 et elle a été téléchargée à partir du serveur FTP du centre de compétence SAP (<ftp://ccsap.hevs.ch>). Afin d'éviter de devoir configurer manuellement les divers paramètres de connexion aux différents systèmes, il est possible de télécharger puis de copier dans le répertoire C:\WINDOWS le fichier saplogon.ini. En effet, ce fichier contient déjà toutes les informations de connexion nécessaires. Tout comme SAP Logon, il est disponible sur le serveur FTP du centre de compétence SAP.

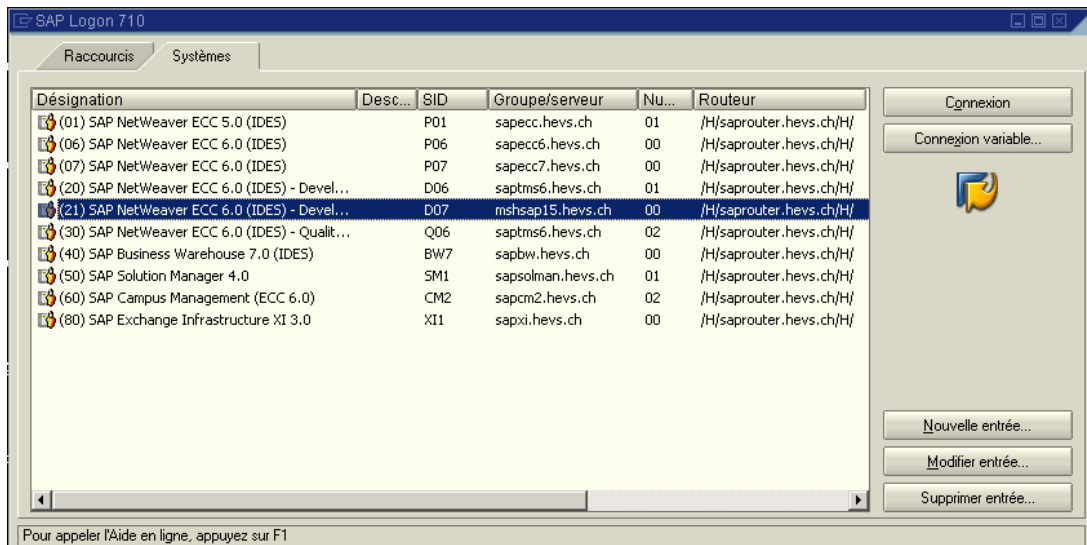


Figure 31 - SAP Logon 7.10

5.2.2.5 Internet Explorer 6+ (client + développement)

Le développement des formulaires est systématiquement testé avec la version 6 du navigateur de Microsoft. Cependant, quelques tests ont été effectués avec la version 7 et à priori il n'y a pas de problème de compatibilité. Pour ce qui est de son principal concurrent, Firefox, il faut que le mode d'affichage du formulaire interactif soit paramétré en "Native" et non en "ActiveX" (voir ACF).

5.2.2.6 Active Component Framework (ACF) (client + développement)

L'ACF est ce qui permet d'interagir avec un formulaire interactif à partir d'Internet Explorer. Il s'agit d'un contrôle ActiveX que le navigateur vous propose d'installer lorsque vous affichez pour la première fois un formulaire interactif. Cela peut ressembler à l'image ci-dessous :

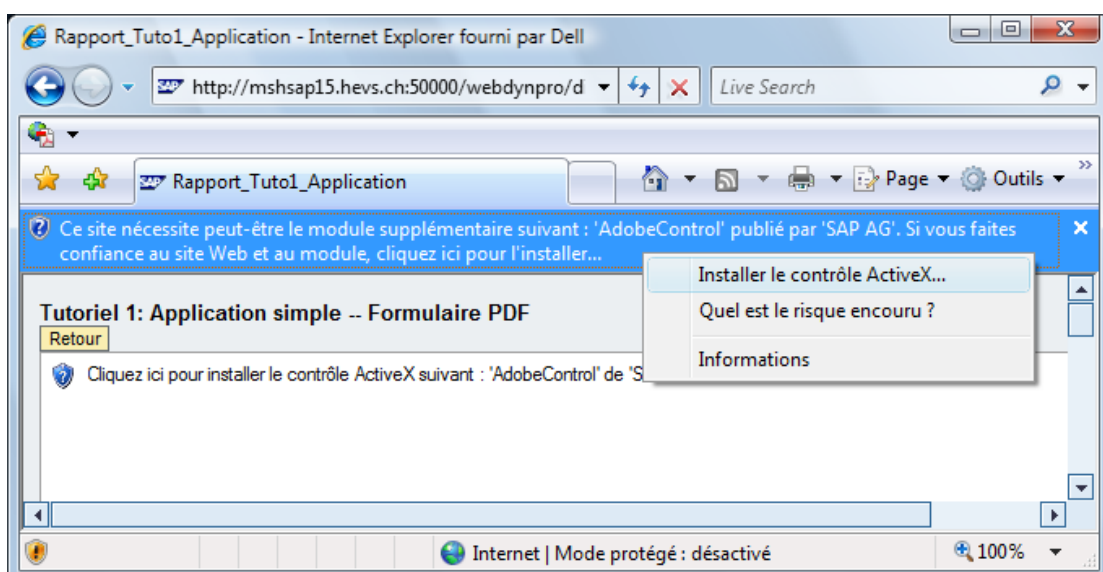


Figure 32 - Active Component Framework: message IE

Cependant, il faut savoir que l'installation de cet élément n'est plus indispensable. En effet, il est à présent possible d'afficher un formulaire en mode "Native", par opposition au mode "ActiveX", ce qui permet, en utilisant uniquement le langage JavaScript, d'éviter de tomber sur le message indiqué sur l'image ci-dessus. Lors de la création d'un formulaire dans SAP NetWeaver Developer Studio, il suffit donc simplement de changer la propriété "displayType" du formulaire et d'utiliser des éléments graphiques (liste déroulante, bouton d'envoi par exemple) libellés « Native ».

L'image ci-dessous correspond à la propriété du formulaire :

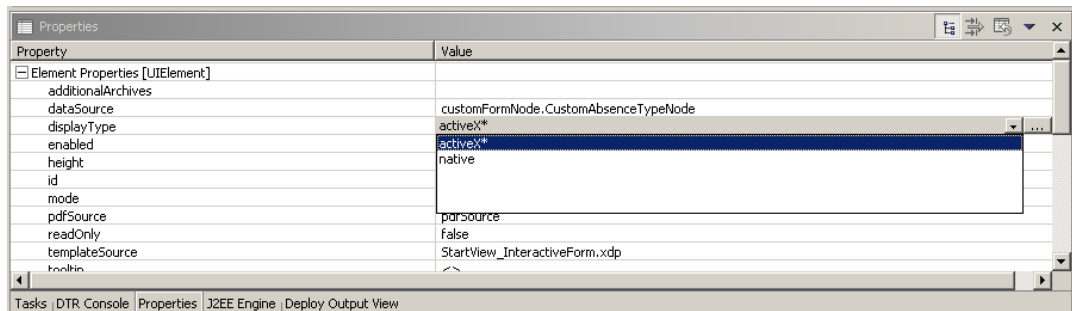


Figure 33 – ACF: s'en passer en changeant le mode d'affichage

Quant à cette image, elle indique quels sont les éléments disponibles dans l'Adobe LiveCycle Designer et propres à ce mode:

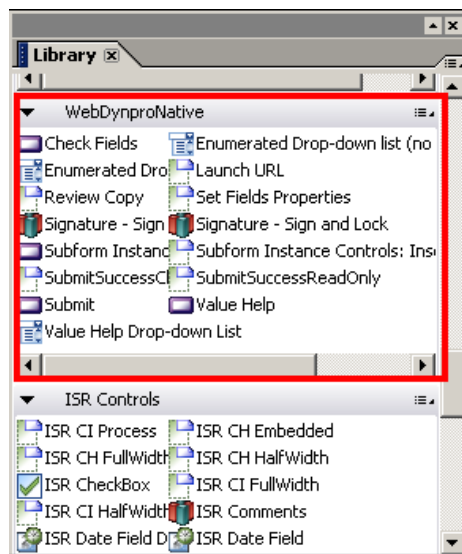


Figure 34 - Mode natif: éléments pour le mode natif

5.2.3 Problèmes rencontrés liés à l'installation

5.2.3.1 D07

L'installation de la D07 a posé quelques problèmes. En effet, à la base l'ADS n'arrivait pas à générer des formulaires interactifs. Lors d'un déploiement, la D07 présentait 2 symptômes:

- Une erreur causée par un exécutable nommé "XMLform.exe" lors du premier déploiement qui suivait le redémarrage du système.
- Une erreur de "Read Time Out" pour tous les déploiements suivants.

Après avoir passé un certain temps à en chercher la cause, il s'est avéré que le problème venait du fait que la D07 utilisait un système d'exploitation 64 bits et que la version de l'ADS installée à ce moment-là comportait des éléments tout simplement incompatibles, comme par exemple l'exécutable XMLform.exe. D'après un message sur les forums SDN, la solution consistait tout simplement à mettre à jour le système et à installer au minimum le Support Package Stack 11 Patch 1 (SPS 11 Patch 1).

Avec l'aide d'Ivan Schnyder, nous avons donc entrepris les démarches pour mettre à jour la D07. Nous avons vu comment sélectionner les Support Package sur le SAP Service Marketplace, comment valider certains téléchargements via le Solution Manager et enfin comment importer les différents Packages sur la D07 via le code transaction SPAM. Tout se passait bien jusqu'à ce que l'importation soit bloquée par une erreur que personne n'a semblé être en mesure de corriger.

Finalement, il a été décidé qu'il valait mieux restaurer un backup de la D07 et de laisser de côté l'installation de certains composants inutiles. Durant cette étape, je n'ai joué aucun rôle si ce n'est de terminer la mise à jour de la partie Java via le JSPM (Java Support Package Manager).

Une fois la mise à jour terminée, un premier formulaire basique a pu être déployé avec succès. La version actuellement installée est la 13.

System D07			
Software Components			
Vendor	Name	Version	Provider
sap.com	SAP-JEECOR	7.00 SP13 (1000.7.00.13.0.20070907082334)	SAP AG
sap.com	BI_MMR	7.00 SP13 (1000.7.00.13.0.20070812013749)	SAP AG
sap.com	CAF-UM	7.00 SP13 (1000.7.00.13.0.20070809092324)	SAP AG
sap.com	KM-KW_JIKS	7.00 SP13 (1000.7.00.13.0.20070812014519)	SAP AG
sap.com	SAP_JTECHF	7.00 SP13 (1000.7.00.13.0.20070812015644)	SAP AG
sap.com	SAP-JEE	7.00 SP13 (1000.7.00.13.0.20070812015311)	SAP AG
sap.com	ADSSAP	7.00 SP13 (1000.7.00.13.0.20070812011854)	SAP AG
sap.com	EPBC	7.00 SP13 (1000.7.00.13.0.20070812123744)	SAP AG

Figure 35 - Version des composants SAP-JEECOR, SAP-JEE, ADSSAP

5.2.3.2 Client & Développement

Pour la machine WHK37006, le principal problème a été l'installation de l'Adobe LiveCycle Designer. Je ne savais pas:

- Quelle version installer
- Comment l'intégrer à Web Dynpro

Etrangement, cette information n'était pas si facile à trouver. Finalement, il s'est avéré qu'il existait deux versions disponibles depuis le SAP Service Marketplace: la 7.1 et 8.0. La particularité de ces versions est qu'elles sont faites pour être installées avec SAP NetWeaver Developer Studio.

Un nouveau problème est survenu après ça: mon compte sur le SAP Marketplace ne disposait pas des droits suffisants pour pouvoir télécharger ces fichiers. Après avoir contacté le centre de compétence SAP, ces droits m'ont été donnés.

Enfin, un 3^{ème} problème est survenu après l'installation de la version 7.1 d'Adobe LiveCycle Designer. En effet, dès que je lançais l'éditeur dans SAP NetWeaver Developer Studio, la machine ralentissait de plus en plus et la charge mémoire montait en flèche, ce qui l'éditeur tout simplement inutilisable. J'ai donc désinstallé la version 7.1 et l'ai remplacée par la version 8.0. Après ça, plus aucun problème n'est survenu.

6 Configuration

6.1 Remarque

La partie suivante détaille les étapes de configuration qui ont été effectuées sur le système D07 après que les Support Packages nécessaires aient été installés. Elle est essentiellement basée sur le guide de configuration de l'AdobeDocumentServices. Avant de suivre les étapes ci-dessous, je recommande donc de suivre le guide au moins une fois pour augmenter les chances que les éventuels problèmes rencontrés soient similaires aux miens.

6.2 Pré-requis

Pour réaliser cette configuration il faut:

- Un compte sur le mandant 100 de la D07 avec le rôle "SAP_J2EE_ADMIN" et le profil "SAP_ALL".
- Les identifiants nécessaires pour une connexion via bureau à distance à la D07
- Les Adobe Reader Rights Credentials (plus d'informations dans la partie les concernant)

6.3 Configuration sur la D07

Les opérations qui suivent ont été réalisées via une connexion de bureau à distance depuis le pc de développement (WHK37006) vers la D07 (adresse: mshsap15.hevs.ch)

6.3.1 Ajout d'une ligne dans les propriétés de démarrage Java pour l'ADS

Cette propriété, indiquée dans le guide de configuration, a du être rajoutée. Voici la marche à suivre:

- Lancer le Configuration Tool pour WebAS Java:
(Chemin: `E:\usr\sap\D07\DVEBMGS00\j2ee\configtool\configtool.bat`)
- Sélectionner Cluster data → instance_ID79779 → server_ID79779
- L'écran qui suit apparaît:

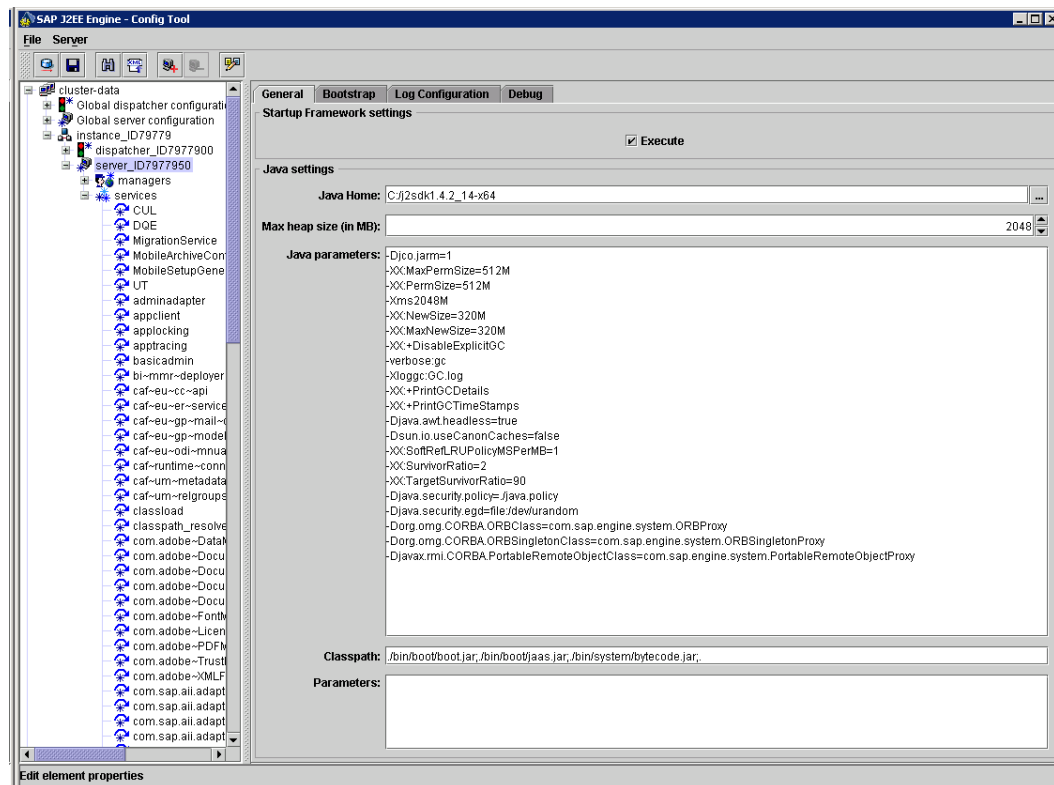


Figure 36 - Configuration Tool: paramètres Java

- Dans la partie "Java parameters" à droite, ajouter la ligne suivante:

-Dorg.omg.PortableInterceptor.ORBInitializerClass=com.sap.engine.services.ts.jts.ots.PortableInterceptor.JTSInitializer

Cela donne:

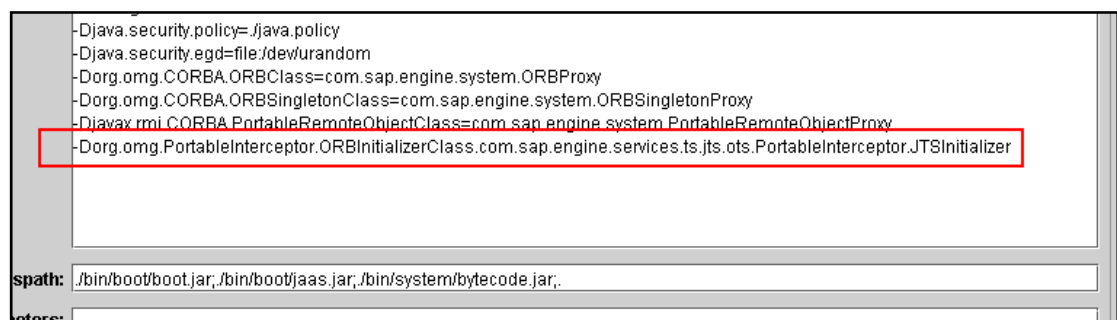


Figure 37 - Configuration Tool: ligne ajoutée

- Sauvegarder.
- A partir de la SAP Management Console, relancer le système D07. Tout d'abord il faut l'arrêter comme ceci:

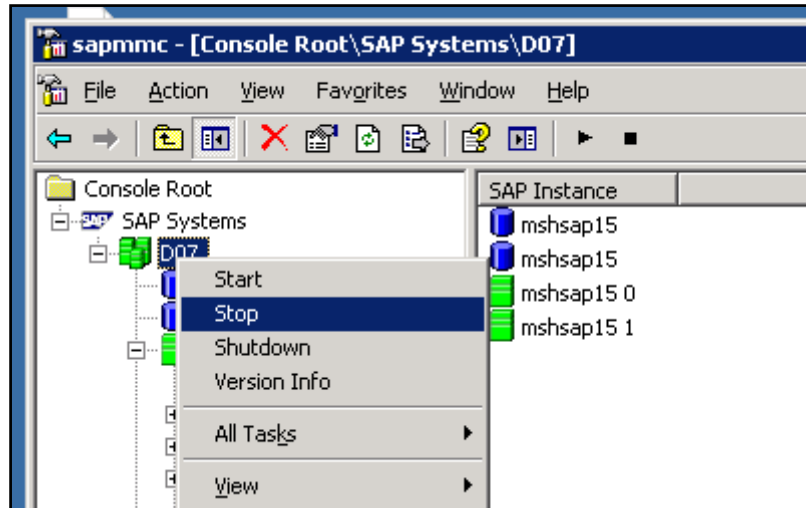


Figure 38 - SAP Management Console: arrêt D07

- Une fois l'icône du système D07 grisée, le relancer comme ceci:

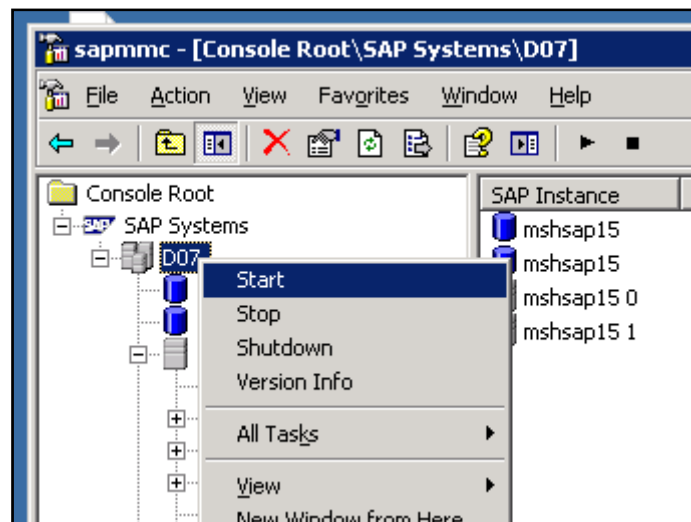


Figure 39 - SAP Management Console: démarrage D07

- Attendre ensuite que l'icône du système D07 soit verte avant de continuer. Cela peut prendre un certain temps (un dizaine de minutes dans mon cas).

6.4 Vérifier le web service AdobeDocumentServices

6.4.1 Remarque

Cette étape se fait à partir de la machine de développement. Nous allons nous assurer que le web service AdobeDocumentServices fonctionne correctement et, en cas de problème, proposer les étapes d'une solution.

6.4.2 Test de l'AdobeDocumentServices

Tout d'abord, on accède à la page concernant l'AdobeDocumentServices dans le Web Service Navigator. L'adresse est la suivante:

<http://mshsap15.hevs.ch:50000/AdobeDocumentServices/Config>

La page suivante apparaît:

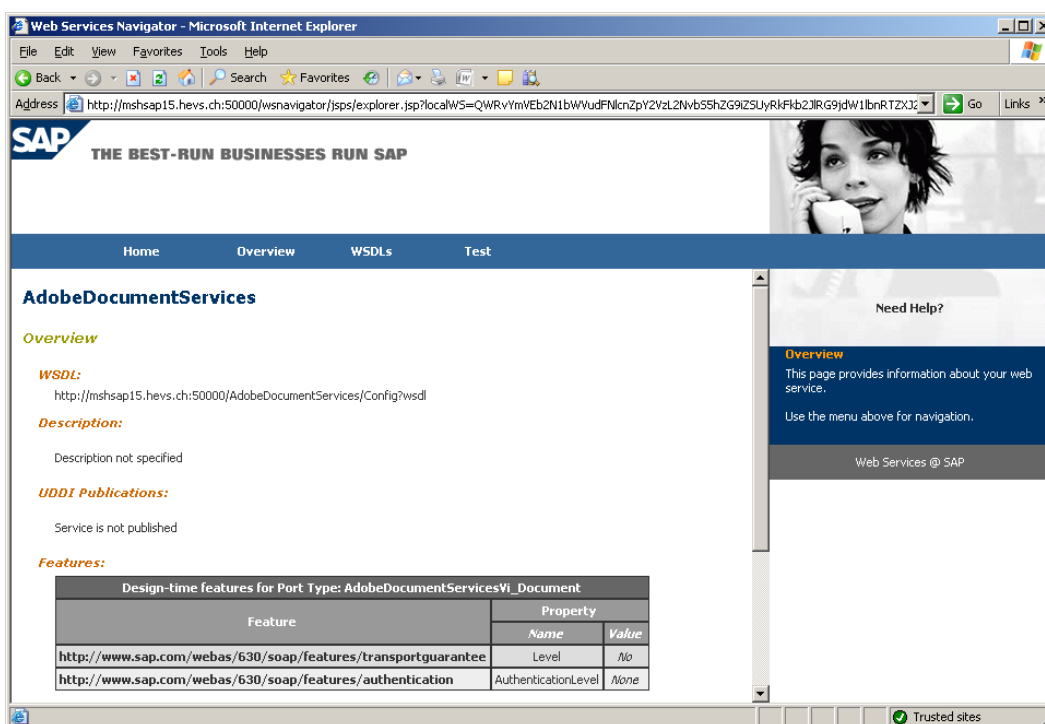


Figure 40 - AdobeDocumentServices: Page d'accueil

- On clique sur Test puis sur l'opération rpData proposée sur la page suivante:

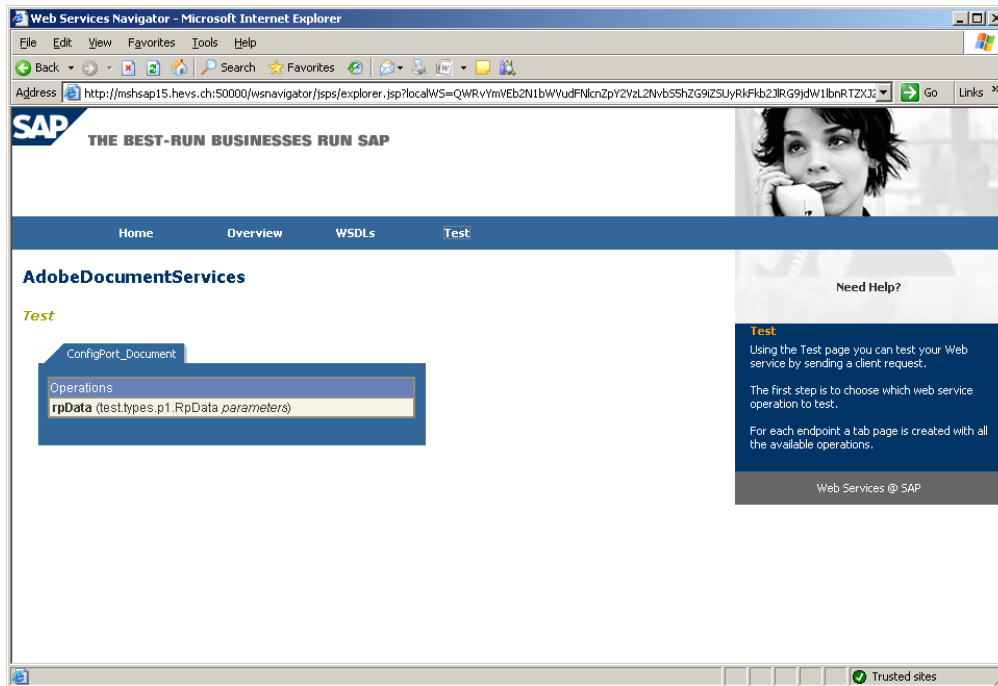


Figure 41 - AdobeDocumentServices: test de l'opération rpData

- A l'écran suivant, il suffit de cliquer sur le bouton "Send". Pas besoin de saisir autre chose:

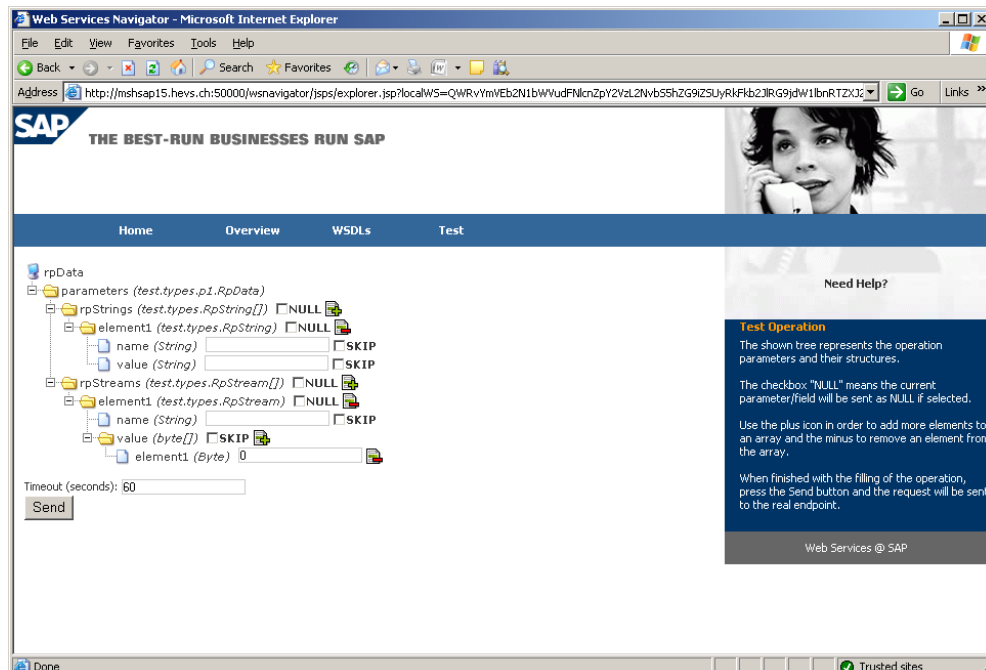


Figure 42 - AdobeDocumentServices: saisie des paramètres pour le test

- La fenêtre ci-dessous apparaît alors. Ici, il faut s'authentifier avec l'utilisateur ADSUser, puis cliquer sur Submit (le mot de passe correspond au mot de passe de l'administrateur):

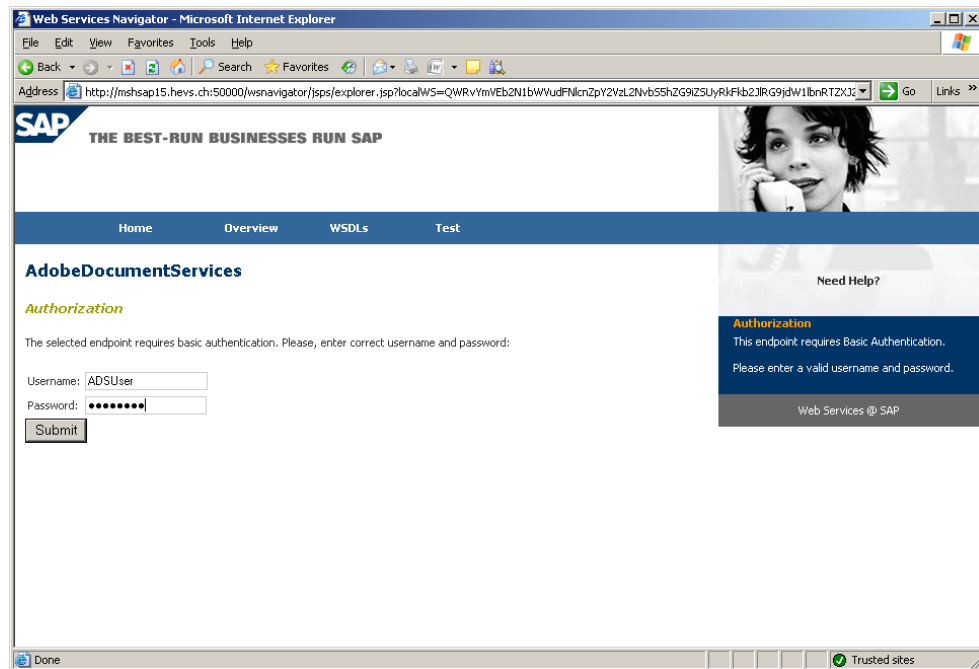


Figure 43 - AdobeDocumentServices: identification

- Si le test a fonctionné, l'écran suivant apparaît et nous pouvons passer à la partie sur le Reader Rights Credential.

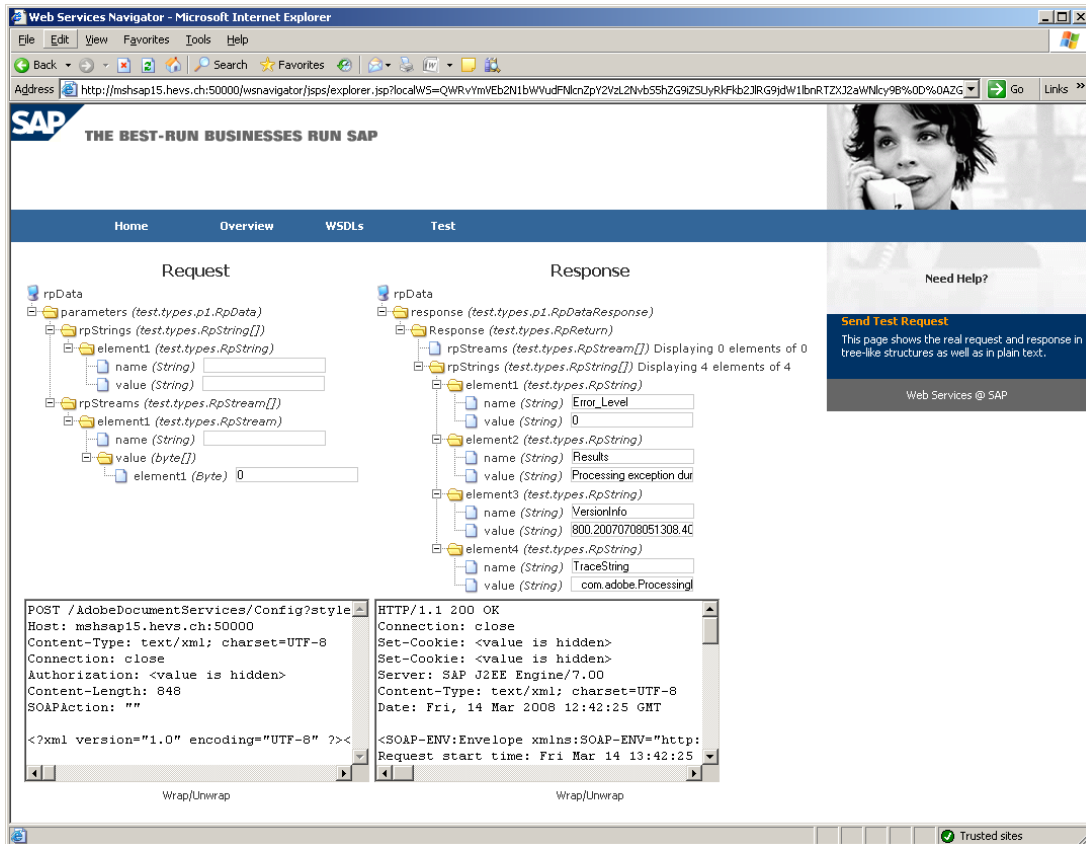


Figure 44 - AdobeDocumentServices: résultats du test

Si le fait de cliquer sur le bouton "Submit" ne produit rien à part vous ramener sur la page d'authentification, il y a un problème avec l'utilisateur ADSUser (mot de passe incorrect, rôle, etc..) et il faut donc suivre la solution décrite ci-dessous.

6.4.2.1 Définir ADSUSER en tant que Technical User, confirmation du mot de passe

- Tout d'abord, allez sur le portail J2EE de la D07. L'adresse est:
<http://mhsap15.hevs.ch:50000>
- Cliquez sur "User Management".

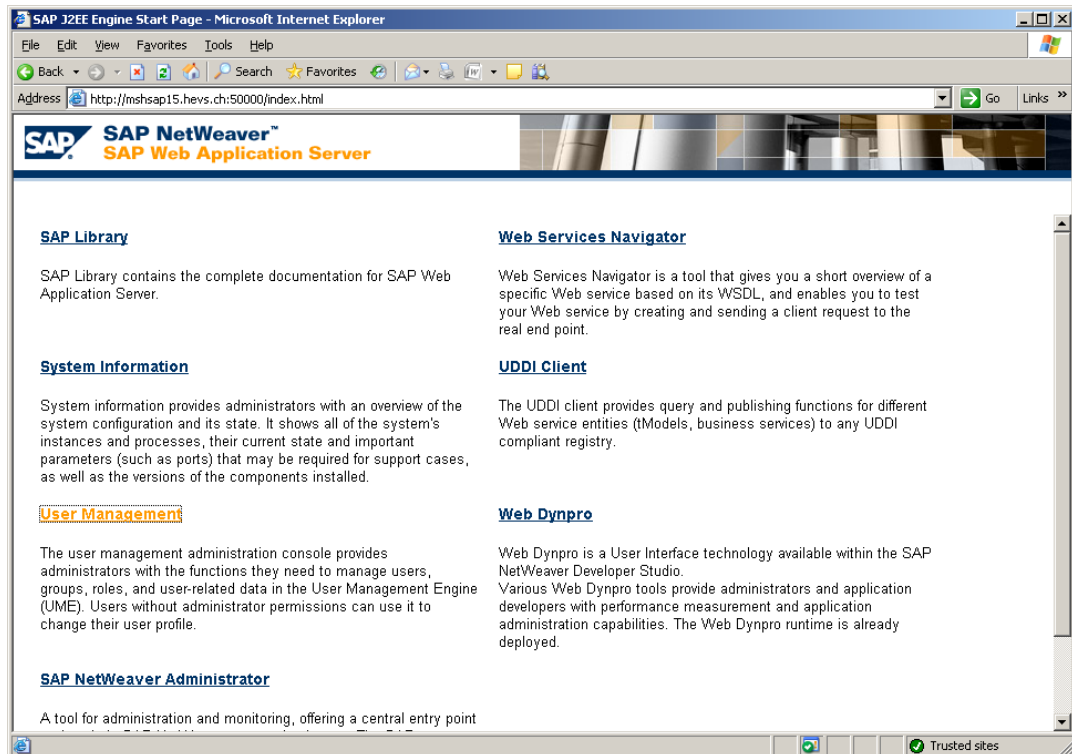


Figure 45 - SAP J2EE Engine Start Page

- Il faut alors s'identifier avec un compte disposant des droits nécessaires.



Figure 46 - User Management: identification

- On arrive ensuite sur la page suivante:

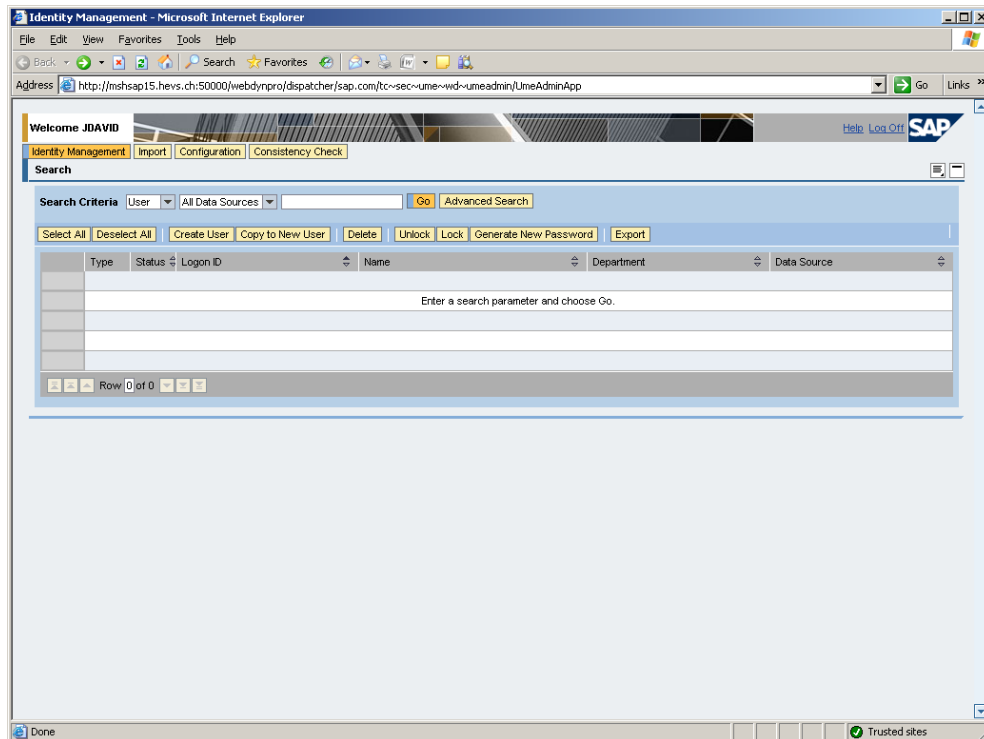


Figure 47 - User Management: écran principal

- Recherchez l'utilisateur ADSUser, affichez-le, puis cliquez sur le bouton Modify:

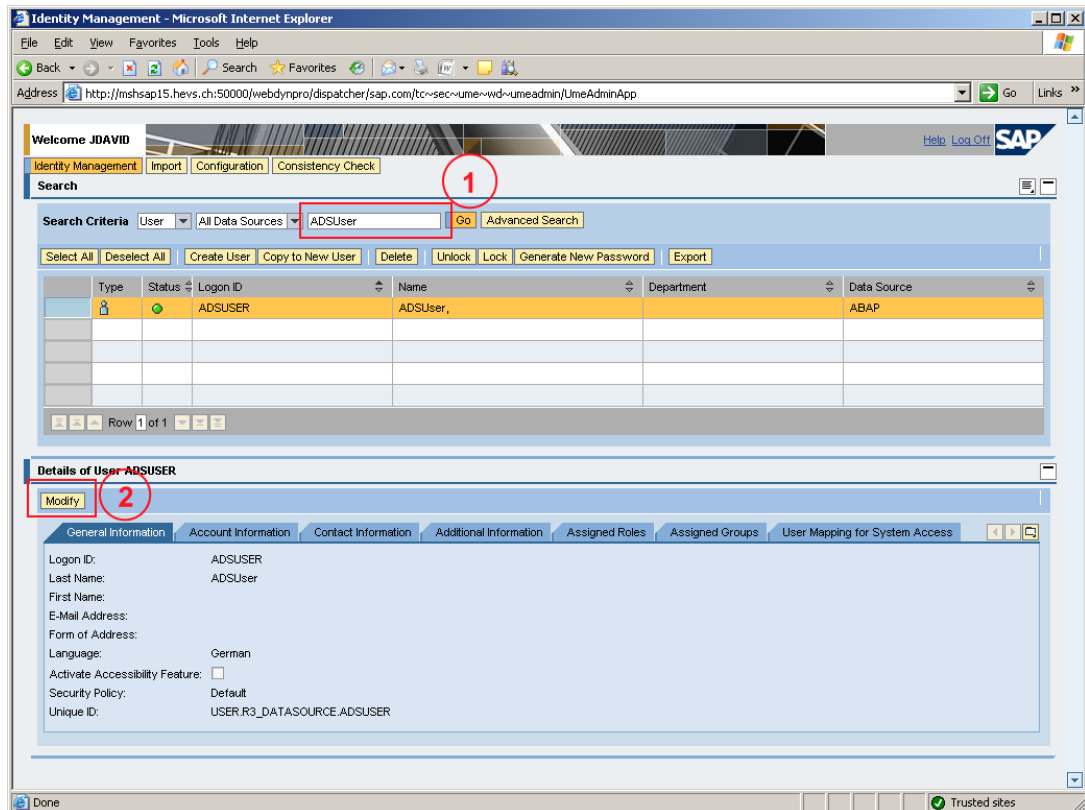


Figure 48 - User Management: recherche, modification

Il faut alors effectuer 2 réglages:

- Ressaisir le mot de passe prévu par les administrateurs. En effet, il se pourrait que le mot de passe actuel ait une faute de frappe.
- Sous "Security Policy", sélectionner "Technical User"

Figure 49 - User Management: modification de ADSUSER

- Une fois ces 2 réglages effectués, cliquez sur le bouton "Save".

Nous pouvons ensuite retourner sur la page du web service AdobeDocumentServices et effectuer un nouveau test.

Pour ce travail de diplôme, les étapes décrites ci-dessus ont suffi pour que le test fonctionne. Dans le cas contraire, le guide d'installation de l'Adobe Document Services propose des réglages supplémentaires qui ne seront cependant pas abordés ici.

6.5 Installer le Reader Rights Credential

6.5.1 Remarque

Pour cette partie, nous retournons sur la D07 via notre connexion de bureau à distance.

6.5.2 Présentation

Cet élément est essentiel pour pouvoir déployer et utiliser des formulaires interactifs avec l'Adobe Reader. Pour l'obtenir, il faut envoyer un message à SAP selon la SAP Note 736902 et faire une demande de credential pour les scénarios d'Adobe Interactive Forms. Pour ce travail de diplôme, Olivier Tosi a effectué la demande et m'a ensuite transmis le credential.

Le credential se compose de 2 éléments indispensables:

- Un fichier .pfx (contenu dans un fichier .zip quand on le reçoit)
- Un mot de passe contenu dans le "Customer Message".

Une fois ces éléments en main, nous pouvons passer à l'installation du Reader Rights Credential (aussi appelé PKCS #12 Credential)

6.5.3 Procédure d'installation

Tout d'abord, sur la D07, copier le fichier .pfx dans le répertoire suivant:

E:\usr\sap\D07\SYS\global\AdobeDocumentServices\TrustManagerService\trust\credentials

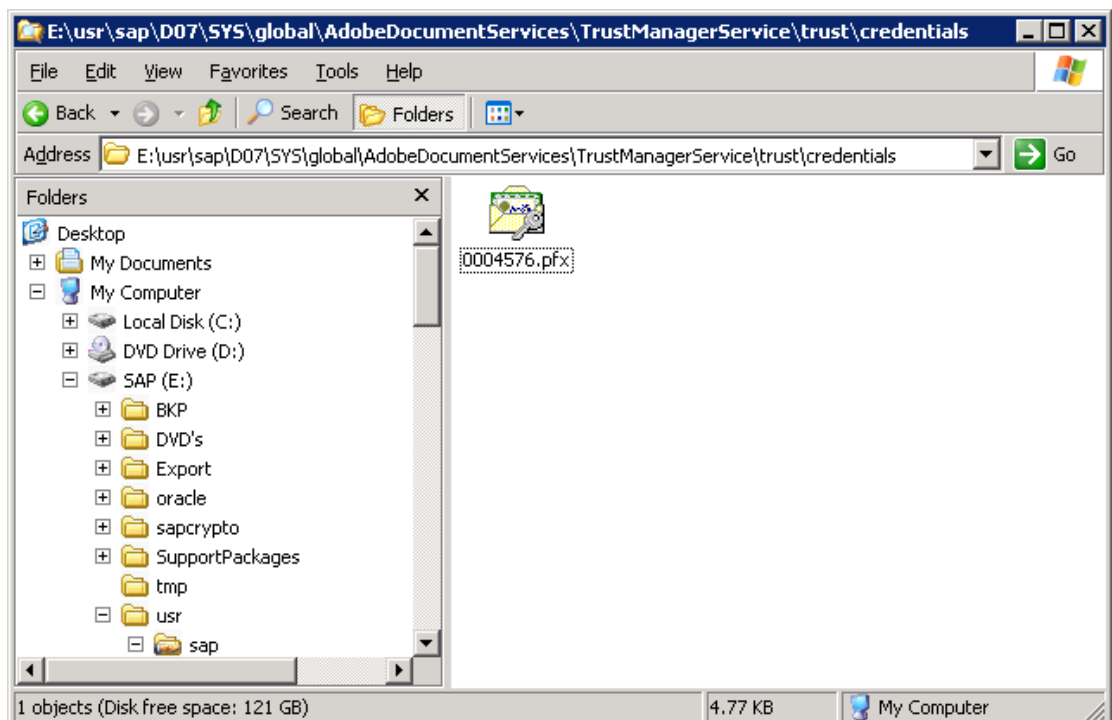


Figure 50 - Répertoire de destination du fichier .pfx

- A présent, il faut lancer le Visual Administrator en ouvrant le fichier suivant:

E:\usr\sap\D07\DVEBMGS00\j2ee\admin\go.bat

- Voici les informations de connexion:

- o User Name: J2EE_ADMIN
- o Host: mshsap15
- o Port: 50004

- Après votre authentification, sélectionnez dans l'onglet Cluster:

Server 0 0_79779 (mshsap15) → Services → Document Services Configuration

L'écran suivant apparaît alors:

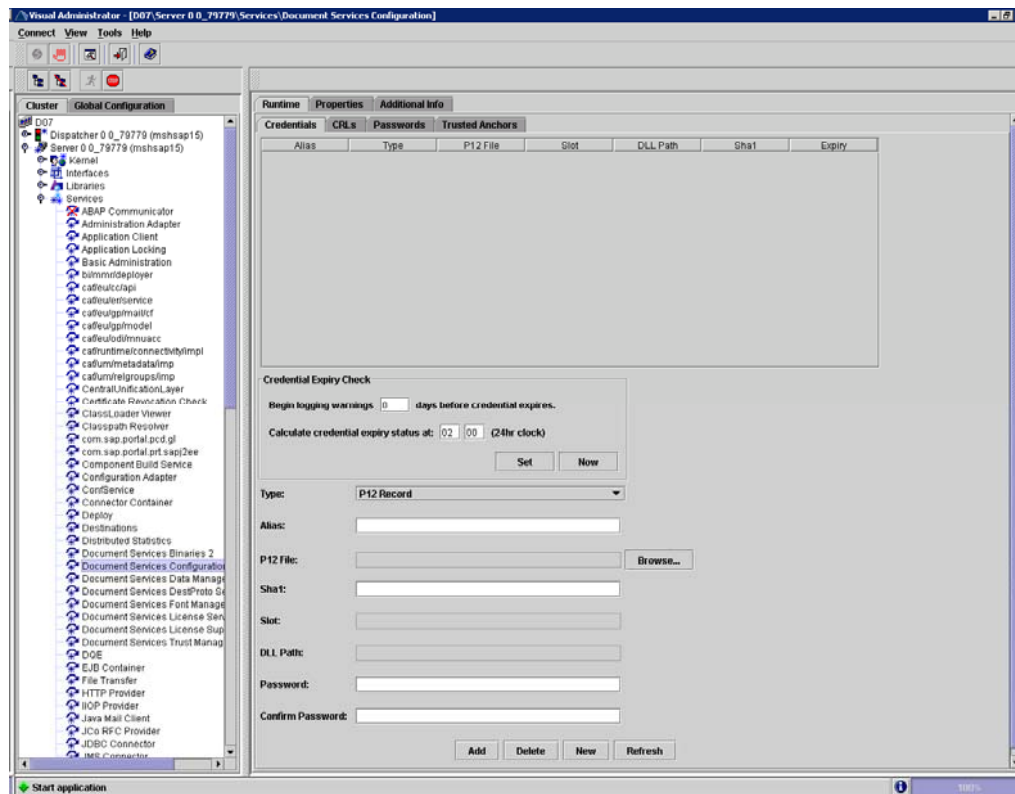


Figure 51 - Reader Rights Credential: Ecran de configuration

Il faut alors saisir plusieurs choses:

- Pour Type, sélectionner "P12 Record"
- Pour Alias, taper "ReaderRights" (sans les guillemets)
- Pour P12 File, cliquer sur Browse puis sélectionner le fichier .pfx qui vient d'être copié.
- Pour Sha1, laisser vide.
- Pour Password et Confirm Password, saisir le mot de passe contenu dans le "Customer Message".

Dans notre cas, cela donne la chose suivante:

Figure 52 - Reader Rights Credential: saisie des paramètres

- Une fois les informations saisies, cliquez sur le bouton "Add". Le nouveau credential apparaît alors dans la liste au dessus:

Alias	Type	P12 File	Slot	DLL Path	Sha1	Expiry
ReaderRights	P12	0004576.pfx				Jan 8, 2023

Figure 53 - Adobe Reader Right Credential: credential ajouté

- Pour terminer, il faut à nouveau relancer la D07 afin que les changements soient pris en compte.

La configuration du système est à présent terminée. Nous pouvons maintenant revenir sur la machine de développement et aborder la partie du travail de diplôme relative à l'implémentation.

6.6 Problèmes possibles

6.6.1 Déploiement d'un formulaire: erreur "401: Unauthorized")

Cette erreur peut venir du fait que l'ADSUser n'utilise pas le bon mot de passe ou n'a pas le bon rôle. Les étapes décrites au point 7.4.2.1 devraient corriger ce problème. Si ce

n'est pas le cas, vérifiez la configuration de l'authentification basique dans le Visual Administrator (cf. guide de configuration ADS, p.15)

6.6.2 Visual Administrator: la fenêtre Web Services Security ne s'affiche pas

En suivant les étapes du guide de configuration de l'ADS, on doit à un moment donné accéder à la fenêtre Web Services Security. Or, lorsque l'on clique sur son nom, la fenêtre ne s'affiche pas et la barre indiquant son chargement reste coincée à 1%.

Bien que la solution trouvée soit destinée à NetWeaver 2004s SP12 & SP20, l'application de la SAP Note 1066586 a corrigé le problème.

7 Développement

7.1 Introduction

Une fois l'installation terminée, nous pouvons commencer à développer des formulaires à l'aide du SAP NetWeaver Developer Studio. Ce point du rapport se divisera essentiellement en 4 parties. Chaque partie abordera les différents aspects théoriques qui seront ensuite repris par un tutoriel immédiatement après. De cette manière, il sera plus aisé de comprendre les différentes étapes du développement.

Les quatre tutoriels sont basés sur les différents tutoriels existants sur le SDN ainsi que sur toute une panoplie de messages postés sur ses forums.

7.2 Bases du développement

Nous allons à présent aborder les différents éléments auxquels nous ferons appel durant le tutoriel qui conclura cette partie.

7.2.1 Utilisation de la perspective Webdynpro

Commençons par ouvrir l'outil de développement SAP NetWeaver Developer Studio. La première chose à faire est de s'assurer que l'on s'apprête à développer en utilisant la bonne perspective. En effet, bien qu'Eclipse en propose plusieurs, la seule qui nous intéresse est la "Web Dynpro Perspective". Il est possible de la sélectionner comme indiqué sur la capture d'écran ci-dessous:

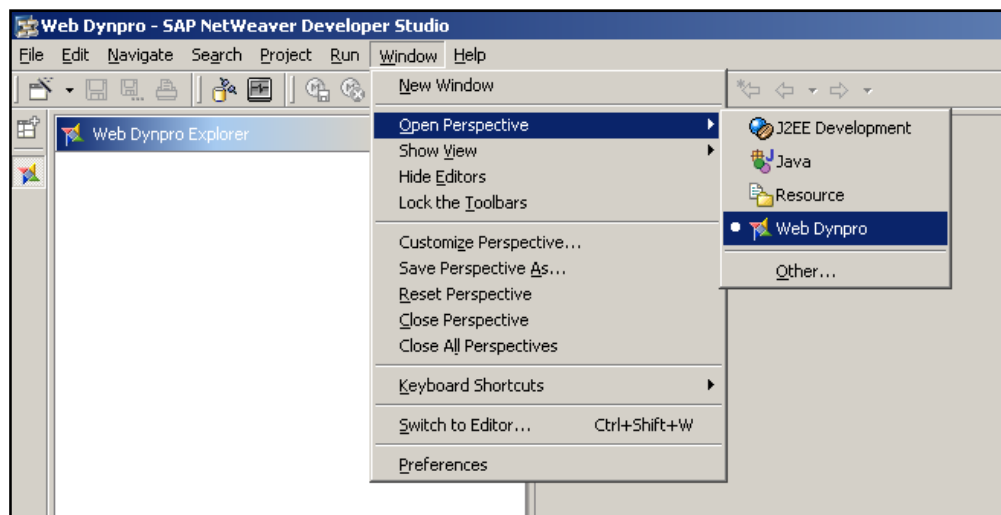



Figure 54 - SAP NetWeaver Developer Studio: changement de perspective

A noter qu'il est aussi possible de la sélectionner en cliquant sur l'icône  sur la gauche.

7.2.2 Réglages de SAP NetWeaver Developer Studio

Etant donné que les applications sont déployées sur le serveur, il faut aussi s'assurer que l'adresse de ce dernier soit saisie correctement. Pour ce faire, il faut se rendre dans les préférences et indiquer l'adresse et le port du serveur J2EE:

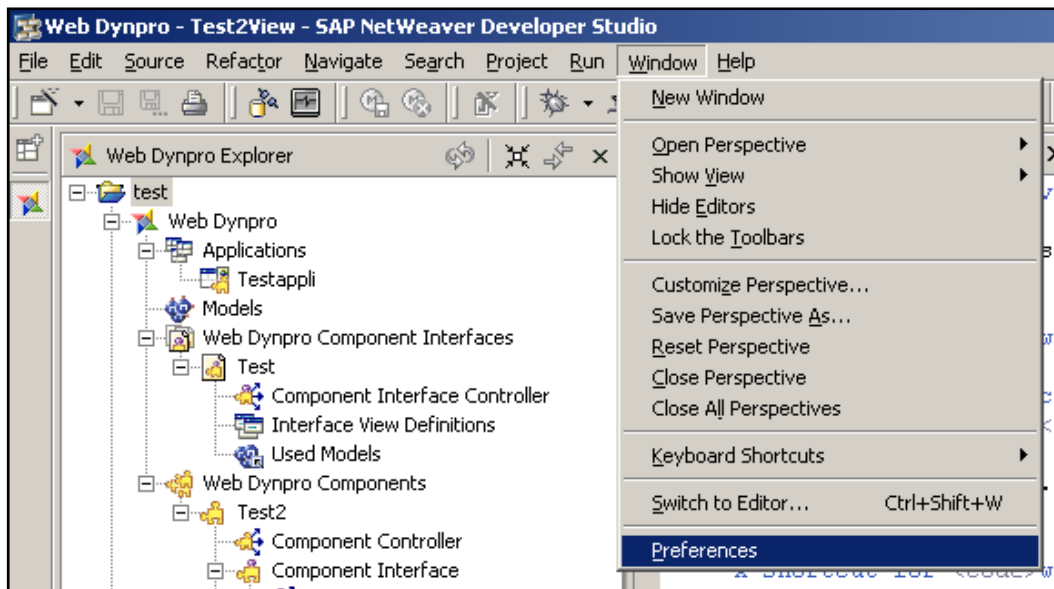


Figure 55 - SAP NetWeaver Developer Studio: accès aux préférences

Dans notre cas, les informations sont les suivantes:

- Adresse: mshsap15.hevs.ch
- Port: 3901

On saisit ces informations comme indiqué ci-dessous et on valide.

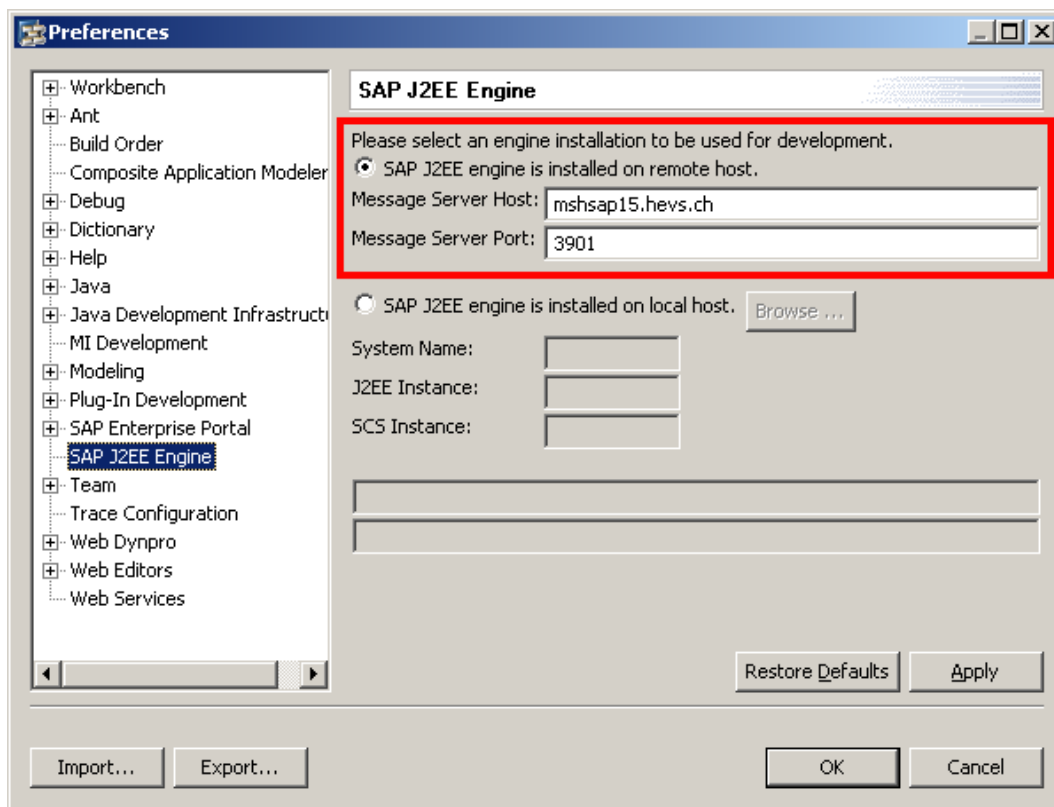


Figure 56 - SAP NetWeaver Developer Studio: paramètres J2EE Engine

7.2.3 Interface de SAP NetWeaver Developer Studio

L'interface ressemble typiquement à cela quand on développe une application Web Dynpro, on remarque qu'elle se divise en 4 parties distinctes:

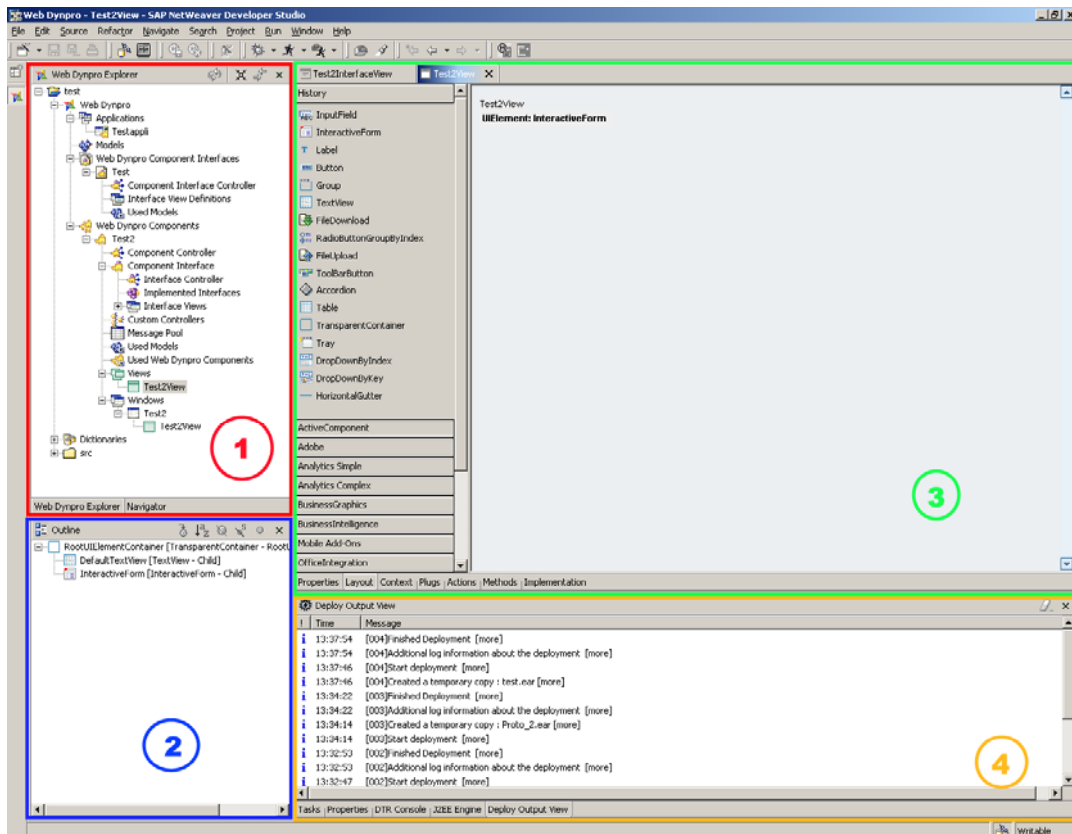


Figure 57 - SAP NetWeaver Developer Studio: interface

7.2.3.1 Web Dynpro Explorer (1)

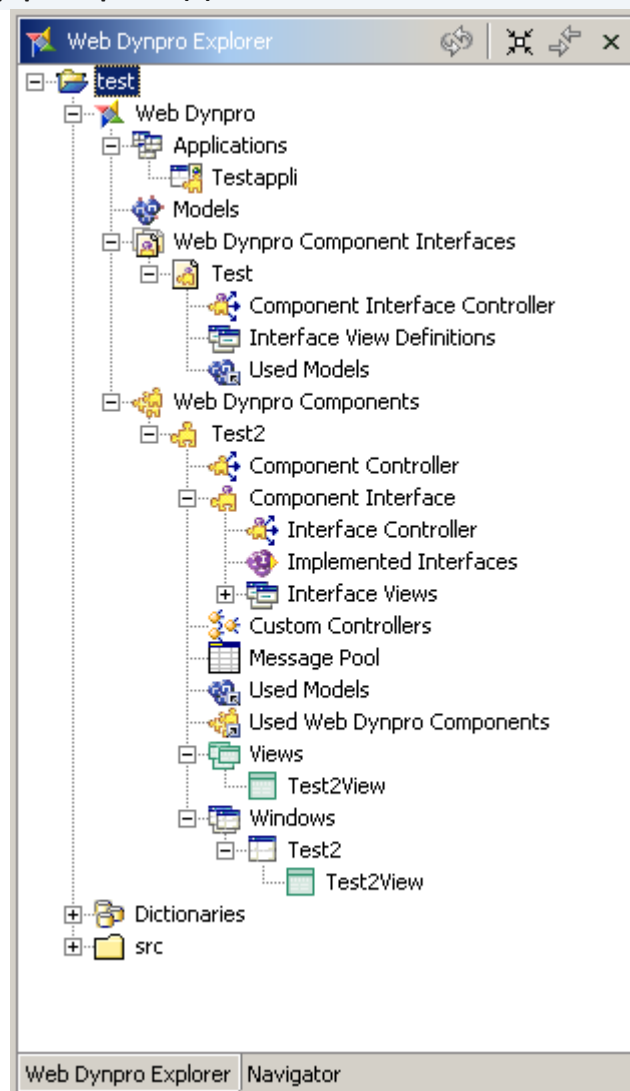


Figure 58 - SAP NetWeaver Developer Studio: Web Dynpro Explorer

Cette fenêtre permet d'accéder aux différents éléments qui composent la structure du projet. On y retrouve notamment les éléments suivants:

- **Applications:** applications que l'on peut déployer
- **Models:** utilisés lors de l'utilisation de RFC/BAPIs.
- **Web Dynpro Components:** il s'agit des différents composants d'une application.
- **Views, ou vues :** correspond à ce que l'on voit à l'écran lorsque l'application est déployée.

7.2.3.2 Outline (2)

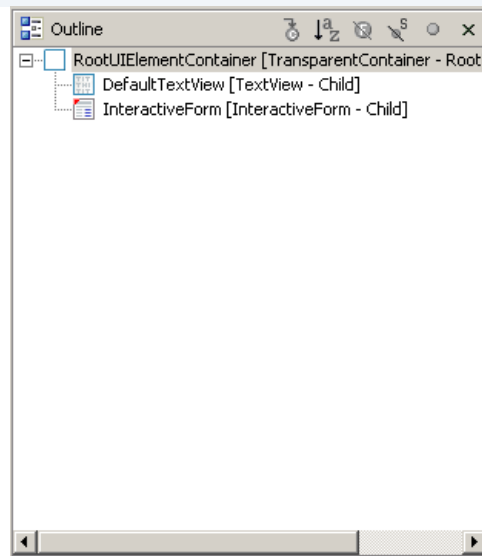


Figure 59 - SAP NetWeaver Developer Studio: outline

Cette fenêtre permet de sélectionner les différents éléments relatifs à la partie de l'application sur laquelle on travaille. Par exemple, lorsque l'on développe l'interface graphique, il permet de gérer les champs de texte, les boutons, les listes déroulantes, etc. Lorsqu'on travaille sur le code, il permet d'accéder rapidement aux méthodes par exemple.

7.2.3.3 Fenêtre de travail (3)

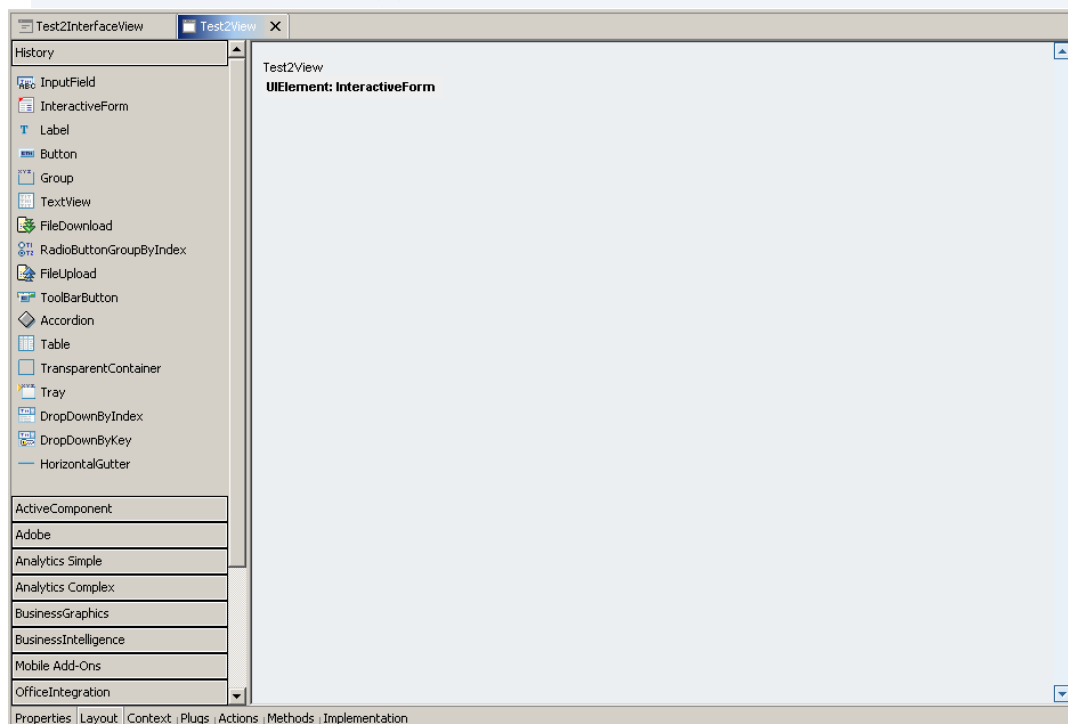


Figure 60 - SAP NetWeaver Developer Studio: fenêtre de travail

La fenêtre de travail permet de gérer l'élément de l'application sur lequel on travaille. Cela peut être, comme le montre la capture d'écran ci-dessus, une "View", c'est-à-dire ce qu'on verra à l'écran une fois l'application déployée (champs, boutons, tableaux, listes déroulantes, etc..), ou alors le code qui se cache derrière cette vue, comme le représente l'image ci-dessous.

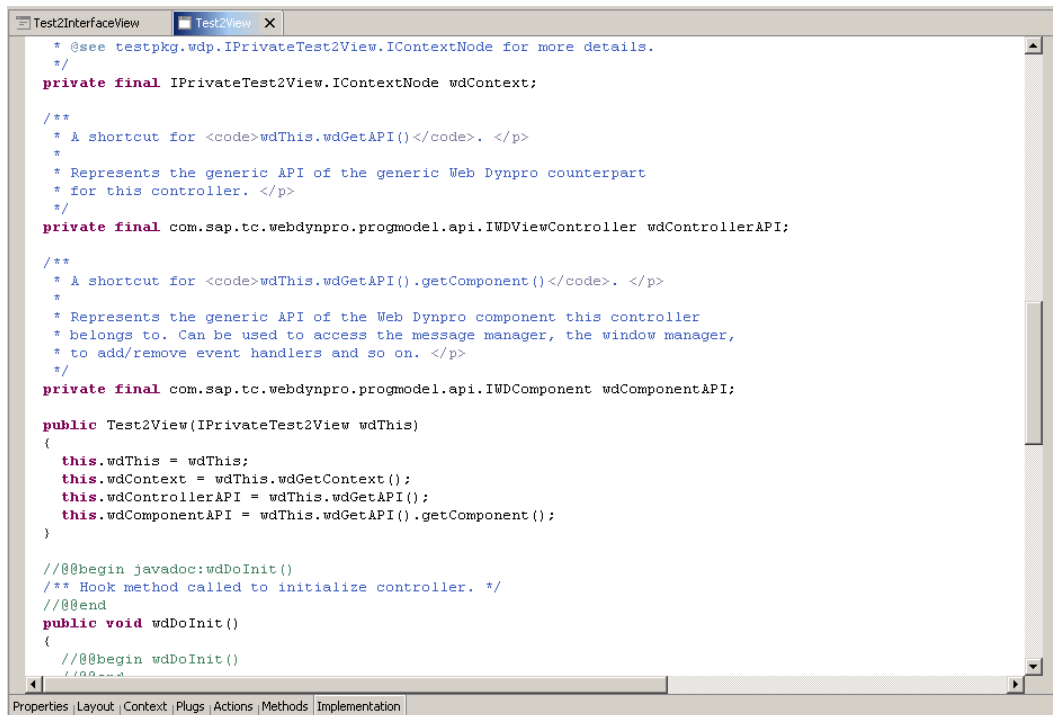


Figure 61 - SAP NetWeaver Developer Studio: fenêtre de travail (code)

En utilisant les onglets disponibles en bas de la fenêtre, il est ainsi possible d'atteindre et de gérer tout ce qui touche à une vue ou un contrôleur par exemple.

7.2.3.4 Fenêtre de rapports (4)

La fenêtre de rapport permet de visualiser rapidement l'état de ce sur quoi on travaille. Les onglets en bas de la fenêtre permettent de sélectionner ce que l'on souhaite voir. Les 4 plus importants étant:

- **Tasks:** indique toutes les erreurs dans le code

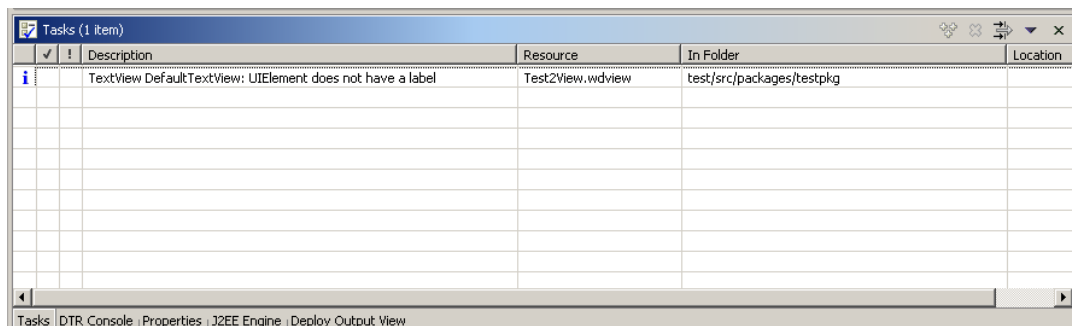


Figure 62 - SAP NetWeaver Developer Studio: fenêtre de rapport (Tasks)

- **Properties:** pour un élément sélectionné (bouton, champ, attribut, etc..), possibilité de modifier toutes ses propriétés. Ca peut être sa taille, le texte qu'il affiche, une liaison entre un champ et un modèle de données ou encore sa mise en page. Tout dépend de l'élément sélectionné.

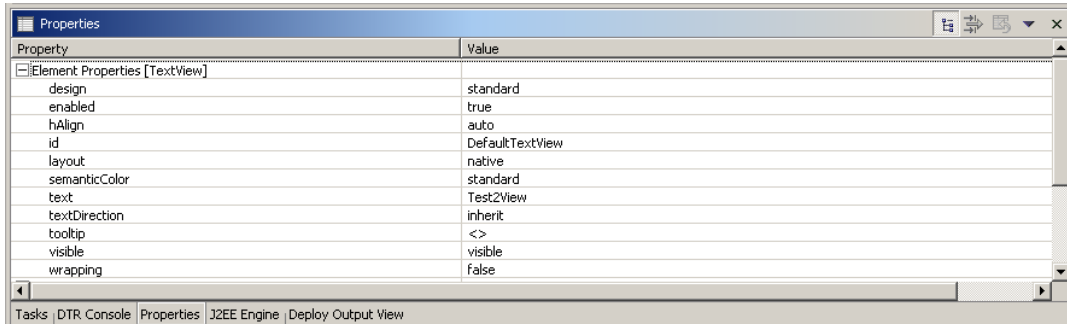


Figure 63 - SAP NetWeaver Developer Studio: fenêtre de rapport (Properties)

- **J2EE Engine:** permet d'afficher l'état du serveur sur lequel les applications sont déployées. Il s'agit du serveur dont les informations de connexion ont été saisies dans les préférences quelques pages plus haut. Pour pouvoir déployer une application, il faut que tous les voyants soient au vert.

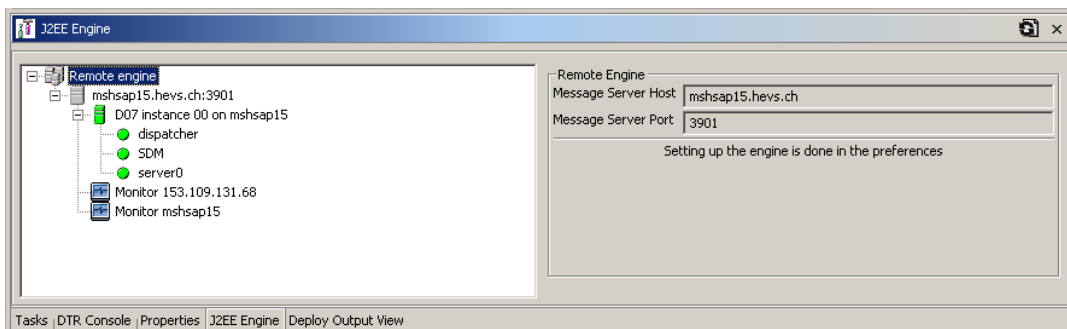


Figure 64 - SAP NetWeaver Developer Studio: fenêtre de rapports (J2EE Engine)

- **Deploy Output View:** lors du déploiement d'une application, il indique les étapes effectuées et signale toute erreur qui pourrait empêcher ce dernier. Il est possible d'obtenir plus d'information en double-cliquant sur une ligne.

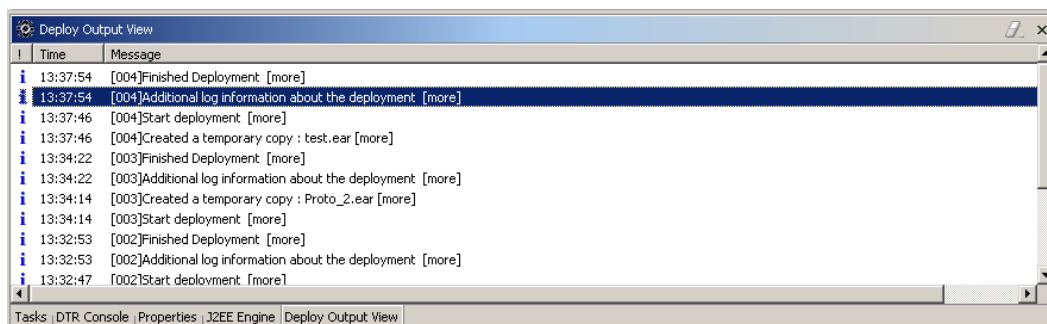


Figure 65 - SAP NetWeaver Developer Studio: fenêtre de rapport (Deploy Output View)

7.2.4 Les nœuds de données, contextes

7.2.4.1 Nœuds de données

Dans une application Web Dynpro, l'utilisation et la compréhension des nœuds de données est essentielles. En effet, une fois configurés correctement, ces derniers permettent de manipuler très simplement les différentes données (ex: nom, prénom, prix, dates, etc..) que l'on retrouve dans une application ou un formulaire et par la même occasion de garantir une certaine fiabilité du code tout en diminuant ce qu'un développeur doit écrire de lui-même.

La manière la plus simple de les comprendre est de les voir comme des objets. Si l'on prend un exemple basique comme la gestion de personnes, on retrouvera à un moment donné un objet nommé "Personne" et qui contiendra divers attributs, tels que le nom, le prénom, la date de naissance, l'adresse et tout autre information utile à l'application. C'est pareil pour un nœud de donnée: on trouvera un nœud nommé "Personne" et les mêmes attributs seront définis.

Tout comme les objets, il est aussi possible d'en réunir plusieurs du même type pour en faire une collection. Ainsi, un nœud "Personne" peut définir entre 0 et n personnes à la fois. Il suffit alors de faire appel à un code itératif pour les parcourir un par un. Le taille de cette collection peut-être définie au moyen de la propriété "cardinality" d'un nœud dans NetWeaver Developer Studio, comme l'indique l'image ci-dessous:

cardinality	0..1
collectionType	0..1
initializeLeadSelection	0..n
name	1..1
selection	1..n
singleton	true

Figure 66 - Nœuds de données: cardinalités

Il y a 4 cardinalités possibles:

- 0..1: un nœud contient 0 ou 1 élément
- 0..n: un nœud contient entre 0 et n éléments
- 1..1: un nœud contient exactement 1 élément
- 1..n: un nœud contient entre 1 et n éléments

7.2.4.2 Contexte

Un nœud se définit dans ce que l'on appelle un "contexte". Un contexte est généralement rattaché à une vue ou à un contrôleur et permet, une fois configuré, d'accéder aux données qu'il contient très facilement. Le schéma ci-dessous résume ce que nous venons de voir:

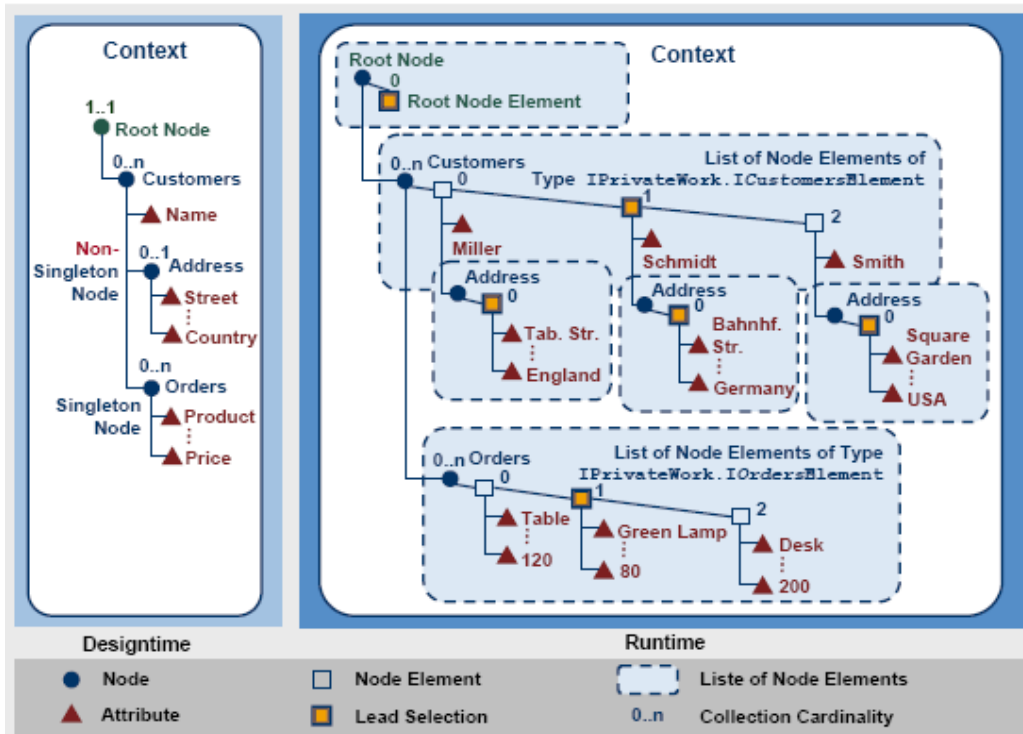


Figure 67 - Contextes: structure¹³

Pour illustrer cela, nous reprenons l'exemple d'un nœud définissant une personne.

¹³ Source: PDF: Application of Context Programming and Data Binding

Dans NetWeaver Developer Studio, ça donne:

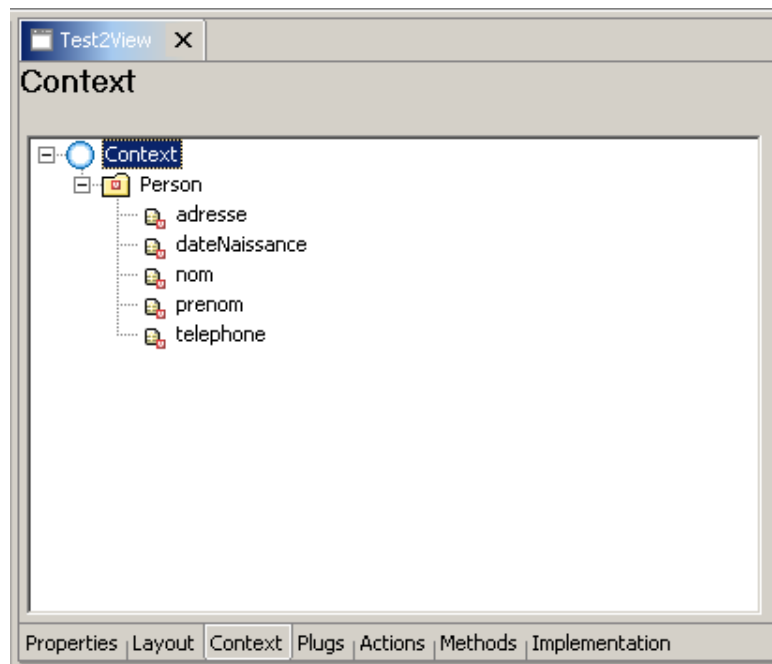


Figure 68 - Contexte: exemple

L'image ci-dessus représente le contexte de la vue "Test2View". Nous pouvons y voir le nœud "Person" qui définit plusieurs attributs d'une personne.

Une fois le contexte défini, le développement de la vue et la liaison d'un champ avec un attribut du nœud devient très aisée.

Avant d'effectuer la liaison, imaginons que notre vue ressemble à ça:

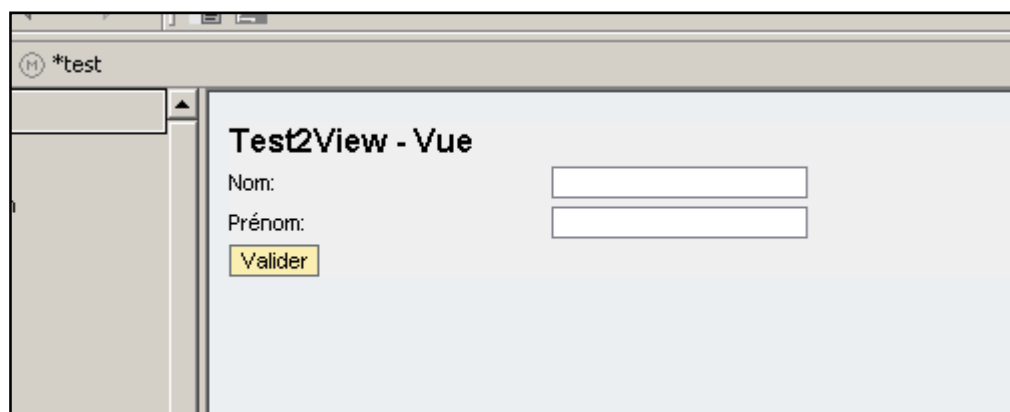


Figure 69 - Exemple de vue

En cliquant sur un champ de saisie et en parcourant ses propriétés en bas de l'écran, il est possible de créer en 2 clics la liaison.

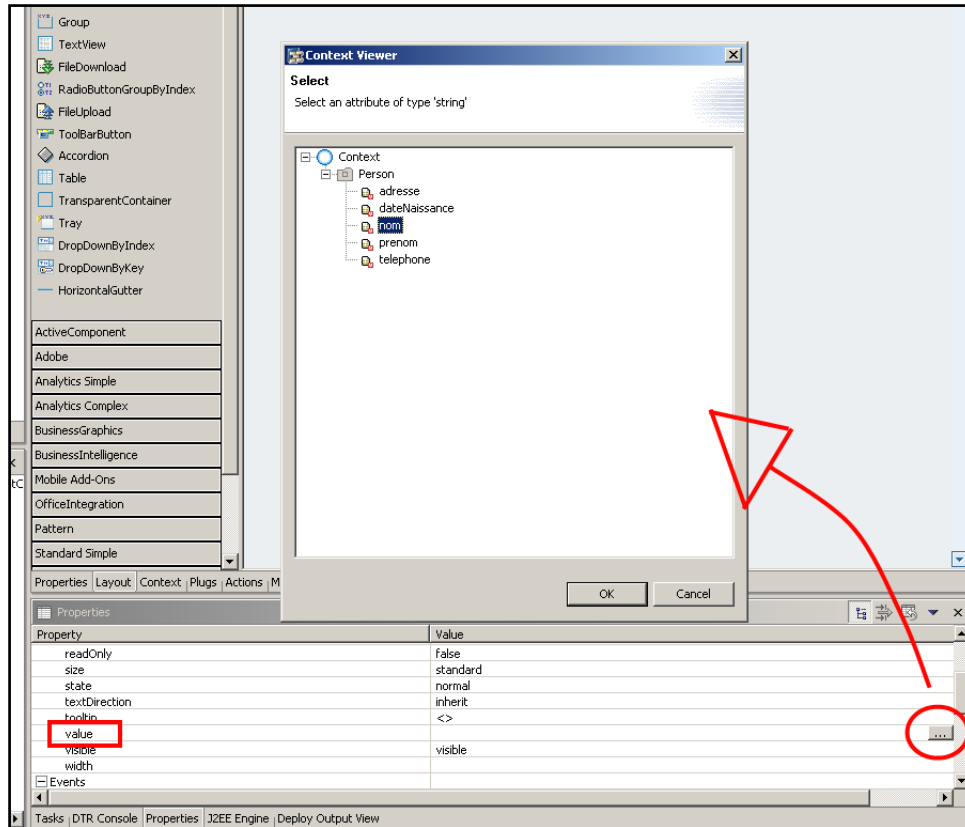


Figure 70 - Liaison d'un champ à un attribut

Il suffit alors de répéter l'opération pour tous les attributs que l'on souhaite lier dans notre vue. Une fois effectué, tous les champs affichent leurs attributs respectifs, comme le montre l'image ci-dessous:

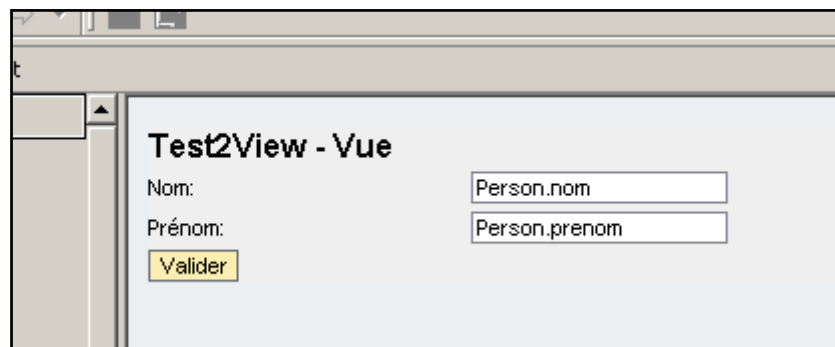


Figure 71 - Liaison d'un champ

De cette manière, si la valeur d'un attribut est modifiée, tous les champs qui y sont liés seront automatiquement mis à jour.

Au niveau du code, la valeur se récupère très simplement. Il suffit de saisir la ligne de code respectant le format suivant:

```
wdContext.current<NomDuNoeud>Element().get<NomDeL'Attribut>();
```

Exemple:

```
String nom;  
String prenom;  
  
nom = wdContext.currentPersonElement().getNom();  
prenom = wdContext.currentPersonElement().getPrenom();
```

7.2.5 Incorporation d'un formulaire interactif

Un élément qui reste à présenter et l'ajout et le développement d'un formulaire interactif dans une application Web Dynpro. Pour ce faire, 2 méthodes simples:

Lors du design de l'application, cliquer sur "Adobe" et à gauche, cliquer sur "InteractiveForm" et effectuer un drag & drop vers l'application.

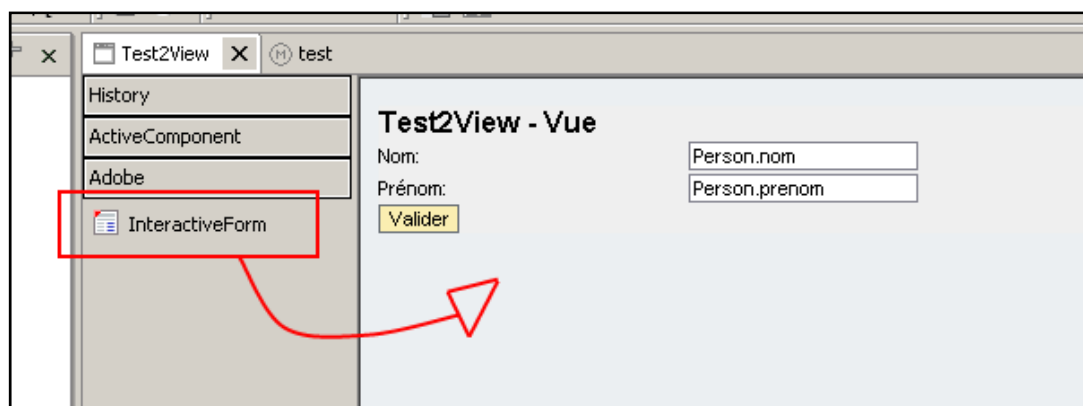


Figure 72 - Interactive Form: via drag & drop

L'autre solution consiste à effectuer un clic droit dans la fenêtre "Outline" lorsque celle-ci affiche les divers éléments graphiques de l'application et de sélectionner "Insert Child":

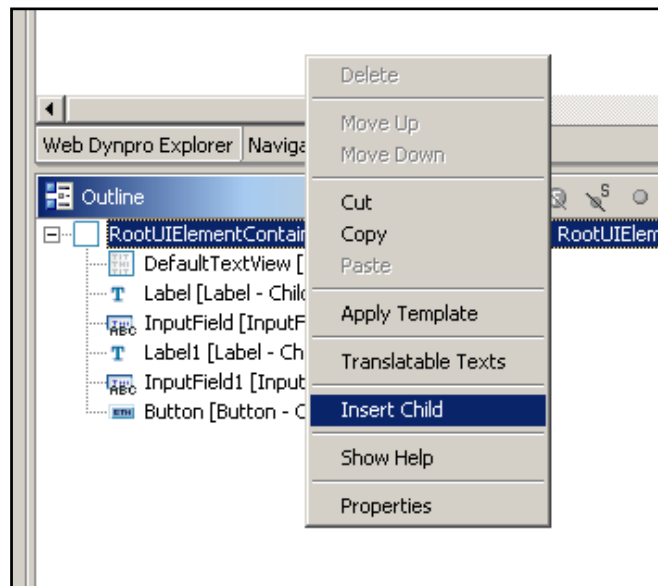


Figure 73 - Interactive Form: insertion via Outline

Il suffit alors de sélectionner "InteractiveForm" dans la liste qui apparaît. A noter qu'il est aussi possible de modifier l'id de l'élément si celui qui apparaît par défaut ne convient pas (dans l'exemple ci-dessous, l'élément a été renommé "demoForm").

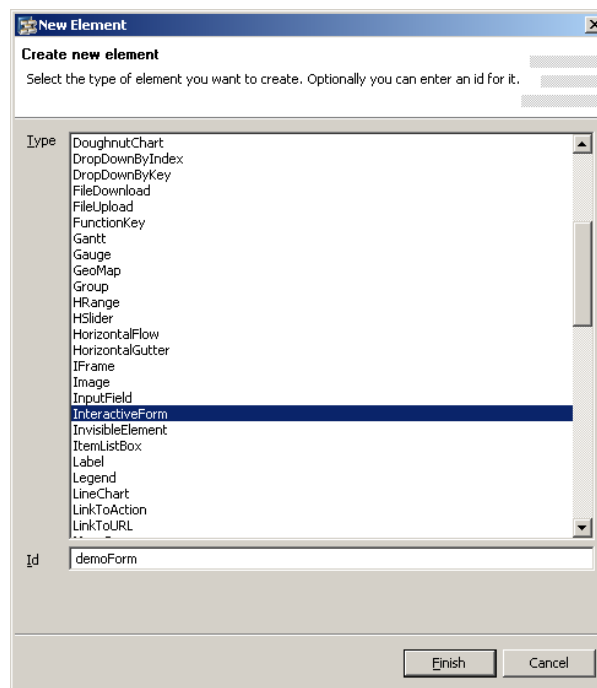


Figure 74 - Liste des éléments insérables

Une fois créé, le formulaire apparaît dans la fenêtre de design de l'application sous une forme visuellement similaire à un label.

Figure 75 - Interactive Form: affichage dans la fenêtre de design

Lors de l'ajout d'un formulaire interactif, il est indispensable de pouvoir retrouver dans le contexte de la vue dans laquelle il est utilisé un attribut qui servira de "source". En effet, c'est dans cet attribut que le PDF est sauvegardé et sans lui, l'application ne pourra tout simplement pas être déployée. Voici la marche à suivre:

Tout d'abord, afficher le contexte de la vue:

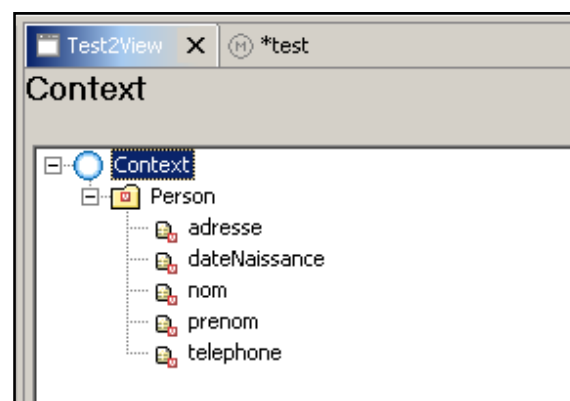


Figure 76 - Contexte

Faire un clic droit sur "Context" et sélectionner New → Value Attribute:

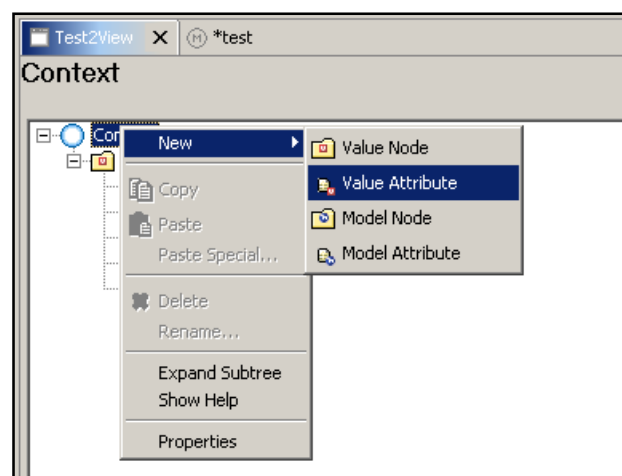


Figure 77 - Context: création d'un attribut

Dans la fenêtre qui apparaît, on saisit simplement un nom, comme par exemple "pdfSource", puis on valide.

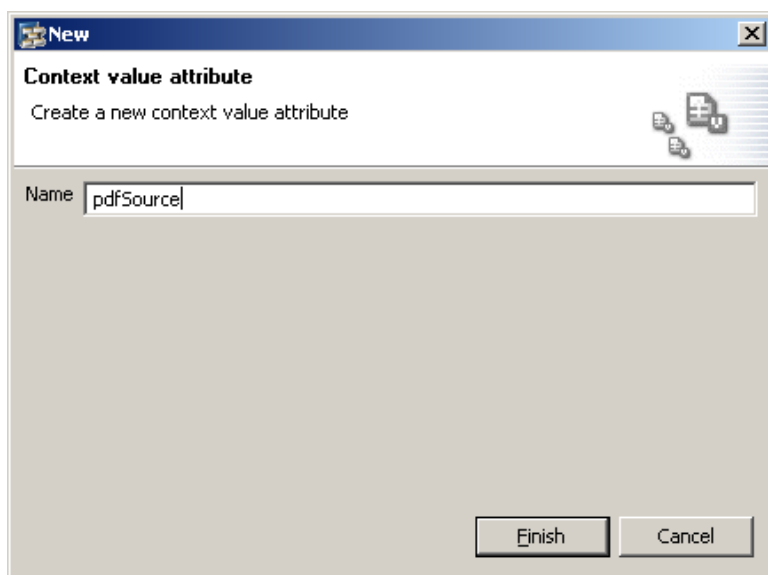


Figure 78 - Contexte: nommage de l'attribut

L'attribut est ainsi créé. Il faut ensuite se rendre dans ses propriétés qui apparaissent automatiquement au bas de l'écran juste après sa création et modifier le type en "binary" étant donné que l'attribut correspond à un fichier:

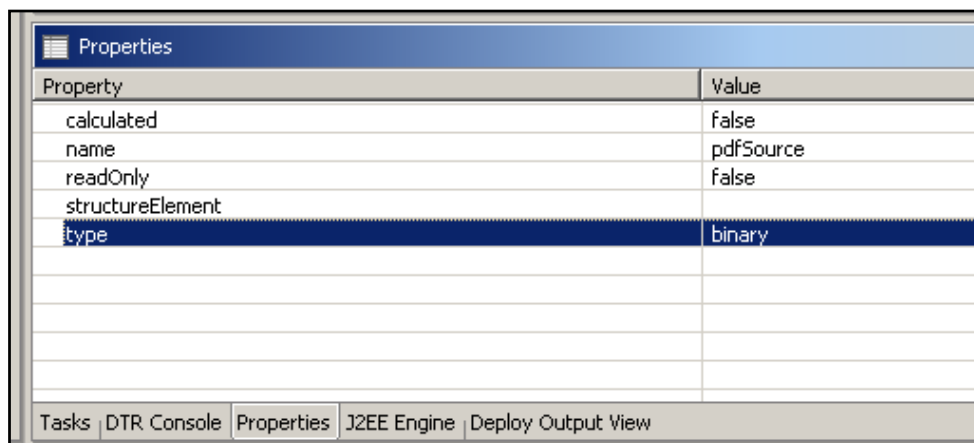


Figure 79 - Attribut "pdfSource": type binaire

7.2.6 Présentation d'Adobe LiveCycle Designer

Nous allons à présent survoler Adobe LiveCycle Designer, l'outil qui nous permettra par la suite de créer le contenu et l'aspect visuel d'un formulaire interactif. L'idée est ici de présenter brièvement l'outil. Nous y reviendront plus en détail lors des divers tutoriels.

Pour lancer Adobe LiveCycle Designer, sélectionner tout d'abord le formulaire interactif dans la fenêtre "Outline", faire un clic droit, puis sélectionner "Edit":

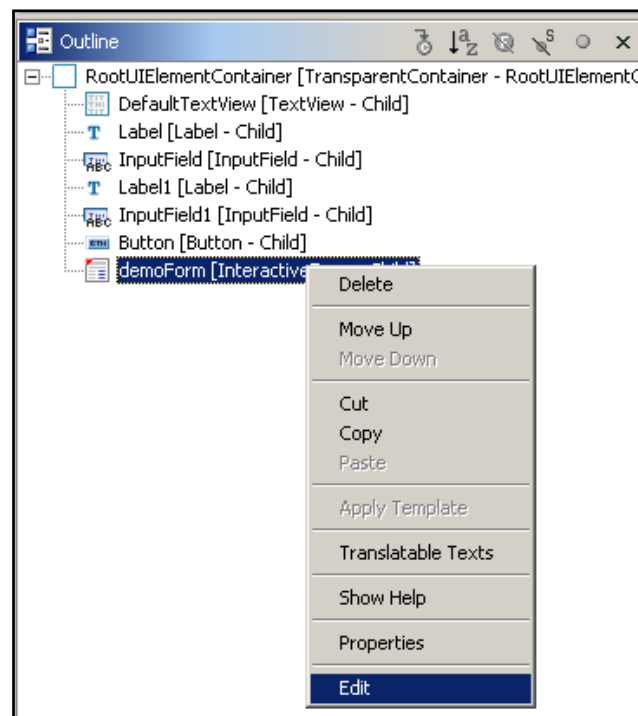


Figure 80 - Interactive Form: lancement d'Adobe LiveCycle Designer

L'éditeur apparaît:

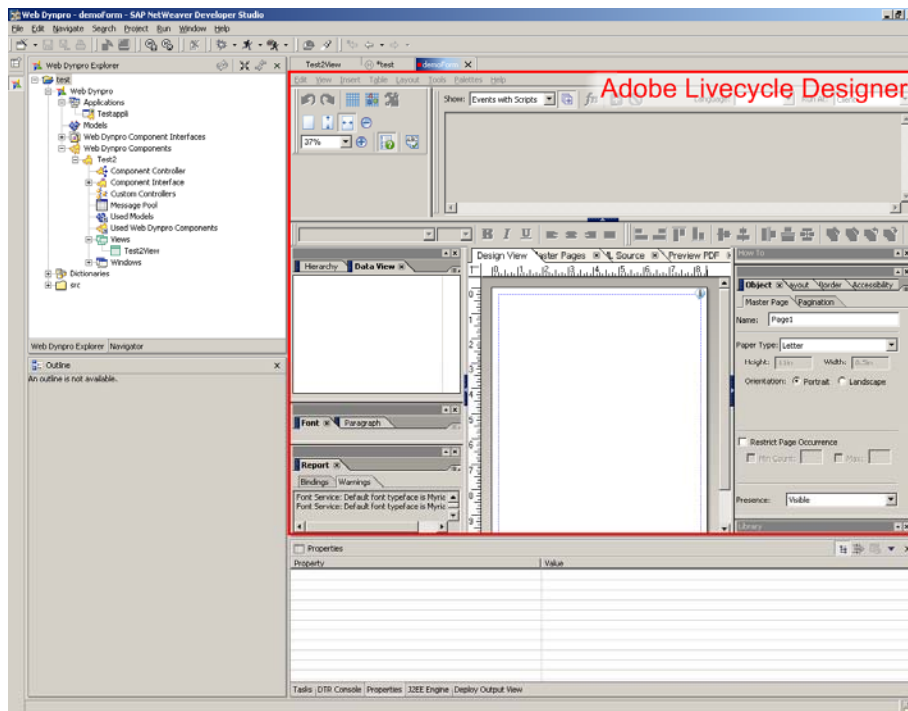


Figure 81 - Adobe LiveCycle Designer dans SAP NetWeaver Developer Studio

Comme nous pouvons le voir, l'éditeur affiche un bon nombre de choses dans un espace assez restreint. En effectuant un double clic sur l'onglet portant le nom du formulaire, il est possible d'afficher l'éditeur sur tout l'espace disponible (pour effectuer l'opération inverse, il suffit de refaire le même double clic)

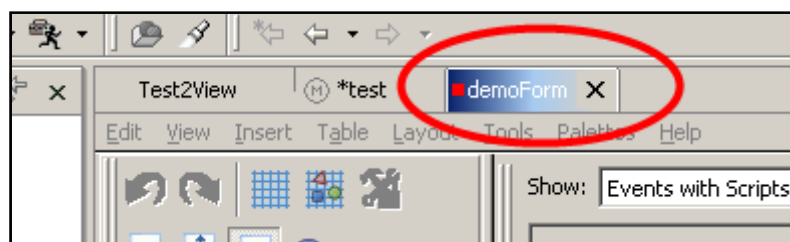


Figure 82 - Interactive Form: onglet dans SAP NetWeaver Developer Studio

L'éditeur se présente ainsi:

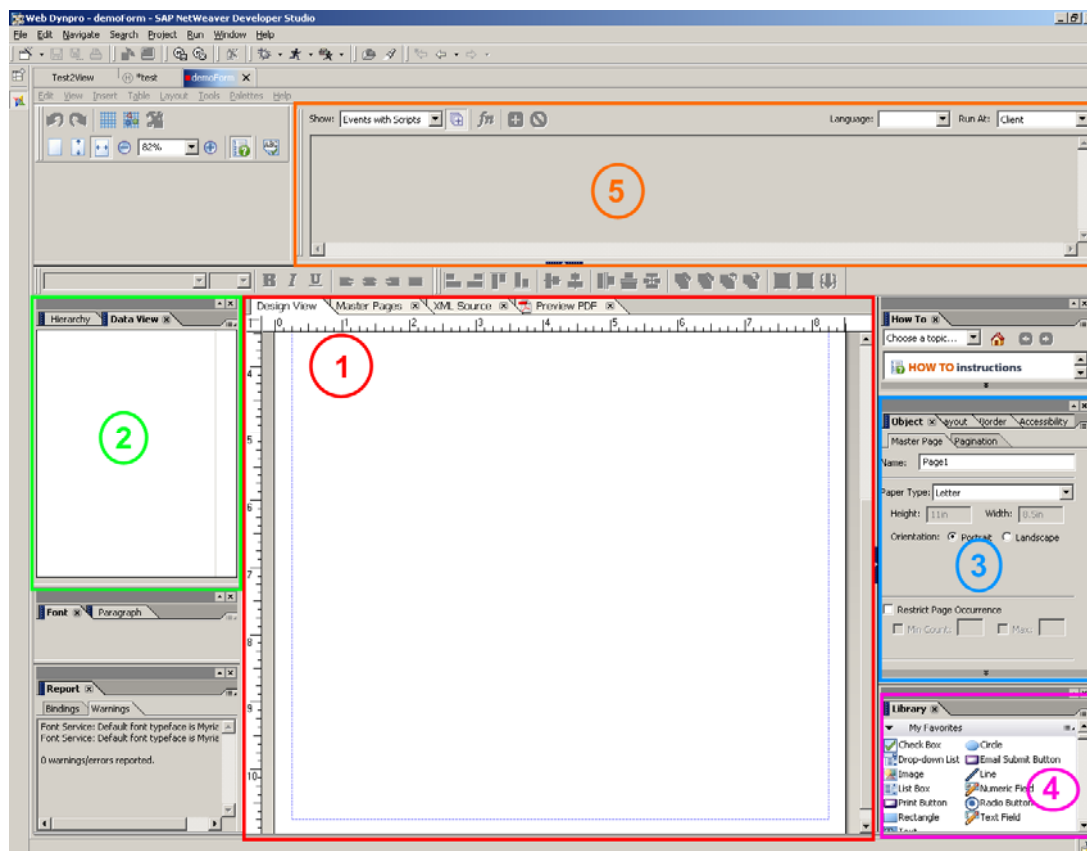


Figure 83 - Adobe LiveCycle Designer: interface principale

Fenêtre de design (1)

C'est ici que l'on gère et met en page le formulaire. Plusieurs onglets sont disponibles, notamment "XML Source" qui permet d'afficher le XML qui se cache derrière notre formulaire ou encore "Preview PDF" qui, sans avoir à déployer l'application, permet d'afficher un aperçu du formulaire et d'effectuer quelques tests, comme par exemple le calcul d'une valeur.

Data View (2)

Une fois le formulaire attaché à un nœud de données, sa structure apparaît dans la fenêtre de "Data View" et il est ainsi possible pour chaque élément d'effectuer un drag & drop vers la fenêtre de design, créant ainsi un champ de saisie immédiatement prêt à l'emploi.

Fenêtre d'objet (3)

Permet de gérer les propriétés d'un élément sélectionné dans la fenêtre de design, comme par exemple l'apparence visuelle, la taille, la liaison des données, le format (dont celui d'affichage) de la donnée, la visibilité de l'élément (uniquement à l'écran,

uniquement à l'impression, les deux), la valeur par défaut, comment une valeur doit être saisie, les éventuels messages d'erreur liés à la saisie, etc...

Librairie (4)

Liste tous les éléments graphiques que l'on peut rajouter via drag & drop dans la fenêtre de design. On y retrouve des éléments classiques tels que des boutons, des listes déroulantes ou encore des cases à cocher, mais aussi des objets plus spécifiques tels que des champs de saisie pré-remplis (liste de pays) ou encore des codes-barres.

Fenêtre de script (5)

Ici, il est possible d'ajouter du code pour n'importe quel élément graphique, mais aussi en fonction de certains événements (lorsque le formulaire est chargé, lors d'un clic, etc..), comme le montre la capture d'écran ci-dessous:

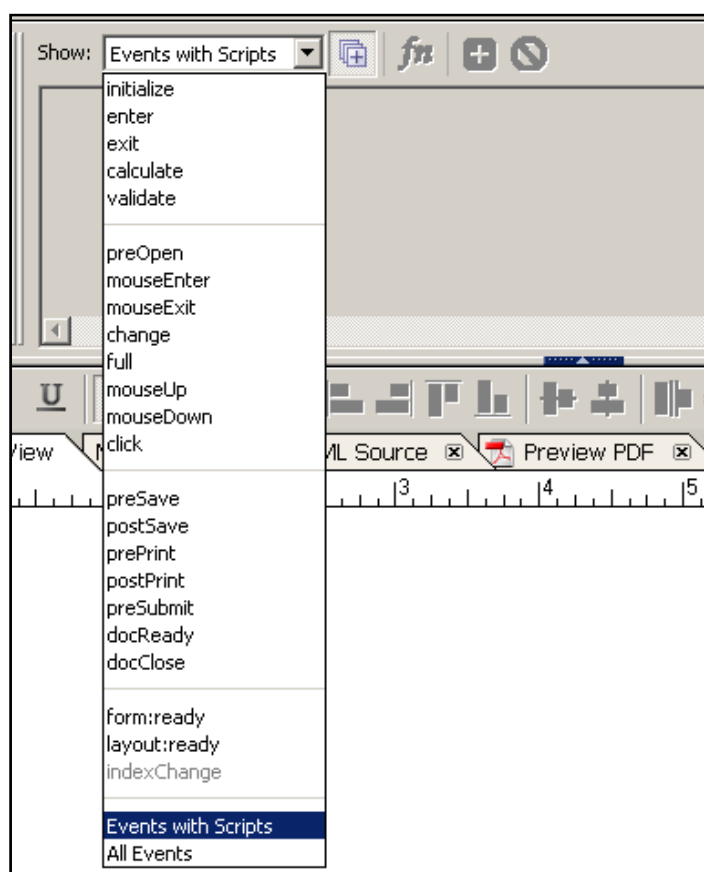


Figure 84 - Interactive Form: événements pour les scripts

Deux langages sont disponibles: FormCalc et JavaScript :

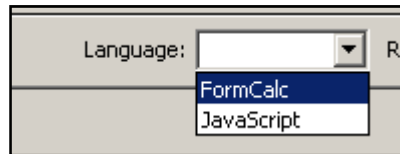


Figure 85 - Interactive Form: langages des scripts

Grâce à ça, il est possible de personnaliser davantage le formulaire, d'effectuer des calculs (ex: somme des prix pour calculer un total) ou encore de renforcer le contrôle des informations saisies.

Dernier point concernant LiveCycle Designer: il est possible d'importer un formulaire existant. Pour ce faire, il suffit de cliquer sur Tools → Import...:

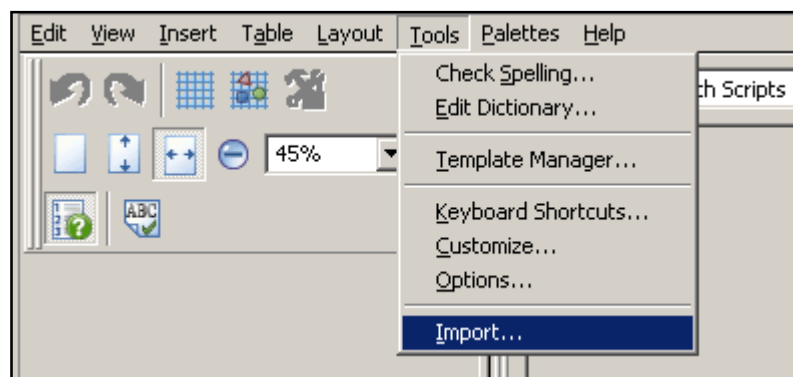


Figure 86 - Interactive Form: importation

Une fenêtre vous demandant quel fichier vous souhaitez importer apparaît alors:

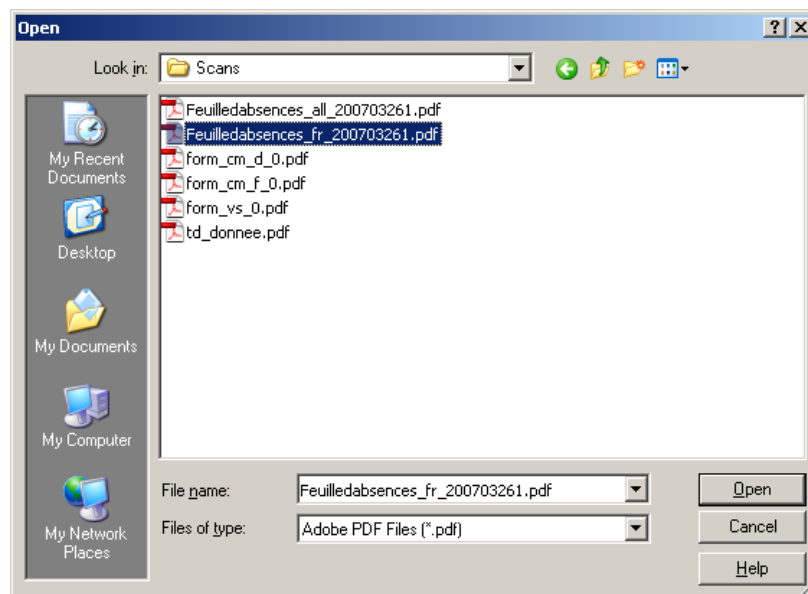


Figure 87 - Sélection du .pdf à importer

Sélectionnez le fichier. La fenêtre des options d'importation s'affiche ensuite:

Import Options

This document was not created in Adobe LiveCycle Designer. A copy of the original document will be created. Choose how you want to work with the imported PDF content.

[Learn More](#)

☐ Create an Interactive Form with Fixed Pages
Import the content of the original document as artwork. This preserves its appearance and retains existing interactive form fields. You can place new form fields over this artwork but cannot edit the artwork in LiveCycle Designer.

☒ Create an Interactive Form with a Flowable Layout
Makes the content in the original document editable and lets you create flowable elements in your form design. You may need to touch up the new document to restore its original appearance.

Custom Options for Importing PDF

Place content in: Master Pages

Join text as: Text Blocks

Tolerance for joining text: Auto Select

Tolerance for joining text lines: Auto Select

Tolerance for joining paragraphs: Auto Select

☒ Embed images Format: BMP

☐ Generate a log file in the Temporary folder

☒ Display a Summary report when conversion issues are found

Page Range

☐ All

☒ Pages: 1

Enter page numbers and/or page ranges separated by commas.
For example: 1,3,5-12

Help OK Cancel

Figure 88 - Interactive Form: options d'importation

Vous pouvez ici régler comment votre fichier PDF sera importé. Une fois les réglages effectués, il suffit de cliquer sur OK pour terminer l'importation.

Le fichier PDF est alors visible dans la fenêtre de design:

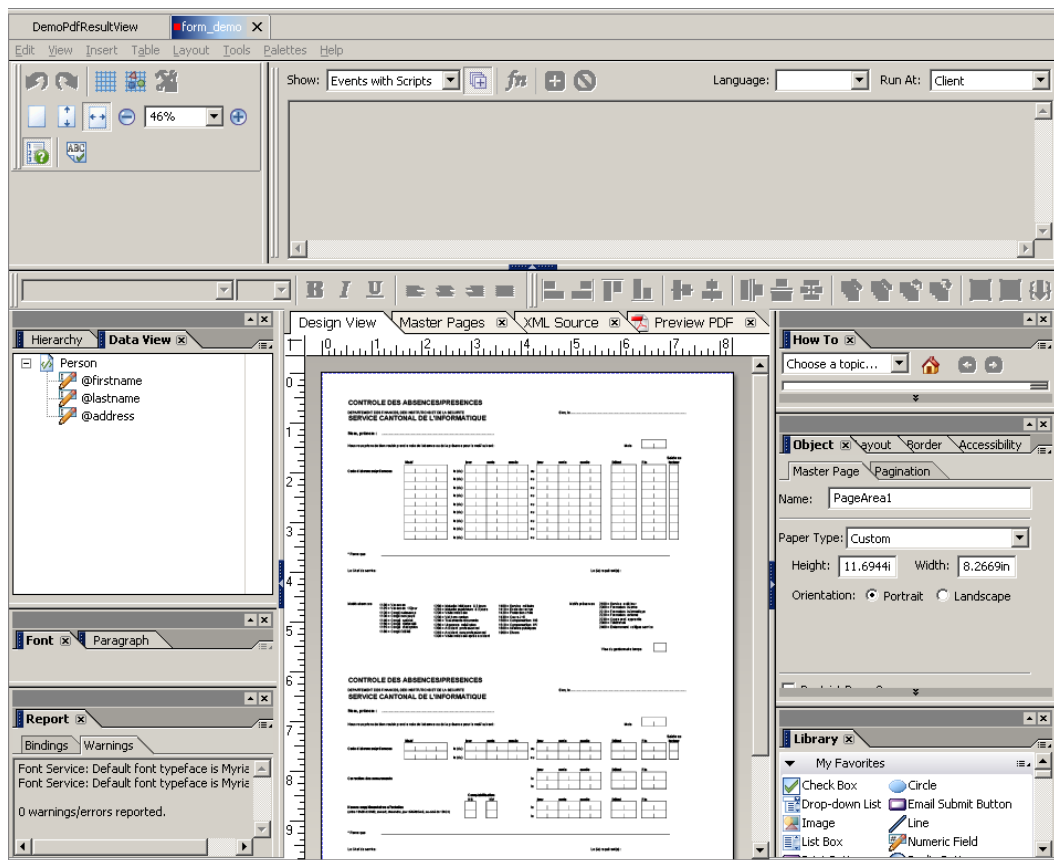


Figure 89 - Interactive Form: exemple de formulaire importé

7.2.7 Déploiement d'une application

La dernière chose à voir est le déploiement.

Pour déployer une application, assurez-vous tout d'abord que la fenêtre de rapports n'indique aucune erreur concernant le serveur ainsi que l'application (les informations et les avertissements peuvent généralement être ignorés).

Ensuite, faites un clic droit sur le nom de l'application et sélectionnez "Deploy New Archive and Run":

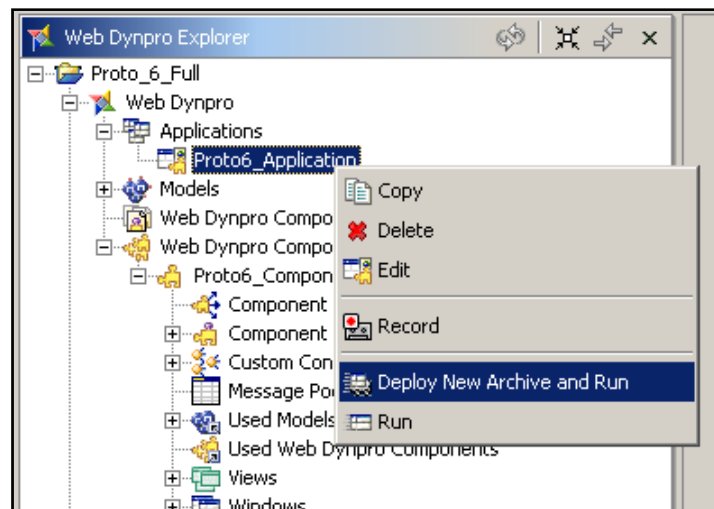


Figure 90 - SAP NetWeaver Developer Studio: déploiement

Lors du premier déploiement après l'ouverture du SAP NetWeaver Developer Studio, un mot de passe est demandé avant de pouvoir continuer. Celui-ci a été défini par la personne qui a installé le serveur d'application web (WebAS).

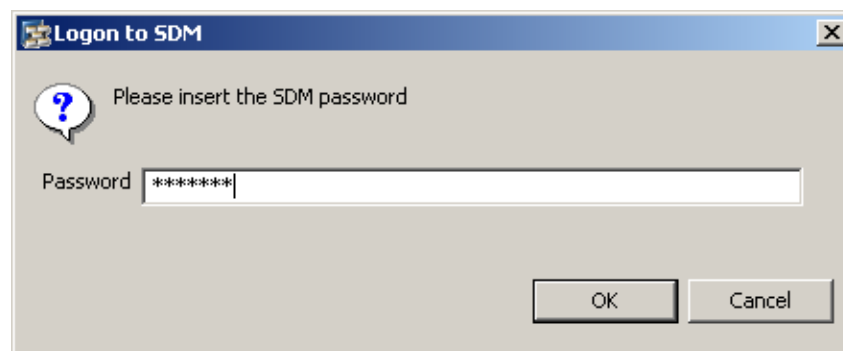


Figure 91 - SAP NetWeaver Developer Studio: mot de passe pour le déploiement

Une fois le déploiement effectué, Internet Explorer affiche l'application en ouvrant une nouvelle fenêtre ou en utilisant la fenêtre la plus récente. Exemple:

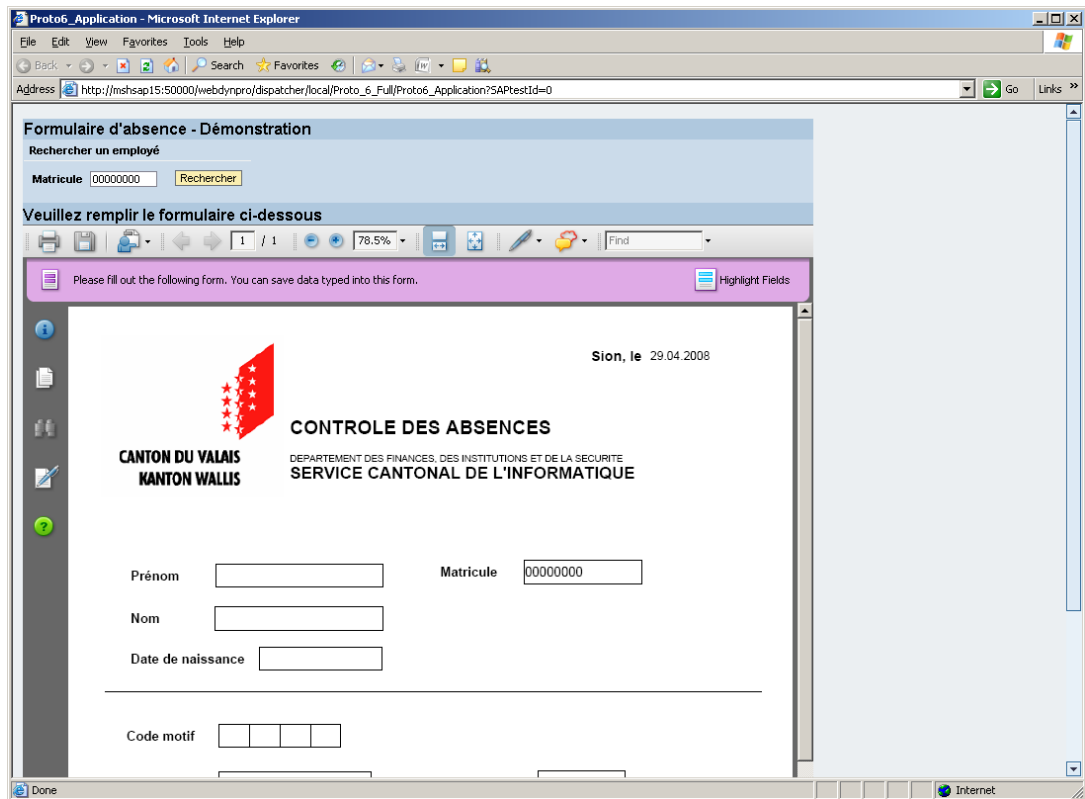


Figure 92 - SAP NetWeaver Developer Studio: exemple d'une application déployée

Il faut aussi savoir qu'une fois déployée, l'application est disponible en passant simplement par le portail J2EE. Pour ce faire, rendez-vous sur la page du portail à l'adresse <http://mshsap15.hevs.ch:50000>, puis cliquez sur "Web Dynpro":

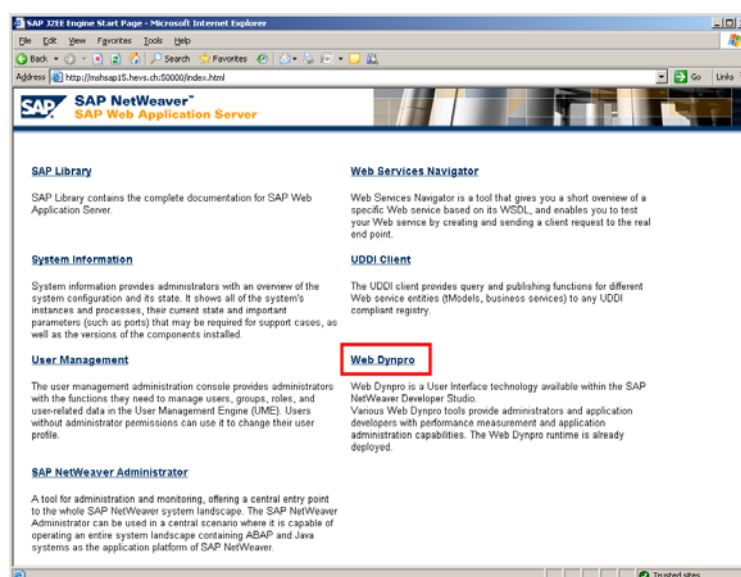


Figure 93 - SAP J2EE Start Page

Cliquez ensuite sur "Content Administrator":

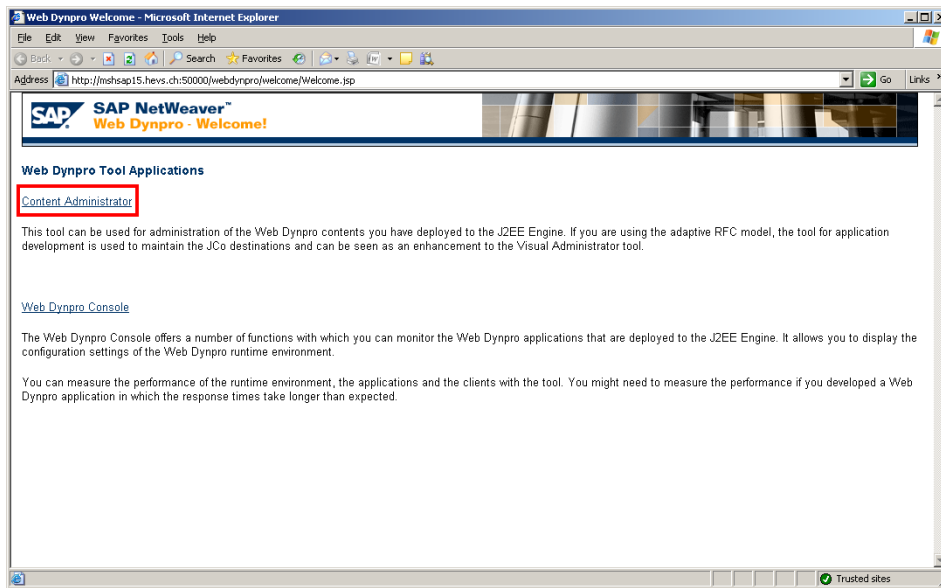


Figure 94 - Outils Web Dynpro

Connectez-vous avec votre identifiant habituel:

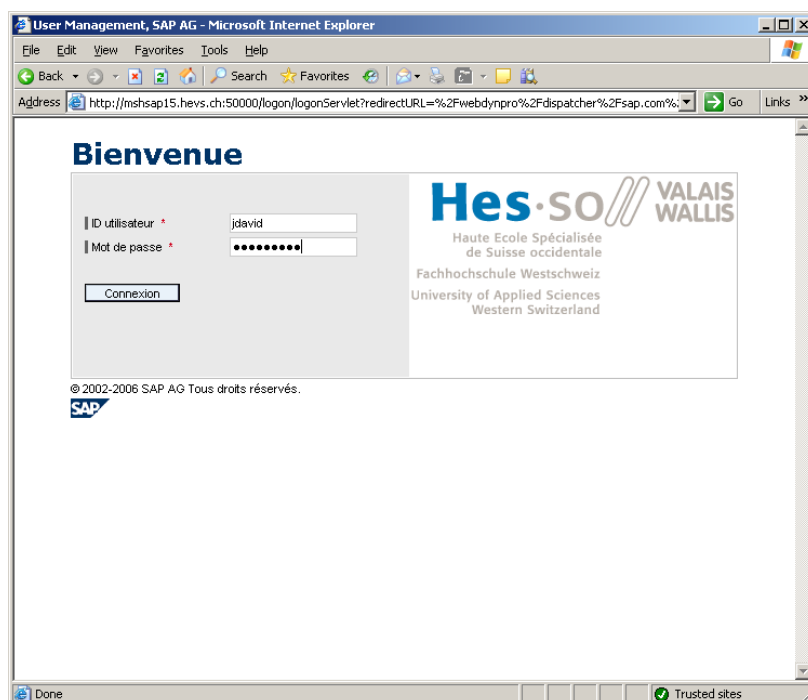


Figure 95 - Identification pour le Web Dynpro Content Administrator

Le Content Administrator apparaît alors:

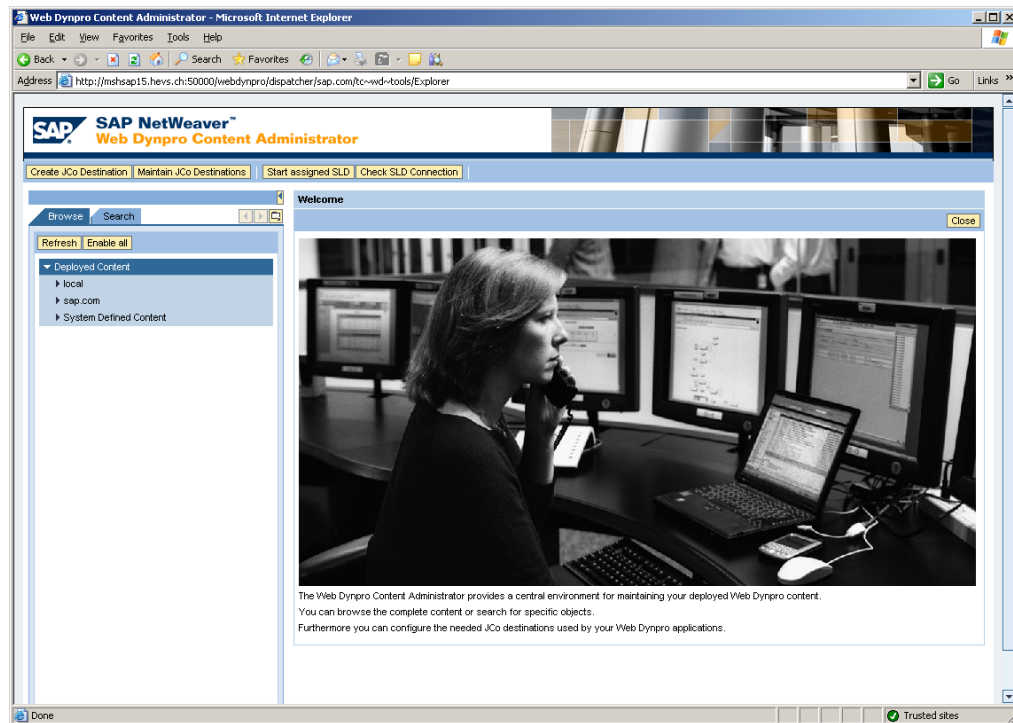


Figure 96 - Web Dynpro Content Administrator: écran principal

Pour retrouver l'application déployée, il suffit de cliquer sur "local" dans la partie de gauche et de rechercher notre application. Une fois sélectionnée, le Content Administrator affiche quelque chose de similaire à la capture d'écran ci-dessous:

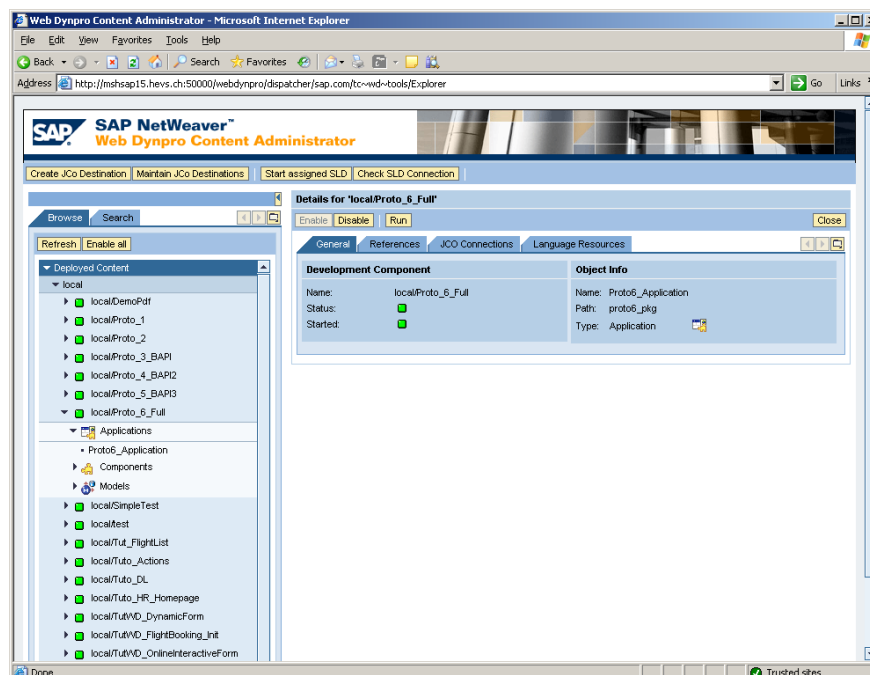


Figure 97 - Web Dynpro Content Administrator: liste des applications déployées

Il suffit alors de cliquer sur Run pour pouvoir lancer l'application.

Un élément important qu'il est intéressant de remarquer dès maintenant est l'onglet "JCO Connections". En effet, il faudra passer par là afin de terminer la configuration des BAPIs utilisées lors d'une liaison entre un formulaire et SAP. Pour l'heure, nous allons simplement garder cela en tête; nous y reviendrons beaucoup plus en détail lors du tutoriel portant sur les BAPIs.

Maintenant que les bases du développement des formulaires ont été parcourues, nous pouvons suivre un premier tutoriel qui a pour but de créer une application simple incorporant un formulaire interactif, l'idée étant notamment de mettre en pratique et de relier les points que nous venons de voir.

7.3 Tutoriel 1: réalisation d'une application simple

7.3.1 Description

Cette application utilisera 2 vues. La première permettra de saisir des informations dans des champs de saisie, puis, après avoir cliqué sur le bouton "Suivant", nous accéderons à la 2^{ème} vue qui incorporera un formulaire interactif pré-rempli avec les valeurs qui ont été saisies dans la vue précédente. Il sera aussi possible de modifier ces valeurs dans le formulaire lui-même, puis de retourner à la première vue qui affichera alors les données modifiées.

Ce premier tutoriel se veut le plus complet possible. Il dispose d'une grande quantité de captures d'écran et détaille précisément chaque étape du développement. L'idée est qu'une personne ne connaissant pas du tout cette technologie ne soit perdue à aucun moment et puisse déployer l'application développée sans avoir à consulter d'autres sources. Les prochains tutoriels aborderont des points plus spécifiques et complexes et n'offriront pas la même quantité de détails sur les manœuvres de base.

7.3.2 Création du projet

Pour commencer, créons un projet en cliquant sur File→New→Web Dynpro Project:

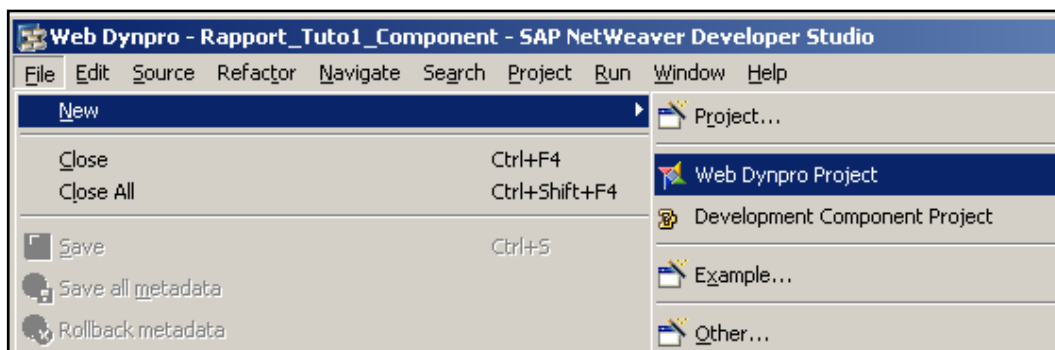


Figure 98 - Tutoriel 1: création d'un nouveau projet

On le nomme "Rapport_Tuto1" et on clique sur "Finish"

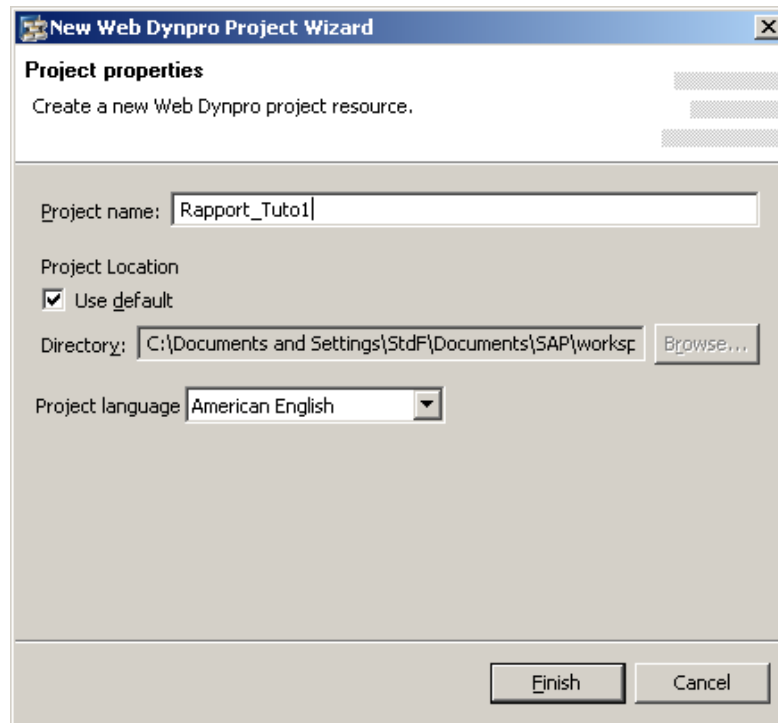


Figure 99 - Tutoriel 1: nommage du nouveau projet

La structure de notre projet apparaît dans l'explorateur Web Dynpro:

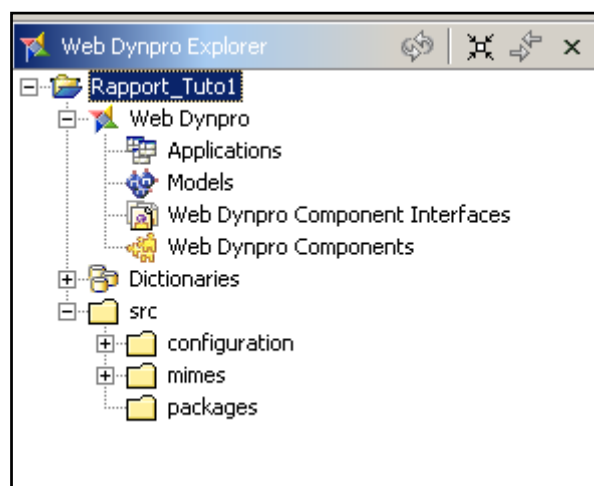


Figure 100 - Tutoriel 1: Structure du projet

7.3.3 Création du composant

La première chose à faire est de créer un "Web Dynpro Component" pour notre projet:

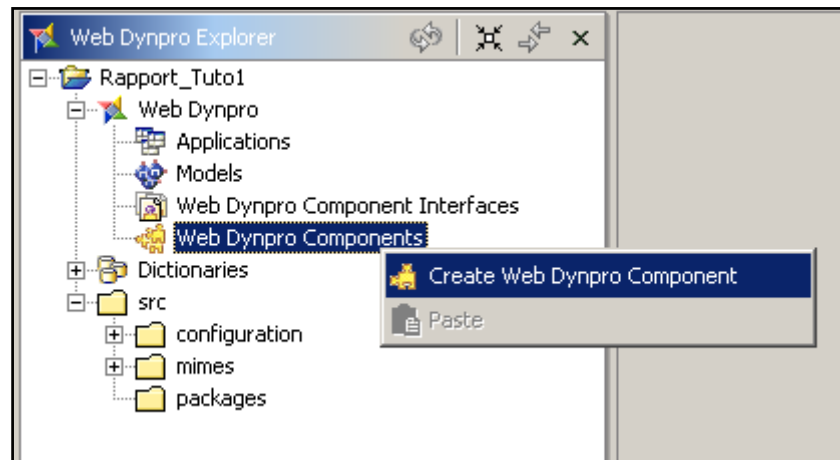


Figure 101 - Tutoriel 1: Création d'un composant

La fenêtre qui s'affiche demande de saisir plusieurs informations: le nom du composant ainsi que le nom du package. Nous pouvons saisir ici ce que nous voulons, mais par mesure de clarté nous allons utiliser des noms très parlants. En saisissant ces 2 noms, on peut s'apercevoir que NetWeaver Developer Studio remplit les autres champs (telle que la vue qui sera créée en même temps que le composant). Pour cette application, nous allons utiliser 2 vues, il est donc judicieux d'en donner un qui soit explicite à la vue qui sera créée. Nous l'appellerons donc "StartView".

Une fois les informations saisies comme sur l'image ci-dessous, nous pouvons cliquer sur "Finish":

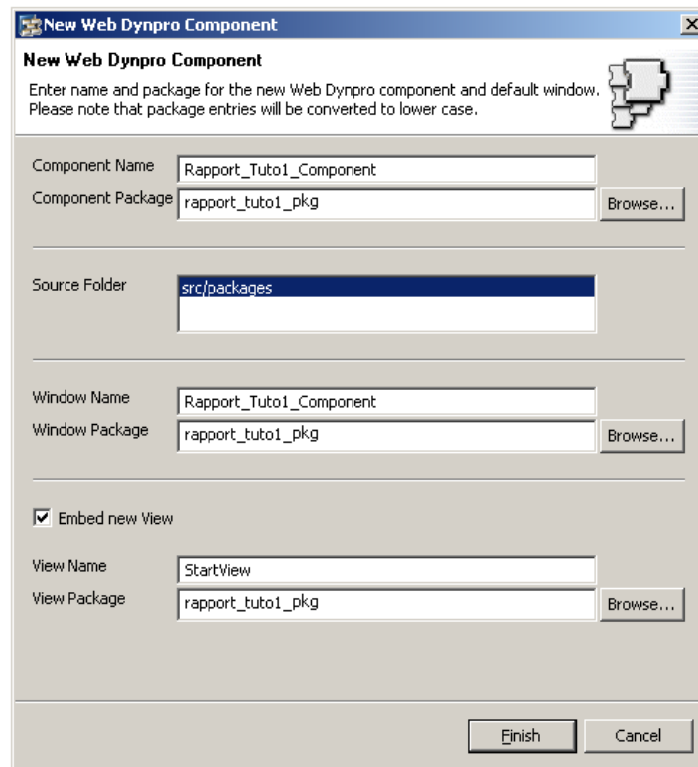


Figure 102 - Tutoriel 1: Nommage du component, package, vue par défaut

Le composant et sa structure sont alors ajoutés dans l'explorateur Web Dynpro:

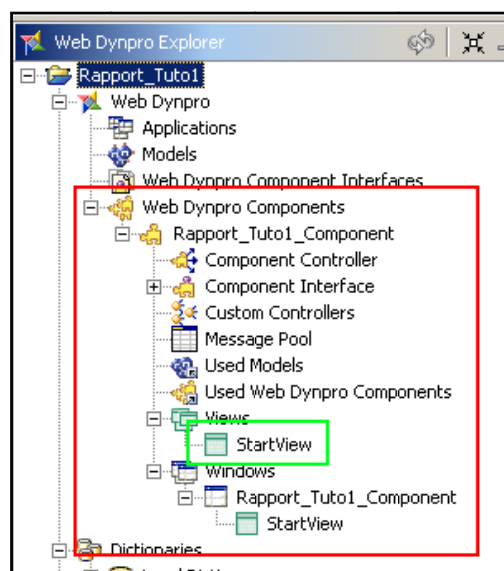


Figure 103 - - Tutoriel 1: structure du projet

Nous pouvons constater que la vue "StartView" (encadrée en vert) a bien été créée automatiquement.

7.3.4 Création d'une nouvelle vue

Nous créons à présent la vue qui incorporera le formulaire interactif. Pour ce faire, on effectue un clic droit sur Views et on sélectionne "Create View":

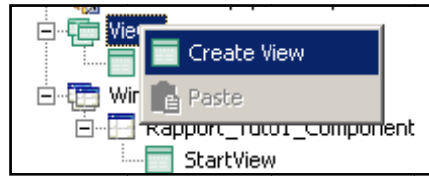


Figure 104 - Tutoriel 1: création d'une vue

On la nomme ensuite "ResultView" et on clic sur "Finish":

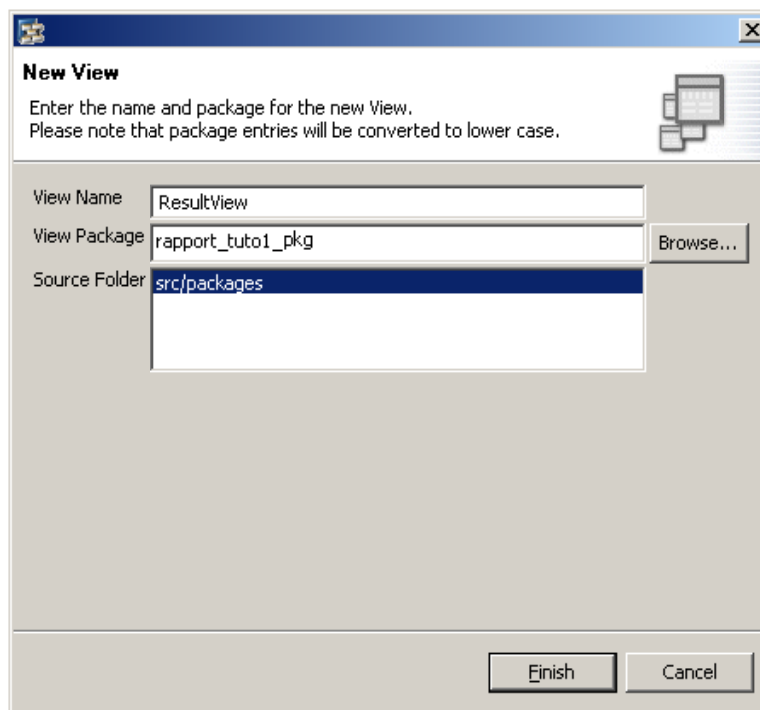


Figure 105 - Tutoriel 1: nommage d'une nouvelle vue

La nouvelle vue apparaît alors dans la structure du projet:

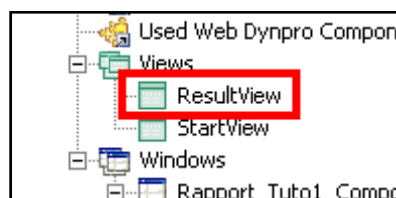


Figure 106 - Tutoriel 1: nouvelle vue

7.3.4.1 Navigation

Avant de nous occuper de l'interface graphique de ces vues, nous allons nous intéresser au schéma de navigation entre ces dernières, c'est-à-dire définir les points d'entrée et de sortie des vues ainsi que la manière dont elles sont liées.

A l'heure actuelle, l'interface de travail devrait ressembler à ceci:

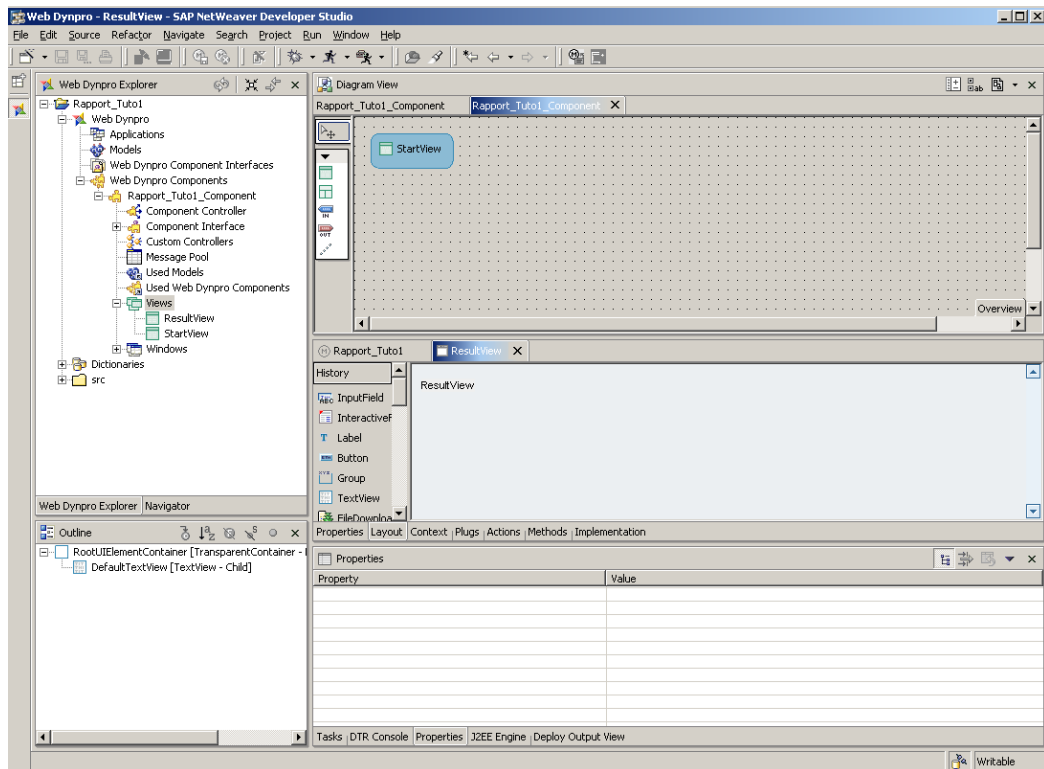



Figure 107 - Tutoriel 1: interface de travail

Comme nous pouvons le voir, la partie supérieure de la fenêtre comporte ce que l'on appelle le diagramme de navigation. C'est la partie où le développeur définit comment les vues sont liées entre elles. On remarque au passage que la vue que nous venons de créer, "ResultView", n'apparaît pas sur le diagramme.

Nous allons donc rajouter la vue "ResultView" en cliquant sur  dans la barre d'outil puis en cliquant n'importe où sur le diagramme. La fenêtre ci-dessous apparaît et nous demande quel type de vue nous souhaitons ajouter. Nous sélectionnons donc "Embed existing view" puis nous cliquons sur le bouton "Next":

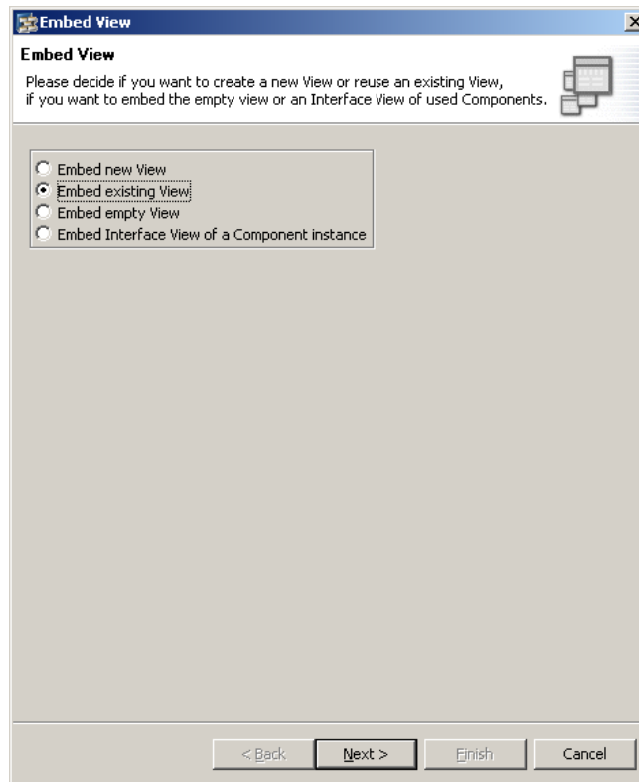


Figure 108 - Tutoriel 1: ajout d'une vue au diagramme

Sur la fenêtre suivante, on sélectionne "ResultView" et on valide en cliquant sur "Finish". La vue apparaît alors sur le diagramme:

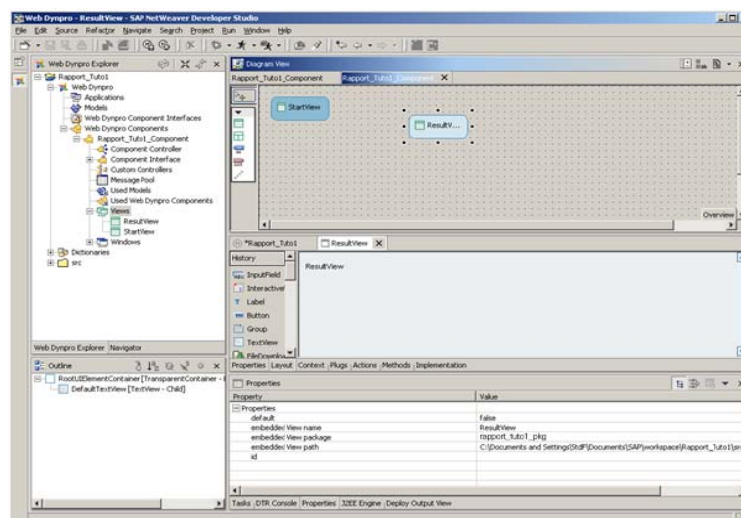



Figure 109 - Tutoriel 1: "Result View" dans le diagramme de navigation

Nous pouvons à présent définir les divers points d'entrée et de sortie entre les 2 vues. Commençons par les points de sortie (Outbound Plug) en cliquant sur  dans la barre d'outil puis en cliquant sur la vue "StartView". Une fenêtre nous demande alors de nommer ce point de sortie, nous l'appelons donc "toResultView" indiquant par la même occasion dans quelle direction il pointe. Pour finir, on valide en cliquant sur "Finish":

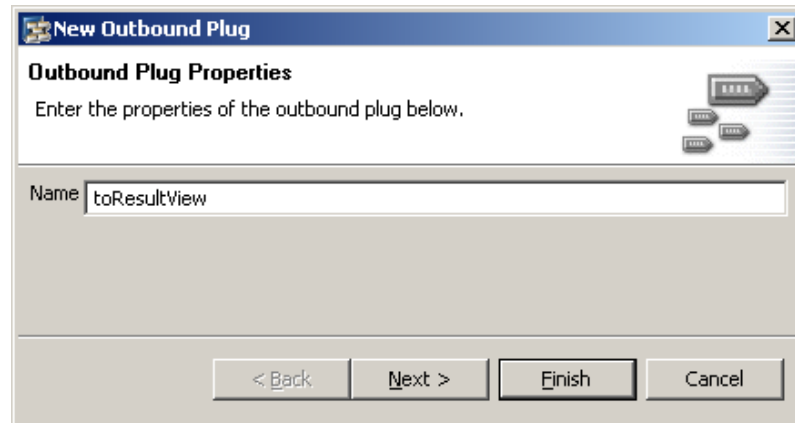


Figure 110 - - Tutoriel 1: nommage du point de sortie

Nous constatons alors que la vue "StartView" dispose à présent d'un point de sortie:

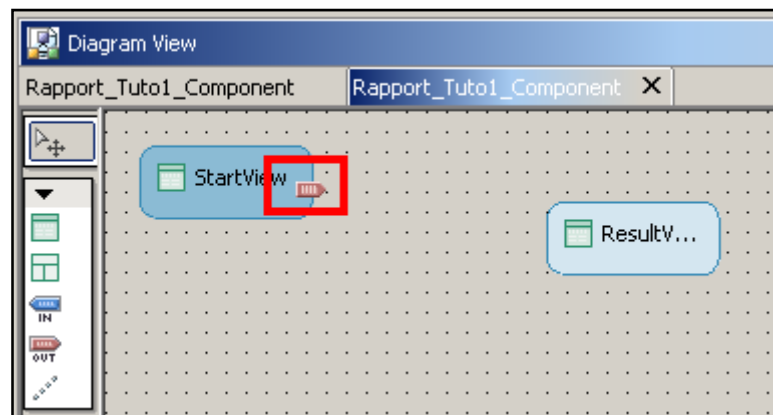





Figure 111 - Tutoriel 1: point de sortie sur le diagramme de navigation

Pour les points de sortie suivants, nous répétons la même opération en suivant le schéma de nommage suivant:

	Point de sortie ()	Point d'entrée ()
StartView	toResultView (déjà fait)	fromResultView
ResultView	toStartView	fromStartView

Remarque: bien que la fenêtre d'ajout des points d'entrée (Inbound Plug ) soit un peu différente, il suffit là aussi de saisir le nom du point et de valider en cliquant sur "Finish".

Une fois ces étapes effectuées, notre schéma se présente ainsi, sans les légendes:

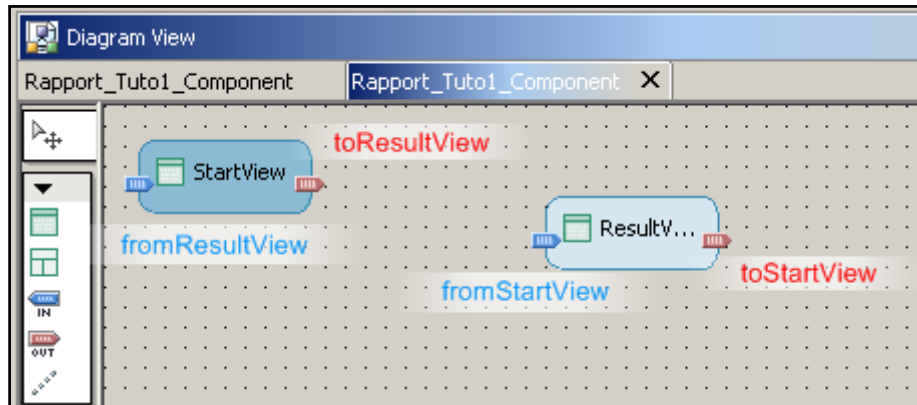



Figure 112 - Tutoriel 1: points d'entrée/sortie

Nous pouvons à présent configurer les liaisons entre ces deux vues. Il y aura 2 liaisons, l'une allant de "StartView" vers "ResultView", l'autre dans le sens inverse. En cliquant sur  puis en effectuant un drag & drop depuis le point de sortie "toResultView" vers le point d'entrée "fromStartView", une liaison entre ces 2 points apparaît sur le schéma:

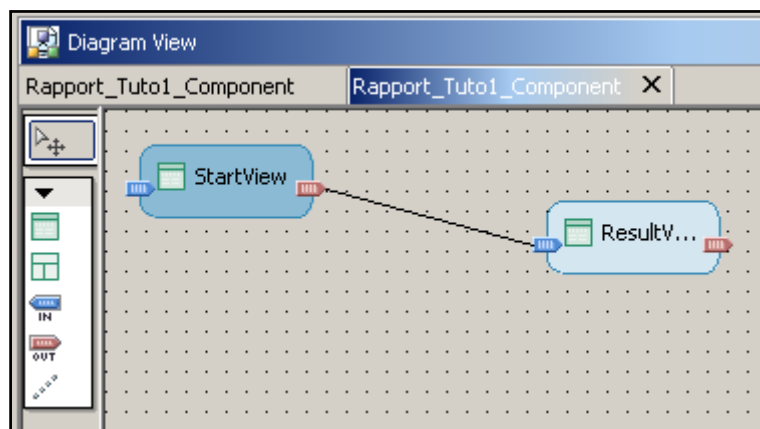


Figure 113 - Tutoriel 1: liaison sortie → entrée

Nous répétons ensuite la même opération en partant cette fois-ci du point de sortie "toStartView" et en allant vers le point d'entrée "fromResultView".

Au final, notre schéma ressemble à cela:

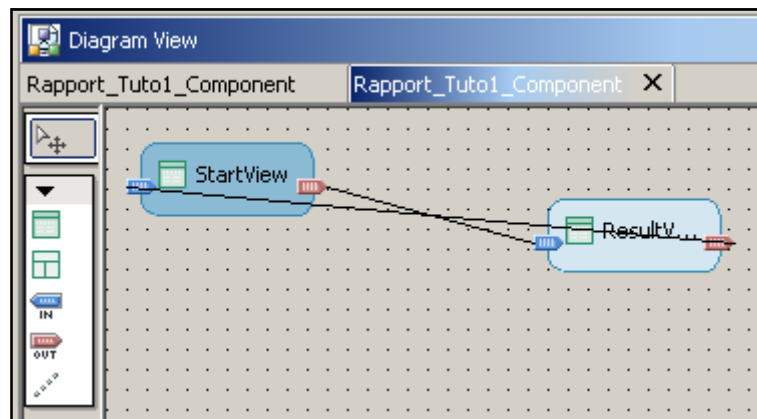


Figure 114 - Tutoriel 1: liaison sortie → entrée

7.3.4.2 Actions

L'étape qui suit la création de la navigation consiste à créer les actions. Celles-ci permettront de configurer très facilement les boutons de l'application qui mèneront d'une vue à une autre. Il y aura deux actions:

- Pour la vue "StartView", nous allons créer une action "toPDF" qui nous renverra vers la vue incorporant le formulaire interactif.
- Pour la vue "ResultView", nous allons créer une action "back" qui nous ramènera vers la vue de départ.

Commençons par la vue "StartView" en double cliquant sur son nom dans la structure du projet:



Figure 115 - Tutoriel 1: StartView dans Web Dynpro Explorer

Nous cliquons ensuite sur l'onglet "Actions" de la vue. La liste (pour l'instant vide) de ses actions apparaît alors:

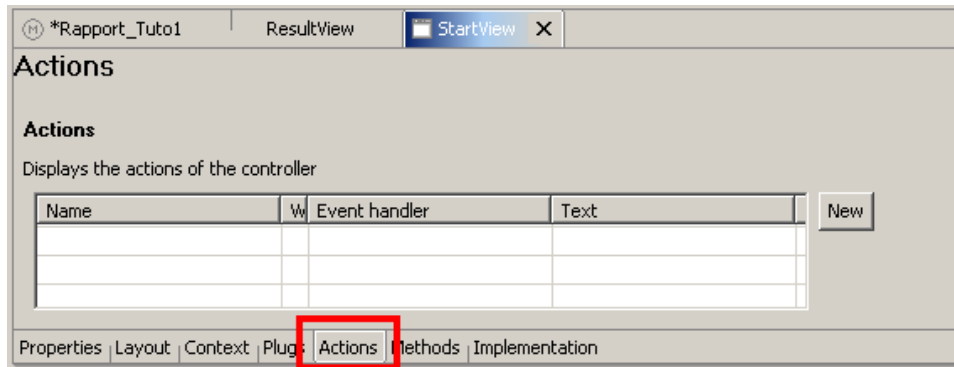


Figure 116 - Tutoriel 1: onglet "Action"

En cliquant sur le bouton "New", une fenêtre nous invite à saisir un nom, un texte ainsi qu'à choisir un "Fire Plug". Nous saisissons les informations suivantes puis cliquons sur le bouton "Finish":

- **Name:** toPDF
- **Text:** Vers le PDF (note: ce texte sera repris par le bouton assigné à cette action)
- **Fire Plug:** toResultView

Cela donne:

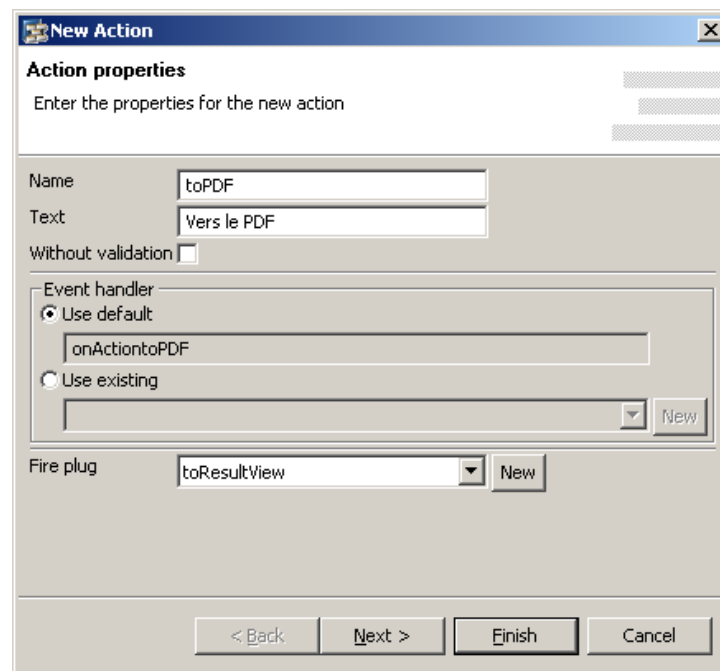


Figure 117 - Tutoriel 1: création de l'action toPDF

Nous répétons ensuite la même étape pour la vue "ResultView". Les informations de l'action sont les suivantes:

- **Name:** back
- **Text:** Retour
- **Fireplug:** toStartView

Ce qui donne:

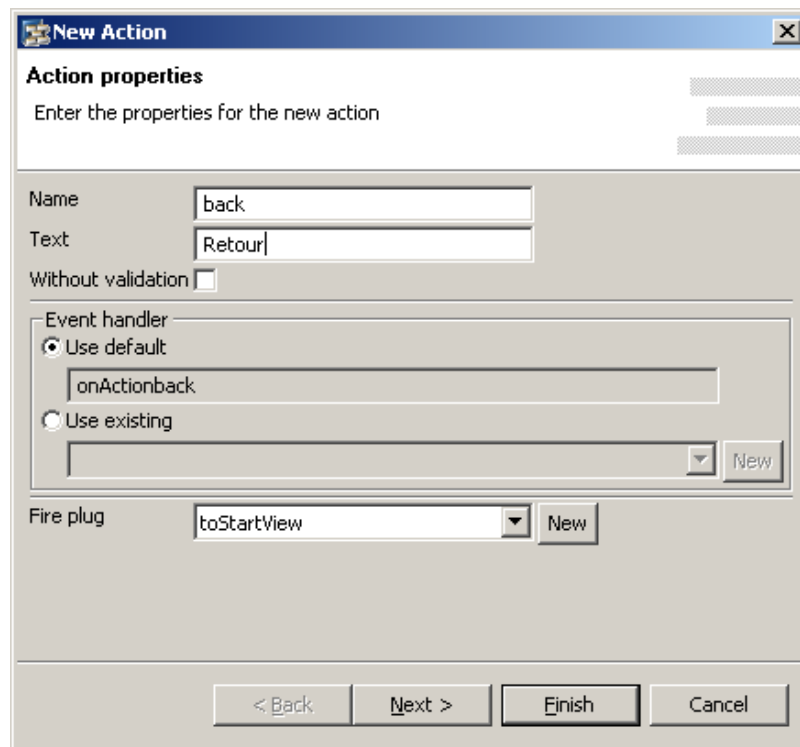


Figure 118 - Tutoriel 1: création de l'action "back"

7.3.4.3 Design de StartView

Une fois les actions créées, nous pouvons passer au design de la vue StartView. Celle-ci comprendra les éléments suivants

- Un titre
- Deux labels
- Deux champs de saisie (InputField)
- Un bouton

Tout d'abord, on double clic la vue "StartView" dans la structure du projet puis sur l'onglet "Layout" afin d'accéder à son interface de design, ce qui donne:

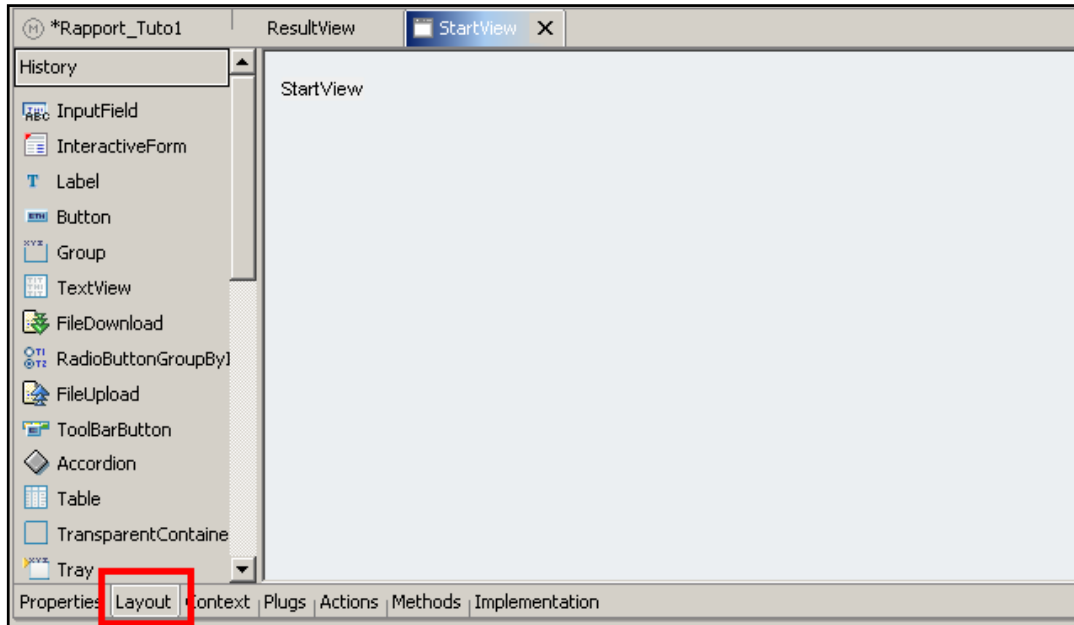


Figure 119 - Tutoriel 1: onglet "Layout"

A l'aide du menu de gauche et de drag & drop, nous déposons dans la vue 2 labels

T Label, 2 champs de saisie **ABC** InputField, et un bouton **BTN** Button.

Une fois ces ajouts effectués, nous retrouvons les éléments dans la fenêtre d'Outline:

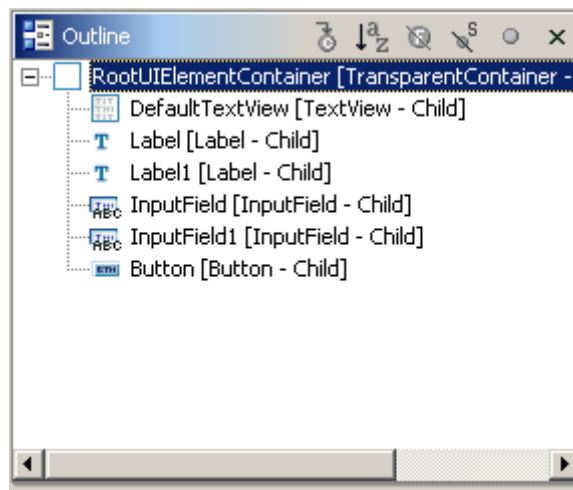


Figure 120 - Tutoriel 1: éléments ajoutés dans la fenêtre d'

Comme nous pouvons le voir dans la fenêtre du designer, les éléments ne sont pas bien disposés:

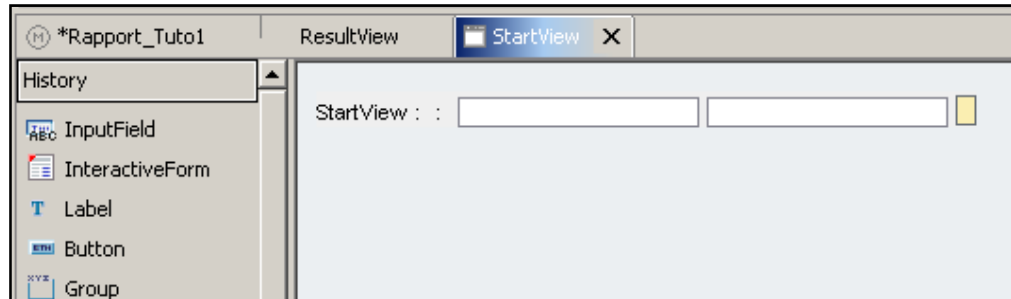


Figure 121 - Tutoriel 1: design de StartView

Nous allons corriger ça en modifiant leurs propriétés et obtenir ainsi une mise en page plus propre.

Pour commencer, nous allons dans les propriétés du conteneur RootUIElementContainer en cliquant sur son nom dans la fenêtre d' puis en cliquant sur l'onglet "Properties" en bas de l'écran. Nous changeons juste 2 propriétés:

- **layout:** GridLayout
- **colCount:** 2

En faisant appel à un GridLayout à 2 colonnes, nous limitons le nombre d'éléments d'interface à 2 par ligne au maximum (typiquement: un label suivi d'un champ de saisie). Si on ajoute un troisième élément, il passera automatiquement à la ligne suivante. Si l'on souhaite avoir uniquement un élément sur une ligne, il suffira d'aller dans ses propriétés et modifier le "colSpan", c'est-à-dire le nombre de colonnes qu'il couvre, de 1 à 2.

Une fois les propriétés du conteneur modifiées, le design évolue mais nécessite visiblement encore quelques réglages:

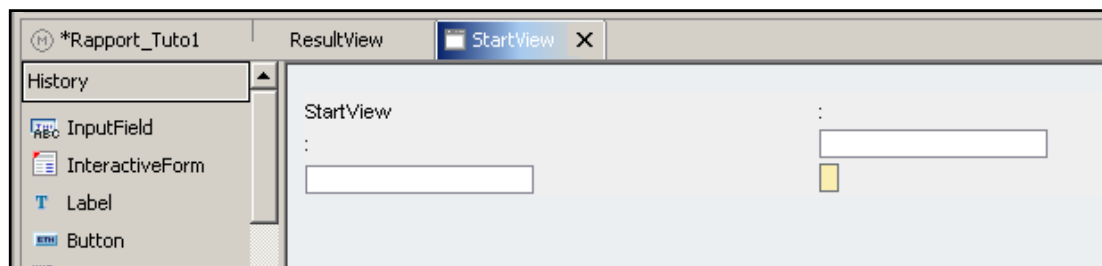


Figure 122 - Tutoriel 1: design de StartView

Nous allons à présent sélectionner quelques éléments dans la fenêtre d' et modifier leurs propriétés. Commençons par l'élément "DefaultTextView":

Cet élément fera office de titre, par conséquent nous allons l'embellir un peu, saisir un texte différent et faire en sorte qu'il occupe les deux colonnes du conteneur. Les propriétés à modifier sont donc:

Properties	
Property	Value
[-] Element Properties [TextView]	
design	header2
enabled	true
hAlign	auto
id	DefaultTextView
layout	native
semanticColor	standard
text	Tutoriel 1: application simple
textDirection	inherit
tooltip	<>
visible	visible
wrapping	false
[-] Layout Data [GridData]	
cellBackgroundDesign	transparent
colSpan	2
hAlign	left
height	
paddingBottom	none
paddingLeft	none
paddingRight	none
paddingTop	none

Figure 123 - Tutoriel 1: propriétés de DefaultTextView

Une fois ces propriétés modifiées, on constate que le design évolue un peu:

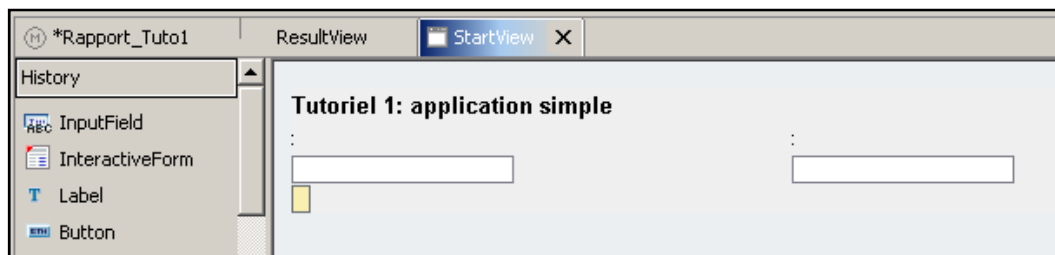


Figure 124 - Tutoriel 1: design de StartView

A présent, la disposition des éléments est correcte mais ils ne sont pas dans le bon ordre. Pour corriger cela, il suffit d'aller dans la fenêtre d'Outline et de réorganiser les éléments via drag & drop, ce qui donne:

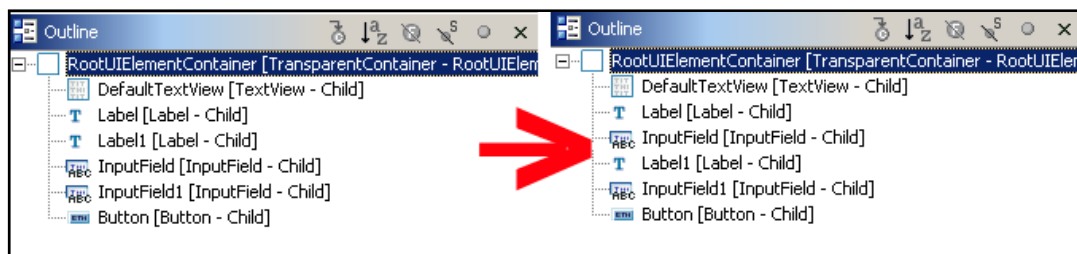


Figure 125 - Tutoriel 1: réorganisation des éléments dans la fenêtre d'

On constate que la vue change:

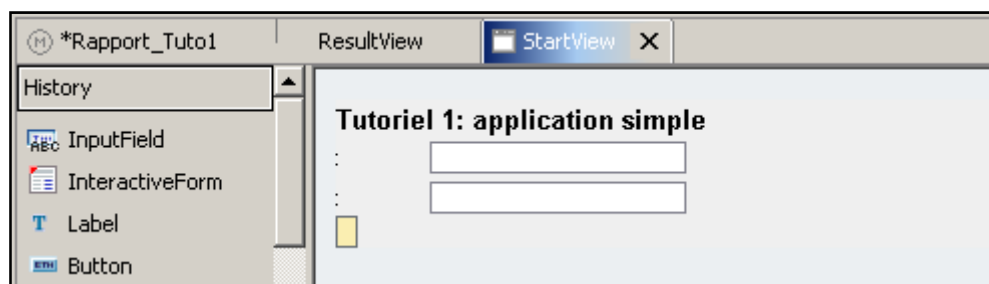


Figure 126 - Tutoriel 1: design de StartView

Les éléments sont à présent organisés comme on le souhaite. Nous allons maintenant modifier les propriétés des labels, à savoir:

Élément	Propriété	Valeur
Label	text	Nom
Label1	text	Prénom

Cela donne:

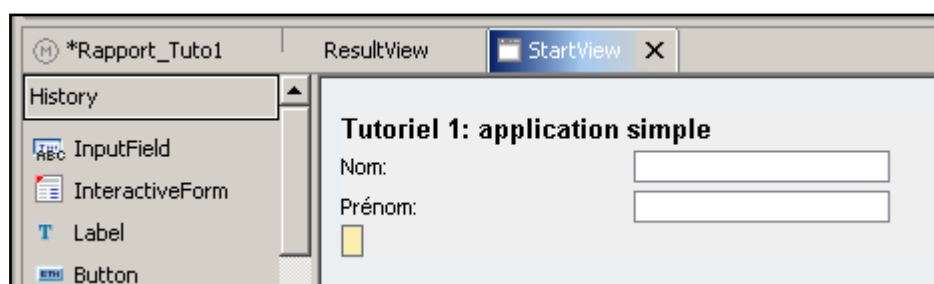


Figure 127 - Tutoriel 1: design de StartView

C'est au tour du bouton pour lequel nous modifions 2 propriétés:

Élément	Propriété	Valeur
Button	onAction	ToPDF
	colSpan	2

La vue ressemble alors à ça:

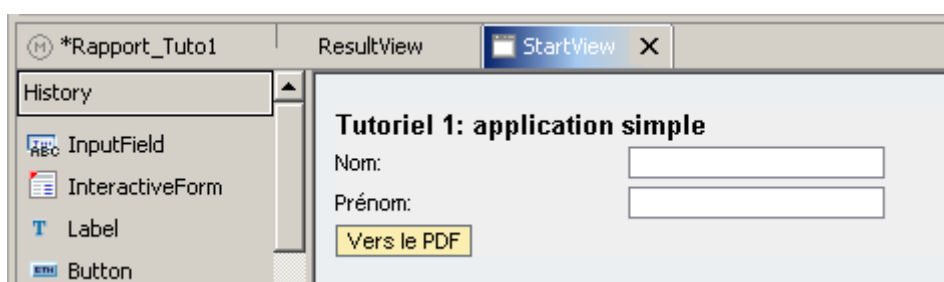


Figure 128 - Tutoriel 1: design de StartView

Nous remarquons au passage que le bouton a hérité du texte que nous avons défini pour l'action "toPDF".

Nous laissons pour l'instant les propriétés des champs de saisie de côté. En effet, ils feront l'objet de plusieurs réglages plus tard et il est préférable de tout faire en une fois.

7.3.4.4 Définition du contexte

Passons maintenant au contexte. Nous allons créer un nœud de données très simple qui contiendra uniquement le nom et le prénom d'une personne. Nous allons aussi en profiter pour créer un attribut de type binaire qu'on définira comme source du futur fichier PDF.

Etant donné que les données du contexte devront être disponibles pour les deux vues de l'application, nous allons créer un contexte au niveau du "Component Controller". Cela nous permettra d'obtenir une centralisation des données vers lesquelles il suffira d'effectuer un mapping personnalisé en fonction de la vue.

Pour ce faire, on double clic "Component Controller" dans la structure du projet:

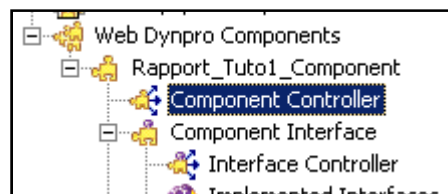


Figure 129 - Tutoriel 1: Component Controller

On clique ensuite sur l'onglet "Context" pour afficher le contexte du contrôleur:

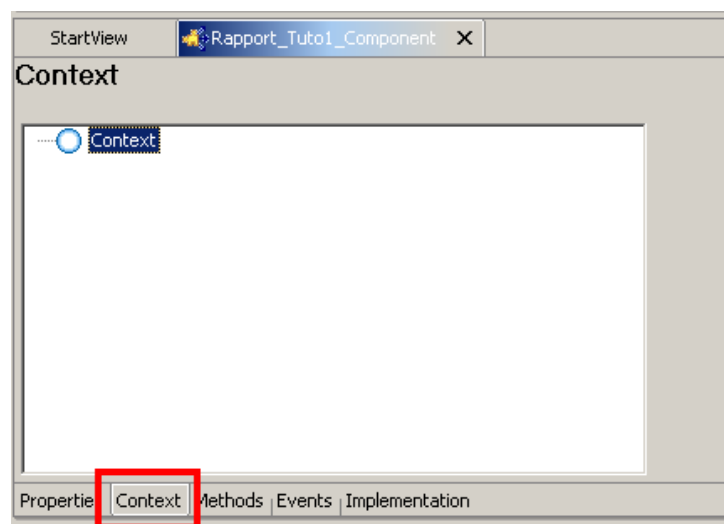



Figure 130 - Tutoriel 1: Onglet Context

On ajoute ensuite un nœud en effectuant un clic droit sur  Context et en sélectionnant New → Value Node:

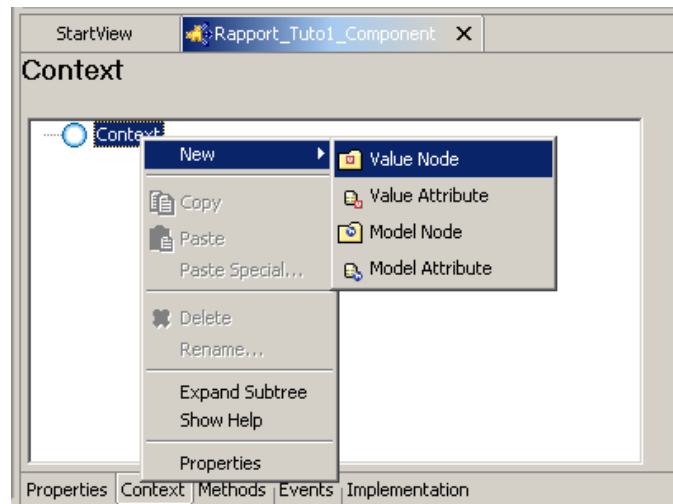


Figure 131 - Tutoriel 1: Création d'un nœud de valeurs

On le nomme "Personne" puis on clique sur "Finish":

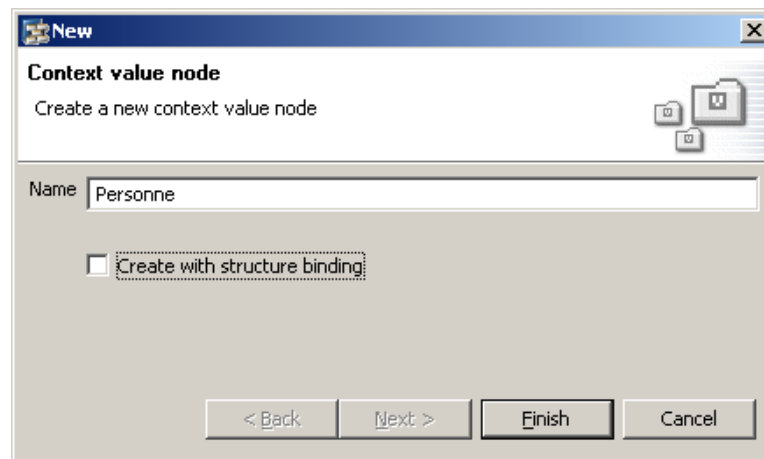


Figure 132 - Tutoriel 1: nommage du nœud

On fait ensuite un clic droit sur le nœud nouvellement créé et on sélectionne New → Value Attribute:

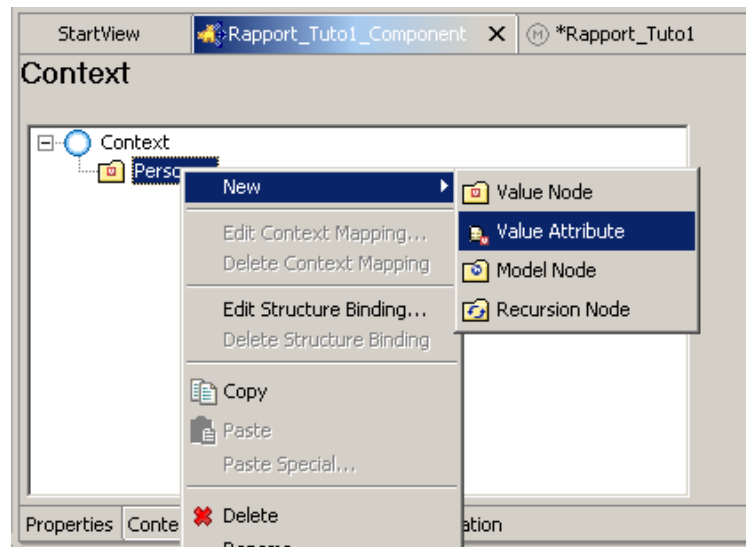


Figure 133 - Tutoriel 1: création d'un attribut

On ajoute ainsi un attribut que l'on nomme "Nom" puis un autre que l'on appellera "Prenom". Notre nœud ressemblera alors à ça:

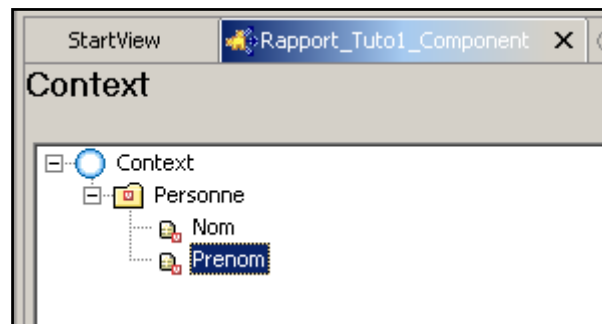


Figure 134 - Tutoriel 1: Context du component

Il ne reste plus qu'à ajouter un attribut qu'on nommera "pdfSource". Attention à 2 choses:

- l'attribut doit être créé à la racine du contexte et non dans le nœud "Personne".
- Une fois créé, il faut modifier son type dans ses propriétés et sélectionner "binary" étant donné que cet attribut correspond à un fichier.

Le contexte de la vue est ainsi terminé:

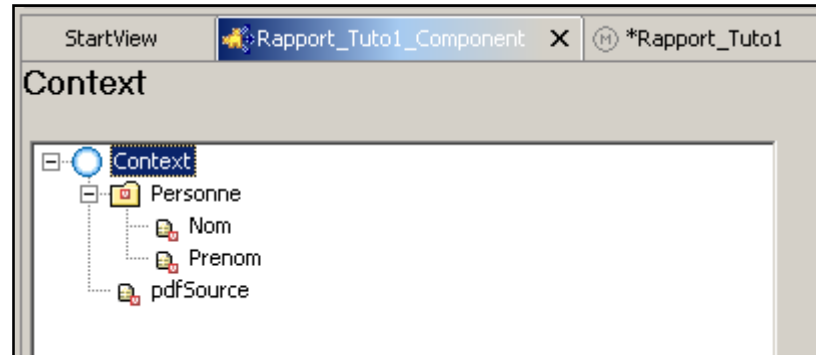


Figure 135 - Tutoriel 1: Contexte du composant

Nous avons maintenant un contexte centralisé.

7.3.4.5 Mappings

L'étape suivante consiste à créer deux mappings, l'un pour la vue "StartView", l'autre pour "ResultView". Une manière simple d'effectuer cette opération consiste à faire un clic droit sur "Rapport_Tuto1_Component" dans la structure du projet et d'ouvrir le "Data Modeler":

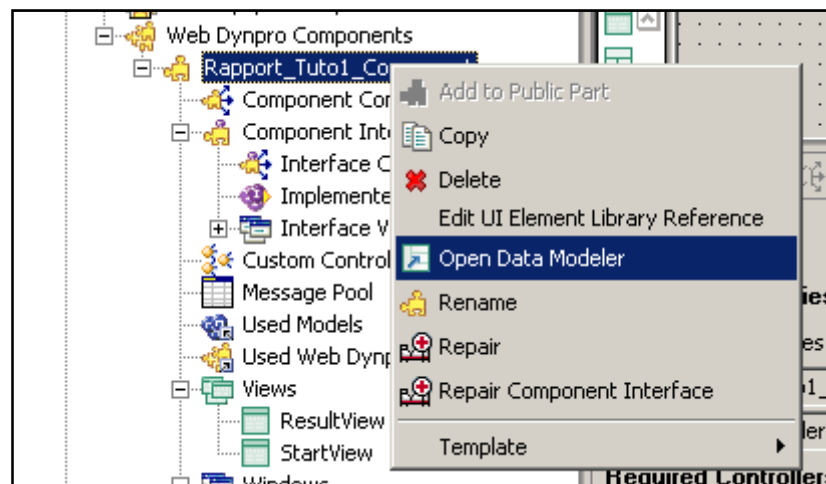


Figure 136 - Tutoriel 1: Ouverture du Data Modeler

Un schéma de l'application s'affiche alors. Nous y retrouvons notamment les 2 vues et le contrôleur qui dispose du contexte que nous venons de créer:

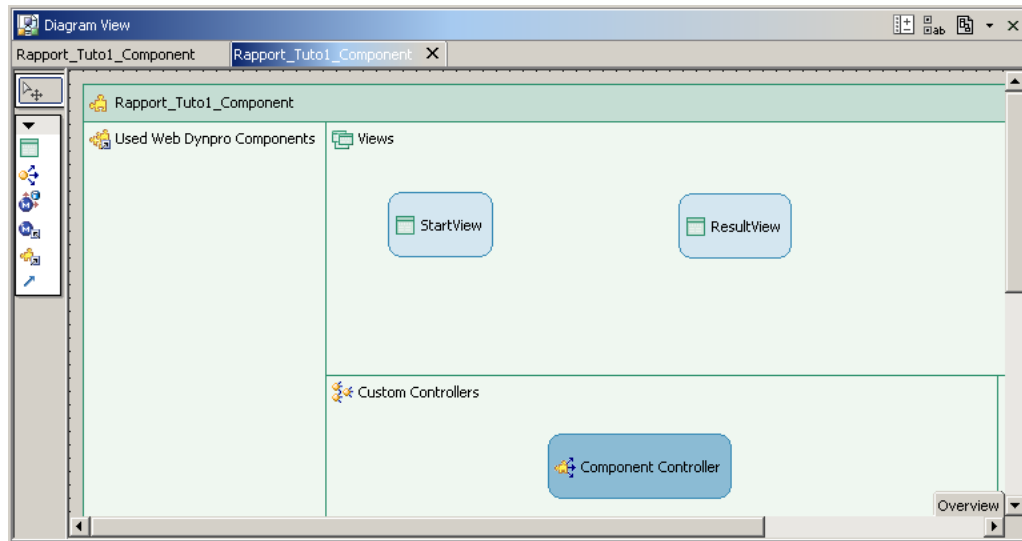



Figure 137 - Tutoriel 1: Data modeler, schéma de l'application

Premièrement, nous allons créer le mapping entre "StartView" et le contexte du contrôleur. En cliquant sur  puis en effectuant un drag & drop depuis la vue jusqu'au contrôleur, une fenêtre s'affiche et nous propose de configurer le mapping:

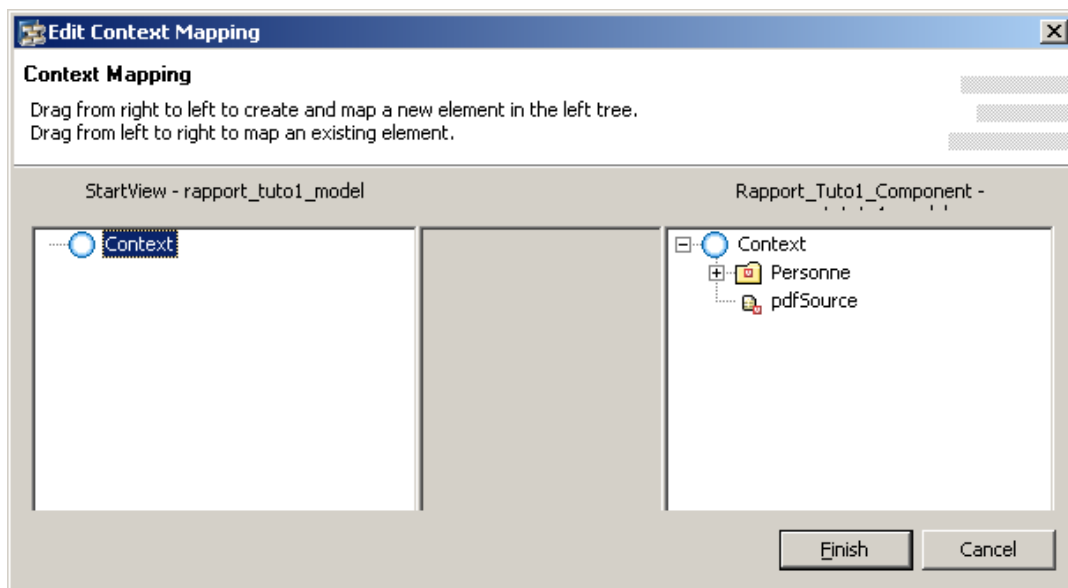



Figure 138 - Tutoriel 1: étape pré-mapping

Pour ajouter un élément au contexte de la vue "StartView", il suffit simplement d'effectuer un drag & drop depuis la partie de droite de la fenêtre vers la racine¹⁴ du contexte de la vue. Dans le cas présent, nous voulons effectuer un mapping avec le nœud "Personne". Après avoir fait le drag & drop, une fenêtre s'affiche automatiquement et permet de personnaliser davantage le mapping en sélectionnant les attributs qui nous intéressent:

¹⁴ Correspond à  Context.

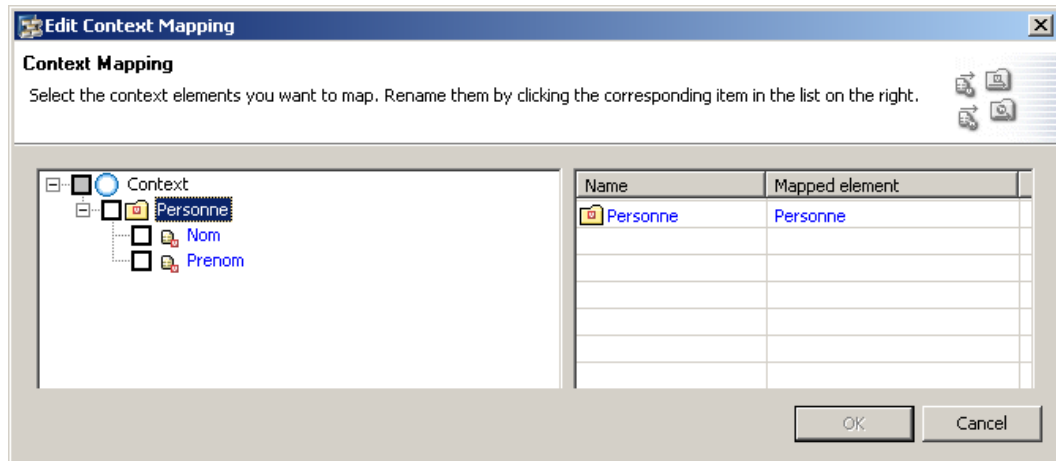


Figure 139 - Tutoriel 1: choix des nœuds/attributs à mapper

Etant donné que nous avons besoin de tous les attributs, nous cliquons simplement sur le nœud "Personne" pour tous les sélectionner. On valide ensuite en cliquant sur "OK":

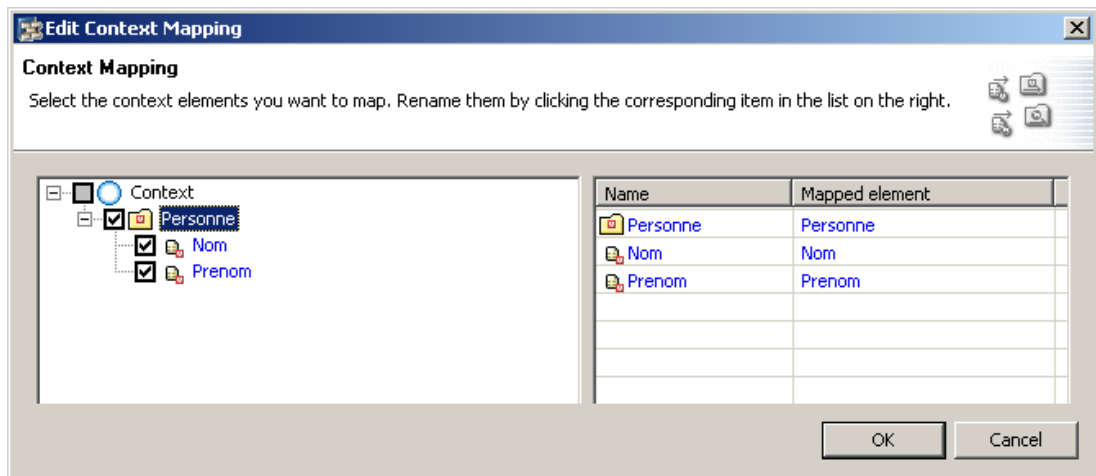


Figure 140 - Tutoriel 1: cochages des nœuds/attributs à mapper

Nous sommes ensuite renvoyés vers la fenêtre précédente qui montre qu'un mapping a bien été créé:

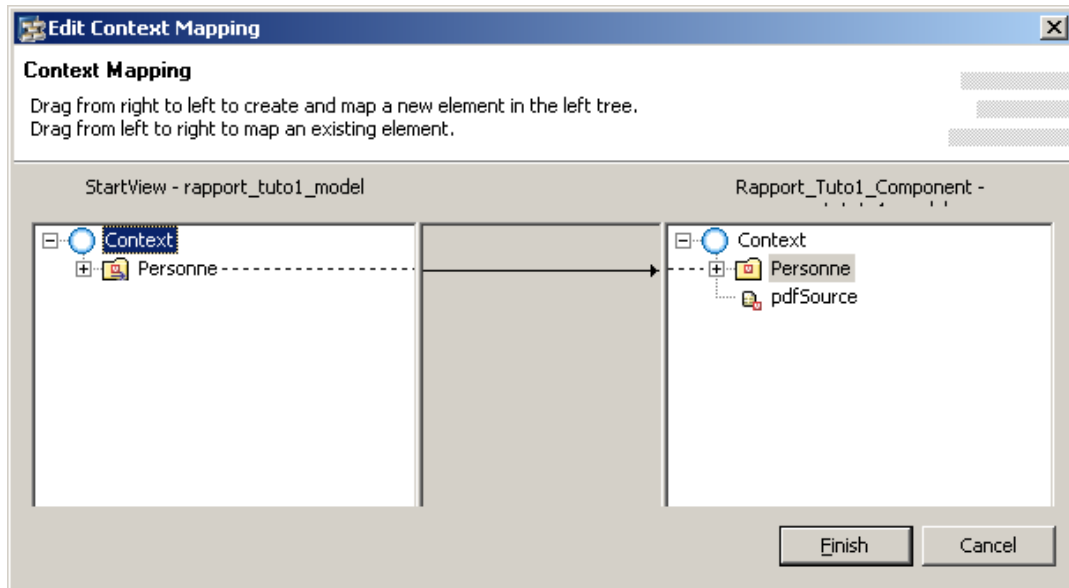


Figure 141 - Tutoriel 1: mapping entre la vue et le component

Nous n'avons pas besoin d'autres attributs, nous pouvons donc cliquer sur "Finish" pour terminer la configuration du mapping. On constate alors que le schéma de l'application a légèrement changé et indique à présent une flèche allant de la vue "StartView" vers le contrôleur:

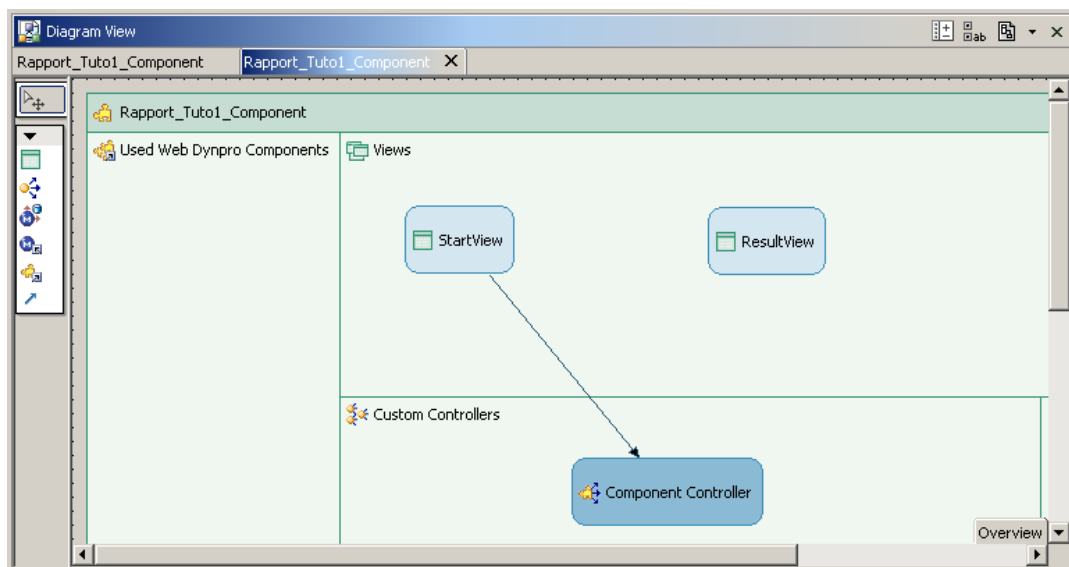


Figure 142 - Tutoriel 1: Data Modeler

Nous répétons maintenant la même étape pour la vue "ResultView". Pour le mapping, on a besoin de la totalité du nœud "Personne" ainsi que de l'attribut "pdfSource" étant donné que cette vue contiendra un formulaire. Avant sa validation, le mapping pour "ResultView" ressemblera donc à ça:

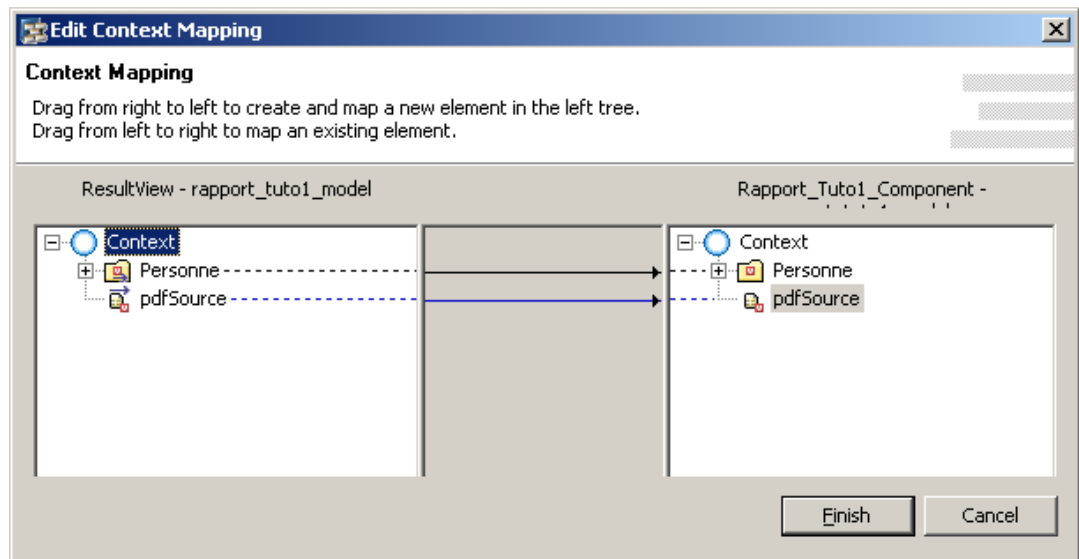


Figure 143 - Tutoriel 1: mapping entre la vue et le component

Le schéma de l'application est à nouveau mis à jour et indique le nouveau mapping:

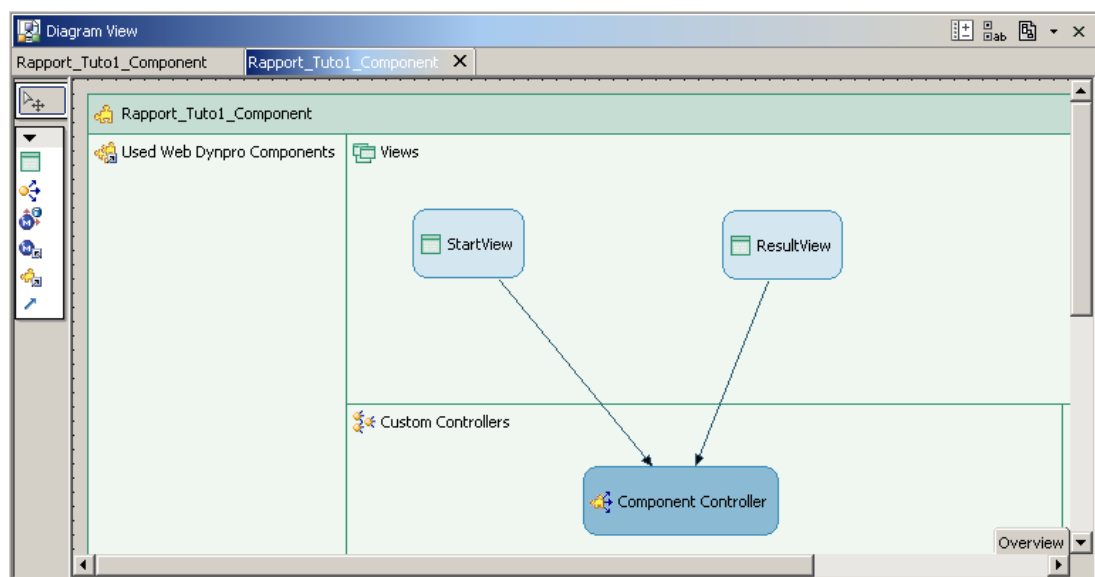


Figure 144 - Tutoriel 1: Data Modeler

Le mapping est à présent terminé et les données que nous avons mappées sont disponibles d'une vue à une autre.

La suite consiste à lier les différents éléments de saisie des deux vues à leurs contextes respectifs afin de pouvoir garantir une certaine persistance de l'information.

Commençons par la vue "StartView" en double cliquant sur son nom dans la structure du projet et en cliquant ensuite sur l'onglet "Layout". Nous retrouvons alors le design de la vue:

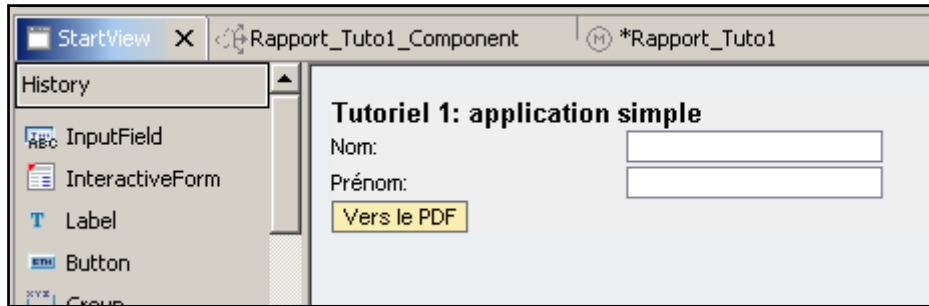



Figure 145 - Tutoriel 1: vue StartView

Rappelez-vous, nous avons laissé de côté la modification des propriétés des champs de saisie (InputField). Maintenant que le contexte existe, nous pouvons terminer l'opération. Pour ce faire, il suffit de sélectionner un champ, d'aller dans ses propriétés, de cliquer sur "value" et enfin sur le bouton  qui apparaît. Une fenêtre nous demande alors de sélectionner quel attribut nous souhaitons lier à l'InputField que nous modifions:

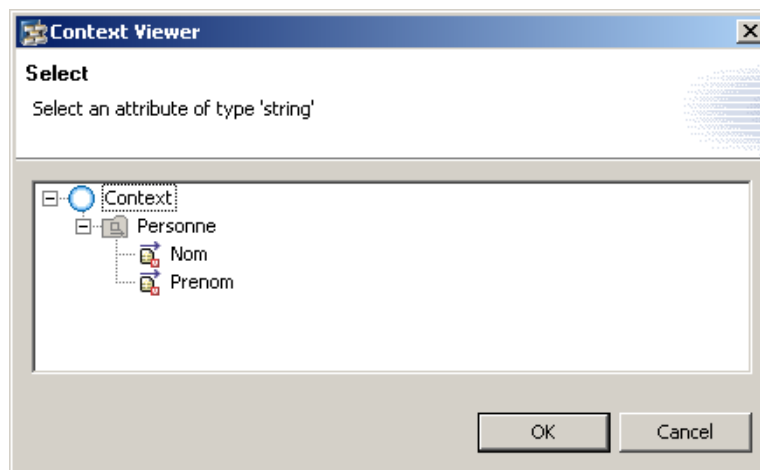


Figure 146 - Tutoriel 1: sélection de l'attribut à lier au champ

Pour un champ nous sélectionnons l'attribut "Nom" et pour l'autre l'attribut "Prenom". Une fois l'opération effectuée, les champs de saisie que l'on voit dans le design affichent leurs liaisons:

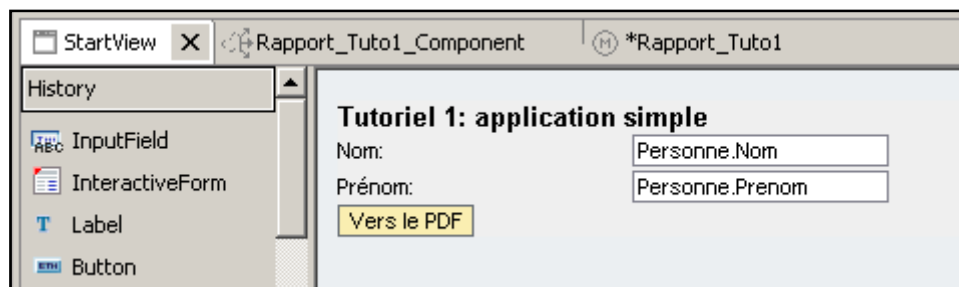



Figure 147 - Tutoriel 1: vue StartView

Remarque: Lors d'une liaison, il se pourrait qu'une croix sur fond rouge soit visible à côté de la valeur de la propriété "value", comme par exemple:

tooltip	<>
value	✖ Personne.Nom
visible	visible

Figure 148 - Tutoriel 1: valeur d'un champ (invalide)

Pour remédier à ça, il suffit simplement de sauvegarder le projet en cliquant sur le bouton  (Save all metadata) au sommet de la fenêtre. Et revenant sur les propriétés du champ de saisie, on pourra alors voir un vu sur fond vert, indiquant ainsi une liaison valide:

tooltip	<>
value	✔ Personne.Nom
visible	visible

Figure 149 - Tutoriel 1: valeur d'un champ (valide)

7.3.4.6 Design de la 2^{ème} vue

Nous pouvons maintenant nous occuper de la vue "ResultView". Son design n'a pas encore été fait, nous allons donc en profiter pour remédier à ça en double cliquant sur son nom dans la structure du projet et en cliquant ensuite sur l'onglet "Layout". Le design apparaît alors:

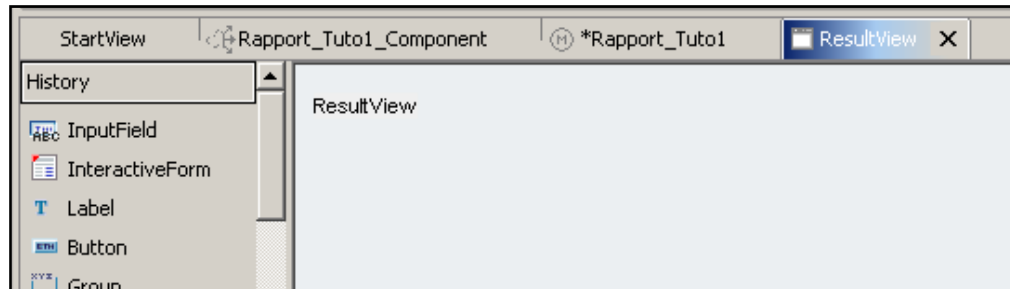




Figure 150 - Tutoriel 1: vue ResultView

Nous allons commencer par ajouter via drag & drop un bouton , puis un formulaire interactif  :

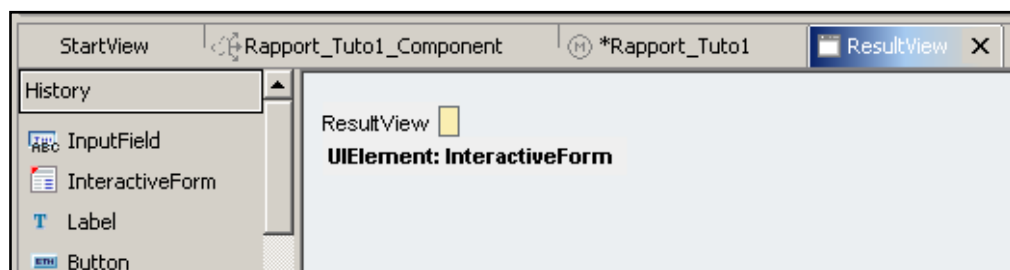


Figure 151 - Tutoriel 1: vue ResultView

En cliquant sur les différents éléments dans la fenêtre d' puis sur l'onglet "Propriétés" en bas de l'écran, nous changeons les propriétés suivantes:

Elément	Propriété	Valeur
RootUIElementContainer	Layout	gridLayout
	colCount	1
DefaultTextView	Design	Header2
	Text	Tutoriel 1: Application simple – Formulaire PDF
Button	onAction	Back
InterActiveForm	dataSource	Personne
	pdfSource	pdfSource
	Height	600px
	Width	700px

Le design de la vue ressemble alors à ça:

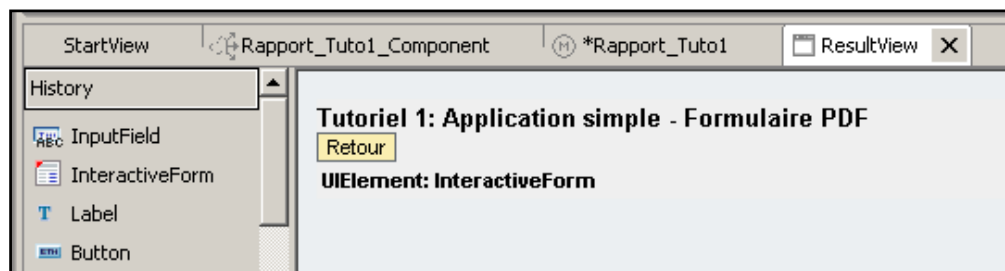


Figure 152 - Tutoriel 1: vue ResultView

7.3.4.7 Design du formulaire

En termes de design, il ne reste plus qu'à réaliser celui du formulaire lui-même. Pour ouvrir Adobe LiveCycle Designer, il suffit de cliquer sur le formulaire interactif dans la fenêtre d' en bas à gauche et de sélectionner "Edit":

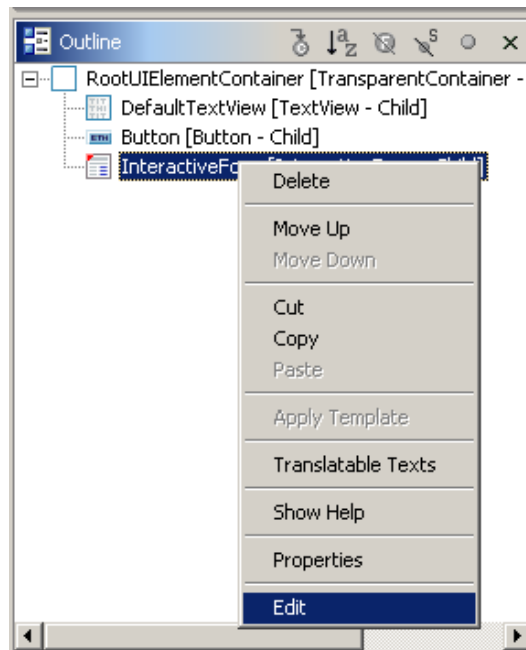


Figure 153 - Tutoriel 1: lancement d'Adobe LiveCycle Designer

Une fois Adobe LiveCycle Designer lancé, la première chose à faire est de double cliquer sur l'onglet portant le nom du formulaire afin de laisser l'éditeur occuper tout l'espace disponible:

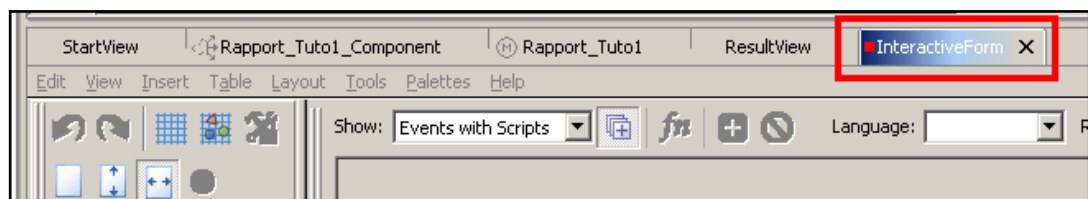


Figure 154 - Tutoriel 1: onglet du formulaire interactif

Pour le design du formulaire, nous allons en réaliser un très simple. Il contiendra:

- Un titre
- 2 champs de saisie pour les attributs "Nom" et "Prenom"
- 1 bouton "Submit"
- Optionnel: un calcul simple (nécessite 3 champs de saisie supplémentaires et une ligne de code). Cette étape peut être sautée.

L'espace de travail ressemble donc à cela pour le moment:

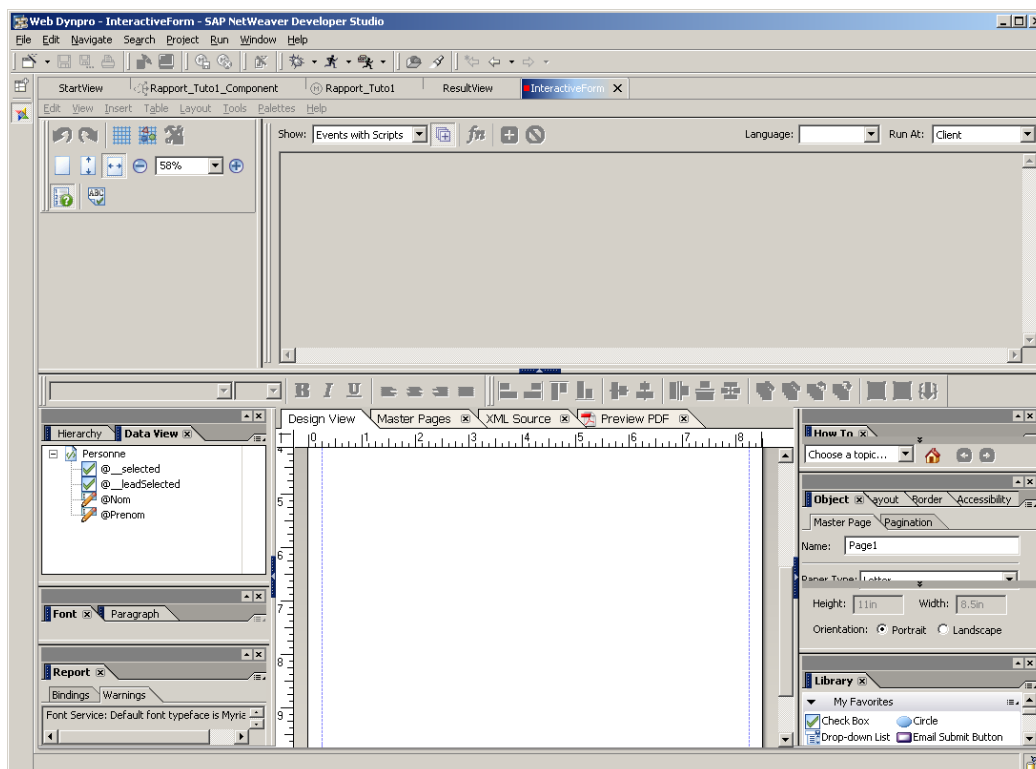



Figure 155 - Tutoriel 1: Adobe LiveCycle Designer

Le premier élément que nous allons créer est un titre. Pour ce faire, nous sélectionnons  Text dans la fenêtre "Library" (en bas à droite sur l'image précédente) et cliquons en haut à gauche de la page PDF afin de déposer cet élément:

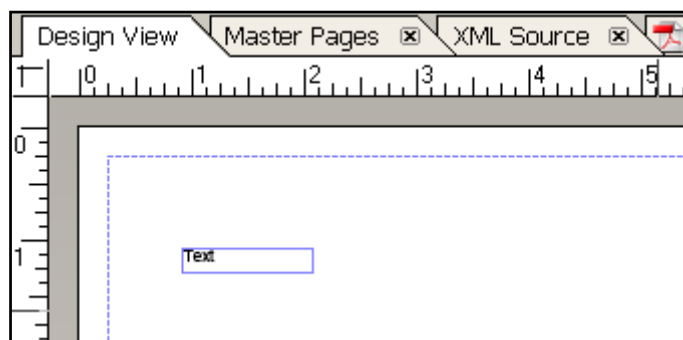


Figure 156 - Tutoriel 1: élément Text

Ensuite, on joue avec la taille du titre, son texte ainsi que celle de sa police pour obtenir un résultat plus visible, comme par exemple:

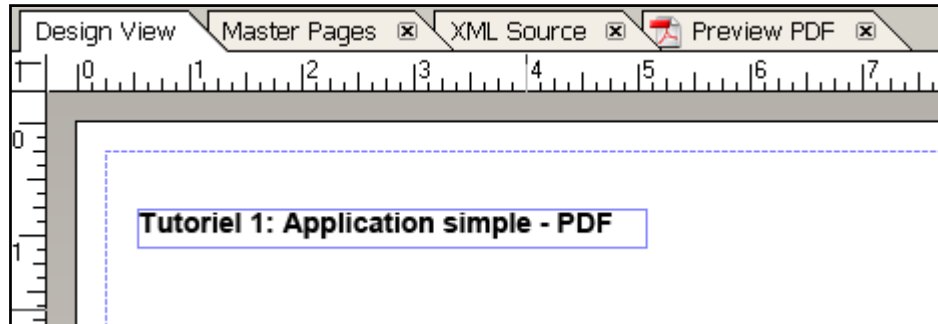


Figure 157 - Tutoriel 1: modification du titre

Note: si un champ, peu importe lequel, affiche un petit carré rouge à son extrémité inférieure droite, cela signifie que l'élément n'a pas la place d'afficher tout ce qu'il contient et qu'il faudrait l'agrandir un peu. Exemple:

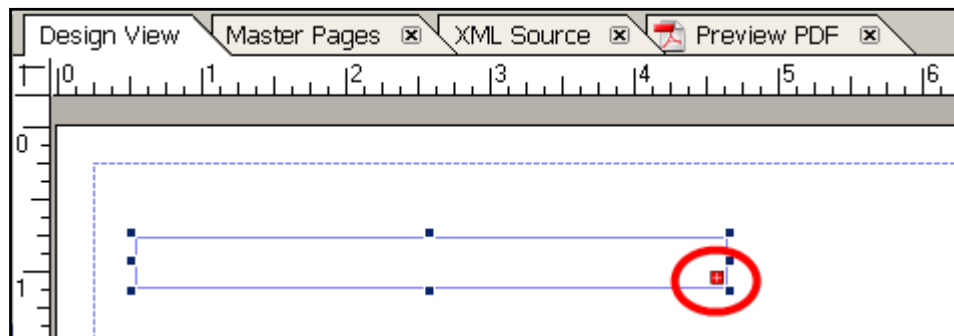


Figure 158 - Tutoriel 1: dépassement d'un champ

A présent, nous allons ajouter les 2 champs de saisie reliés aux attributs "Nom" et "Prénom". Contrairement au titre, nous n'allons pas sélectionner un élément dans la librairie. En effet, en liant le formulaire au nœud "Personne", tous les attributs disponibles s'affichent dans la fenêtre "Data View" dans la partie gauche de l'espace de travail:

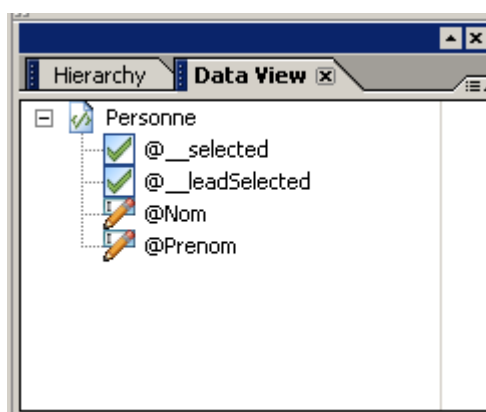


Figure 159 - Tutoriel 1: Data View

Cela a l'avantage de permettre au développeur de créer des champs de saisie immédiatement utilisables. A partir de la fenêtre "Data View", il suffit d'effectuer un drag & drop pour chaque attribut désiré vers la page PDF. Nous obtenons ainsi des champs déjà liés aux données et qui font appel dès le départ au bon type de donnée (ici: données de type texte). Nous voulons ajouter un champ pour les attributs "Nom" et "Prenom", par conséquent nous glissons "@Nom" puis "@Prenom" vers la page PDF. Le résultat est comparable à celui de l'image ci-dessous:

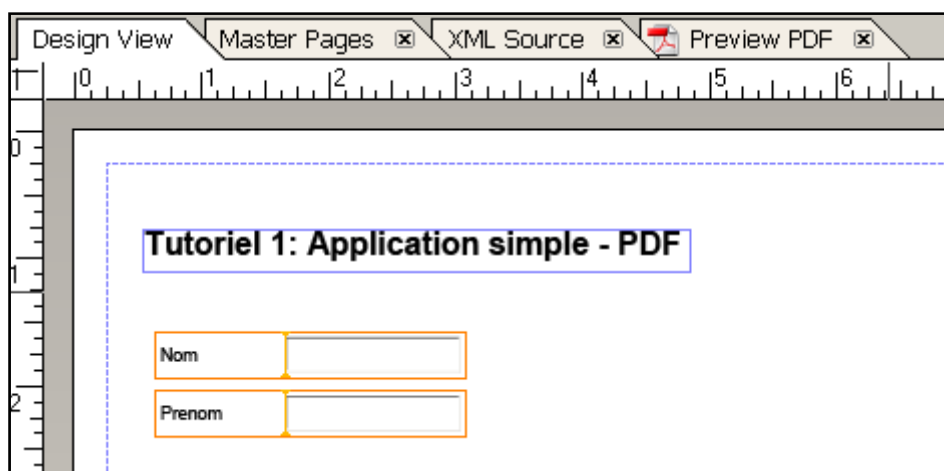


Figure 160 - Tutoriel 1: champs de saisie

Une fois les champs ajoutés, nous pouvons constater que 2 petites icones ont été rajoutées dans la fenêtre "Data View", attestant ainsi que ces attributs sont bien utilisés dans le formulaire:

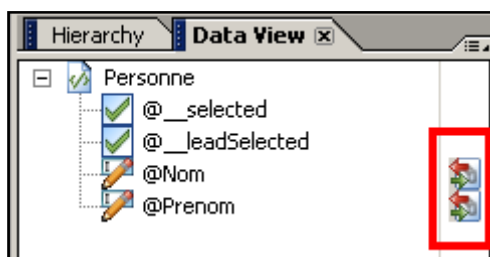


Figure 161 - Tutoriel 1: attributs liés dans le formulaire

Il ne reste plus qu'à ajouter un bouton qui permettra de sauvegarder les données saisies. Pour ce faire, nous retournons dans la librairie et ajoutons un bouton "Submit" disponible dans la catégorie "WebDynproActiveX":

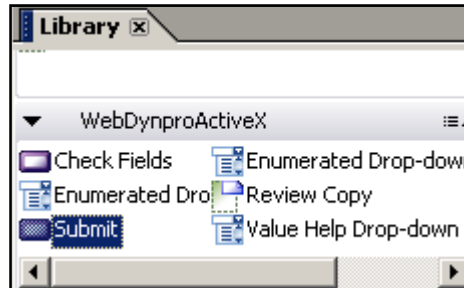


Figure 162 - Tutoriel 1: bouton (ActiveX)

Nous obtenons alors un résultat similaire à ceci:

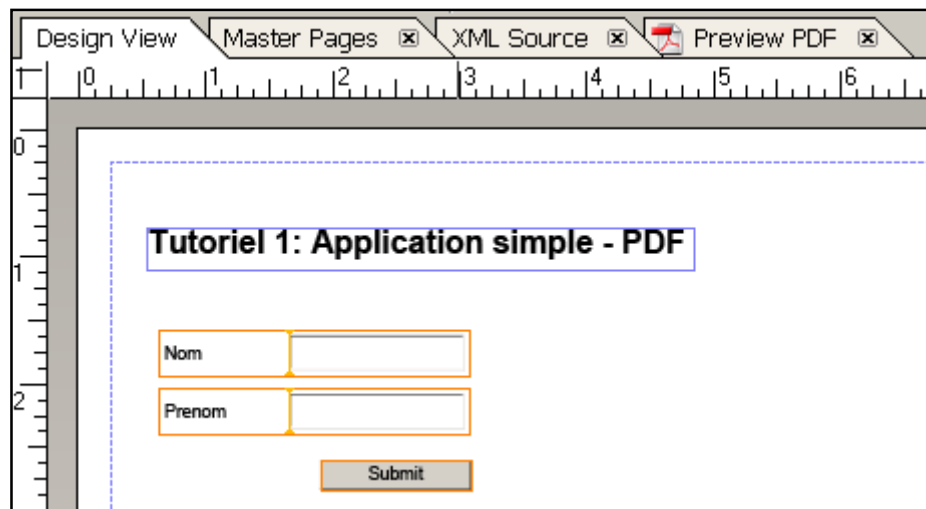


Figure 163 - Tutoriel 1: ajout du bouton dans le formulaire

L'avantage de ce bouton "Submit" est qu'il est lui aussi prêt à l'emploi. En effet, si on clique sur lui, une ligne de code apparaît dans la partie supérieure de l'éditeur:

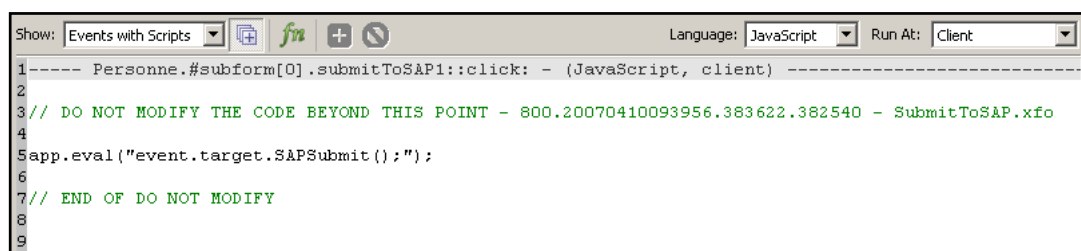


Figure 164 - Tutoriel 1: code du bouton "Submit"

Il faut aussi savoir que l'application qui contient le formulaire interactif peut récupérer l'évènement qui consiste à cliquer sur un bouton "Submit" dans le formulaire. En effet, si on regarde les propriétés d'un formulaire interactif, nous y trouvons 2 évènements possibles, notamment "onSubmit":

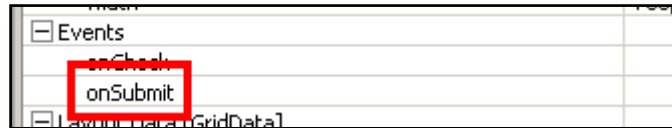


Figure 165 - Tutoriel 1: événement onSubmit du formulaire dans SAP NetWeaver Developer Studio

A partir de là, il devient alors possible d'effectuer des opérations supplémentaires dans l'application en fonction de nos besoins, comme par exemple traiter des données saisies en vue de les sauvegarder d'une manière spécifique dans un nœud.

Le design du formulaire est donc déjà terminé.

7.3.4.8 Calcul simple

L'étape suivante est optionnelle, dans la mesure où elle sert uniquement à voir comment on peut réaliser un calcul simple dans un formulaire. Cette étape ne sera pas reprise dans la suite du tutoriel.

Nous allons réaliser un calcul qui consiste à additionner les valeurs de 2 champs et d'afficher le résultat dans un 3^{ème} champ. Nous allons donc dans la librairie et ajoutons 3 champs numériques:

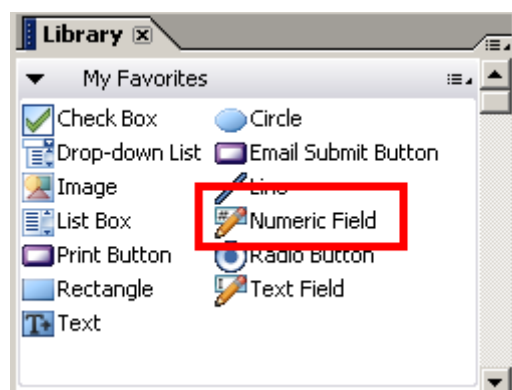


Figure 166 - Tutoriel 1: champ numérique

Nous renommons ensuite les labels des champs afin de les rendre plus parlants. Il suffit de cliquer sur eux:

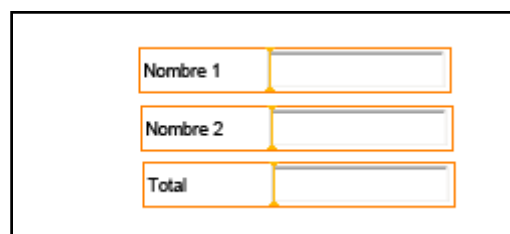


Figure 167 - Tutoriel 1: champs de saisie

Ensuite, on clique le champ qui correspond au total et on sélectionne "calculate" dans la liste déroulante de la partie relative aux scripts:

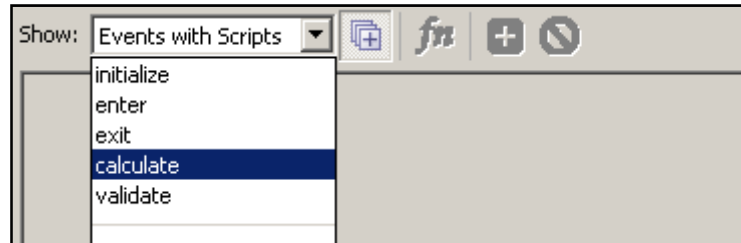


Figure 168 - Tutoriel 1: choix de l'évènement "calculate"

On change ensuite le langage, étant donné qu'on souhaite utiliser du FormCalc:

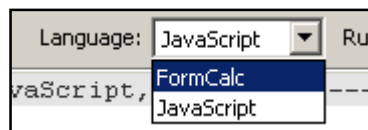


Figure 169 - Tutoriel 1: choix du langage

Il suffit alors d'ajouter une ligne de calcul respectant le format suivant:

`<NomDuChampTotal> = <NomDuChamp1> + <NomDuChamp2>`

Pour retrouver le nom d'un champ, il suffit de cliquer sur l'un d'eux et de cliquer sur l'onglet "Object" dans la partie droite de la fenêtre, puis sur l'onglet "Binding". Le nom apparaît alors dans le champ "Name":

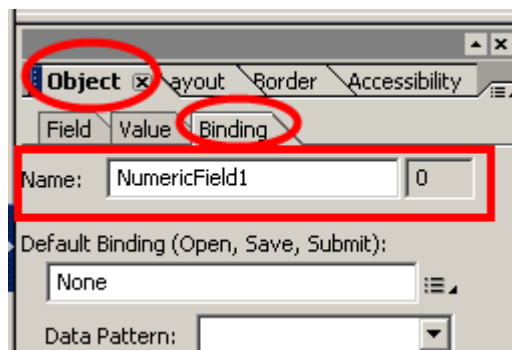


Figure 170 - Tutoriel 1: nom du champ numérique

Le code du calcul sera donc:

```
NumericField3=NumericField2+NumericField1
```



Une fois la ligne ajoutée, il suffit de cliquer sur l'onglet  Preview PDF  pour pouvoir tester le calcul. On s'aperçoit qu'à chaque fois que l'un des 2 premiers champs est modifié, le total est automatiquement mis à jour:

Figure 171 - Tutoriel 1: vérification du calcul

Le développement du formulaire interactif est à présent terminé.

7.3.4.9 Implémentation

La prochaine étape consiste à ajouter quelques lignes de code à l'application. Si ce n'est pas encore fait, on double clique sur l'onglet portant le nom du formulaire interactif afin de revenir dans l'espace de travail Web Dynpro.

Nous allons implémenter qu'une seule chose dans ce tutoriel, la méthode wdDoInit() du contrôleur. Nous double cliquons tout d'abord "Component Controller" dans la structure du projet:

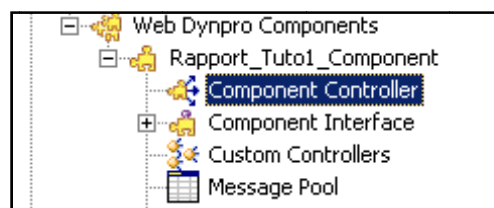


Figure 172 - Tutoriel 1: Component Controller

Ensuite, nous cliquons sur l'onglet "Implementation" pour afficher le code qui se cache derrière le contrôleur:

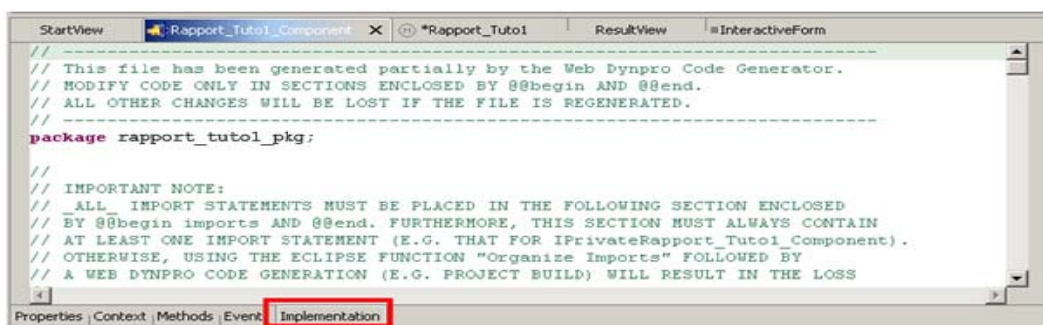


Figure 173 - Tutoriel 1: onglet Implementation

Dans la fenêtre d', nous pouvons voir la structure du code. On y retrouve notamment la méthode que nous allons modifier, wdDoInit(). Nous cliquons alors dessus pour en afficher le code:

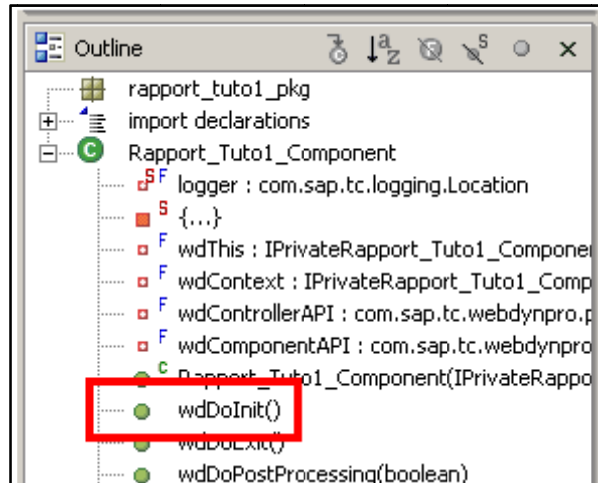


Figure 174 - Tutoriel 1: méthode wdDoInit() dans l'

La méthode wdDoInit() est exécutée lorsque l'application charge le contrôleur. Dans le cas de notre application, cela se produira simplement lors du démarrage.

Le code que nous allons implémenter va ajouter un élément au nœud "Personne". En effet, le fait de déployer une application comportant un nœud vide la rend inutilisable, comme le montre l'exemple ci-dessous (champs grisés):

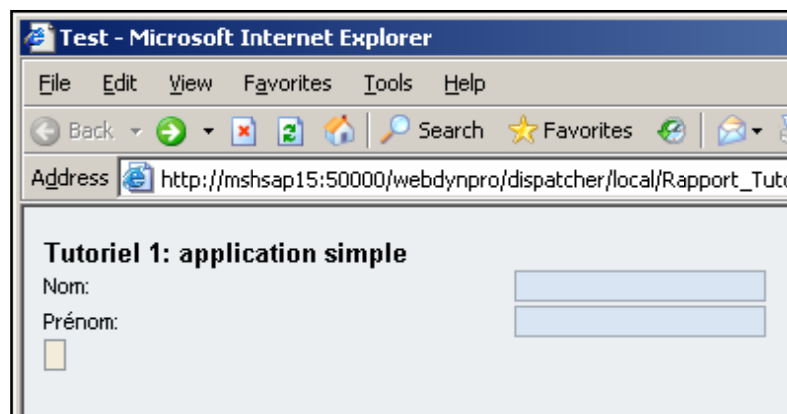


Figure 175 - Tutoriel 1: nœud vide: conséquence

Il existe 2 manières de remédier à ce problème: ajouter quelques lignes de code dans la méthode wdDoInit() afin de rendre le nœud non-vide, ou alors modifier les propriétés du nœud "Personne" et régler la propriété "cardinality" sur "1..n".

Bien que la 2^{ème} option semble être plus simple et plus rapide, elle ne fonctionnera pas plus tard, lorsque nous utiliserons des BAPIs pour échanger des données entre un formulaire et SAP. Par conséquent, il est plus judicieux de prendre dès maintenant les bonnes habitudes et de faire appel à l'option qui consiste à ajouter du code.

Le code à ajouter dans la méthode wdDoInit() est donc le suivant:

```
public void wdDoInit()
{
  //@@begin wdDoInit()
  IPublicRapport_Tuto1_Component.IPersonneElement nodeInput =
    wdContext.createPersonneElement();

  wdContext.nodePersonne().bind(nodeInput);
  //@@end
}
```

La première instruction consiste à créer un élément destiné au nœud "Personne", la seconde le lie au nœud, permettant ainsi d'éviter que "Personne" soit vide. Certes, l'élément "nodeInput" que nous venons d'ajouter ne contient aucune valeur mais cela suffit pour que l'application considère le nœud non vide.

Remarque importante concernant la modification du code: il faut toujours faire attention à ce que l'on code se situe entre les balises //@@begin et //@@end d'une méthode. En effet, si du code est ajouté en dehors de ces dernières, il sera supprimé sans avertissement lors de la prochaine sauvegarde.

7.3.4.10 Création de l'application

Il ne reste plus qu'une seule étape à présent: créer l'application. Pour ce faire, on effectue un clic droit sur "Applications" dans la structure du projet, puis on sélectionne "Create Application":

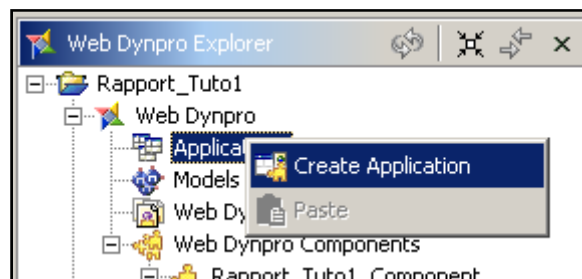


Figure 176 - Tutoriel 1: création de l'application

La fenêtre qui apparaît nous demande de saisir un nom d'application et un nom de package, nous saisissons donc les valeurs suivantes:

- Name: Rapport_Tuto1_Application
- Package: rapport_tuto1_pkg

Cela donne:

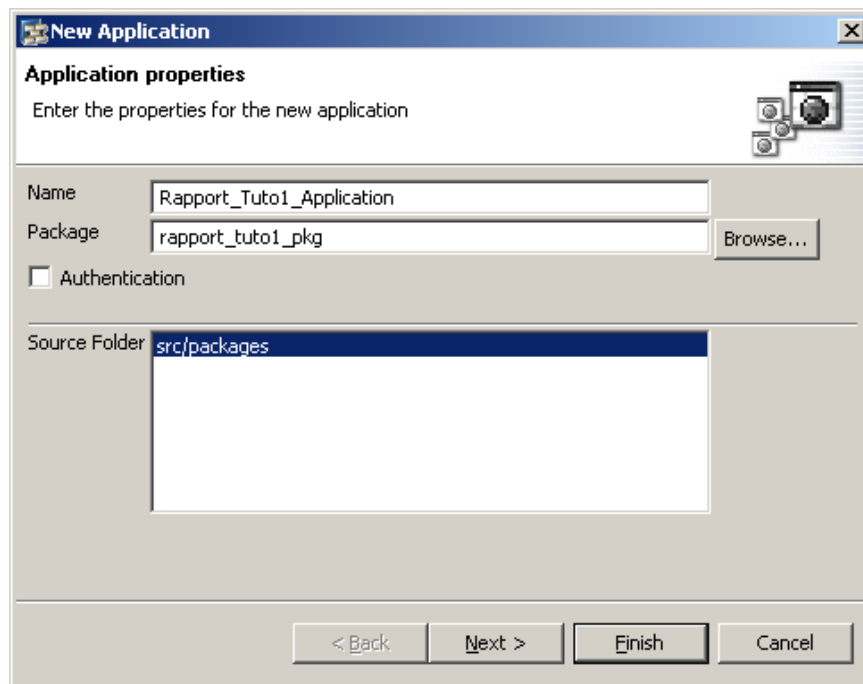


Figure 177 - Tutoriel 1: nommage de l'application

Sur la fenêtre suivante, on indique qu'on veut utiliser un composant existant et on clique sur "Next":

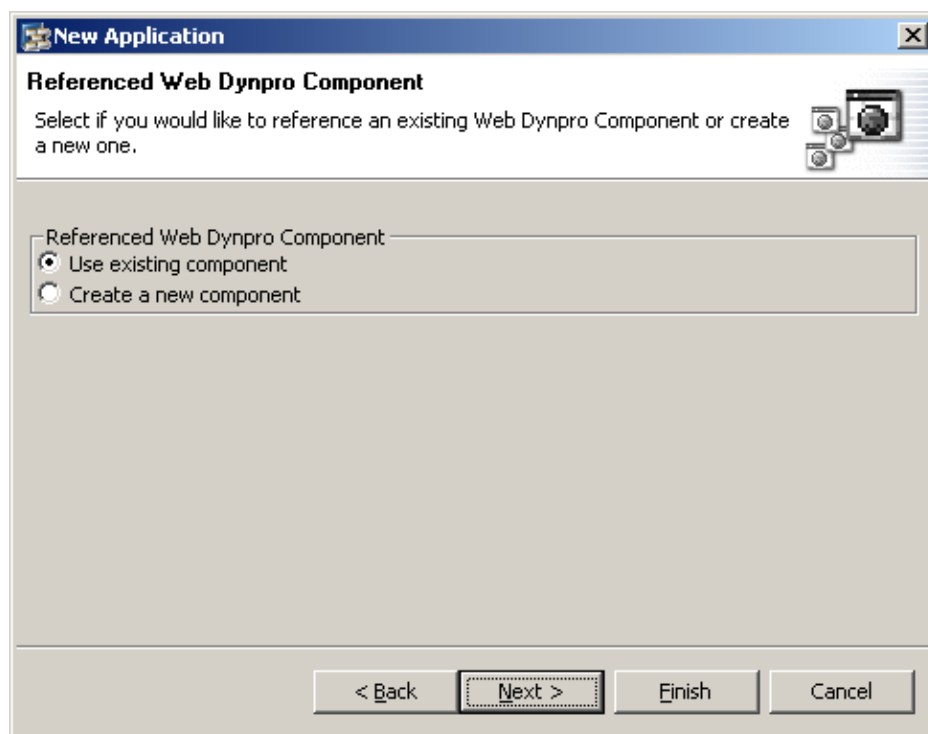


Figure 178 - Tutoriel 1: choix du component

Et enfin on valide en cliquant sur "Finish":

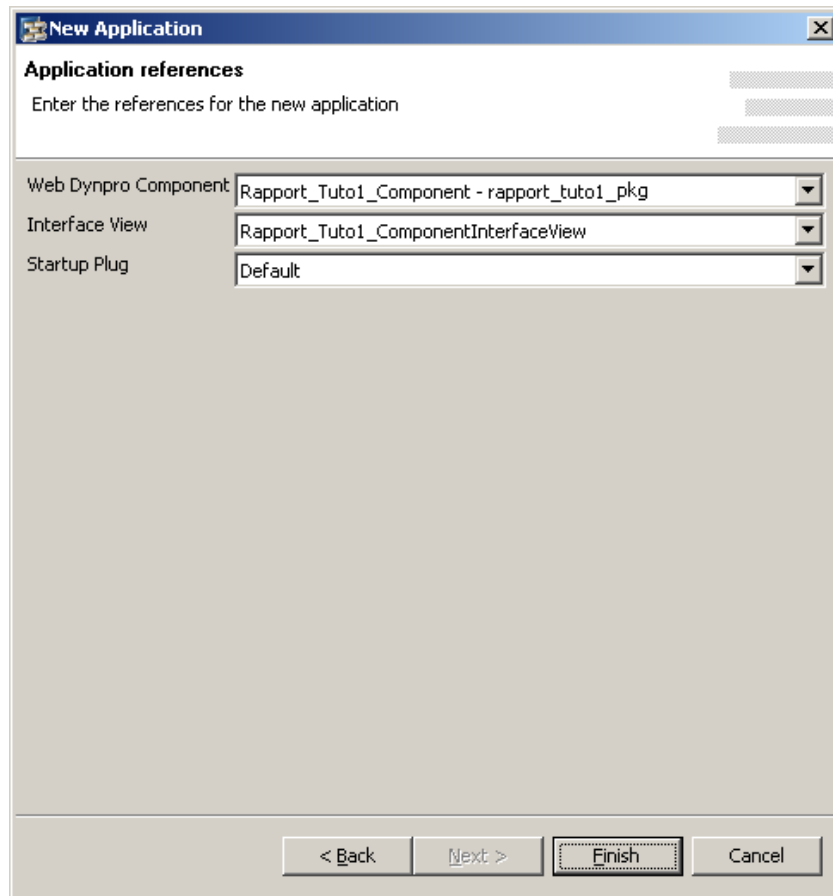


Figure 179 - Tutoriel 1: sélection du component

L'application apparaît alors dans la structure du projet:

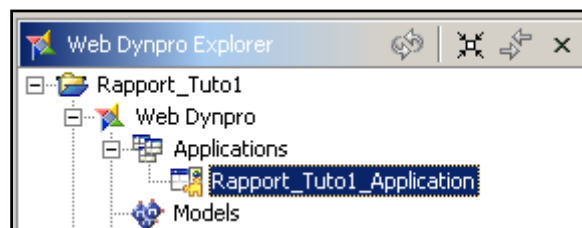


Figure 180 - Tutoriel 1: Application dans Web Dynpro Explorer

7.3.4.11 Déploiement, test, résultat

Il ne reste plus qu'à la déployer et à l'exécuter en faisant un clic droit sur son nom et en sélectionnant "Deploy New Archive and Run":

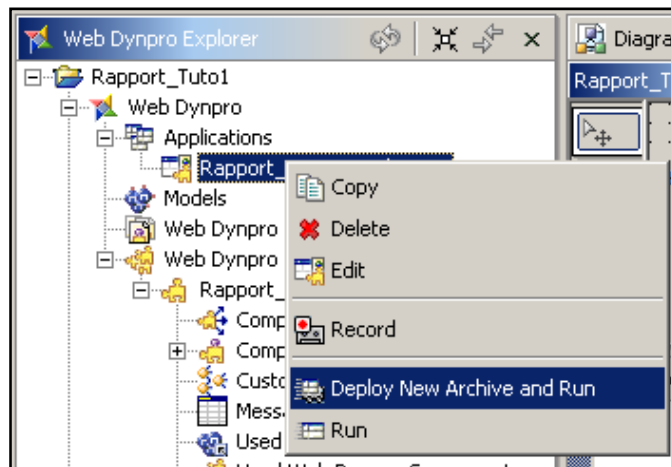


Figure 181 - Tutoriel 1: déploiement de l'application

Si c'est le tout premier déploiement, un mot de passe mis en place par la personne qui a installé le serveur J2EE sera demandé.

Après quelques instants, une fenêtre Internet Explorer s'ouvrira automatiquement et nous amènera vers la première vue de l'application:

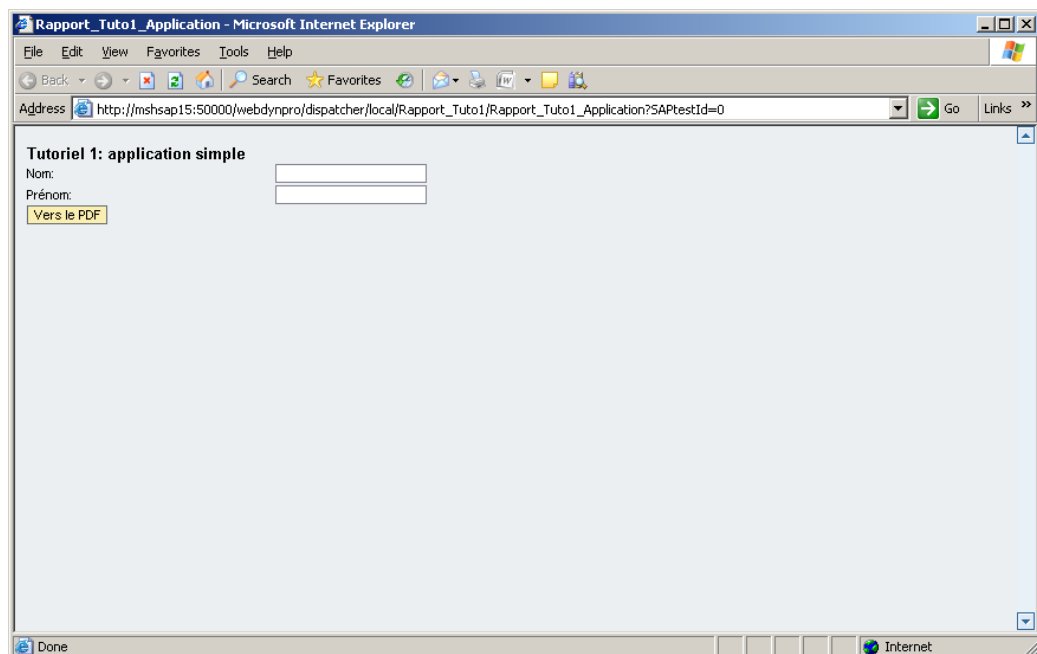


Figure 182 - Tutoriel 1: affichage de l'application dans Internet Explorer

Pour tester, nous saisissons un nom et un prénom et cliquons sur le bouton "Vers le PDF":

Figure 183 - Tutoriel 1: saisie de données

La page contenant le formulaire apparaît alors et nous constatons qu'il est déjà rempli avec les informations que nous venons de saisir:

Figure 184 - Tutoriel 1: pré-remplissage dans le formulaire

Pour finir, nous testons la modification des données dans le formulaire, pour voir si le changement se propage dans le reste de l'application. Comme nom, nous saisissons "Doe", comme prénom "John", nous cliquons ensuite sur le bouton "Submit" et enfin sur le bouton "Retour":

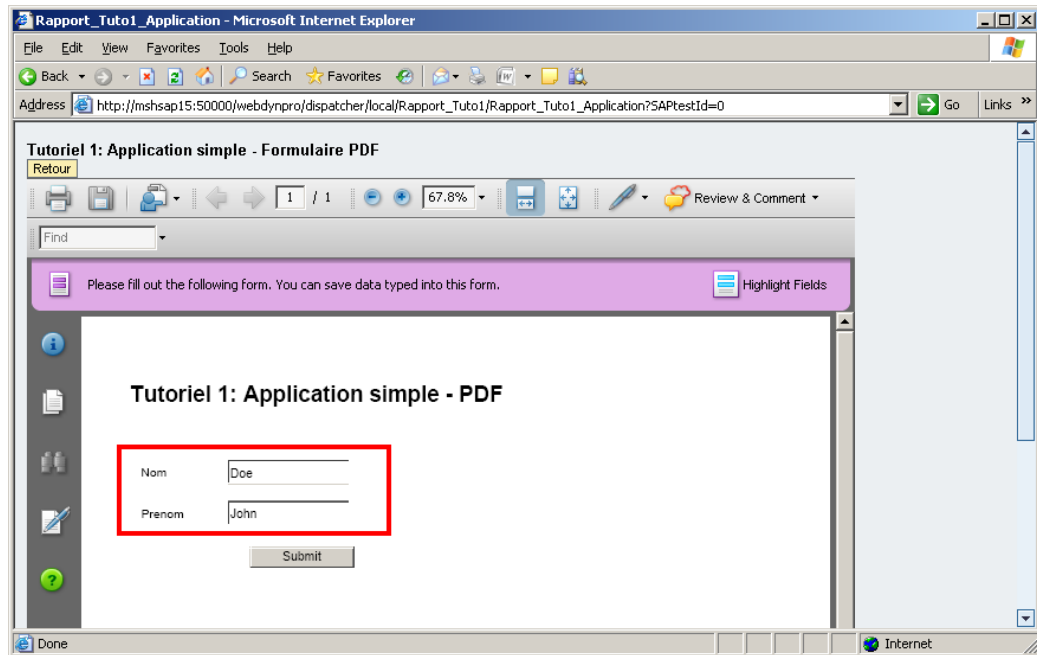


Figure 185 - Tutoriel 1: modification des données du formulaire

Une fois de retour sur la première page, nous constatons que les valeurs ont bien été mises à jour:

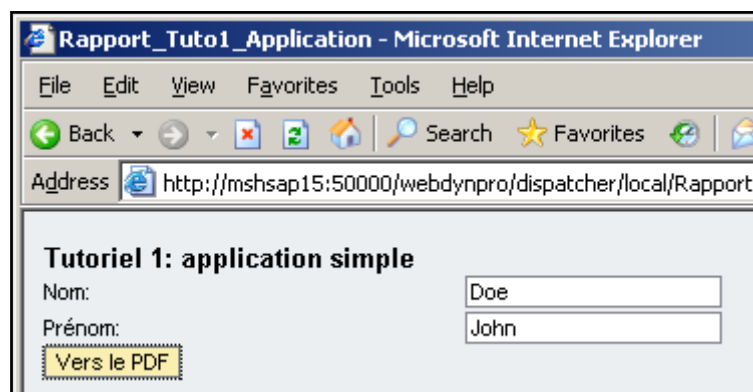


Figure 186 - Tutoriel 1: affichage des données modifiées

Ceci conclut ce premier tutoriel sur l'incorporation d'un formulaire interactif dans une application Web Dynpro.

7.4 Les BAPIs/RFC

Nous allons à présent aborder l'une des parties les plus importantes de ce travail de diplôme: il s'agit de l'utilisation des BAPIs/RFC.

7.4.1 Définitions

- **BAPI:** Business Application Programming Interfaces
- **RFC:** Remote Function Call

7.4.2 Fonctionnement

En faisant appel aux BAPIs dans une application, il devient possible de dialoguer et d'effectuer des transactions avec un système SAP. Concrètement, elles se présentent sous la forme de fonctions que l'on peut exécuter et qui peuvent retourner un résultat en fonction d'éventuels paramètres que l'on a passé lors de l'appel. Le schéma ci-dessous illustre les possibilités offertes par les BAPIs:

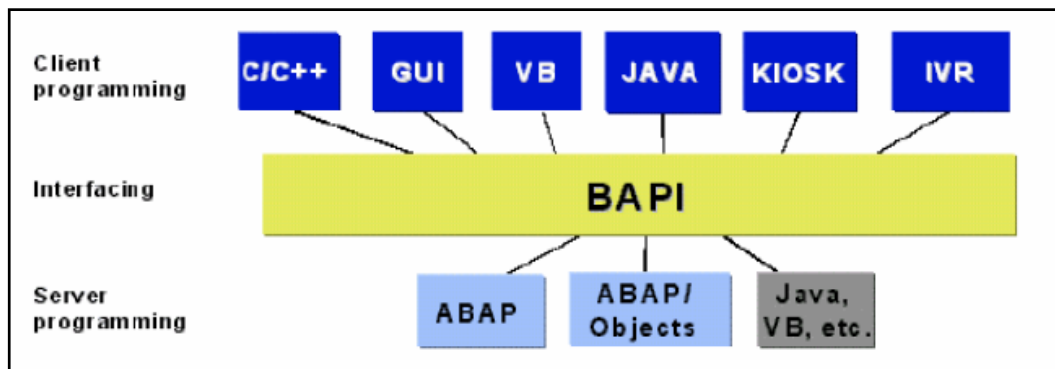


Figure 187 - Tutoriel 1: BAPI ¹⁵

Comme nous pouvons le voir, les BAPIs sont utilisables par n'importe quelle application elle-même codée dans n'importe quel langage. Cela a comme principal avantage qu'il est possible d'incorporer une application non-SAP dans un environnement SAP, offrant ainsi une réelle ouverture sur la manière de gérer et d'élargir l'architecture actuelle.

Enfin, il est important de préciser la principale différence entre une BAPI et une RFC. Elle est très simple: une BAPI permet d'établir un dialogue entre une application non-SAP et un système SAP alors qu'une RFC ne peut être utilisée que par une application SAP.

Il faut aussi savoir qu'un développeur n'est pas obligé de créer une BAPI afin de pouvoir l'utiliser dans son application. En effet, il existe une panoplie très vaste et complète de BAPIs prêtes à l'emploi ce qui présente 2 avantages très intéressants:

- Gain de temps: il suffit de les importer.
- Ne nécessite aucune connaissance du langage ABAP vu qu'il ne faut pas les créer.

¹⁵ Source: PDF: BAPI Step-by-Step

7.4.3 Comment trouver une BAPI?

Pour ce travail de diplôme, 3 manières ont été utilisées:

7.4.3.1 Via le code transaction BAPI

Cela affiche l'explorateur des BAPIs qui permet de parcourir les BAPIs disponibles classées soit par groupe, soit alphabétiquement. Il fournit toutes les informations nécessaires telles que les noms des BAPIs, les paramètres nécessaires, les informations retournées, etc..

Exemple: on cherche une BAPI pour créer une absence. En naviguant dans l'arborescence dans la partie gauche de la fenêtre, on peut la trouver sous :

Personnel Time Management → EmployeeAbsence → Create

On trouve ainsi son nom (encadré en rouge): BAPI_ABSENCE_CREATE, ainsi que les paramètres (encadrés en vert): numéro de l'employée, heures de début et de fin, dates de début et de fin, type d'absence, etc...

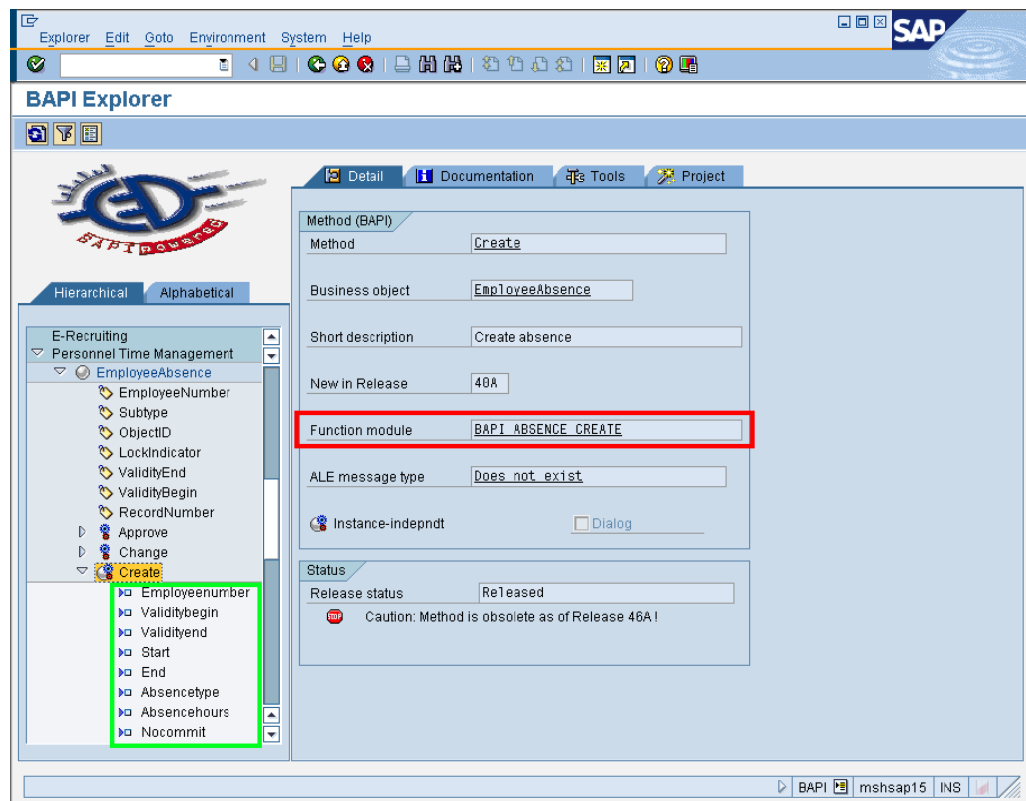


Figure 188 - BAPI: BAPI Explorer

7.4.3.2 Via le code transaction SE37 (Function Builder Initial Screen)

Cette manière de chercher peut être très intéressante si l'on est sûr que le nom d'une BAPI contient un mot en particulier. En effet, en arrivant sur la page de démarrage du Function Builder et en saisissant une partie du nom entouré d'astérisques (*) suivi par la touche F4, on obtient une liste de toutes les BAPIs qui contiennent la partie de nom que nous avons saisie.

Bien que cette manière de faire relève plus de l'astuce, elle offre un avantage non négligeable: une fois qu'on pense avoir trouvé la BAPI, on peut immédiatement la tester, c'est-à-dire saisir des paramètres et s'assurer qu'on obtient en retour les informations escomptées.

Exemple: on imagine qu'on cherche une BAPI qui retourne toutes les absences d'un employé. On peut donc supposer que son nom contient l'expression "ABSENCE". On utilise alors le code transaction SE37, on saisit "*ABSENCE*", comme présenté ci-dessous:

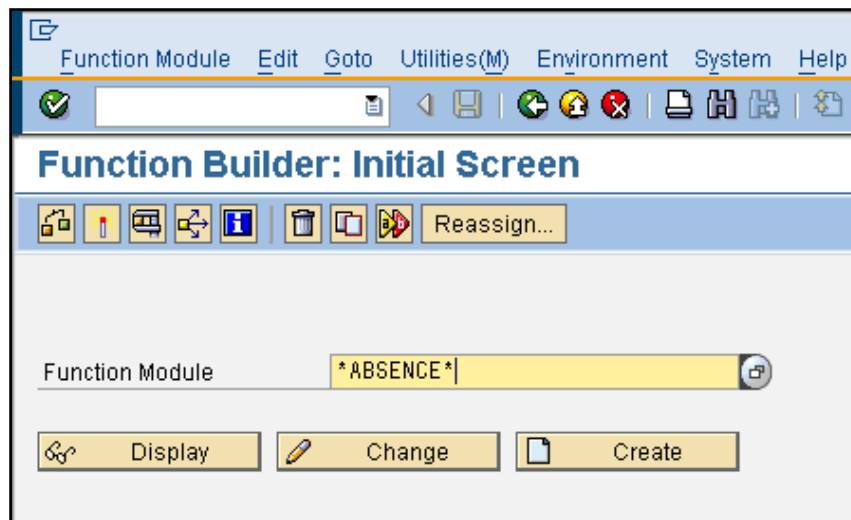


Figure 189 - BAPI: Function Builder

On appuie ensuite sur la touche F4, ce qui a pour effet de retourner la liste de toutes les BAPIs dont le nom contient "ABSENCE":

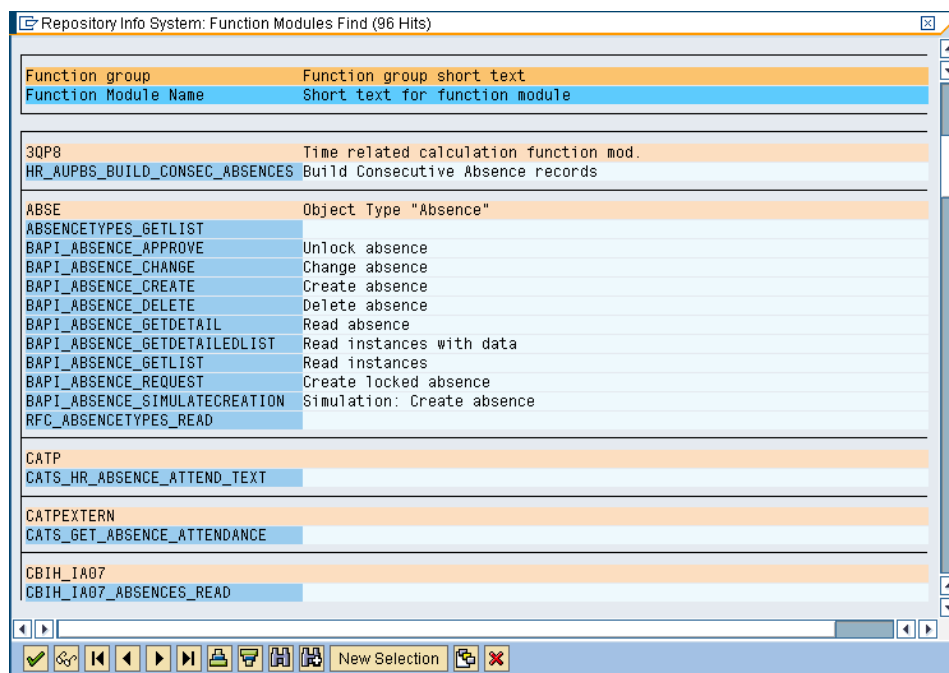


Figure 190 - BAPI: exemple de résultats de recherche d'une BAPI

On sélectionne alors BAPI_ABSENCE_GETDETAILEDLIST en double cliquant dessus étant donné que son nom est plutôt générique et qu'il sous-entend un retour d'informations détaillées.

Le nom de la BAPI est alors inséré dans le champ de saisie:

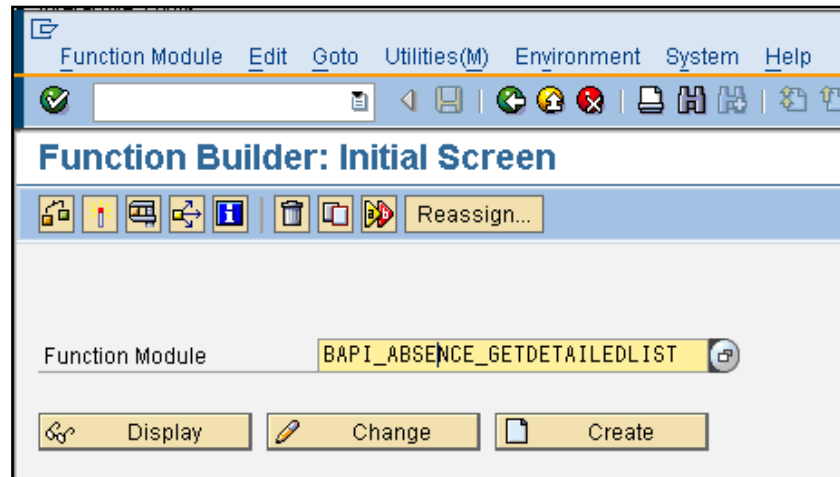




Figure 191 - BAPI: Function Builder

L'étape suivante consiste à s'assurer qu'on ait bien trouvé la bonne BAPI. Pour ce faire, on appuie sur la touche F8 ou alors sur l'icône . La fenêtre qui apparaît ensuite nous demande de saisir des paramètres. Dans le cas présent, l'employé qui nous intéresse est celui dont le numéro est 11199130. On saisit donc l'information comme indiqué ci-dessous et on clique ensuite sur  ou sur la touche F8 pour exécuter la BAPI:

Function modules Edit Goto Utilities(M) System Help

Test Function Module: Initial Screen

Debugging Test data directory

Test for function group ABSE
Function module BAPI_ABSENCE_GETDETAILEDLIST
Uppercase/Lowercase ☐
RFC target sys:

Import parameters	Value
EMPLOYEENUMBER	11199130
SUBTYPE	
TIMEINTERVALLOW	01.01.1800
TIMEINTERVALHIGH	31.12.9999

Tables	Value
ABSENCE	0 Entries

Figure 192 - BAPI: saisie d'un paramètre pour test

On arrive alors sur l'écran des résultats. On remarque que la BAPI a retourné une absence:

Tables	Value
ABSENCE	1 Entry

Result: 1 Entry

Figure 193 - BAPI: résultat du test de la BAPI

Finalement, on clique sur l'icône afin d'afficher la liste des résultats qui se présente ainsi:

Structure Editor: Display ABSENCE from Entry 1

Column Entry Metadata

1 Entry

EMPLOYEE	SUBT	DB	L	VALIDEND	VALIDBEGIN	REC	START	END	ABSE	NAMEOFABSENCETYPE	ABSENCED	ABSENCEHO
11199130	0230			16.04.2008	16.04.2008	000	10:00:00	11:30:00	0230	Doctor's appointment	0,20	1,50

Figure 194 - BAPI: informations retournées

7.4.3.3 Via le SDN

La dernière méthode de recherche utilisée est simplement la recherche sur le SDN. En effet, à moins de travailler sur quelque chose de très précis et peu supporté, il y a de grandes chances à ce que l'on trouve d'autres personnes qui ont eu besoin de la même information que nous dans le passé. Même si cette technique tient elle aussi de l'astuce, elle permet très souvent de tomber sur des messages, guides ou blogs qui parlent de ce que l'on cherche ainsi que d'autres éléments complémentaires que l'on pourrait éventuellement utiliser plus tard.

7.4.4 Implementation d'une BAPI

7.4.4.1 Pré-requis: utilisateur technique

Avant d'importer une BAPI, il est indispensable de passer par la création d'un utilisateur technique. En effet, l'importation et l'exécution d'une BAPI ne se fait pas sans authentification: il faut un compte qui dispose des autorisations nécessaires. Cet utilisateur est qualifié de "technique" car il existe dans le seul but de permettre une communication entre deux systèmes ou un système et une application. Ce type d'utilisateur a aussi une autre caractéristique: une fois configuré, il ne pourra plus se connecter au système via le SAP Logon par exemple. La connexion ne sera possible que via les BAPIs ou RFCs.

Pour créer cet utilisateur, nous utilisons le code transaction SU01 qui correspond à la gestion des utilisateurs du système. Pour ce travail de diplôme, l'utilisateur JDAVID_TECH a été créé avec les propriétés suivantes:

- Logon data:
- User Type: Communications Data
- User group: TECHNICAL
- Initial et Repeat Password: le mot de passe désiré

The screenshot shows the SAP SU01 transaction in the 'Logon data' tab. The 'User Type' is set to 'Communications Data'. The 'User Group for Authorization Check' is set to 'TECHNICAL'. The 'Initial password' and 'Repeat password' fields are filled with asterisks. The 'Password Status' is 'Initial password (set by administrator)'. The 'User Group for Authorization Check' is also set to 'TECHNICAL'.

Figure 195 - Utilisateur technique: détails de connexion

- Profiles: SAP_ALL (par mesure de simplicité)

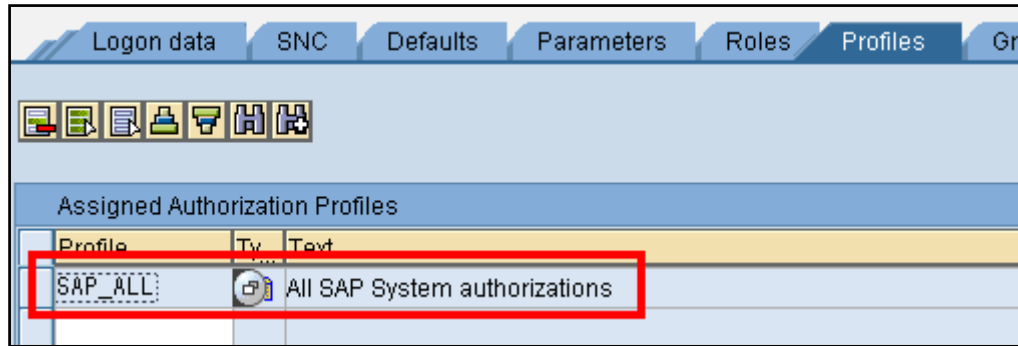



Figure 196 - Utilisateur technique: profile

Il ne reste alors plus qu'à sauvegarder ().

Note: il n'est pas nécessaire de créer un utilisateur technique à chaque fois que l'on souhaite utiliser une BAPI. Pour ce travail de diplôme, le même utilisateur (JDAVID_TECH) a été utilisé pour tous les différents développements & tests.

7.4.5 Déploiement d'une application contenant au moins une BAPI.

Contrairement à l'application développée dans le tutoriel précédent, une application qui contient au moins une BAPI doit être déployée différemment la première fois sans quoi une erreur se produira lors de son exécution. Ce déploiement se passe en 4 étapes une fois le développement terminé:

- Création d'une archive
- Déploiement de l'archive
- Création des connexions JCO dans le Web Dynpro Content Administrator
- Exécution de l'application

7.4.5.1 Création d'une archive dans SAP NetWeaver Developer Studio:

Il suffit d'effectuer un clic droit sur le nom du projet et de sélectionner "Create Archive":

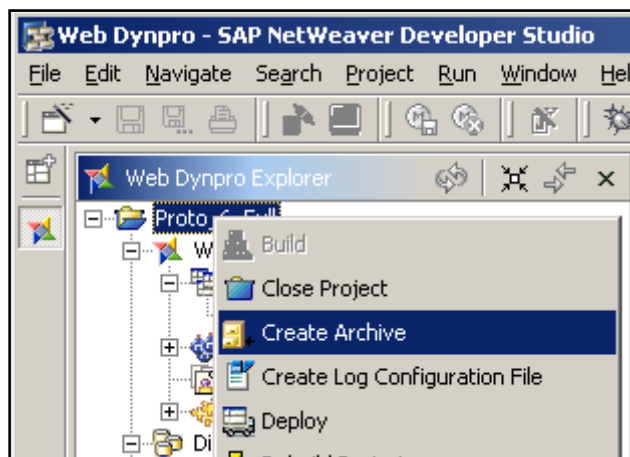


Figure 197 - Création d'un archive

7.4.5.2 Déploiement de l'archive vers le serveur J2EE:

Il suffit d'effectuer un clic droit sur le nom du projet et de sélectionner "Deploy ":

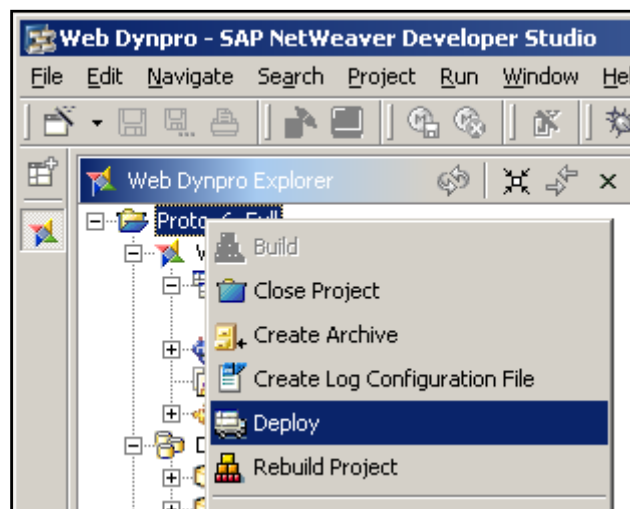


Figure 198 - Déploiement d'une archive

7.4.5.3 Création des connexions JCO dans le Web Dynpro Content Administrator.

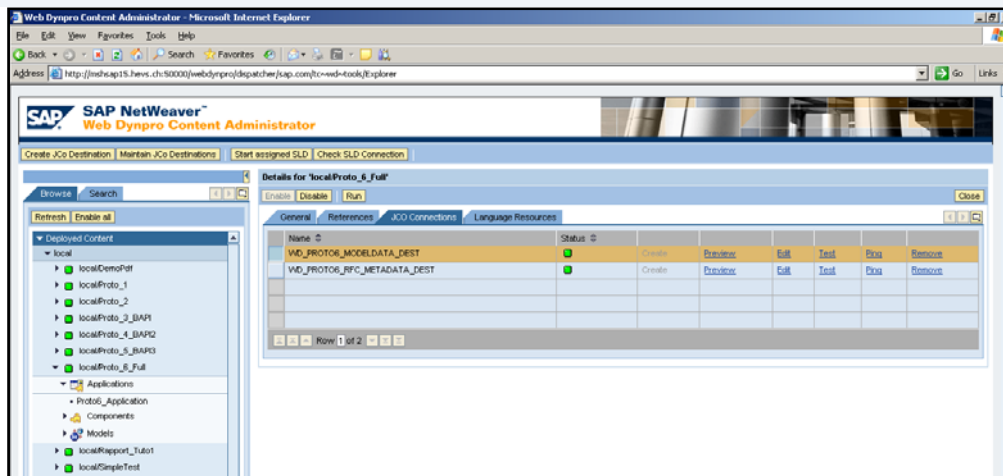


Figure 199 - Connexions JCO

Pour plus d'informations concernant la création des connexions JCO, le tutoriel 2 en montre un exemple très détaillé.

7.4.5.4 Exécution de l'application

Depuis SAP NetWeaver Developer Studio ("Run" la première fois, "Deploy New Archive And Run" les fois suivantes):

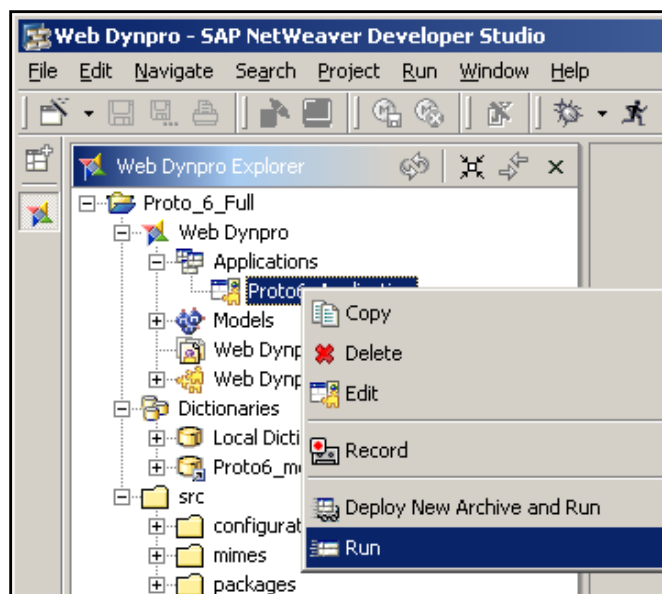


Figure 200 - Lancement de l'application

151

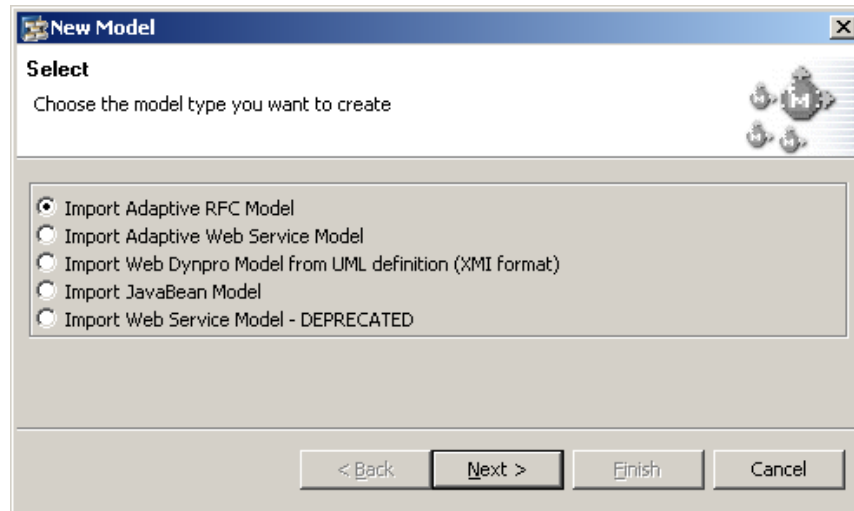


Figure 202 - Tutoriel 2: Types de modèles

Une fenêtre nous demande alors de nommer plusieurs choses. Elles peuvent être nommées comme sur l'image ci-dessous par exemple. On clique ensuite sur le bouton "Next":

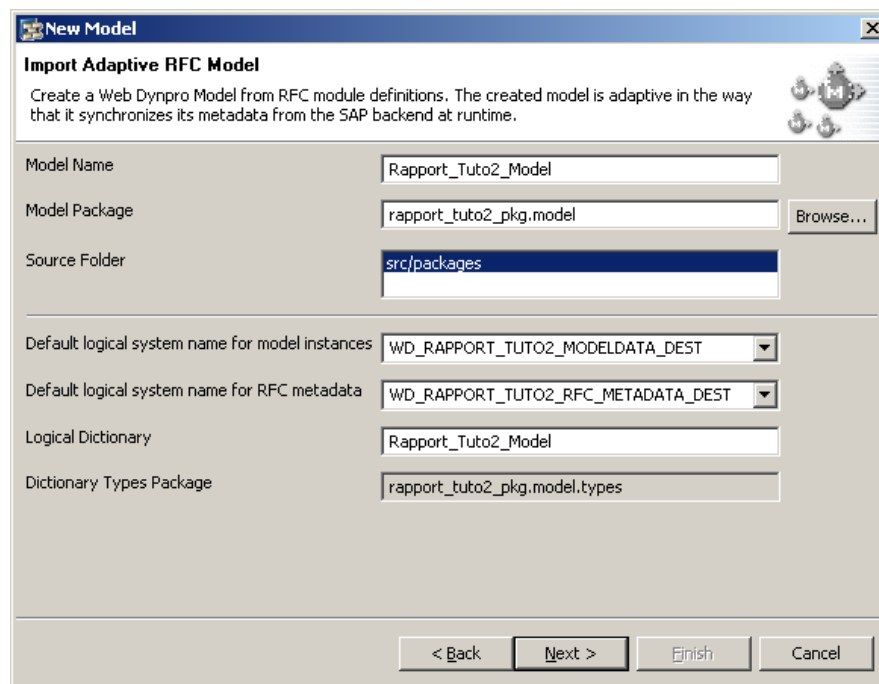


Figure 203 - Tutoriel 2: nommages pour le modèle

La fenêtre suivante demande d'indiquer les informations de connexion au système SAP. Nous utilisons ici les informations de l'utilisateur technique que nous avons créé au préalable. Le système indiqué sur la capture d'écran ci-dessous correspond à la D07. Une fois les informations saisies on clique sur "Next":

The screenshot shows a Windows-style dialog box titled "New Model". Inside, there's a section "SAP Logon Information" with the instruction "Specify the SAP Server you want to connect to and enter the logon information". Below this, there are two tabs: "Single Server" (selected) and "Load Balancing". The "Single Server" tab contains three input fields: "Host Name" with the value "mshsap15.hevs.ch", "System Number" with the value "00", and "SAP Router" with a dropdown menu showing "/H/saprouter.hevs.ch/H/". Below these is a "User Account" section with four input fields: "Client" with the value "100", "Logon Name" with the value "jdavid_tech", "Password" with the value "*****", and "Language" with the value "en". At the bottom of the dialog, there are four buttons: "< Back", "Next >", "Finish", and "Cancel".

Figure 204 - Tutoriel 2: connexion à SAP

La fenêtre suivante nous demande de sélectionner le ou les modules RFC que nous souhaitons ajouter à notre modèle. Pour ce faire, il faut tout d'abord les rechercher en tapant leur nom exact ou alors une partie du nom entouré d'astérisques. Pour ce tutoriel, nous voulons utiliser la BAPI "BAPI_PERSDATA_GETDETAILEDLIST". Cela donne par exemple:

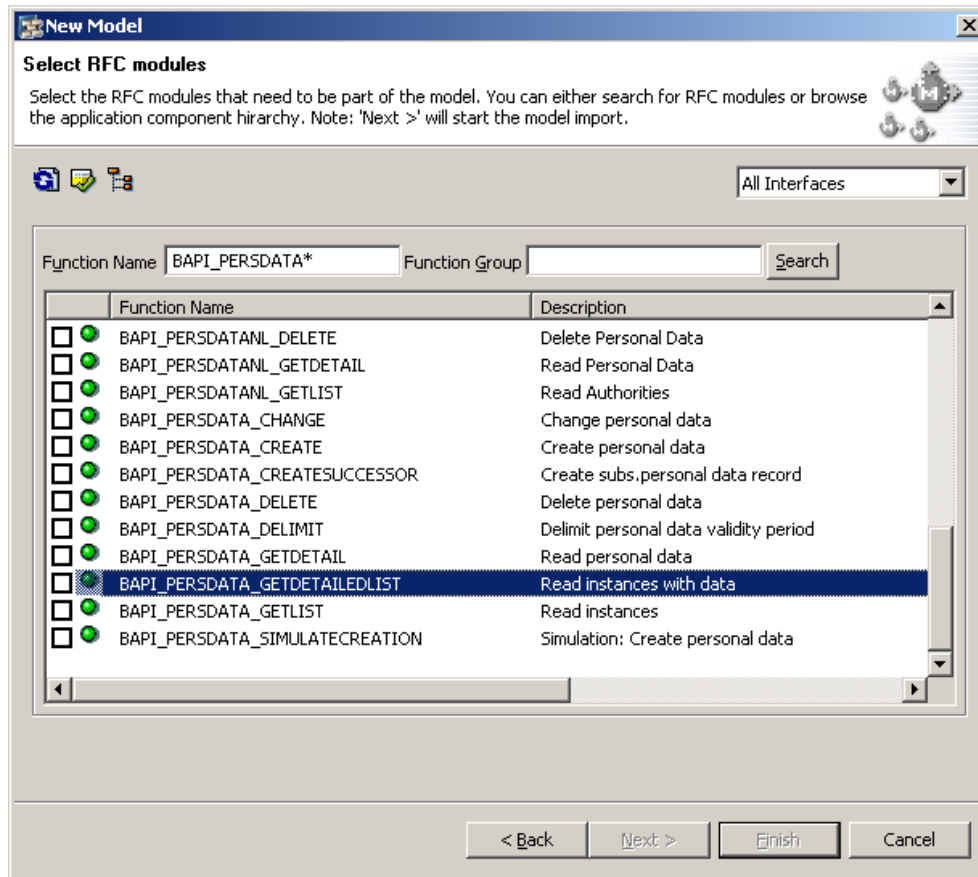


Figure 205 - Tutoriel 2: liste des BAPIs

La BAPI désirée apparaît dans la liste, par conséquent on coche la case à côté de son nom et on clique sur le bouton "Next" étant donné que nous n'avons pas besoin d'autre BAPI:

<input type="checkbox"/>	<input checked="" type="checkbox"/>	BAPI_PERSDATA_DELIMIT	Delimit personal data validity period
<input type="checkbox"/>	<input checked="" type="checkbox"/>	BAPI_PERSDATA_GETDETAIL	Read personal data
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	BAPI_PERSDATA_GETDETAILEDLIST	Read instances with data
<input type="checkbox"/>	<input checked="" type="checkbox"/>	BAPI_PERSDATA_GETLIST	Read instances

Figure 206 - Tutoriel 2: sélection de la BAPI désirée

La fenêtre suivante affiche le journal d'importation. Il n'y a pas d'erreur, nous pouvons donc cliquer sur "Finish" pour terminer la création du modèle:

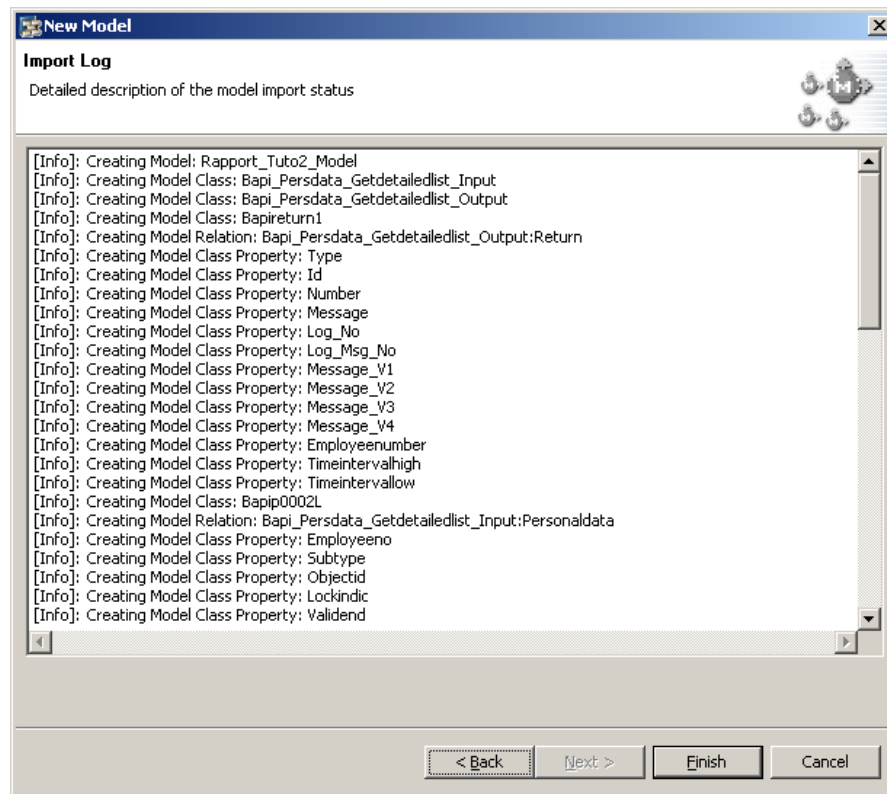


Figure 207 - Tutoriel 2: journal d'importation du modèle

On constate alors que le modèle a bien été créé vu qu'il apparaît dans la structure du projet:

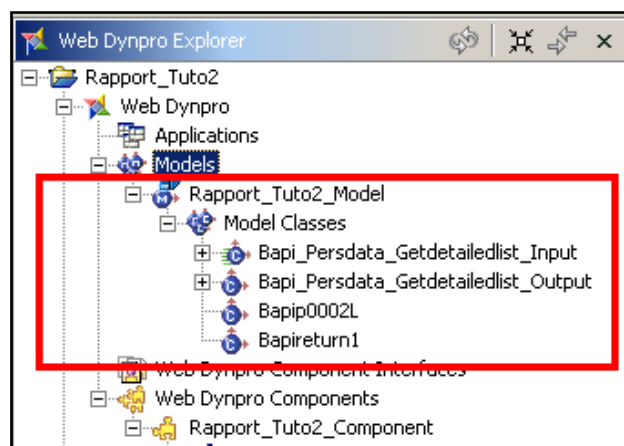


Figure 208 - Tutoriel 2: modèle dans le Web Dynpro Explorer

Nous pouvons ensuite créer le design de la vue. On y ajoute un label, un champ de saisie, un bouton et un formulaire interactif. Visuellement, nous pouvons avoir quelque chose qui ressemble à l'image ci-dessous par exemple:

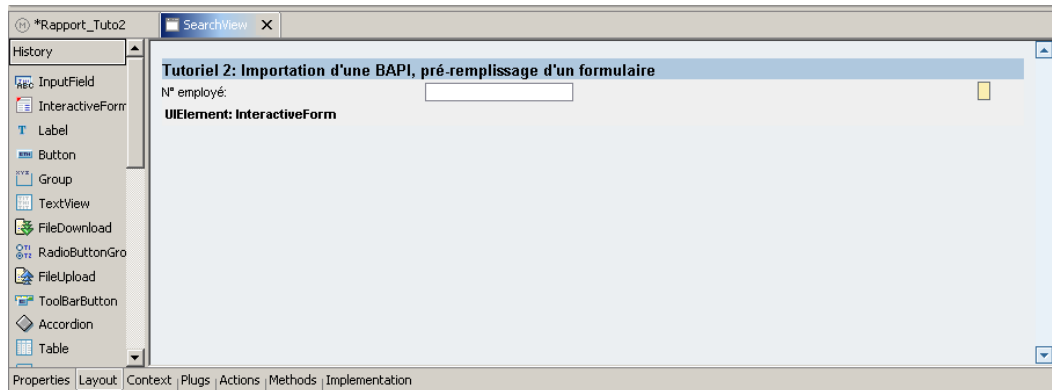


Figure 209 - Tutoriel 2: vue SearchView

A présent, nous créons un Custom Controller dont le contexte sera lié au modèle contenant la BAPI. Tout d'abord, on fait un clic droit sur "Custom Controllers" dans la structure du projet et on sélectionne "Create Custom Controller":

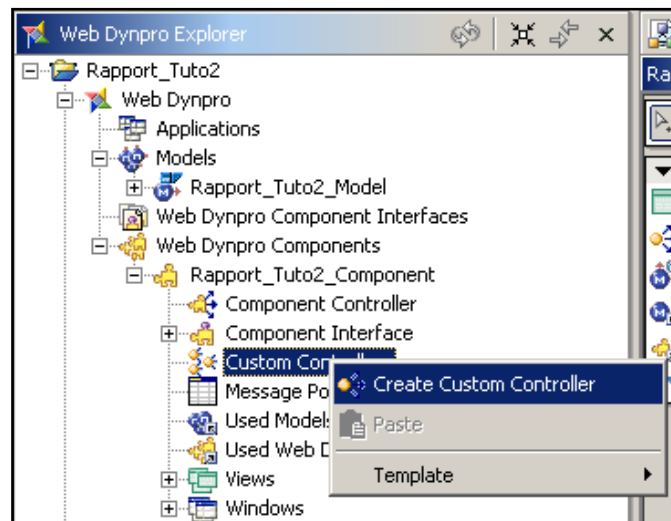


Figure 210 - Tutoriel 2: Création d'un Custom Controller

On lui donne un nom, comme par exemple "Rapport_Tuto2_CustController" et on clique sur "Finish":

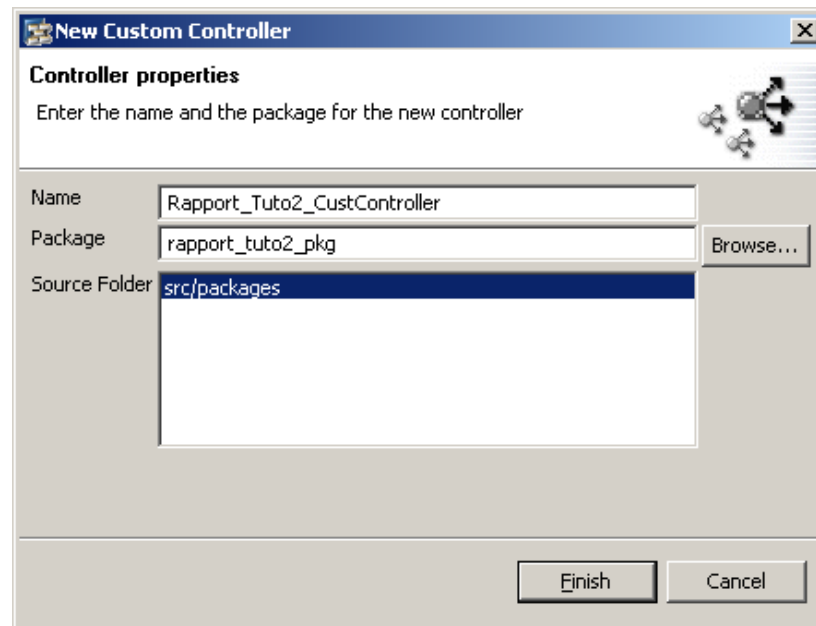


Figure 211 - Tutoriel 2: nommage du Custom Controller

On constate alors que le contrôleur apparaît dans la structure du projet:

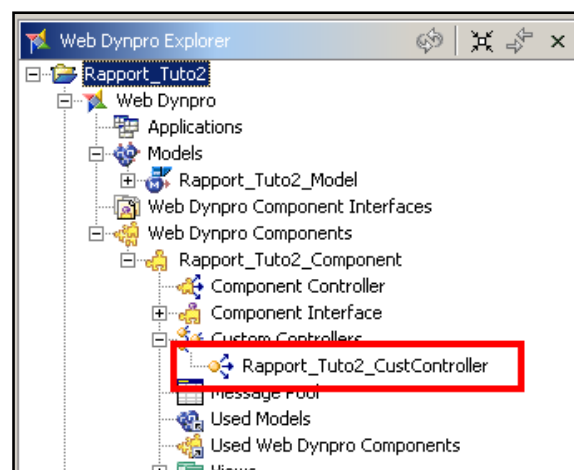


Figure 212 - Tutoriel 2: Custom Controller dans Web Dynpro Explorer

L'étape suivante consiste à "déclarer" l'utilisation d'un modèle. Pour ce faire, on effectue un clic droit sur "Used Models" dans la structure du projet et on sélectionne "Add ":

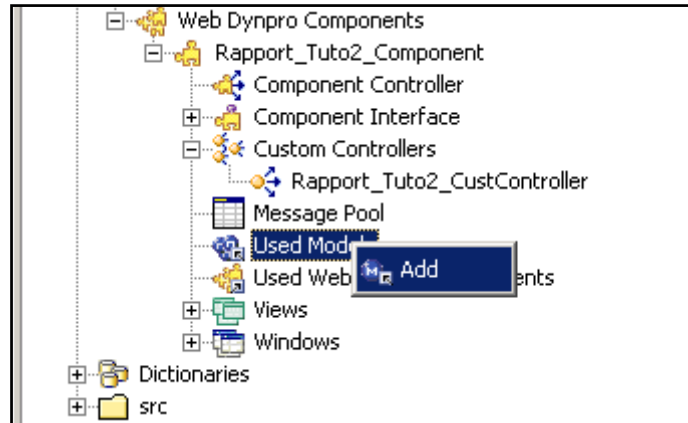


Figure 213 - Tutoriel 2: déclaration d'utilisation d'un modèle

La fenêtre suivante nous demande quel modèle nous souhaitons utiliser. Nous sélectionnons celui que nous avons créé durant les étapes précédentes et on clique sur "OK":

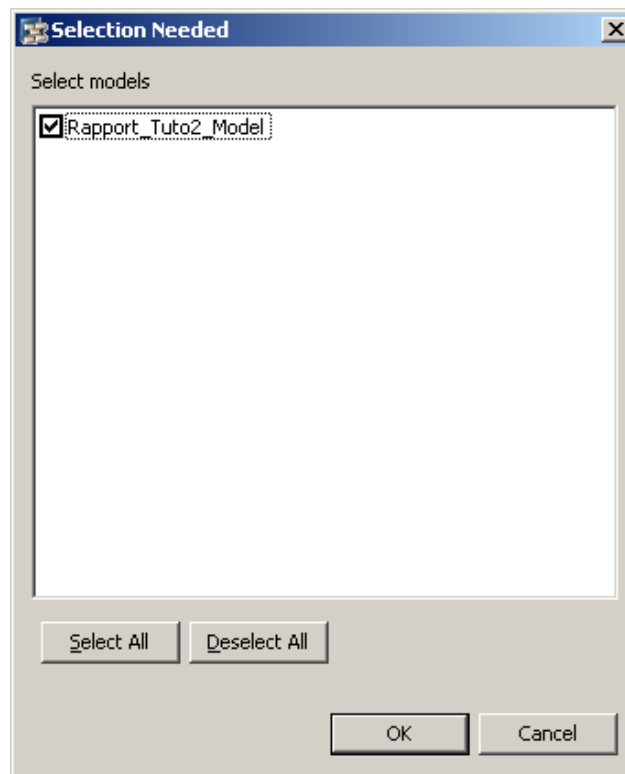


Figure 214 - Tutoriel 2: modèles disponibles

Le modèle apparaît alors dans la liste des modèles utilisés dans la structure du projet:



Figure 215 - Tutoriel 2: modèle utilisé

7.5.4 Mapping Custom Controller → modèle

Nous allons à présent effectuer un premier mapping entre le modèle et le contexte du contrôleur que nous avons créé. Nous affichons donc le schéma représentant la structure de l'application en faisant un clic droit sur le nom du composant dans le Web Dynpro Explorer et en sélectionnant "Open Data Modeler":

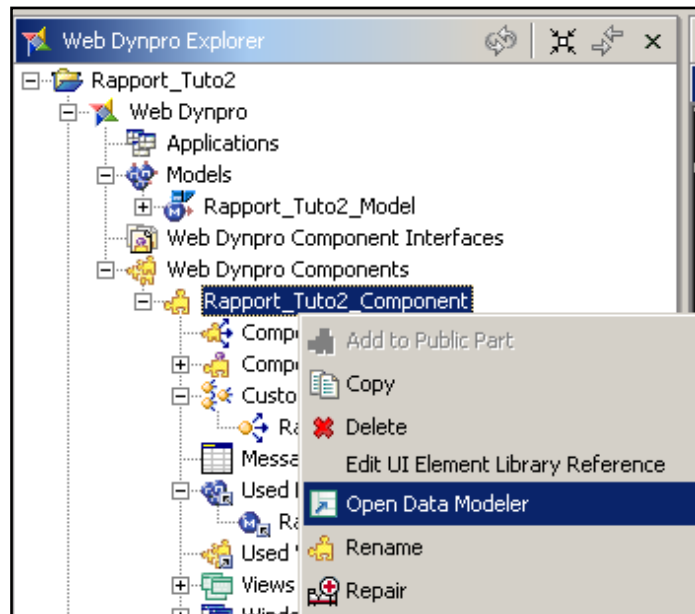


Figure 216 - Tutoriel 2: Lancement du Data Modeler

Nous y retrouvons notamment la vue, le Custom Controller et le modèle:

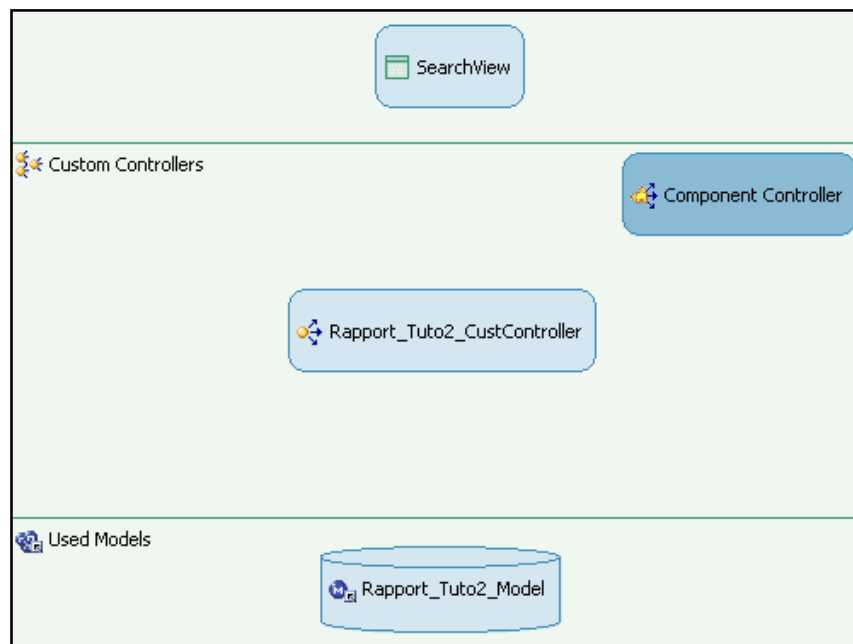


Figure 217 - Tutoriel 2: schéma de l'application

Edit Model Binding

Model Binding

Drag from right to left to create and bind a new element in the left tree.
 Drag from left to right to bind an existing element.

Rapport_Tuto2_CustController -

Context

Rapport_Tuto2_Model

- Rapport_Tuto2_Model
- Bapi_Persdata_Getdetaileddlist_In
- Bapi_Persdata_Getdetaileddlist_Out
- Bapip0002L
- Bapireturn1

Finish Cancel

[illegible]

Jacquin David
HES-SO Valais, mai 2008

On coche les cases comme indiqué sur le schéma explicatif ci-dessous et on clique sur "OK":

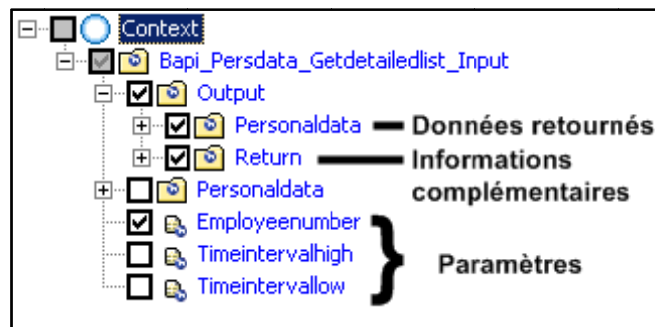


Figure 220 - Tutoriel 2: éléments cochés pour le mapping

Le mapping, symbolisé par des flèches, est créé. Il n'y a plus qu'à cliquer sur "Finish":

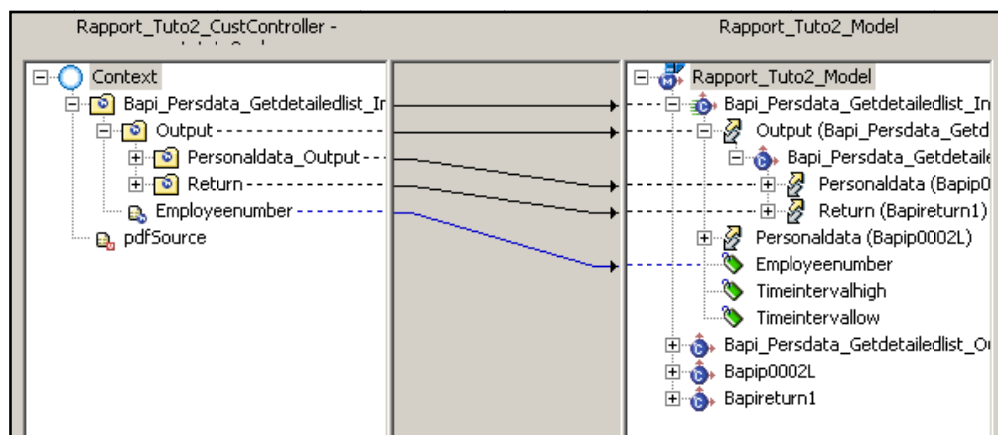


Figure 221 - Tutoriel 2: mapping entre le Custom Controller et la vue

On remarque alors qu'une flèche relie le contrôleur au modèle à présent:

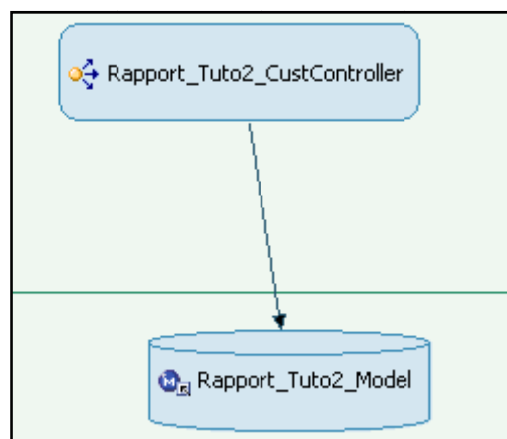


Figure 222 - Tutoriel 2: liaison Custom Controller → modèle

On peut ainsi voir le contexte du contrôleur:

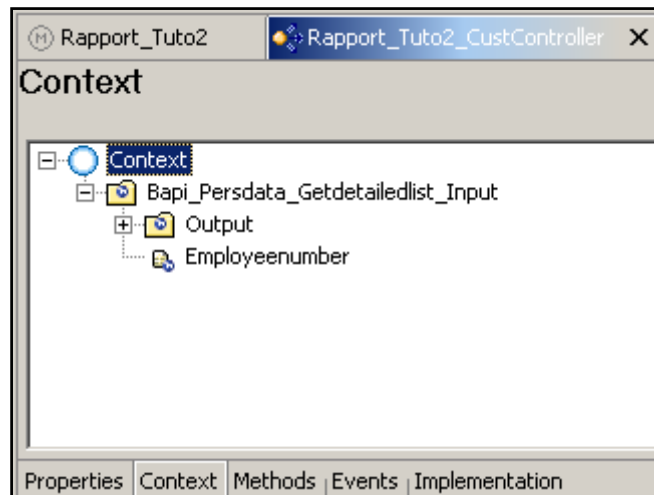


Figure 223 - Tutoriel 2: Contexte du Custom Controller

Etant donné que nous allons utiliser un formulaire interactif, n'y a-t-il pas un élément qui manque dans ce contexte?

En effet, nous devons encore rajouter un "Value Attribute" de type binaire et qu'on nommera pdfSource:

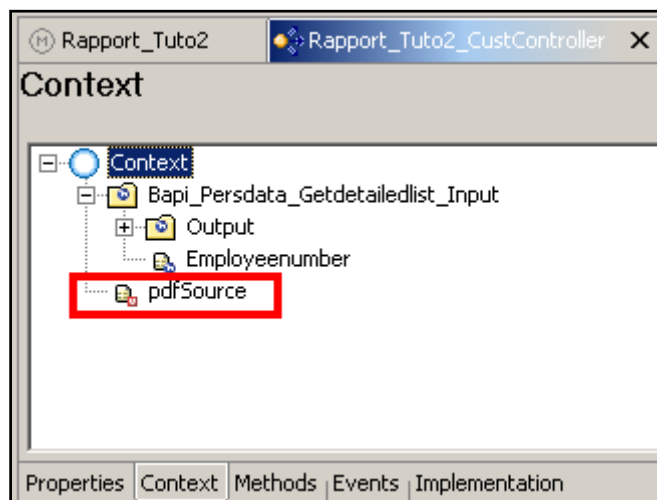



Figure 224 - Tutoriel 2: attribut pdfSource

7.5.5 Mapping SearchView → Custom Controller

L'étape suivante consiste à créer un mapping entre la vue et le contrôleur. Pour ce faire, on sélectionne à nouveau  dans la barre d'outils du schéma et on fait un drag & drop depuis la vue jusqu'au contrôleur. La fenêtre suivante apparaît alors:

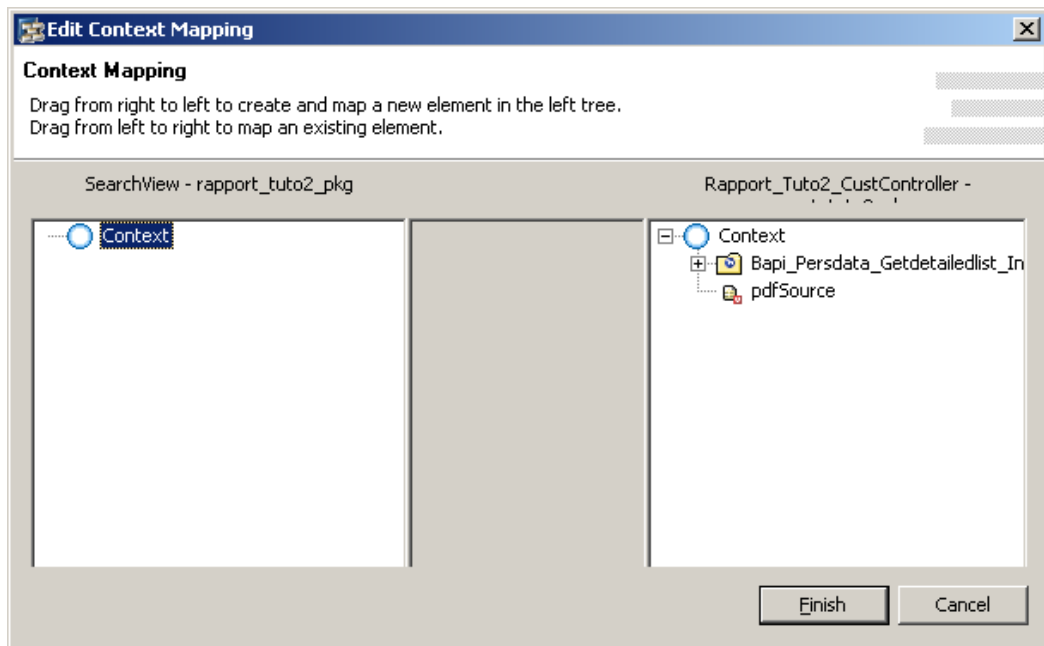


Figure 225 - Tutoriel 2: mapping entre la vue et le Custom Controller

On effectue un premier drag & drop de Bapi_Persdata_Getdetailedlist_Input vers le contexte de la vue. La fenêtre suivante nous demande alors d'indiquer quels éléments nous souhaitons mapper.

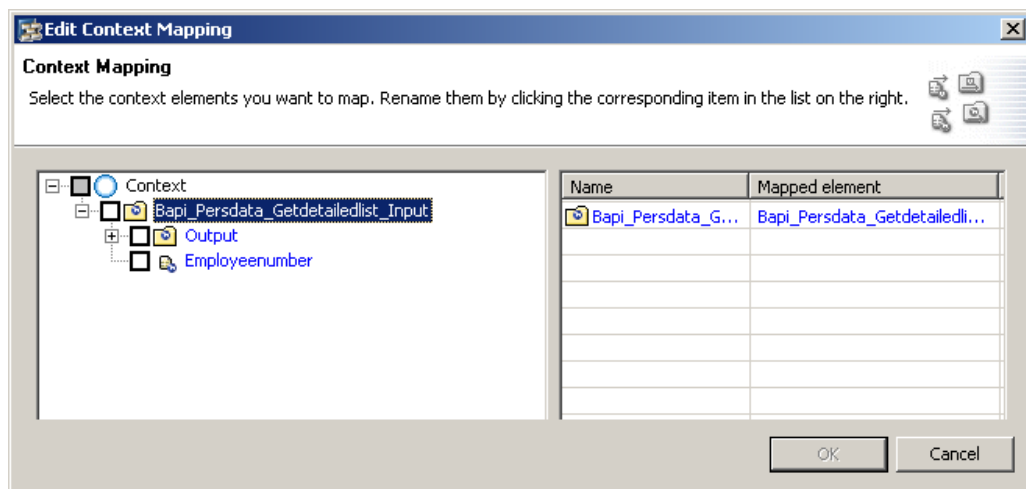


Figure 226 - Tutoriel 2: sélection du mapping

Avons de poursuivre nous devons nous demander de quoi nous aurons besoin dans la vue, que ce soit lors de la recherche de l'employé ou quand ses informations seront affichées. Comme paramètre pour la BAPI, nous aurons uniquement besoin de "EmployeeNumber", par conséquent nous cochons la case correspondante:

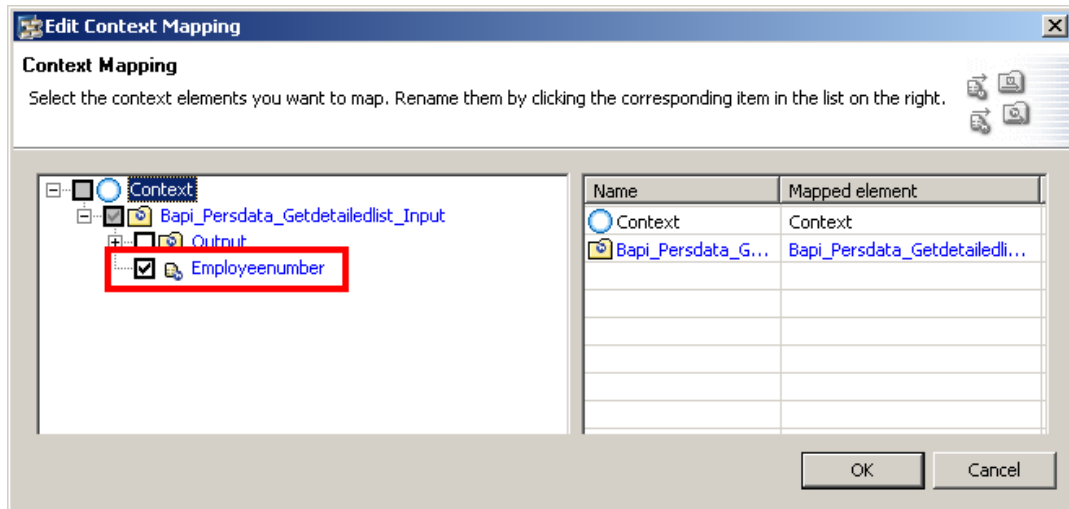


Figure 227 - Tutoriel 2: mapping de l'attribut paramètre

Pour ce qui est des informations retournées, nous voulons pouvoir afficher le nom, prénom et la date de naissance d'un employé. Nous ouvrons donc le nœud "Output", qui correspond à ce qui sera retourné par la BAPI, puis le nœud "Personaldata" et cochons les cases liées aux attributs désirés (DateOfBirth, FirstName, LastName). Une fois les cases cochées, on clique sur "OK":

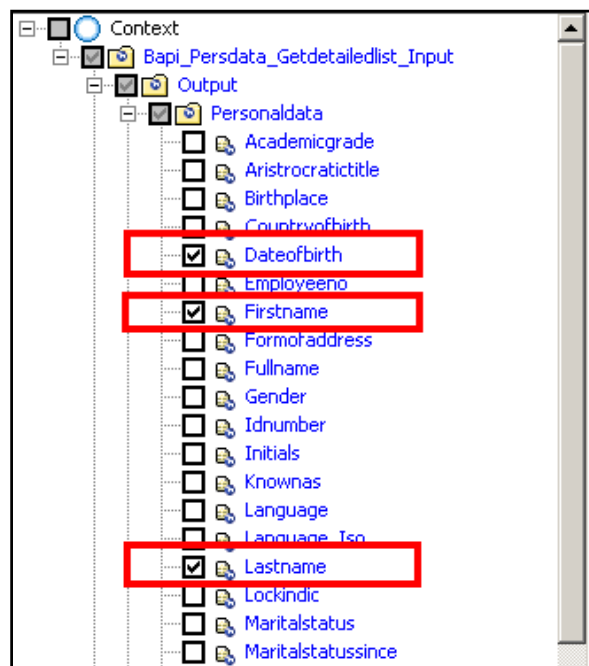


Figure 228 - Tutoriel 2: choix des attributs retournés par la BAPI

Nous obtenons ainsi un premier mapping entre la vue et le contrôleur:

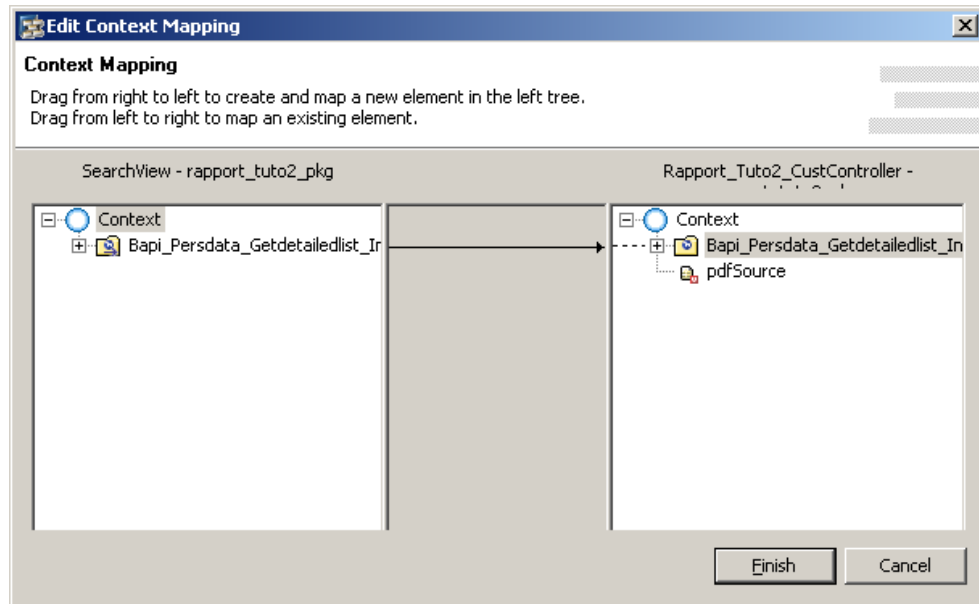


Figure 229 - Tutoriel 2: mapping entre la vue et le Custom Controller

On effectue encore un mapping pour l'attribut pdfSource et on clique ensuite sur "Finish":

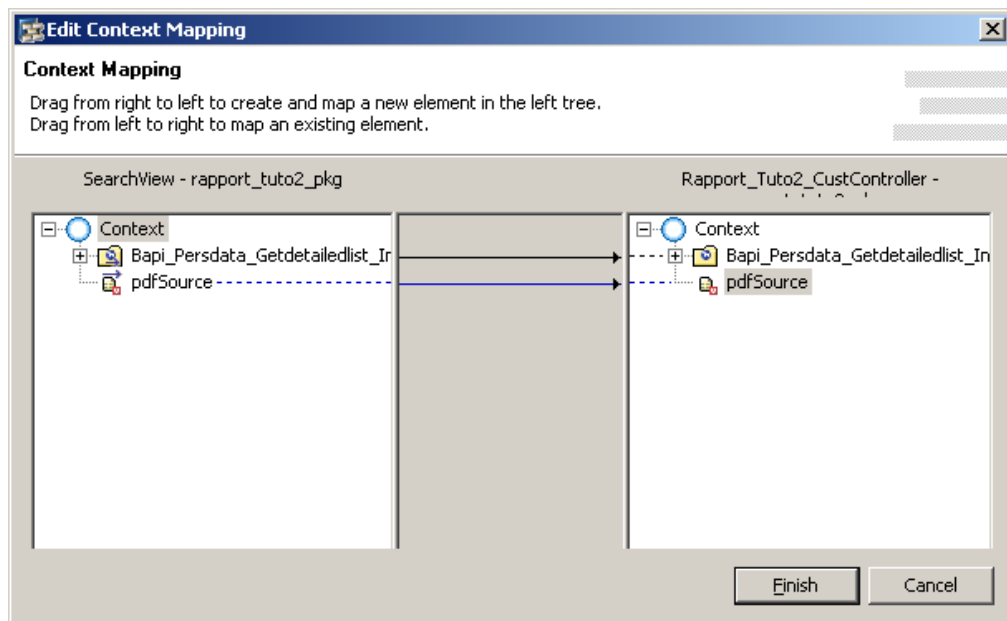


Figure 230 - Tutoriel 2: mapping entre la vue et le Custom Controller

On constate alors que la vue est liée au contrôleur sur le schéma:

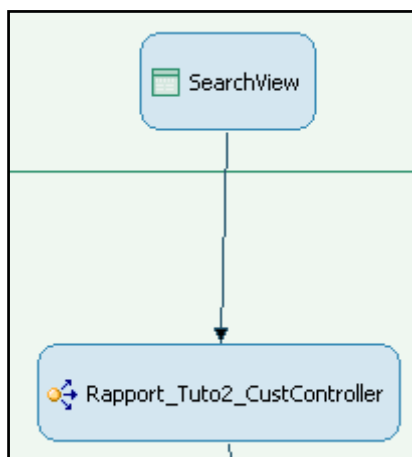


Figure 231 - Tutoriel 2: liaison entre la vue et le Custom Controller

7.5.6 Méthode du Custom Controller

On ajoute ensuite une méthode qui permettra d'exécuter la BAPI. Cette méthode, qu'on nommera "executeBapi_GetEmployeeData", doit être ajoutée au "Custom Controller" "Rapport_Tuto2_CustController" via l'onglet "Methods". Une fois l'étape effectuée, cela donne:

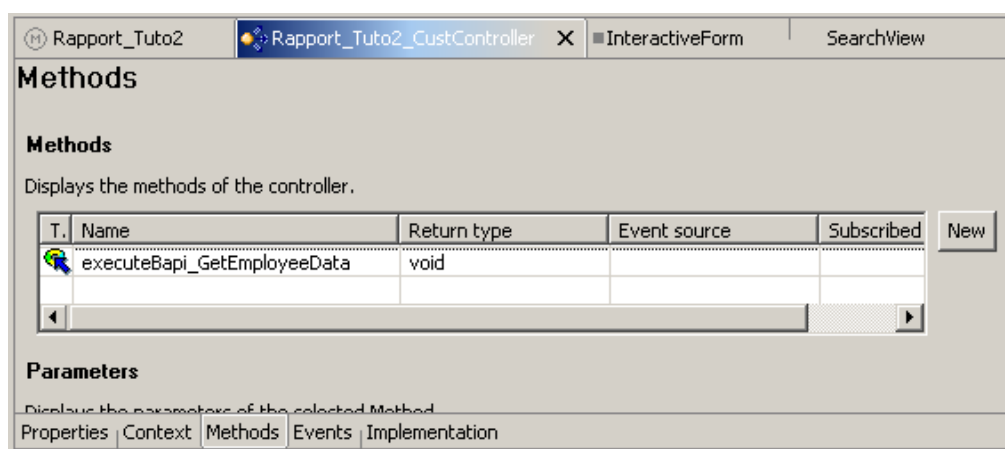


Figure 232 - Tutoriel 2: méthode executeBapi_GetEmployeeData

7.5.6.1 Action de SearchView

On passe ensuite à la vue (dans notre cas "SearchView") et on lui ajoute une action qu'on nommera "Search" via l'onglet "Actions". Dans le cas présent, il n'y pas besoin d'indiquer un "Fire Plug" lors de la création de l'action étant donné qu'il n'y a pas de navigation. Au final, cela donne:

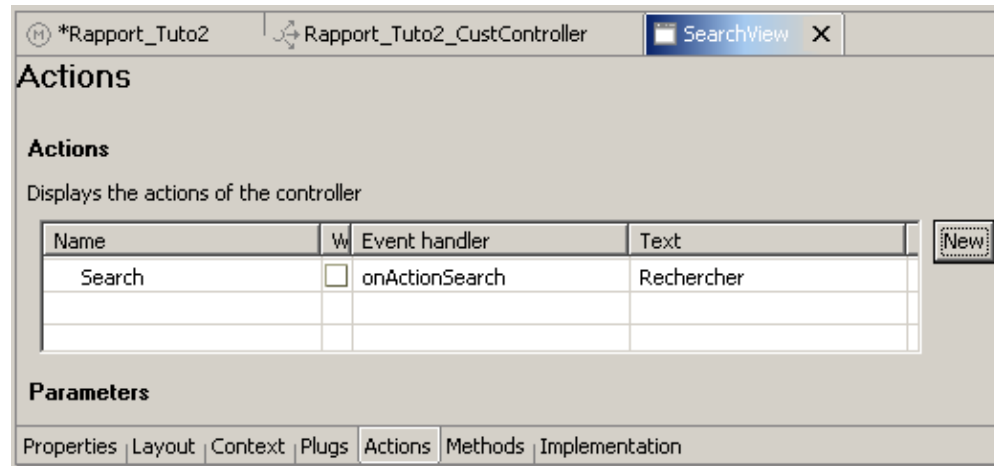


Figure 233 - Tutoriel 2: action Search

Toujours dans la vue, on termine la liaison des différents éléments graphiques. On a donc:

Elément	Propriété	Valeur
InputField	Value	Bapi_Persdata_Getdetailedlist_Input.Employeeenumber
Button	onAction	Search
Interactive Form	dataSource	Bapi_Persdata_Getdetailedlist_Input.Output.PersonalData
	pdfSource	pdfSource

Une fois les liaisons de donnée effectuée, le design de la vue change légèrement:

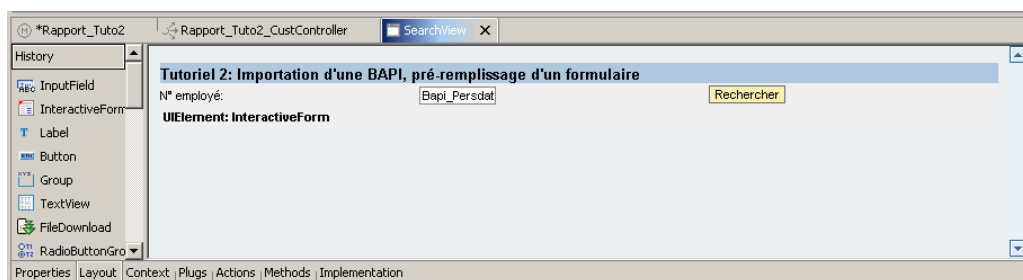


Figure 234 - Tutoriel 2: vue SearchView

7.5.7 Design du formulaire

L'étape suivante consiste à réaliser le design du formulaire. On lance donc l'Adobe LiveCycle Designer (clic droit sur le nom du formulaire interactif dans la fenêtre d'Outline, puis sélectionner "Edit"). On effectue ensuite un drag & drop des données que l'on souhaite voir dans le formulaire et enfin on ajoute un titre. Cela donne par exemple:

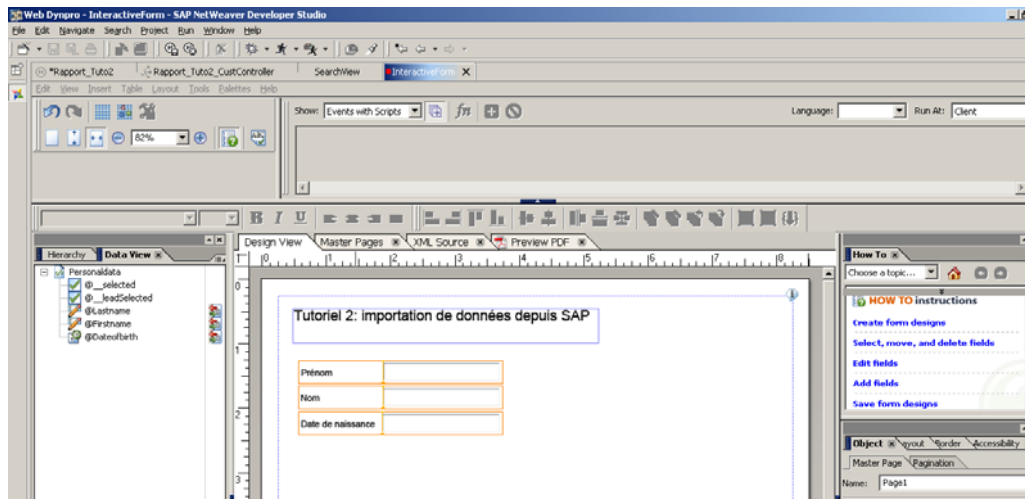


Figure 235 - Tutoriel 2: design du formulaire

7.5.8 Implémentation

Maintenant que le formulaire a été créé, on peut passer à l'implémentation. Nous allons modifier 3 méthodes:

- Custom Controller ("Rapport_Tuto2_CustController"):
 - o wdDoInit()
 - o executeBapi_GetEmployeeData()
- Vue ("SearchView"):
 - o onActionSearch()

Commençons par le Customer Controller et sa méthode wdDoInit(). Cette méthode devra effectuer 2 choses: créer un élément vide pour que le contexte ne le soit plus. Exécuter une fois la BAPI pour que le nœud auquel est lié le formulaire interactif ne soit pas vide. En effet, si on essaie de déployer un formulaire lié à un nœud vide, le serveur interrompt l'exécution et renvoie une page contenant une erreur.

7.5.8.1 wdDoInit() (Custom Controller)

La méthode wdDoInit() du Customer Controller se présente donc ainsi:

```
public void wdDoInit()
{
  //@@begin wdDoInit()
  Bapi_Persdata_Getdetailedlist_Input input =
    new Bapi_Persdata_Getdetailedlist_Input();

  wdContext.nodeBapi_Persdata_Getdetailedlist_Input().bind(input);
  input.addPersonalddata(new Bapip0002L());
  input.setEmployeeNumber("");//valeur vide par défaut

  try {
    wdContext.currentBapi_Persdata_Getdetailedlist_InputElement()
      .modelObject().execute();
  }
```

```
wdContext.nodeOutput().invalidate();
}
catch (Exception e) {
    e.printStackTrace();
}
//@@end
}
```

La première partie du code empêche le nœud utilisé par le champ de saisie d'être vide, évitant ainsi qu'il soit désactivé lors de l'exécution de l'application

La deuxième partie, entourée par une déclaration Try/Catch, exécute la BAPI et synchronise le nœud contenant les informations retournées. De cette manière, même si le numéro d'employé indiqué dans le code est vide, le nœud de retour est rempli avec des données vierges, ce qui empêche le formulaire interactif de causer une erreur lors du déploiement.

Note: lors de l'implémentation de ce code, il se pourrait que SAP NetWeaver Developer Studio indique une à la ligne contenant "new Bapip0002L()". Dans ce cas-là, il faut effectuer un clic droit et sélectionner *Source* → *Organize Imports*.

7.5.8.2 executeBapi_GetEmployeeData() (Custom Controller)

On passe ensuite à la méthode executeBapi_GetEmployeeData() du Custom Controller. Nous allons simplement reprendre le code qui exécute la BAPI dans wdDoInit(). Cette méthode sera appelée à chaque fois que l'utilisateur effectuera une recherche. Le code est donc:

```
public void executeBapi_GetEmployeeData( )
{
    //@@begin executeBapi_GetEmployeeData()
    try {
        wdContext.currentBapi_Persdata_Getdetailedlist_InputElement()
            .modelObject().execute();

        wdContext.nodeOutput().invalidate();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    //@@end
}
```

7.5.8.3 onActionSearch() (SearchView)

Nous passons ensuite à l'implémentation de la méthode onActionSearch() de la vue "SearchView". Cette méthode sera exécutée à chaque fois que l'utilisateur cliquera sur le bouton de recherche. Lors de son exécution, cette méthode appelle tout simplement la méthode du contrôleur qui exécute la BAPI. Cela donne:

```
public void onActionSearch(com.sap.tc.webdynpro.progmodel
```

```
.api.IWDCustomEvent wdEvent )
{
  //@@begin onActionSearch(ServerEvent)
  wdThis.wdGetRapport_Tuto2_CustControllerController().
  executeBapi_GetEmployeeData();
  //@@end
}
```

Ceci conclut l'implémentation de l'application.

7.5.9 Déploiement

Nous pouvons à présent passer au déploiement de cette dernière. Pour commencer, on crée une archive en cliquant sur le nom du projet dans le Web Dynpro Explorer et on sélectionne "Create Archive":

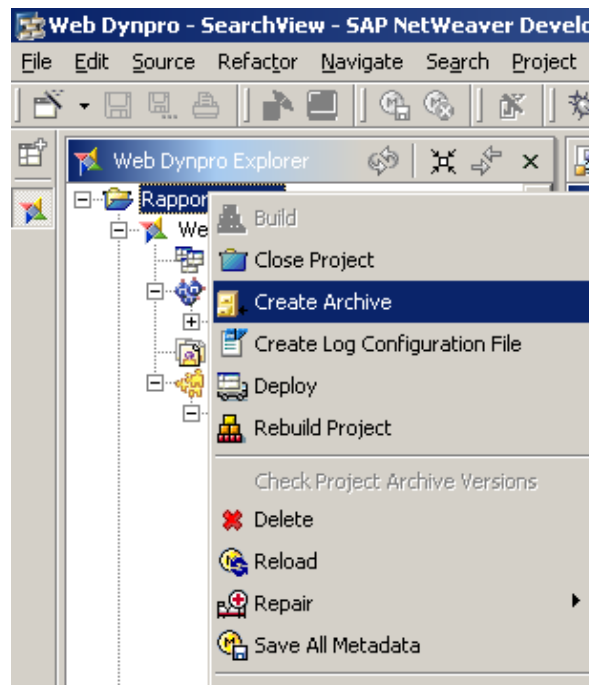


Figure 236 - Tutoriel 2: création de l'archive

Ensuite, on déploie l'archive sur le serveur en sélectionnant dans le même menu "Deploy". Cela a pour effet de déployer l'archive sans toutefois l'exécuter. En effet, nous devons encore créer les connexions JCO avant de pouvoir exécuter notre application. Il serait donc prématuré de le faire maintenant.

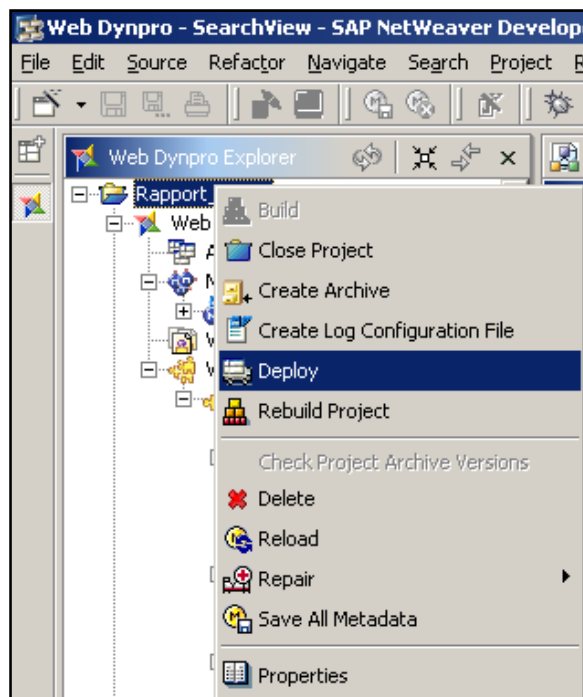


Figure 237 - Tutoriel 2: déploiement de l'archive

7.5.9.1 Connexions JCO

Nous allons à présent créer les connexions JCO. On accède tout d'abord au Web Dynpro Content Administrator:

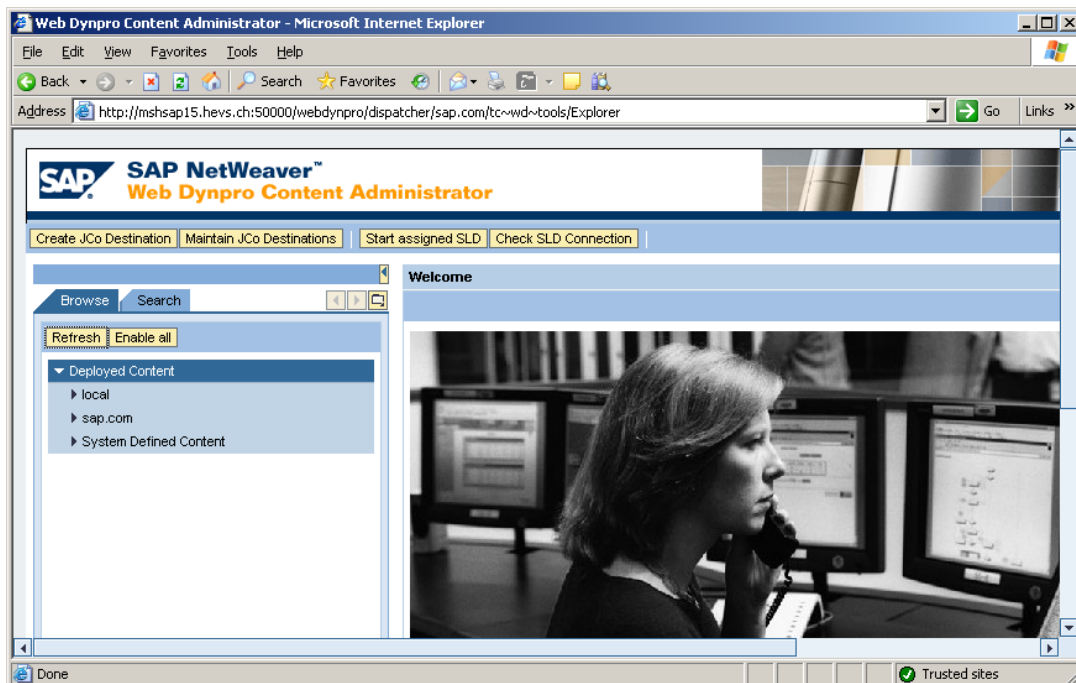


Figure 238 - Tutoriel 2: Web Dynpro Content Administrator

On cherche ensuite notre application et on clique sur son nom pour afficher ses propriétés:

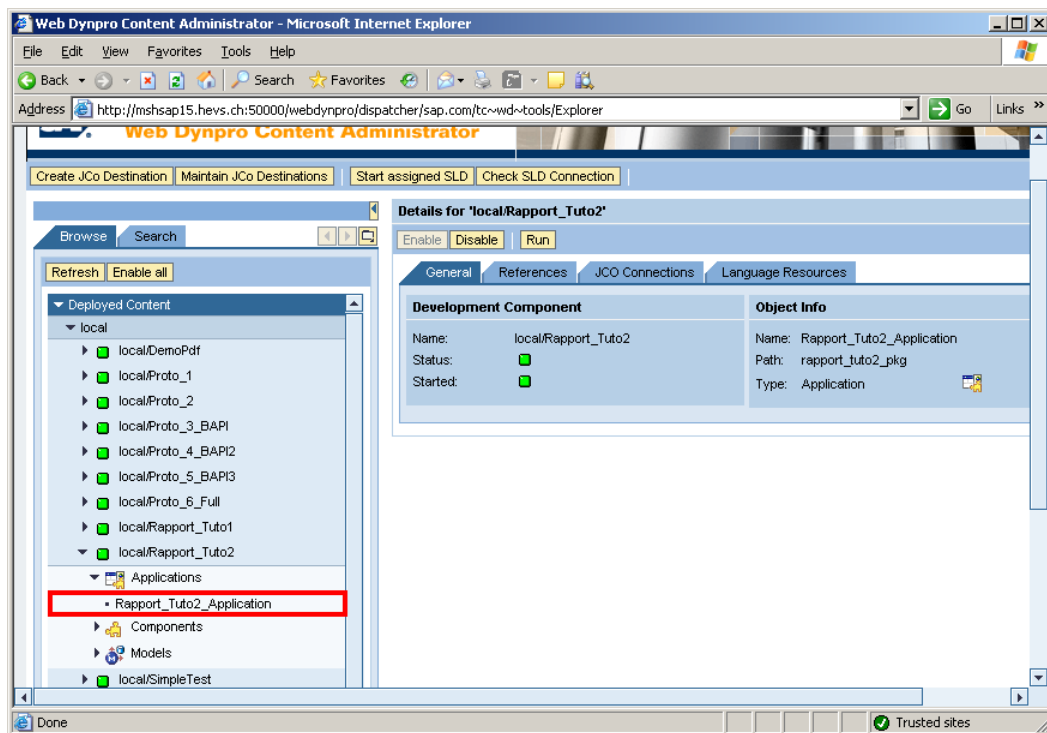


Figure 239 - Tutoriel 2: Connexions JCO

On clique ensuite sur l'onglet "JCO Connections":

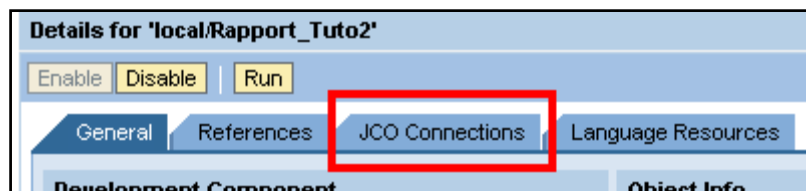


Figure 240 - Tutoriel 2: onglet "JCO Connections"

Les connexions disponibles s'affichent alors:

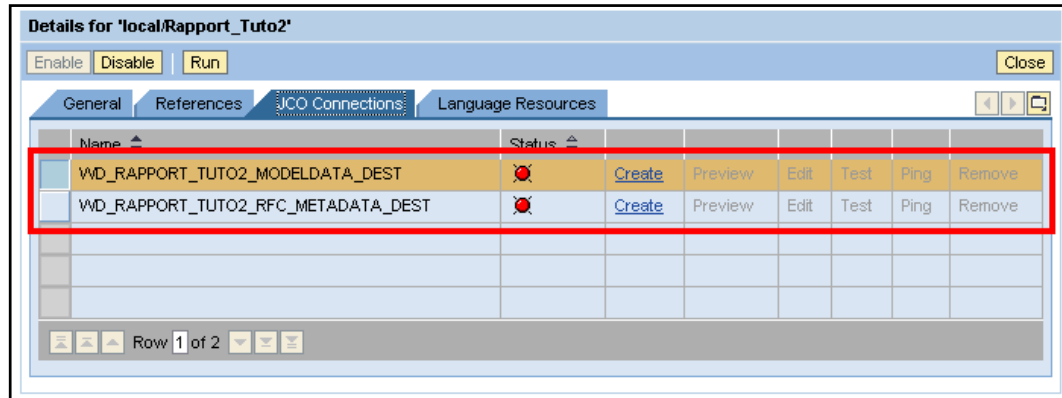


Figure 241 - Tutoriel 2: Connexions JCO (inactives)

Comme nous pouvons le constater, leurs voyants sont rouges, ce qui signifie que nous devons les configurer. Pour ce faire, nous cliquons sur le premier "Create". L'assistant suivant apparaît alors en bas de l'écran:

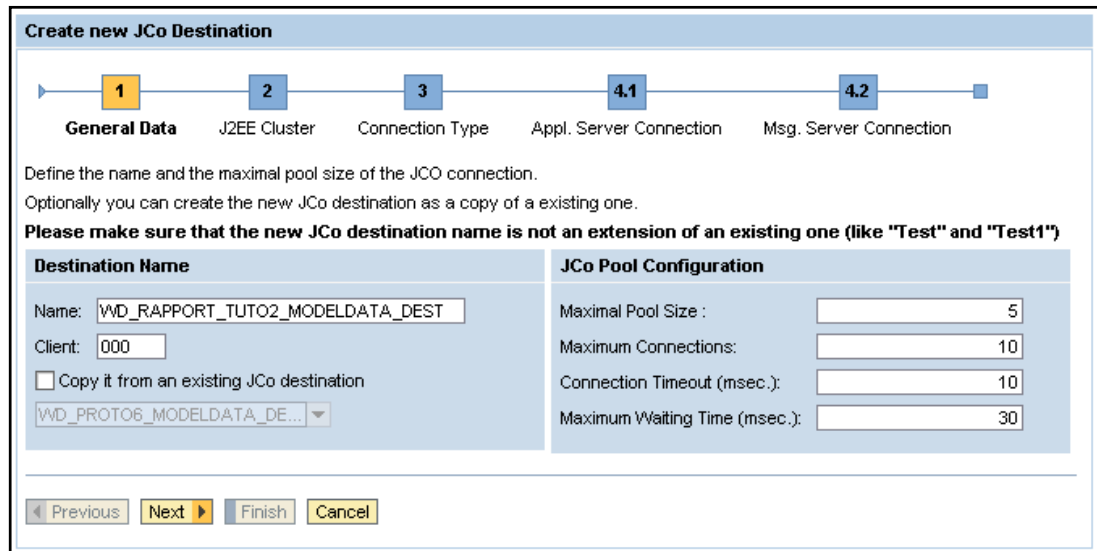


Figure 242 - Tutoriel 2: création de la connexion JCO

Ici, la seule valeur que nous changeons est le mandant ("Client"): on tape 100. On clique ensuite sur **Next**:

Create new JCo Destination

1 General Data 2 J2EE Cluster 3 Connection Type 4.1 Appl. Server Connection 4.2 Msg. Server Connection

Define the name and the maximal pool size of the JCo connection.
Optionally you can create the new JCo destination as a copy of an existing one.
Please make sure that the new JCo destination name is not an extension of an existing one (like "Test" and "Test1")

Destination Name
Name: WD_RAPPORT_TUTO2_MODELDATA_DEST
Client: 100
☐ Copy it from an existing JCo destination
WD_PROTO6_MODELDATA_DE...

JCo Pool Configuration
Maximal Pool Size: 5
Maximum Connections: 10
Connection Timeout (msec.): 10
Maximum Waiting Time (msec.): 30

Previous Next Finish Cancel

Figure 243 - Tutoriel 2: création de la connexion JCO

La fenêtre suivante ne nécessite aucune intervention de notre part, vu que les paramètres sont corrects. Nous pouvons cliquer sur **Next** :

Create new JCo Destination

1 General Data 2 J2EE Cluster 3 Connection Type 4.1 Appl. Server Connection 4.2 Msg. Server Connection

Define the J2EE cluster, to which the JCo destination should be assigned.
You should use your local J2EE cluster as default. In some cases it might be useful to define the JCo destination for another J2EE cluster.
☒ Use local J2EE engine "D07 on mshsap15"
D07 on mshsap15

Previous Next Finish Cancel

Figure 244 - Tutoriel 2: création de la connexion JCO

Même chose pour la fenêtre suivante, nous pouvons cliquer sur **Next** :

Create new JCo Destination

1 General Data 2 J2EE Cluster 3 Connection Type 4.1 Appl. Server Connection 4.2 Msg. Server Connection 5 Security 6 Summary

Data Type
Define the type of data you want to read using the JCo destination.
☒ Dictionary Meta Data
☐ Application Data

Destination Type
Define whether or not you want a load-balanced access.
Note: This is only possible for destinations used to read application data.
☒ Load-balanced Connection (recommended)
☐ Single Server Connection (should be used only for debugging)

Previous Next Finish Cancel

Figure 245 - Tutoriel 2: création de la connexion JCO

L'assistant nous demande alors de sélectionner le "Message Server". Dans notre cas, on sélectionne la D07 puis on clique sur **Next** :

Create new JCo Destination

1 General Data 2 J2EE Cluster 3 Connection Type 4.1 Appl. Server Connection 4.2 **Msg. Server Connection**

Define the message server, system name and the logon group used by the JCO connection.

Message Server: D07 mshsap15

System Name: D07

Logon Group: PUBLIC

☐ Use SAP Router

SAP Router String:

Previous Next Finish Cancel

Figure 246 - Tutoriel 2: création de la connexion JCO

Ensuite, on nous demande de saisir les informations de connexion. Nous saisissons ici les informations du compte de l'utilisateur technique (JDAVID_TECH dans notre cas) et on clique sur **Next** :

Create new JCo Destination

1 General Data 2 J2EE Cluster 3 Connection Type 4.1 **Appl. Server Connection** 4.2 Msg. Server Connection

Define the required authentication method and (optionally) the parameters needed for a secure network communication (SNC).

Define the used authentication method and (optionally) the parameters needed for a secure network connection (SNC).

User Authentication	Secure Network Connection (SNC)
Used Method: User / Password	SNC Mode: Off
Name: jdavid_tech	SNC Partner Name:
Password:	SNC Security Level:
Confirm Password:	SNC Name:
Language: English	SNC Library Path:

Previous Next Finish Cancel

Figure 247 - Tutoriel 2: création de la connexion JCO

Enfin, l'assistant nous affiche un résumé des paramètres. Il n'y a plus qu'à cliquer sur **Finish** pour terminer la création de la connexion JCO:

Figure 248 - Tutoriel 2: résumé des paramètres de la connexion JCO

On constate alors que la première connexion a un voyant vert, symbolisant ainsi une configuration effectuée:

Name	Status	Create	Preview	Edit	Test	Ping	Remove
VWD_RAPPORT_TUTO2_MODELDATA_DEST	Green	Create	Preview	Edit	Test	Ping	Remove
VWD_RAPPORT_TUTO2_RFC_METADATA_DEST	Red	Create	Preview	Edit	Test	Ping	Remove

Figure 249 - Tutoriel 2: connexions JCO (1/2 active)

Nous devons à présent configurer la 2^{ème} connexion dont le voyant est encore rouge. Les étapes de création sont identiques à celles de la première. Il suffit donc de répéter les étapes précédentes.

Une fois la 2^{ème} connexion configurée, les 2 voyants sont verts:

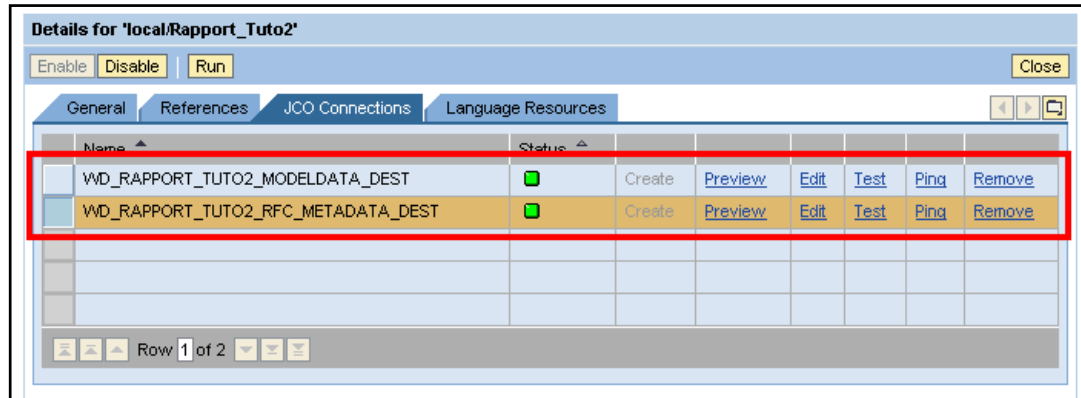


Figure 250 - Tutoriel 2: connexions JCO (actives)

7.5.9.2 Création et lancement de l'application

Il ne reste plus qu'à retourner dans SAP NetWeaver Developer Studio et à créer l'application:

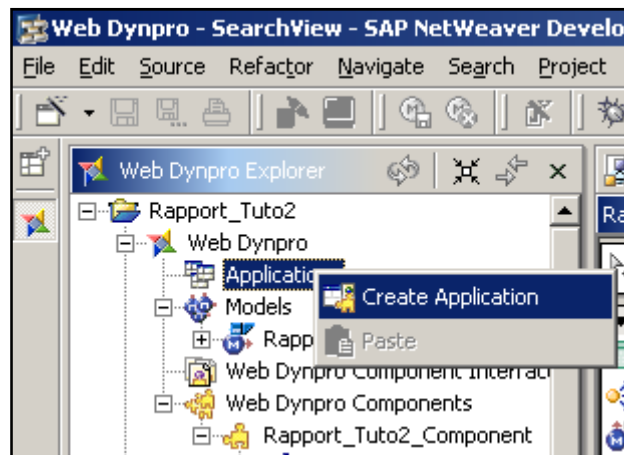


Figure 251 - Tutoriel 2: création de l'application

Une fois créée, nous pouvons l'exécuter. Lors de son premier lancement, il n'est pas nécessaire de la redéployer via "Deploy New Archive and Run", il suffit de sélectionner "Run".

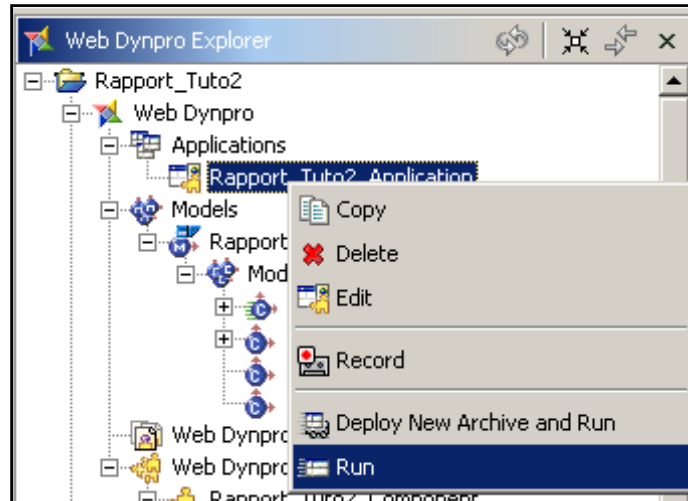


Figure 252 - Tutoriel 2: lancement de l'application

Dans l'éventualité où l'on souhaiterait modifier le projet, il faudra sélectionner "Deploy New Archive and Run" afin que les changements soient pris en compte.

7.5.10 Test & résultat

Une fois l'application lancée, nous pouvons la tester en saisissant un numéro d'employé (ex: 11199130) et en cliquant sur "Rechercher". On constate alors que les informations de cet employé pré-remplissent le formulaire:

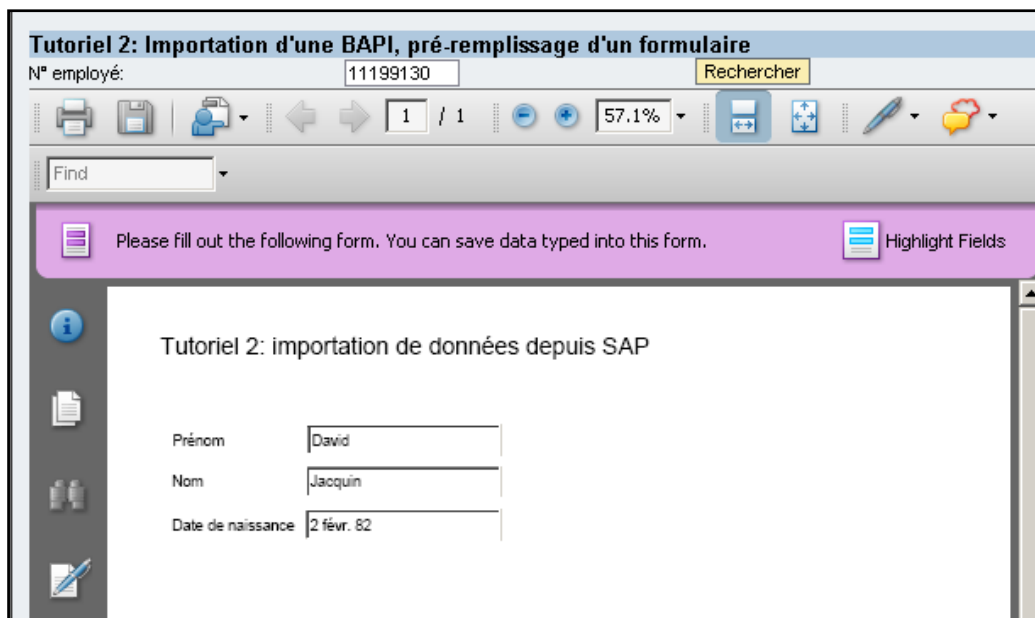


Figure 253 - Tutoriel 2: test de l'application

Ce 2^{ème} tutoriel est à présent terminé, nous avons créé une application qui récupère des informations provenant de SAP.

7.6 Tutoriel 3: réalisation d'une application complète

7.6.1 Objectifs

Nous allons à présent passer au développement d'une application complète. Par "complète" on entend la possibilité de:

- recevoir des données depuis SAP
- pré-remplir certains champs avec ces dernières.
- Ajouter une aide à la saisie sous la forme d'un filtre qui modifie le contenu d'une liste déroulante
- Effectuer un calcul avec 2 valeurs saisies
- Ajouter un contrôle des données au niveau des dates
- Renvoyer des données vers SAP
- Rendre le formulaire exploitable en cas d'impression

Remarque: Ce tutoriel correspond au développement d'un prototype du formulaire destiné à l'Etat du Valais, ainsi qu'au prototype général de ce travail de diplôme.

7.6.2 Description

L'application réalisée proposera un formulaire d'annonce d'absence. L'utilisateur saisit son matricule, choisit une catégorie d'absence puis un type d'absence dans une liste déroulante filtrée ce qui a pour effet de pré-remplir le formulaire avec ces informations. Il saisit ensuite les données relatives à l'absence et clique sur un bouton d'envoi pour les sauvegarder dans ECC 6.0.

Enfin, si l'utilisateur souhaite imprimer le formulaire, il doit pouvoir avoir une liste des types d'absence ainsi que des champs pour pouvoir date et signer le document.

7.6.3 Création du projet

Pour commencer nous créons un nouveau projet que nous allons nommer "Rapport_Tuto3". On crée ensuite un "Component" que l'on nomme "Rapport_Tuto3_Component", qui utilise le package "rapport_tuto3_pkg" et qui utilise une vue (Embed View) "StartView". Dans la structure du projet, cela donne:

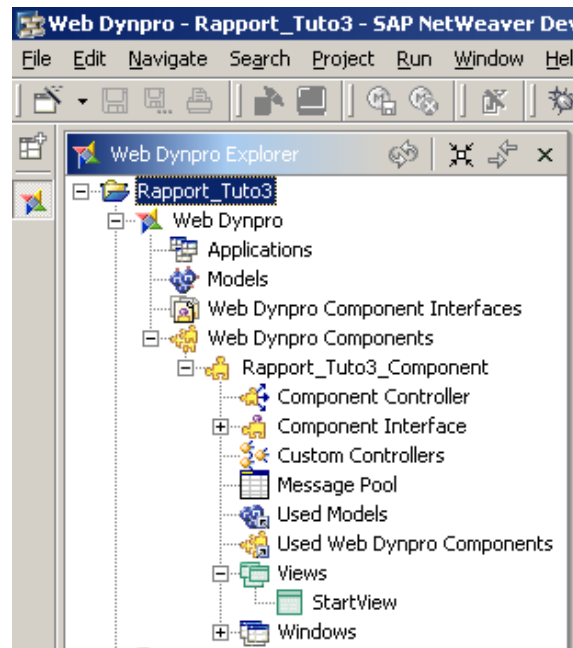


Figure 254 - Tutoriel 3: Web Dynpro Explorer

7.6.4 Importation des BAPIs

Nous allons ensuite importer un modèle qui comportera les RFC/BAPIs suivantes:

- BAPI_PERSDATA_GETDETAILEDLIST: pour obtenir les informations d'un employé.
- BAPI_ABSENCE_CREATE: pour créer une absence pour un employé
- BAPI_EMPLOYEE_ENQUEUE: pour verrouiller un employé
- BAPI_EMPLOYEE_DEQUEUE: pour déverrouiller un employé
- BAPI_TRANSACTION_COMMIT: pour valider l'exécution d'une BAPI
- RFC_ABSENCETYPES_READ: pour obtenir la liste des motifs d'absence.

Remarque:

Bien que les noms de ces RFC/BAPIs soient assez parlants, il convient d'expliquer pourquoi nous avons besoin de BAPI_EMPLOYEE_ENQUEUE et BAPI_EMPLOYEE_DEQUEUE. La raison est simple: lorsque l'on crée une absence pour un employé, ce dernier ne peut pas faire l'objet de modifications tant qu'il n'est pas verrouillé. C'est pourquoi lorsque que notre application appellera la méthode qui envoie les données vers SAP, on ajoutera avant cette dernière un verrouillage de l'employé et un déverrouillage après. Sans ça, bien que la BAPI qui crée l'absence s'exécute normalement, le système refuse la création et on constate par la suite que la table contenant la liste des absences d'un employé reste inchangée.

Une fois les BAPIs importées, notre modèle se présente ainsi:

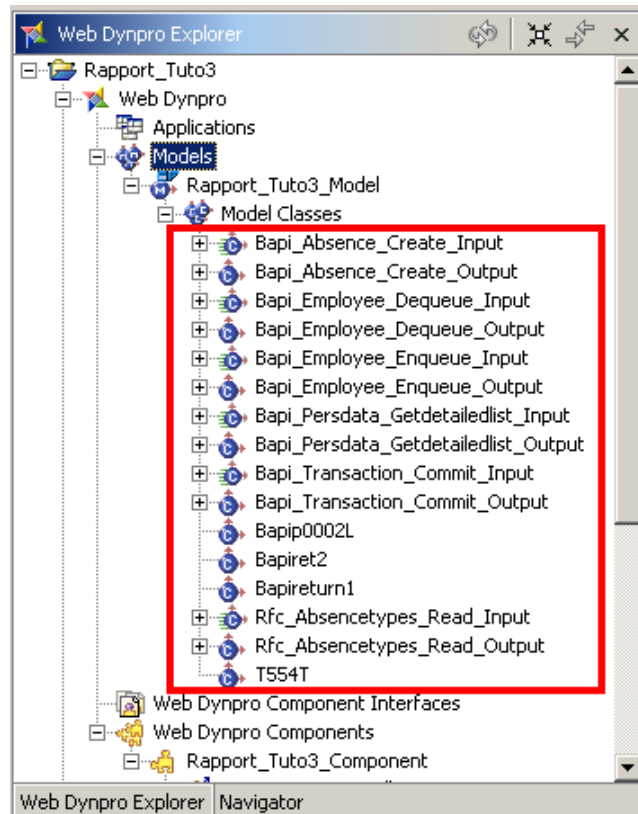


Figure 255 - Tutoriel 3: Modèle dans Web Dynpro Explorer

7.6.5 Création du Custom Controller, déclaration d'usage du modèle

Ensuite, nous créons un Custom Controller (🔗 Custom Controllers) et on déclare l'utilisation d'un modèle (🔗 Used Models). Cela donne:

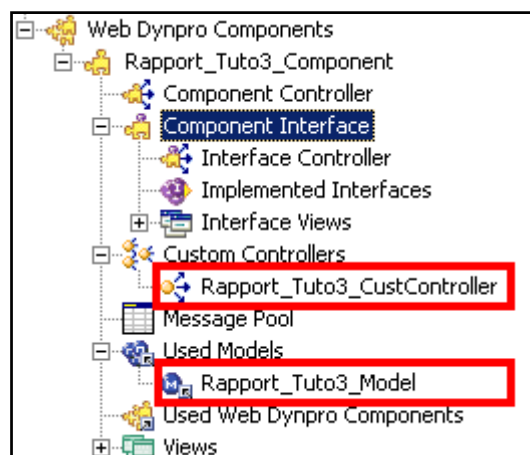


Figure 256 - Tutoriel 3: Custom Controller et modèle utilisé

7.6.6 Mapping

L'étape suivante consiste à créer le contexte du Custom Controller et celui de la vue "StartView". Dans la mesure où nous faisons appel à un bon nombre de BAPIs, il est important de bien savoir comment créer les différents mappings et de ne pas s'y perdre.

Il y a donc 2 conseils à suivre: lors du mapping, renommer les nœuds que l'on retrouverait à double dans le contexte en s'assurant que leurs noms soient très parlants. Ensuite, toujours lors de la création des mappings, surtout pour la vue, se poser à chaque fois la question suivante: "De quoi ai-je besoin?". Exemple: dans la vue "StartView", nous n'aurons pas besoin de la BAPI qui valide la création d'une absence (BAPI_TRANSACTION_COMMIT) étant donné que son exécution aura lieu dans une méthode du contrôleur uniquement.

On ouvre donc le "Data View Modeler" afin de s'occuper des contextes:

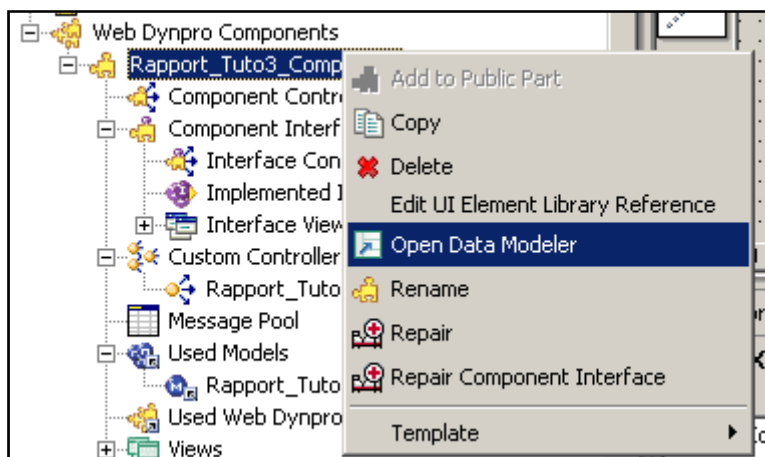



Figure 257 - Tutoriel 3: lancement du Data Modeler

7.6.6.1 Mapping du Custom Controller → Modèle

On va tout d'abord s'occuper du mapping du Custom Controller. A l'aide de l'outil  on établit donc une liaison entre ce dernier et le modèle utilisé.

Ici, nous allons devoir renommer plusieurs nœuds car les BAPIs possèdent toutes un nœud nommé "Output", parfois un nœud "Return" et occasionnellement deux nœuds au nom identique dans une même BAPI. Etant donné qu'un contexte n'autorise pas l'utilisation de 2 noms de nœud similaires, nous ajouterons simplement un suffixe lors du mapping. Suffixe qui devra bien évidemment être clair afin de pouvoir y accéder simplement via le code.

Dans le cas des nœuds "Output" et "Return", nous allons systématiquement ajouter un suffixe qui décrit le rôle de la BAPI. Pour les nœuds que l'on retrouve en double dans une BAPI, on ajoutera en suffixe le nom du nœud parent (le nom se modifie dans la partie droite de la fenêtre, dans la colonne "Name"). Voici quelques exemples:

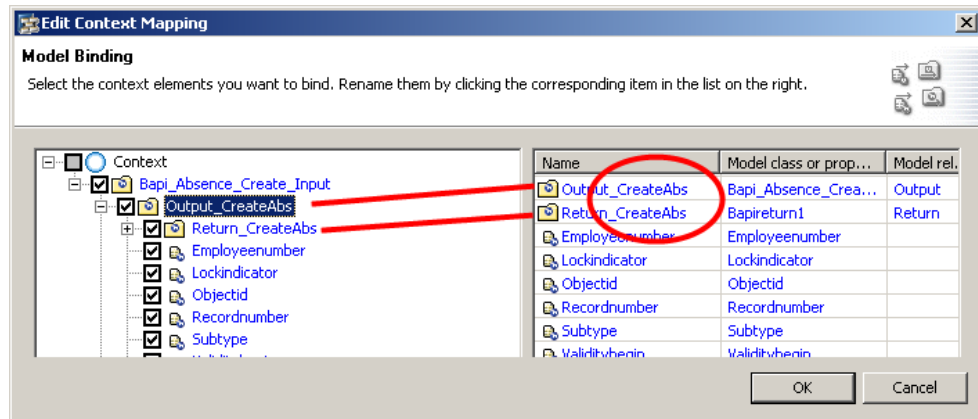


Figure 258 - Tutoriel 3: renommage des nœuds récurrents

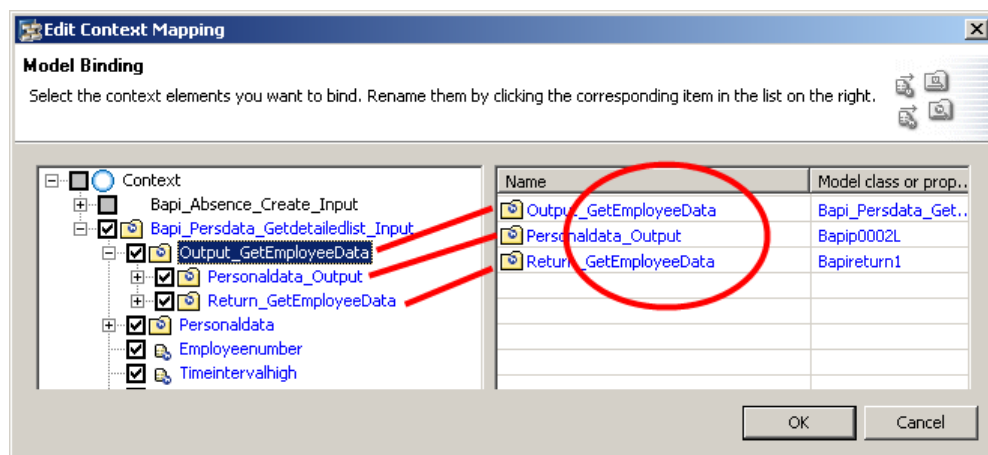


Figure 259 - Tutoriel 3: renommage des nœuds récurrents

Pour le mapping entre le contrôleur et le modèle utilisé, nous allons mapper toutes les BAPIs dont le nom se termine par "_Input", c'est-à-dire toutes celles qui se trouve dans le cadre rouge ci-dessous:

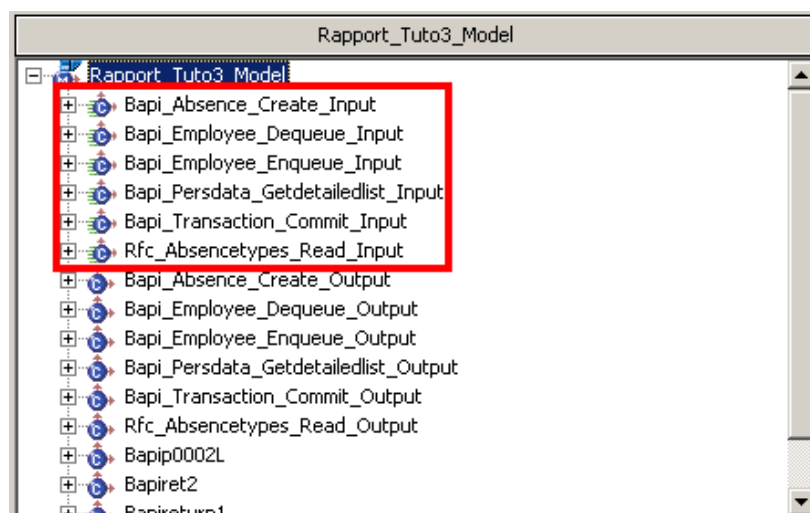


Figure 260 - Tutoriel 3: BAPIs utilisées

Pour toutes les BAPIs, on effectue un mapping de tous les éléments en prenant soin de bien renommer les nœuds comme indiqué quelques lignes plus haut. Une fois cette étape effectuée, le mapping ressemble à ceci et on clique sur "Finish" pour valider:

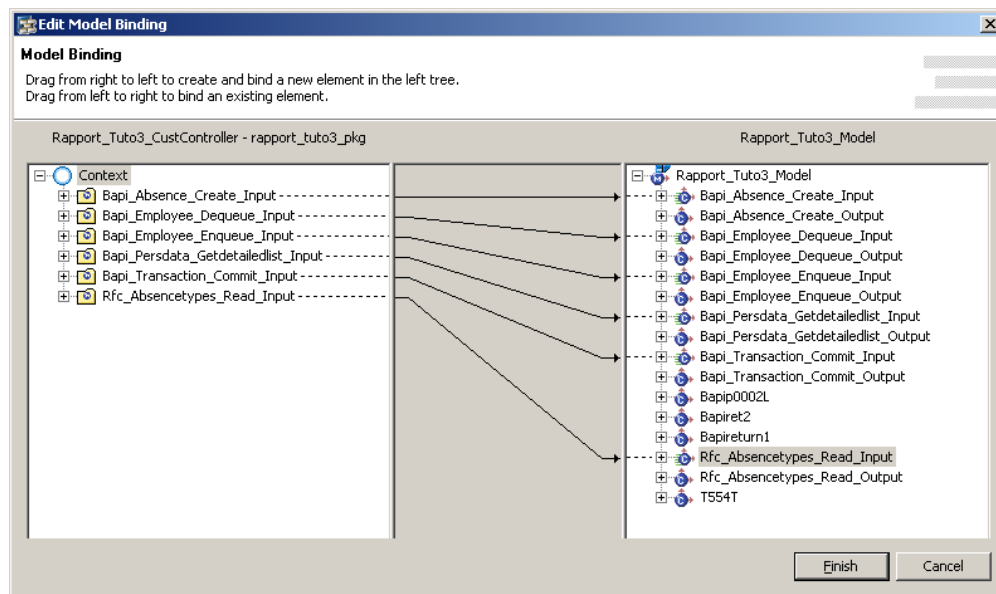


Figure 261 - Tutoriel 3: mapping du Custom Controller vers le modèle

7.6.7 Nœud, attribut personnalisé

7.6.7.1 Explication

Toujours pour le contexte du contrôleur, nous allons à présent réaliser une étape particulière. En effet, nous allons ajouter un nœud personnalisé qui contiendra un attribut que nous allons créer et qui pourra être énuméré d'une manière différente. Cet attribut-là contiendra à la fois le numéro d'un type d'absence et une description contenant le numéro suivi par le descriptif auquel il correspond. Tout cela dans un seul et même attribut, et non 2.

Example:

- Id: 0220
- Descriptif: 0220: Visite chez le médecin.

Cette étape est indispensable si l'on souhaite par la suite remplir une liste déroulante contenant une valeur combinant 2 attributs.

Dans une application Web Dynpro classique, il suffit simplement d'ajouter une liste déroulante (`DropDownListByIndex`) et d'y attacher un attribut pour que celle-ci soit automatiquement remplie par toutes les valeurs de l'attribut. Dans notre cas, cela ne convient pas car nous avons besoin de combiner deux valeurs. En effet, un numéro seul n'est pas forcément parlant et un descriptif sans numéro n'offre pas une information complète.

De plus, si l'on souhaite ajouter une liste déroulante de la même manière dans un formulaire interactif, elle ne récupère et affiche que la valeur actuellement sélectionnée dans le contexte. Il est donc intéressant de mentionner cette étape particulière, que ce soit pour ce formulaire ou pour savoir quoi faire lors du futur développement d'un autre formulaire.

7.6.7.2 Création d'un type simple

Pour commencer, on crée un nouveau type en allant dans la structure du projet, en effectuant un clic droit sur Dictionaries→Local Dictionary→Data Types→Simple Type et en sélectionnant "Create Simple Type":

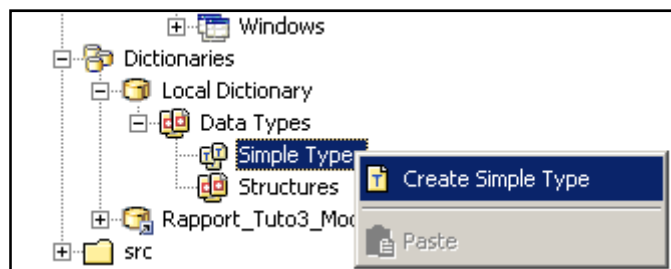


Figure 262 - Tutoriel 3: création d'un type simple

On lui donne un nom, on utilise le package principal de l'application et on valide en cliquant sur "Finish":

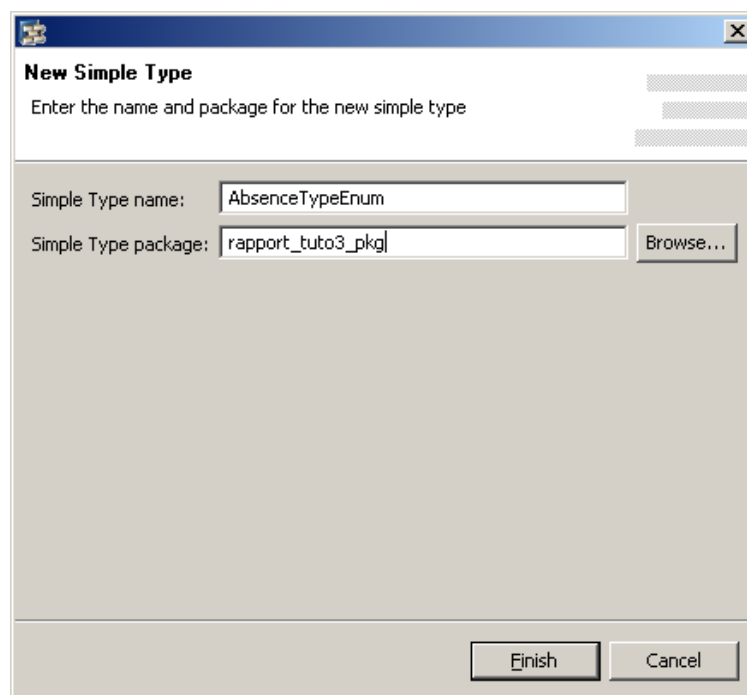


Figure 263 - Tutoriel 3: nommage du type simple

Le type est à présent créé et sa fenêtre de définition s'affiche:

The screenshot shows the 'Simple Type Definition' dialog box in SAP. The title bar includes 'Rapport_Tuto3_CustController', 'StartView', 'Rapport_Tuto3', and 'AbsenceTypeEnum'. The dialog is divided into several sections:

- General Information:** Contains fields for 'Name' (AbsenceTypeEnum), 'Package' (rapport_tuto3_pkg), 'Dictionary' (Local Dictionary), 'Base Type' (with a 'Browse...' button), 'Built-In Type' (string), and 'Description'.
- Length Constraints:** Includes fields for 'Maximum Length', 'Minimum Length', 'Fixed Length', 'Total Digits', and 'Fraction Digits'.
- Value Constraints:** Includes fields for 'Minimum Inclusive', 'Maximum Inclusive', 'Minimum Exclusive', and 'Maximum Exclusive'.
- Navigation:** At the bottom, there are tabs for 'Definition', 'Enumeration', 'Representation', and 'Database'.

Figure 264 - Tutoriel 3: Définition du type simple

Nous n'avons pas besoin de modifier ses paramètres.

Si on clique sur l'onglet "Enumeration", on peut constater qu'il est possible de remplir un tableau avec des valeurs:

Figure 265 - Tutoriel 3: énumération du type simple

7.6.7.3 Création du nœud personnalisé

187

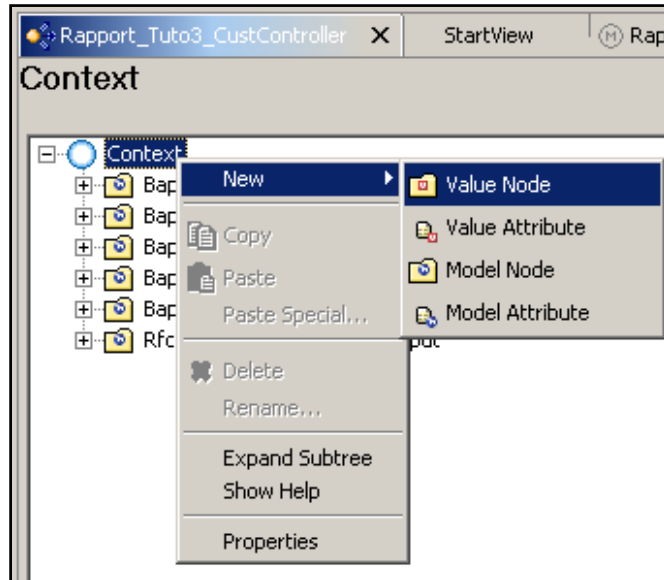


Figure 266 - Tutoriel 3: création d'un Value Node

Une fois le nœud créé, on ajoute un attribut ("Value Attribute") pour le type d'absence. Cela donne:

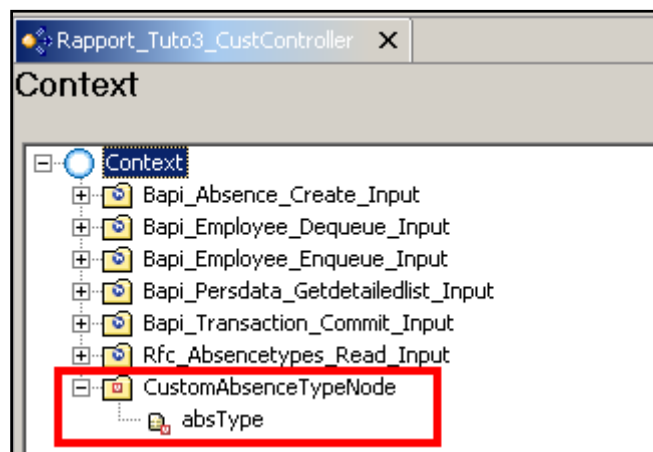


Figure 267 - Tutoriel 3: nœud personnalisé pour les types d'absence

Ensuite, on modifie les propriétés de l'attribut correspondant aux types d'absence ("absType" dans notre cas) et on change le type:

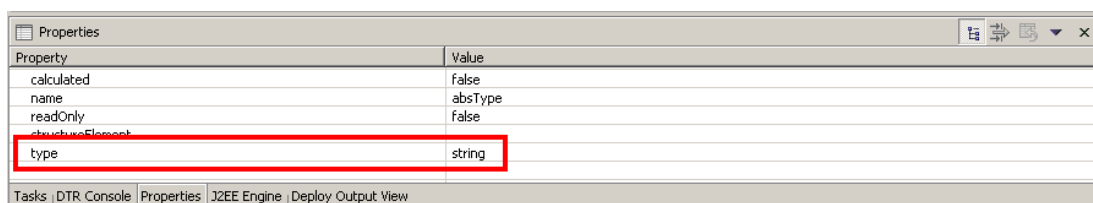



Figure 268 - Tutoriel 3: type de l'attribut absType (avant)

Le type qui nous intéresse n'est pas "string", mais bien celui que nous venons de créer.
Pour ce faire, on clique d'abord sur le bouton  :

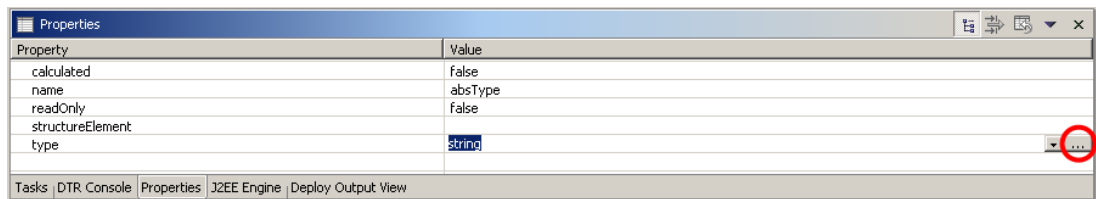


Figure 269 - Tutoriel 3: type de l'attribut absType (modification)

On sélectionne ensuite le type que l'on a créé dans le dictionnaire local puis dans le package de l'application et on valide en cliquant sur "OK":

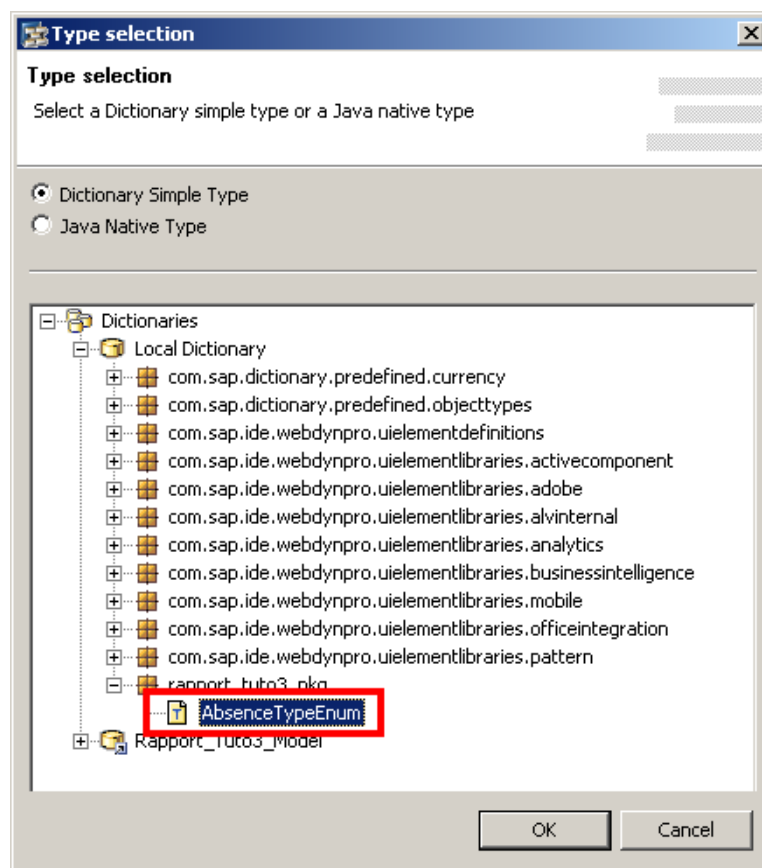


Figure 270 - Tutoriel 3: type de l'attribut absType (résultat)

Le type apparaît alors dans les propriétés de l'attribut:

Properties	
Property	Value
calculated	false
name	absType
readOnly	false
structureElement	
type	rapport_tuto3_pkg.AbsenceTypeEnum
Tasks DTR Console Properties J2EE Engine Deploy Output View	

Figure 271 - Tutoriel 3: type de l'attribut absType (résultat)

Afin de pouvoir filtrer ces types d'absences, nous allons encore créer un "Value Node" contenant un "Value Attribute" qui correspondra aux catégories. Pour cet attribut, on laisse son type inchangé, c'est-à-dire "string". Le contexte du contrôleur se présente alors ainsi:

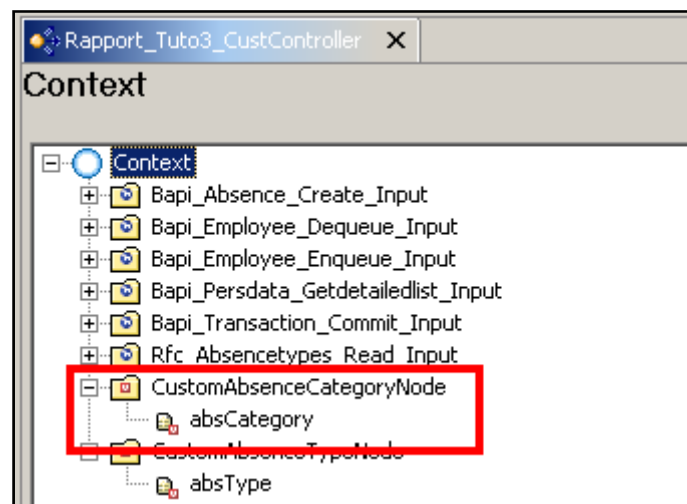


Figure 272 - Tutoriel 3: nœud personnalisé pour les catégories d'absence

7.6.8 Mapping StartView → Custom Controller

L'étape suivante consiste à créer un mapping entre la vue et le contrôleur. Ici, nous n'aurons pas besoin d'effectuer un renommage des nœuds étant donné que nous venons de le faire pour le contrôleur.

Nous allons tout d'abord lier la vue au contrôleur en nous rendant dans les propriétés de la vue:

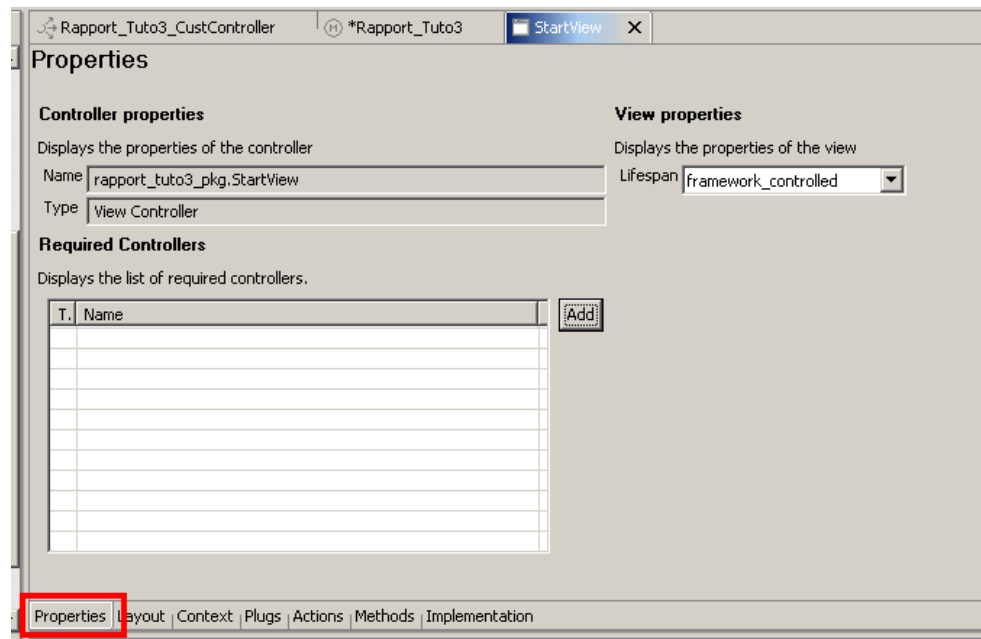


Figure 273 - Tutoriel 3: onglet propriétés de la vue StartView

On clique ensuite sur le bouton "Add". La fenêtre ci-dessous s'affiche:

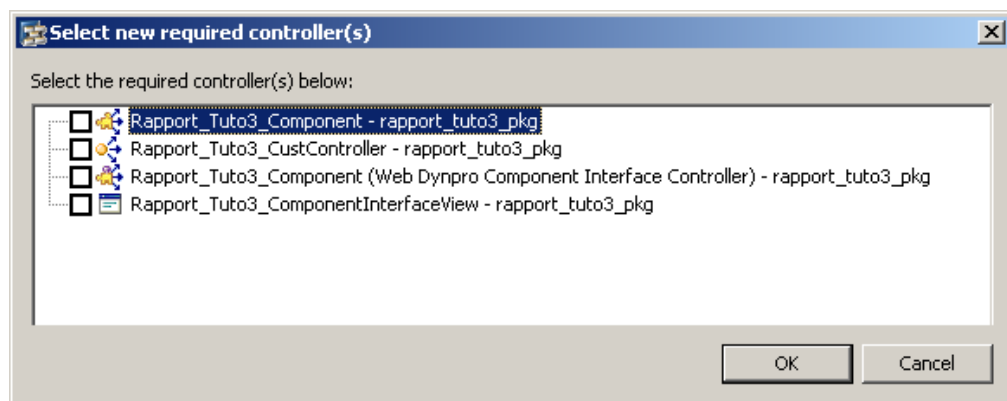


Figure 274 - Tutoriel 3: choix du contrôleur requis pour la vue

On coche la case correspondant au Custom Controller que nous avons créé et on clique sur OK:

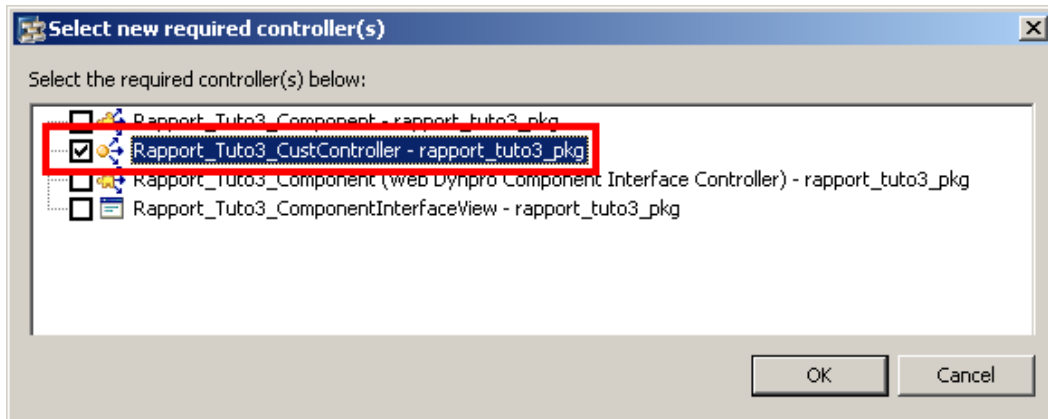



Figure 275 - Tutoriel 3: sélection du Custom Controller

A noter que ce que nous venons de faire est équivalent à la création d'une liaison entre la vue et le Custom Controller dans le schéma de l'application ( Open Data Modeler).

Nous pouvons à présent retourner dans le contexte de la vue.

Cependant, avant de mapper les nœuds un par un, nous devons garder en tête que cette opération a une contrainte: le formulaire interactif. En effet, il faut savoir qu'un formulaire ne peut être lié qu'à un seul et unique nœud. Or il devra être en mesure d'utiliser des attributs provenant de plusieurs nœuds à la fois (type d'absence, nom de la personne, etc..). Par conséquent, nous allons créer un nœud ("Model Node") spécialement pour le formulaire interactif et nous mapperons à l'intérieur de ce dernier tous les attributs dont nous aurons besoin.

Toujours avant d'effectuer le mapping, nous allons en profiter pour créer l'habituel attribut "pdfSource" de type binaire.

Avant le mapping, le contexte de la vue contient le nœud et l'attribut que nous venons de créer et il ressemble donc à ça:

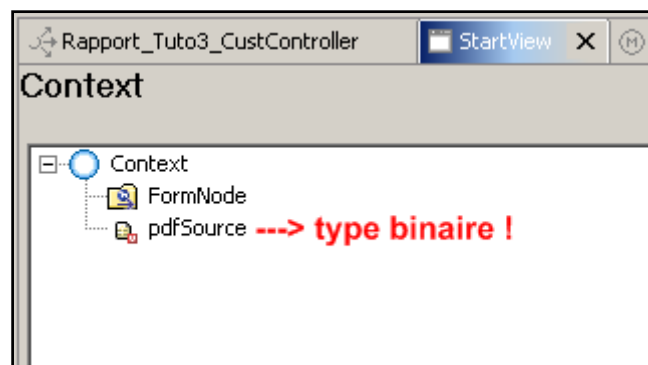


Figure 276 - Tutoriel 3: type de l'attribut pdfSource

Nous allons à présent mapper tous ce qu'il faut pour notre formulaire. Pour commencer, on effectue un clic droit sur le nœud que nous avons créé et on sélectionne "Edit Context Mapping...". A ne pas confondre avec "Edit Model Binding...":

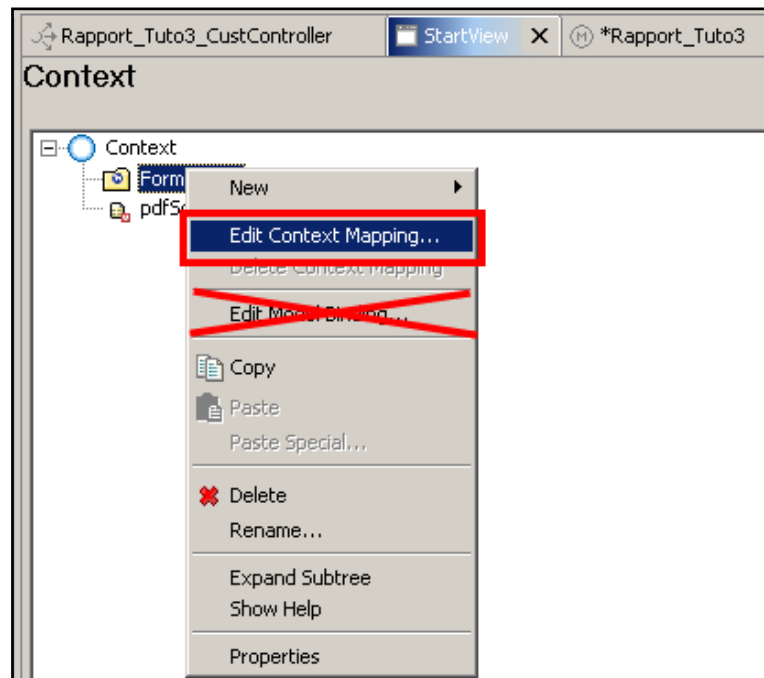


Figure 277 - Tutoriel 3: édition du mapping vue → Custom Controller

Une fenêtre contenant tous les nœuds du contexte du Custom Controller apparaît alors:

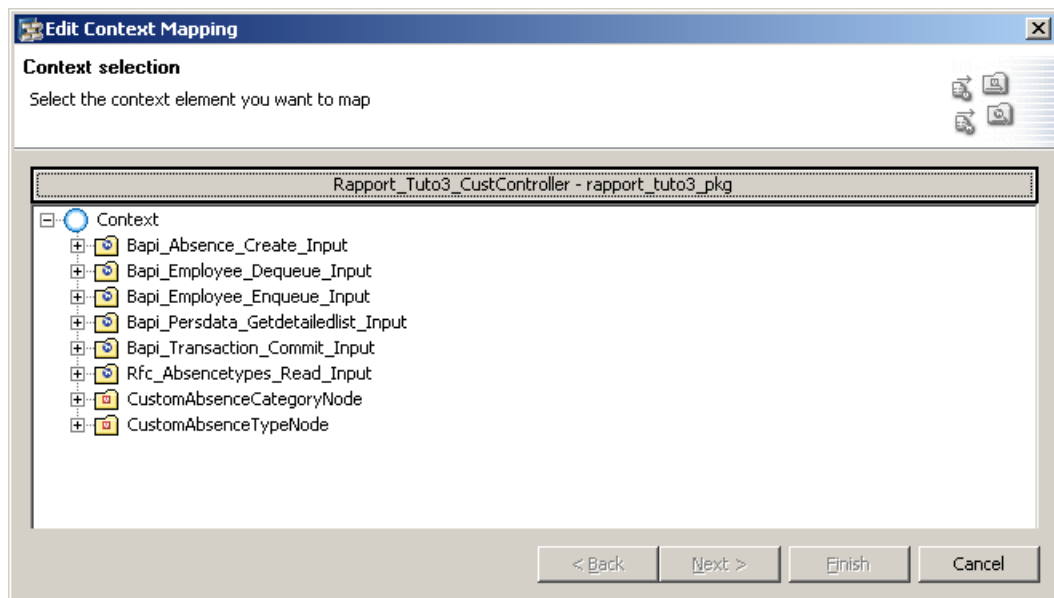


Figure 278 - Tutoriel 3: choix du nœud à mapper

On sélectionne en premier la BAPI qui s'occupe de créer une absence:



Edit Context Mapping

Context Mapping

Select the context elements you want to map. Rename them by clicking the corresponding item in the list on the right.

Left pane (Tree view):

- Context (Selected)
- FormNode
- Output_CreateAbs
- Absencehours
- Absencetype
- Employeeenumber
- End
- Nocommit
- Start
- Validitybegin
- Validityend
- pdfSource

Right pane (Table):

Name	Mapped element
FormNode	Bapi_Absence_Create_Input

Buttons: < Back, Next >, Finish, Cancel

- Absencehours (nombre d'heures d'absence)
- Absencetype (code du motif d'absence)
- End (heure de fin de l'absence)
- Start (heure de début de l'absence)
- Validitybegin (date de début de l'absence)

- Validityend (date de fin de l'absence)

Avant de cliquer sur "Finish" pour valider, le mapping se présente ainsi:

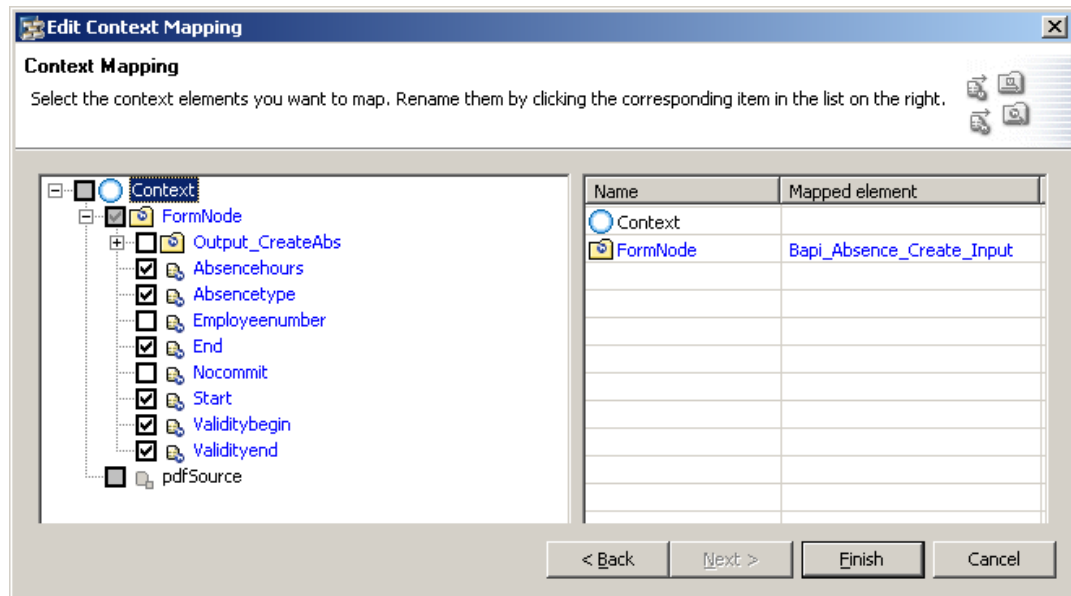


Figure 281 - Tutoriel 3: éléments mapping pour Bapi_Absence_Create_Input

On se retrouve ensuite à nouveau dans le contexte de la vue.

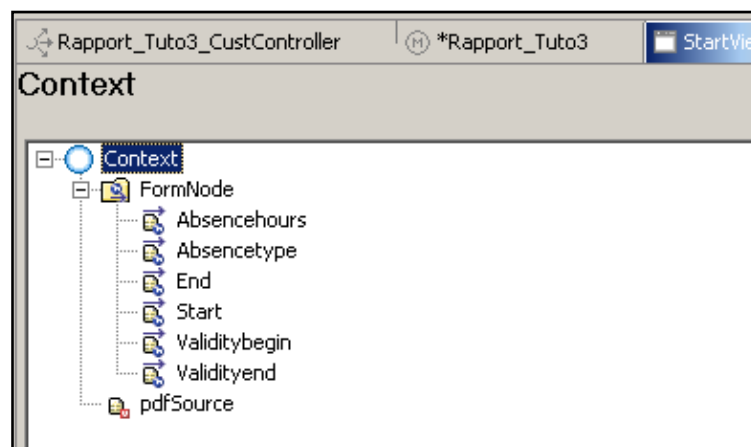


Figure 282 - Tutoriel 3: contexte de StartView

Nous avons à présent besoin d'ajouter tout ce qui touche à la récupération des informations d'un employé. On effectue à nouveau un clic droit sur le nœud du formulaire et on sélectionne "Edit Context Mapping...":

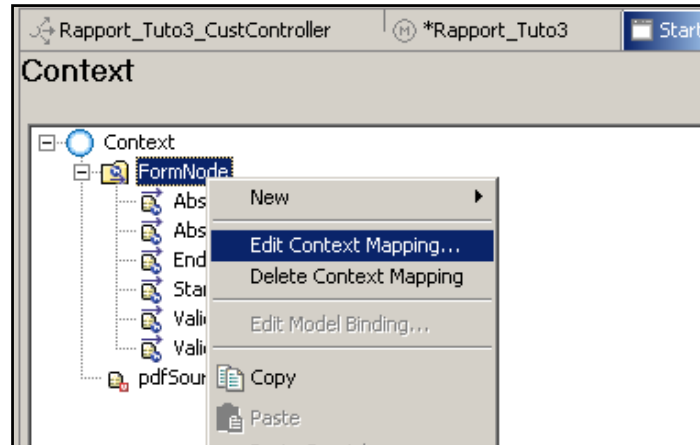


Figure 283 - Tutoriel 3: édition du mapping

Sur la fenêtre suivante, on sélectionne la BAPI qui retourne les informations d'un employé et on clique sur "Next":

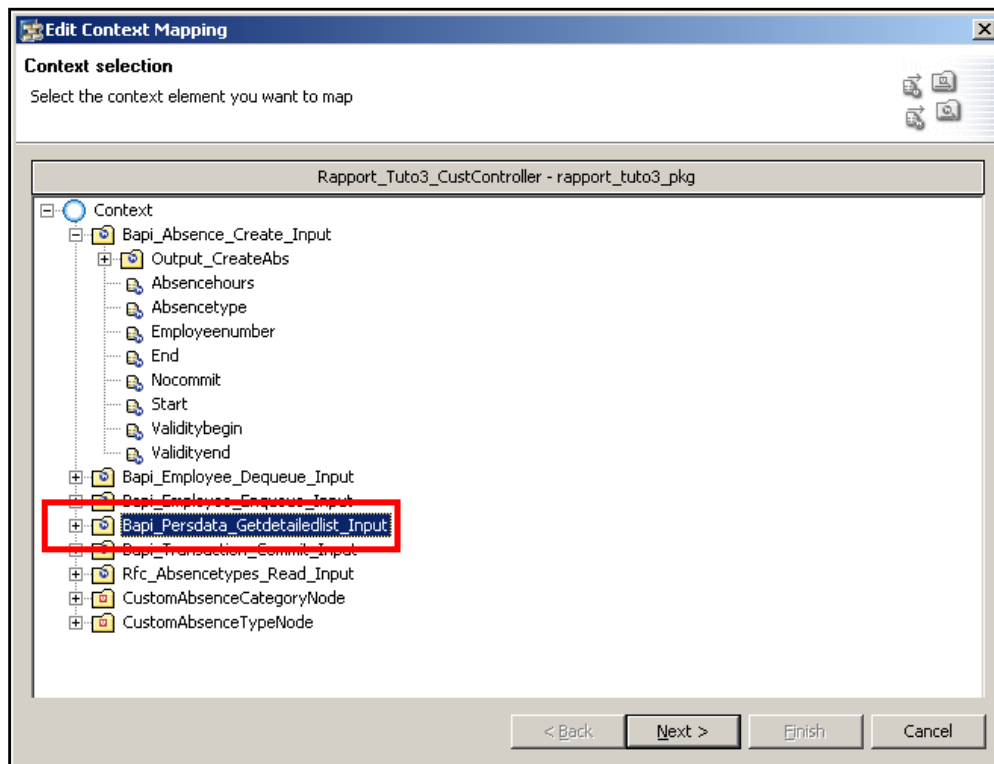


Figure 284 - Tutoriel 3: sélection de Bapi_Persdata_Getdetailelist_Input

La fenêtre suivante nous propose alors de configurer le mapping entre le nœud du formulaire et la BAPI:



- Dateofbirth (retour: date de naissance)
- Employeeeno (retour: numéro de l'employé)
- Firstname (retour: prénom de l'employé)
- Lastname (retour: nom de l'employé)

Edit Context Mapping

Context Mapping

Select the context elements you want to map. Rename them by clicking the corresponding item in the list on the right.

Context

- FormNode
 - Output_GetEmployeeData
 - Personaldata_Output
 - Academicgrade
 - Aristocratictitle
 - Birthplace
 - Countryofbirth
 - Dateofbirth
 - Employeeeno
 - Firstname
 - Formofaddress
 - Fullname
 - Gender
 - Idnumber
 - Initials
 - Knownas
 - Language
 - Language_Iso
 - Lastname**
 - Lockindic
 - Maritalstatus
 - Maritalstatussince
 - Middlename

Name	Mapped element
Lastname	Lastname

< Back Next > Finish Cancel

[illegible]

Jacquin David
HES-SO Valais, mai 2008

On clique alors sur "Finish", ce qui nous ramène vers le contexte de la vue:

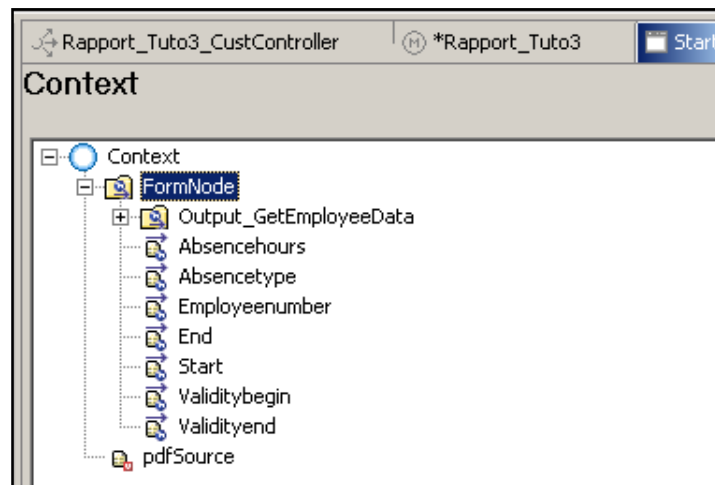


Figure 288 - Tutoriel 3: contexte de la vue StartView

Nous venons d'ajouter l'attribut qui sert de paramètre à la BAPI qui fournit les informations d'un employé ainsi que les attributs qu'on reçoit en retour et qui pré-rempliront le formulaire.

Pour terminer la création du contexte de la vue, on se rend dans le "Data Modeler" en faisant un clic droit sur le nom du Custom Controller dans la structure du projet et en sélectionnant "Open Data Modeler":

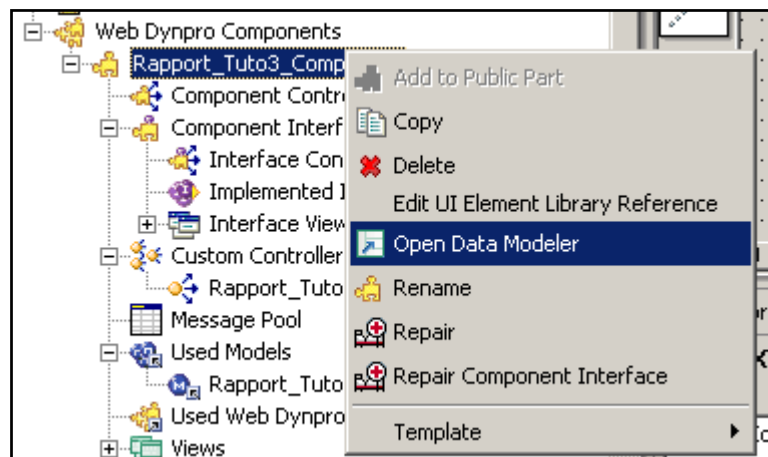


Figure 289 - Tutoriel 3: lancement du Data Modeler

Le schéma du projet apparaît alors. On constate que la vue et le contrôleur sont déjà liés:

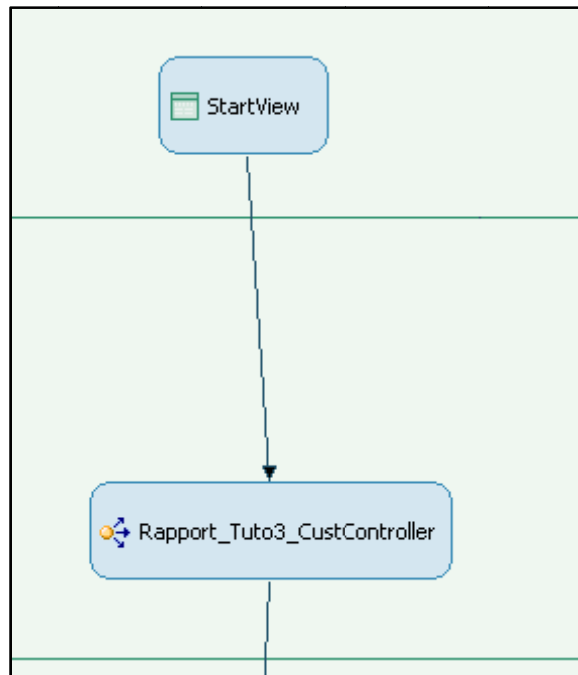


Figure 290 - Tutoriel 3: liaison StartView → Custom Controller

On ouvre ensuite la fenêtre permettant de configurer le mapping en double-cliquant sur la flèche reliant la vue au contrôleur:

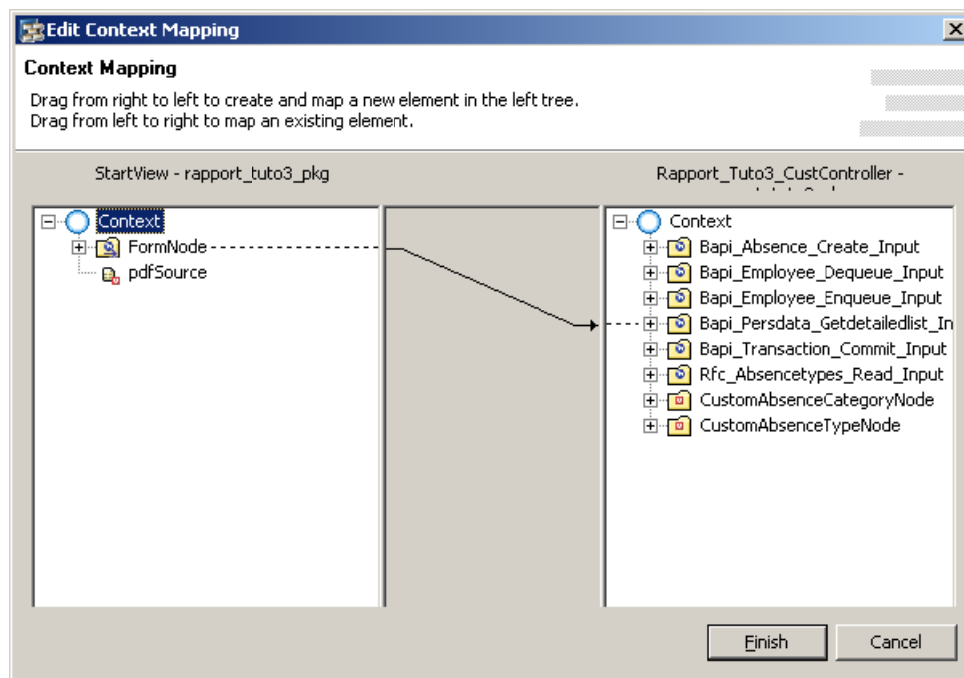



Figure 291 - Tutoriel 3: mapping StartView → Custom Controller

Nous devons encore mapper deux nœuds : celui des catégories d'absence et celui des types d'absence. Pour les deux, on effectue un drag & drop vers la racine du contexte de la vue ( Context) et on coche à chaque fois toutes les cases disponibles. Avant de cliquer sur "Finish" pour valider, cela donne:

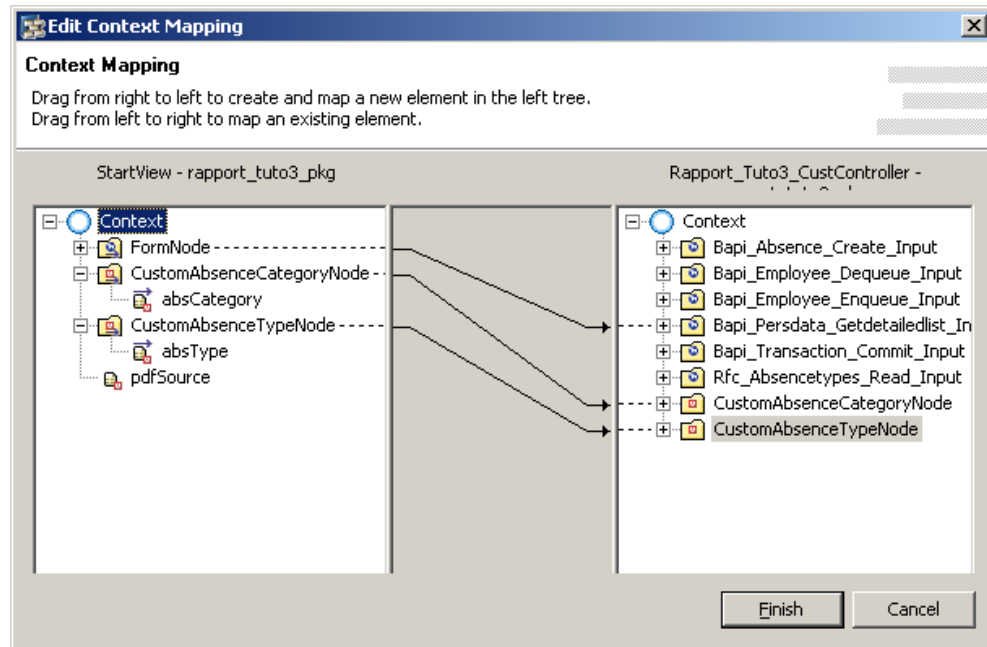


Figure 292 - Tutoriel 3: mapping StartView → Custom Controller

7.6.9 Création des méthodes du Custom Controller

L'étape suivante consiste à ajouter les méthodes qu'utilisera le Custom Controller. Nous aurons besoin de 3 méthodes que l'on nommera:

- execute_Search (pour trouver un employé)
- execute_CreateAbsence (pour créer une absence)
- execute_Filter (pour filtrer la liste déroulante contenant les types d'absence)

Pour execute_Search et execute_CreateAbsence, on accède à la fenêtre des méthodes en cliquant sur l'onglet "Methods" du Custom Controller, on clique sur le bouton "Add", on saisit simplement leurs noms.

Pour la méthode execute_Filter, il faudra définir un paramètre qui correspondra à la catégorie sélectionnée. Pour ce faire, on clique comme d'habitude sur le bouton Add et on saisit son nom:

New Method

Method Properties
Enter the method properties below.

Name:

Return type: ...

☐ Array type

Dimension:

< Back Next > Finish Cancel

Figure 293 - Tutoriel 3: nommage de la méthode

Contrairement aux 2 méthodes précédentes, on clique sur le bouton "Next". On nous propose ensuite de définir les paramètres:

New Method

Create Parameters
Create the parameters.

Parameters
Edit the parameters below.

Name	Type

New
Up
Down

< Back Next > Finish Cancel

Figure 294 - Tutoriel 3: paramètres de la méthode

On clique alors sur le bouton "New", la fenêtre suivante apparaît et nous demande de saisir un nom de paramètre et un type. On le nomme "selectedCategory", on choisit le type "integer" et on clique sur le bouton "Finish":

Figure 295 - Tutoriel 3: création du paramètre

Le nouveau paramètre apparaît alors dans la liste. Il ne reste plus qu'à cliquer sur "Finish" pour valider:

Name	Type
selectedCategory	integer

Figure 296 - Tutoriel 3: paramètres de la méthode

La fenêtre des méthodes du contrôleur se présente alors ainsi:

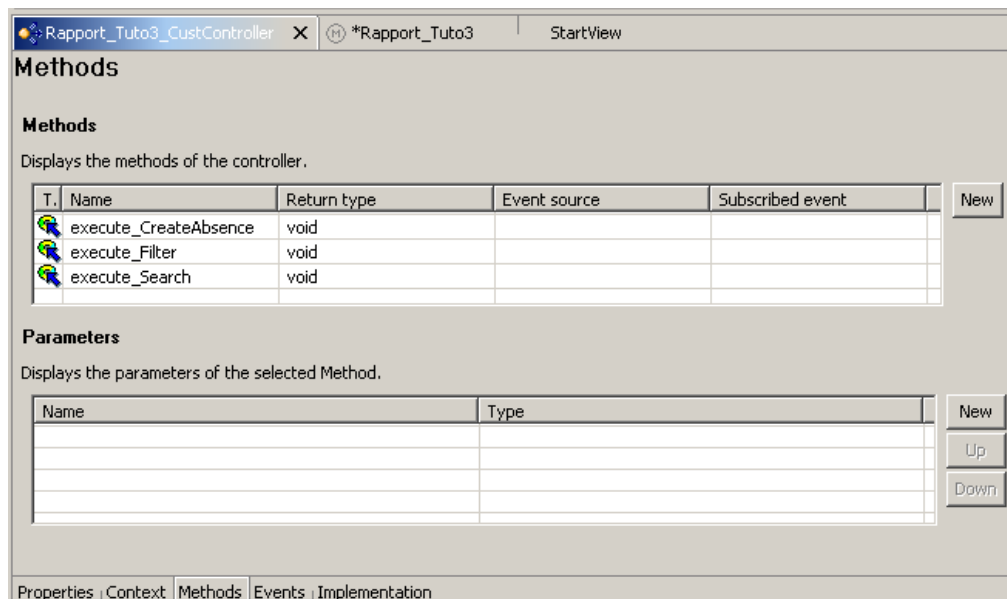


Figure 297 - Tutoriel 3: liste des méthodes du Custom Controller

7.6.10 Implémentation

Nous allons maintenant passer à l'implémentation du contrôleur. Les méthodes que nous allons coder sont, dans l'ordre:

- execute_Filter(int)
- wdDoInit()
- execute_Search()
- execute_CreateAbsence()

Note: lors de l'implémentation de ces méthodes, il sera peut-être nécessaire d'effectuer un clic droit puis Source → Organize Imports si vous obtenez des erreurs indiquant que ce que vous tapez n'est pas défini ("Undefined").

7.6.10.1 execute_Filter(int) (Custom Controller)

Commençons par execute_Filter(int). Cette méthode reçoit en paramètre la catégorie sélectionnée et modifie l'attribut qui contient les types d'absence en conséquence. Voici le code:

```
public void execute_Filter( int selectedCategory )
{
    //@@begin execute_Filter()

    //définition des intervalles de filtrage selon
    //la catégorie sélectionnée
    int minRange = 0;
    int maxRange = wdContext.nodeAbsencetext_Output().size();

    if (selectedCategory == 1) {
        maxRange = 24;
    }
}
```



```
}  
if (selectedCategory == 2) {  
    minRange = 24;  
    maxRange = 60;  
}  
if (selectedCategory == 3) {  
    minRange = 60;  
}  
  
//on définit une variable qui permettra de manipuler l'attribut  
//utilisant un type personnalisé.  
IWDAttributeInfo customAbsTypeInfo =  
    wdContext.nodeCustomAbsenceTypeNode()  
        .getNodeInfo().getAttribute("absType");  
  
ISimpleTypeModifiable customAbsTypeText =  
    customAbsTypeInfo.getModifiableSimpleType();  
  
IModifiableSimpleValueSet customAbsTypeValueSet =  
    customAbsTypeText.getSVServices()  
        .getModifiableSimpleValueSet();  
  
//on vide l'attribut de toute ses valeurs pour  
//pouvoir le remplir par la suite  
customAbsTypeValueSet.clear();  
  
//on définit 2 variables de type string:  
// > type: code du motif d'absence  
// > text: descriptif du motif d'absence  
String type, text;  
  
//on lance une boucle qui parcourt les types d'absence  
//retournés par la RFC, en tenant compte de l'intervalle  
//qui correspond à la catégorie.  
for (int i = minRange; i < maxRange; i++) {  
    wdContext.nodeAbsencetext_Output().setLeadSelection(i);  
    text = wdContext.currentAbsencetext_OutputElement().getAtext();  
    type = wdContext.currentAbsencetext_OutputElement().getAwart();  
    //on ajoute dans l'attribut absType la combinaison type + texte  
    customAbsTypeValueSet.put(type, type + ": " + text);  
}  
//@@end  
}
```

7.6.10.2 wdDoInit() (Custom Controller)

On passe ensuite à la méthode wdDoInit() :

```
public void wdDoInit()  
{  
    //@@begin wdDoInit()  
  
    //=====  
    //Execution de la BAPI qui retourne les types d'absence  
    Rfc_Absencetypes_Read_Input inputAbsTypes =  
    new Rfc_Absencetypes_Read_Input();  
  
    wdContext.nodeRfc_Absencetypes_Read_Input().bind(inputAbsTypes);  
}
```

```
try {
    wdContext.currentRfc_Absencetypes_Read_InputElement()
        .modelObject().execute();
}
catch (Exception e){
    e.printStackTrace();
}

//=====
//Création d'un noeud vide pour les types d'absence customisés
IPublicRapport_Tuto3_CustController
    .ICustomAbsenceTypeNodeElement inputCustAbsType =
        wdContext.createCustomAbsenceTypeNodeElement();

wdContext.nodeCustomAbsenceTypeNode().bind(inputCustAbsType);

//=====
//Création des catégories
String[] categoryNames = new String[] {
    "Toutes", "Catégorie A", "Catégorie B", "Catégorie C",
};

List allCategories = new ArrayList();
for (int i = 0; i < categoryNames.length; i++) {
    IPublicRapport_Tuto3_CustController
        .ICustomAbsenceCategoryNodeElement category =
            wdContext.createCustomAbsenceCategoryNodeElement();

    category.setAbsCategory(categoryNames[i]);
    allCategories.add(category);
}

wdContext.nodeCustomAbsenceCategoryNode().bind(allCategories);

//=====
//Filtrage initial: afficher toutes les catégories
execute_Filter(0);

//=====
//Création d'un noeud vide pour toutes les autres BAPIS
Bapi_Absence_Create_Input inputAbsCreate =
    new Bapi_Absence_Create_Input();

wdContext.nodeBapi_Absence_Create_Input().bind(inputAbsCreate);
Bapi_Persdata_Getdetailedlist_Input inputPersData =
    new Bapi_Persdata_Getdetailedlist_Input();

wdContext.nodeBapi_Persdata_Getdetailedlist_Input()
    .bind(inputPersData);

Bapi_Employee_Enqueue_Input inputEnqueue =
    new Bapi_Employee_Enqueue_Input();

wdContext.nodeBapi_Employee_Enqueue_Input().bind(inputEnqueue);

Bapi_Employee_Dequeue_Input inputDequeue =
    new Bapi_Employee_Dequeue_Input();
```

```
wdContext.nodeBapi_Employee_Dequeue_Input().bind(inputDequeue);

Bapi_Transaction_Commit_Input inputCommit =
    new Bapi_Transaction_Commit_Input();

wdContext.nodeBapi_Transaction_Commit_Input().bind(inputCommit);

//@@end
}
```

7.6.10.3 execute_Search() (Custom Controller)

A présent, la méthode execute_Search():

```
public void execute_Search( )
{
    //@@begin execute_Search()
    try {
        wdContext.currentBapi_Persdata_Getdetailedlist_InputElement()
            .modelObject().execute();

        wdContext.nodeOutput_GetEmployeeData().invalidate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    //@@end
}
```

7.6.10.4 execute_CreateAbsence() (Custom Controller)

Et enfin, la méthode execute_CreateAbsence(). Comme nous pouvons le voir, en plus de créer l'absence, elle s'occupe de verrouiller, puis de déverrouiller un employé, afin que la saisie soit acceptée par le système:

```
public void execute_CreateAbsence( )
{
    //@@begin execute_CreateAbsence()
    //Récupération de l'ID de l'employé pour qu'il soit utilisé par
    //les //BAPIs de création d'absence, de verrouillage et de
    //déverrouillage.
    String employeeId =
        wdContext.currentBapi_Persdata_Getdetailedlist_InputElement
        ()
            .getEmployeeNumber();

    wdContext.currentBapi_Employee_Enqueue_InputElement()
        .setNumber(employeeId);
    wdContext.currentBapi_Employee_Dequeue_InputElement()
        .setNumber(employeeId);
    wdContext.currentBapi_Absence_Create_InputElement()
        .setEmployeeNumber(employeeId);

    //enregistrement de l'absence:
    //verrouillage, validation
    //création, validation
    //déverrouillage, validation
}
```

```
try {
    wdContext.currentBapi_Employee_Enqueue_InputElement()
        .modelObject().execute();

    wdContext.currentBapi_Transaction_Commit_InputElement()
        .modelObject().execute();

    wdContext.currentBapi_Absence_Create_InputElement()
        .modelObject().execute();

    wdContext.currentBapi_Transaction_Commit_InputElement()
        .modelObject().execute();

    wdContext.nodeOutput_CreateAbs().invalidate();

    wdContext.currentBapi_Employee_Dequeue_InputElement()
        .modelObject().execute();

    wdContext.currentBapi_Transaction_Commit_InputElement()
        .modelObject().execute();

} catch (Exception e) {
    e.printStackTrace();
}

//@@end
}
```

7.6.11 Actions de StartView

On passe à présent à la vue. Pour commencer nous allons passer par son onglet "Actions" pour en créer 4. Pour les 4, il suffit juste de cliquer sur "New", saisir un nom, un texte court mais parlant et enfin de cliquer sur "Finish". Les actions à créer sont:

- selectType (utilisée quand on sélectionne un type d'absence dans la liste déroulante. Permet de pré-remplir le formulaire)
- createAbsence (utilisée lorsque l'on cliquera sur le bouton d'envoi du formulaire)
- filter (utilisée lorsque l'on sélectionne une catégorie d'absence)
- search (utilisée lorsque l'on clique sur le bouton de recherche pour trouver un employé)

Une fois cette étape effectuée, la fenêtre des actions de la vue se présente ainsi:

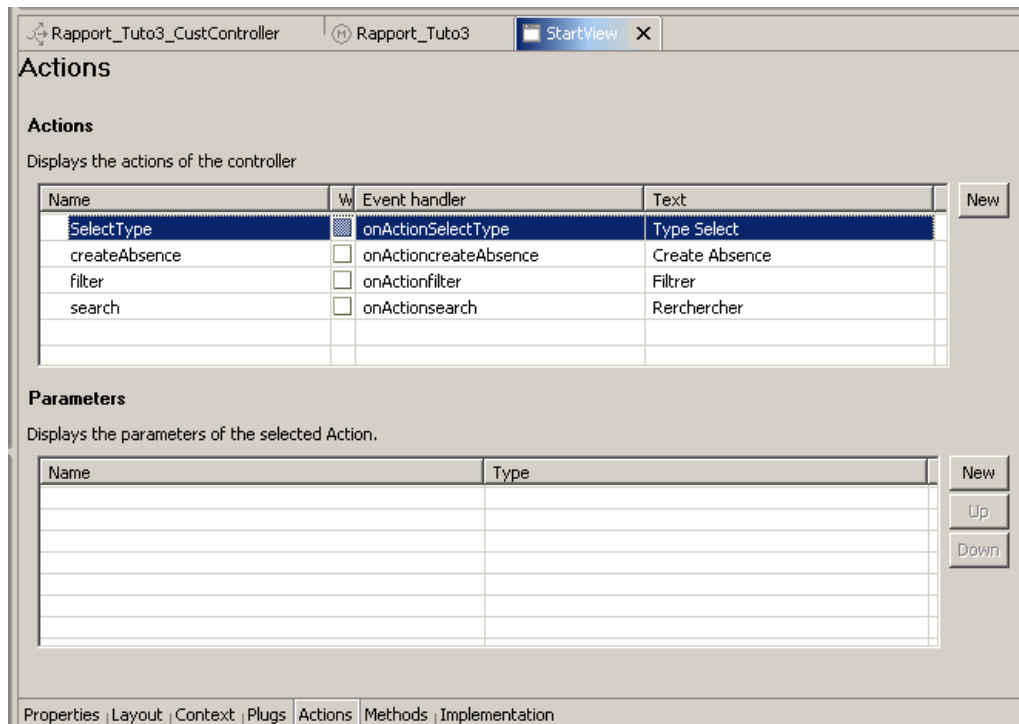


Figure 298 - Tutoriel 3: liste des actions de la vue StartView

7.6.12 Implémentation des actions de StartView

Nous allons immédiatement implémenter ces actions. Une fois qu'on clique sur l'onglet "Implémentation", nous pouvons les retrouver en bas à gauche dans la fenêtre d'Outline. Leur nom est systématiquement précédé par le préfixe "on Action".

7.6.12.1 onActionSelecteType (StartView)

On commence par onActionSelectType:

```
public void onActionsearch(com.sap.tc.webdynpro.progmodel
                           .api.IWDCustomEvent wdEvent )
{
    //@@begin onActionsearch(ServerEvent)
    wdThis.wdGetRapport_Tuto3_CustControllerController()
        .execute_Search();
    //@@end
}
```

7.6.12.2 onActioncreateAbsence (StartView)

On passe à onActioncreateAbsence:

```
public void onActioncreateAbsence(com.sap.tc.webdynpro.progmodel
                                .api.IWDCustomEvent wdEvent )
{
    //@@begin onActioncreateAbsence(ServerEvent)
    wdThis.wdGetRapport_Tuto3_CustControllerController()
        .execute_CreateAbsence();
    //@@end
}
```

7.6.12.3 onActionfilter (StartView)

Ensuite, onActionfilter:

```
public void onActionSelectType(com.sap.tc.webdynpro.progmodel
                               .api.IWDCustomEvent wdEvent )
{
    //@@begin onActionSelectType(ServerEvent)
    String selectedAbsenceType =
        wdContext.currentCustomAbsenceTypeNodeElement().getAbsType();

    wdContext.currentFormNodeElement()
        .setAbsencetype(selectedAbsenceType);
    //@@end
}
```

7.6.12.4 onActionSearch (StartView)

Et enfin onActionSearch:

```
public void onActionsearch(com.sap.tc.webdynpro.progmodel
                           .api.IWDCustomEvent wdEvent )
{
    //@@begin onActionsearch(ServerEvent)
    wdThis.wdGetRapport_Tuto3_CustControllerController()
        .execute_Search();
    //@@end
}
```

7.6.13 Modification du nom du fichier .pdf

Nous allons aussi en profiter pour modifier le nom par défaut du fichier PDF lorsqu'on veut en faire une sauvegarde locale. En effet, sans code, un formulaire se nomme toujours "unknown.pdf", ce qui n'est pas très parlant. Avec le code que l'on va rajouter, il se nommera par défaut "Formulaire_Absence_fr.pdf".

Pour ce faire, on modifie la méthode wdDoInit() de la vue (à ne pas confondre avec la méthode du contrôleur) et on ajoute les lignes suivantes:

```
public void wdDoInit()
{
    //@@begin wdDoInit()
    IWDAAttributeInfo attInfo =
        wdContext.currentContextElement().node()
        .getNodeInfo().getAttribute( "pdfSource" );

    ISimpleTypeModifiable type = attInfo.getModifiableSimpleType();
    IWDModifiableBinaryType binaryType = (IWDModifiableBinaryType)
    type;
    binaryType.setFileName( "Formulaire_Absence_fr.pdf" );
    binaryType.setMimeType( WDWebResourceType.PDF );
    //@@end
}
```

Nous pouvons maintenant réaliser le design de la vue en cliquant sur l'onglet "Layout". Pour la vue, nous aurons besoin au minimum des éléments suivants, avec les propriétés éventuellement mentionnées:

Eléments	Propriétés	Valeurs
InputField	value	FormNode.EmployeeNumber
Button	onAction	search
RadioButtonGroupByIndex	colCount	4
	Texts	CustomAbsencecategoryNode.absCategory
	onSelect	Filter
DropDownByKey	selectedKey	CustomAbsenceTypeNode.absType
	onSelect	SelectType
Interactive Form	dataSource	FormNode
	displayType	native
	pdfSource	pdfSource
	onSubmit	createAbsence

Remarque:

- Lors du design, le groupe de boutons affiche plusieurs lignes. C'est normal, elles n'apparaîtront plus lors du déploiement.

- Pour la propriété "displayType", nous sélectionnons "native" à la place de "ActiveX" afin de créer une application qui ne nécessite pas l'Active Component Framework (ACF) et qui peut fonctionner avec Firefox.

Après avoir ajouté ces éléments, on peut encore insérer différents labels et champs de texte (TextView) pour rendre la vue plus compréhensible.

Nous n'allons pas revenir sur les autres détails de création du design étant donné que ce ne serait qu'une répétition des tutoriels précédents. Au final, le design de notre vue peut ressembler à cela:

Figure 299 - Tutoriel 3: design de StartView

7.6.14 Design du formulaire interactif

Nous passons maintenant au design du formulaire interactif. On effectue donc un clic droit sur son nom dans la fenêtre d'Outline en bas à gauche et sélectionne "Edit", ce qui lance l'Adobe LiveCycle Designer.

7.6.14.1 Exemple de résultat

Avant d'aborder les étapes de sa création plus en détail, voici un exemple de ce à quoi il ressemblera dans la fenêtre de design une fois terminé:

CANTON DU VALAIS
KANTON WALLIS

CONTROLE DES ABSENCES
DEPARTEMENT DES FINANCES, DES INSTITUTIONS ET DE LA SECURITE
SERVICE CANTONAL DE L'INFORMATIQUE

Sion, le

Nom Matricule

Prénom

Date de naissance

Code motif

Motifs absences		
1100 = Vacances	1200 = Maladie intérieure à 3 jours	1400 = Service militaire
1105 = Vacances 1/2 jour	1210 = Maladie extérieure à 3 jours	1410 = Ecole de recrue
1120 = Congé naissance	1220 = Voie médicale	1420 = Protection civile
1130 = Congé non payé	1230 = VM hors canton	1430 = Cours J+5
1140 = Congé spécial	1240 = Traitements récurrents	1500 = Compensation HB
1150 = Congé maternité	1250 = Urgences médicales	1510 = Compensation HV
1155 = Congé d'adoption	1300 = Accident professionnel	1600 = Affaires publiques
1160 = Congé légal	1310 = Accident non professionnel	1900 = Divers
	1320 = Voie médicale après accident	

Date de début Heure de début

Date de fin Heure de fin

Heures d'absence **Calculer**

Envoyer

Lieu, date: Signature:

Figure 300 - Tutoriel 3: formulaire, aperçu du résultat

7.6.14.2 Liaisons des éléments de saisie

Pour les liaisons entre les attributs du nœud et les éléments de saisie, voici un schéma qui explique comment les disposer via le drag & drop:

Figure 301 - Tutoriel 3: liaison des données dans le formulaire

7.6.14.3 Partie supérieure

Regardons plus en détail la partie supérieure:

Figure 302 - Tutoriel 3: formulaire: partie supérieure

En haut à droite, on place un champ de date. Pour ce faire, on sélectionne un "TextField" dans la librairie et on change son type en "Date/Time Field":

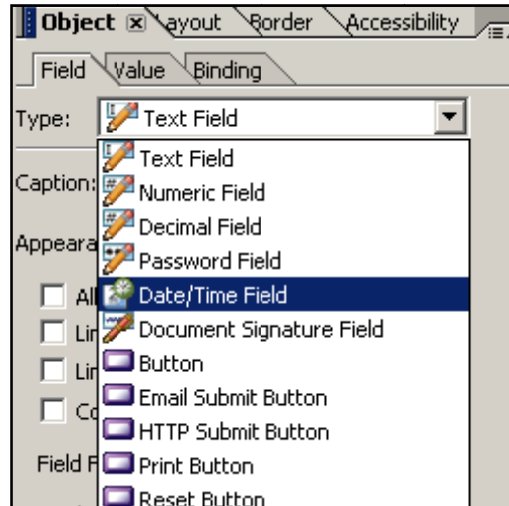


Figure 303 - Tutoriel 3: formulaire, sélection du type de champ de saisie

On change retire aussi ses bordures, on règle son format d'affiche ("Display Format") et son format d'édition ("Edit Format") en DD.MM.YYYY:

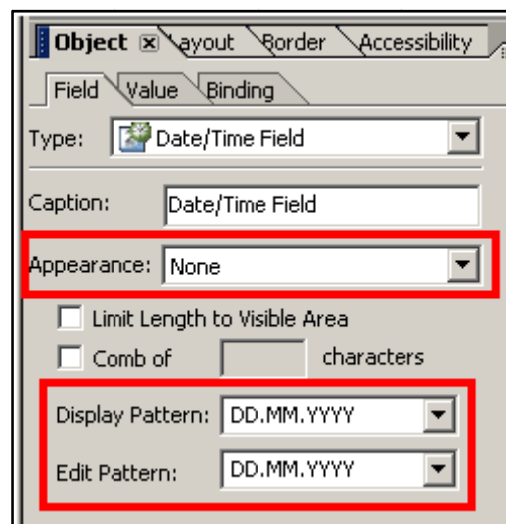


Figure 304 - Tutoriel 3: formulaire, modification de l'affichage du champ

On souhaite que ce champ affiche automatiquement la date du jour, sans qu'il soit possible de le modifier. On sélectionne donc "Calculated – Read Only" dans l'onglet "Value":

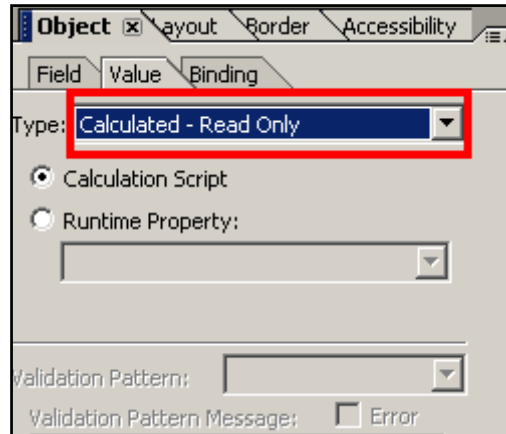


Figure 305 - Tutoriel 3: formulaire, modification du type de valeur

Enfin, on saisit dans la partie supérieure le code qui va générer la date du jour en utilisant l'évènement "layout:ready" et le langage "FormCalc". Le code est:

```
$.rawValue = Num2Date(Date(), "DD.MM.YYYY")
```

On a donc:

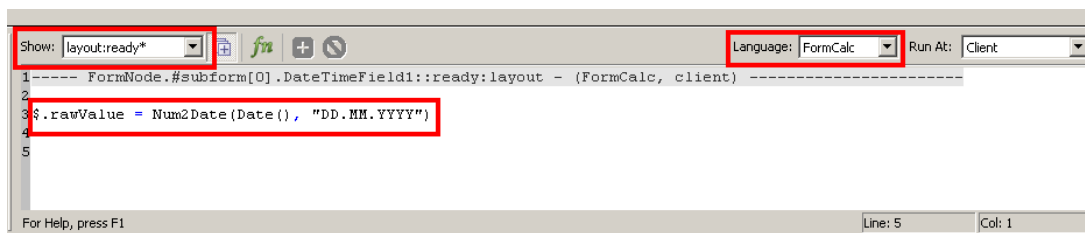


Figure 306 - Tutoriel 3: formulaire, calcul de la date du jour

On ajoute ensuite une image (ici, le logo de l'Etat du Valais) en cliquant sur Insert→Standard→Image. Un champ apparaît alors:

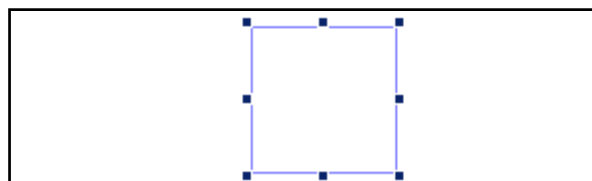


Figure 307 - Tutoriel 3: formulaire, insertion d'une image

En double-cliquant dessus, une première fenêtre apparaît. On clique sur "OK":

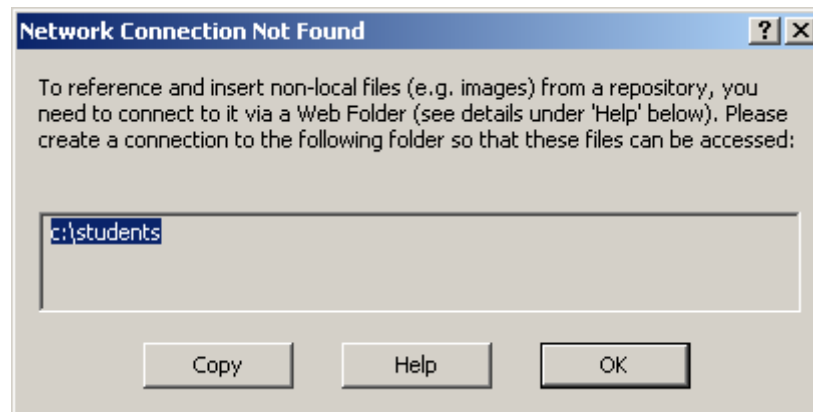


Figure 308 - Tutoriel 3: formulaire, message sur la connexion réseau

Un explorateur de fichier s'ouvre alors et nous demande d'indiquer quelle image on souhaite afficher:

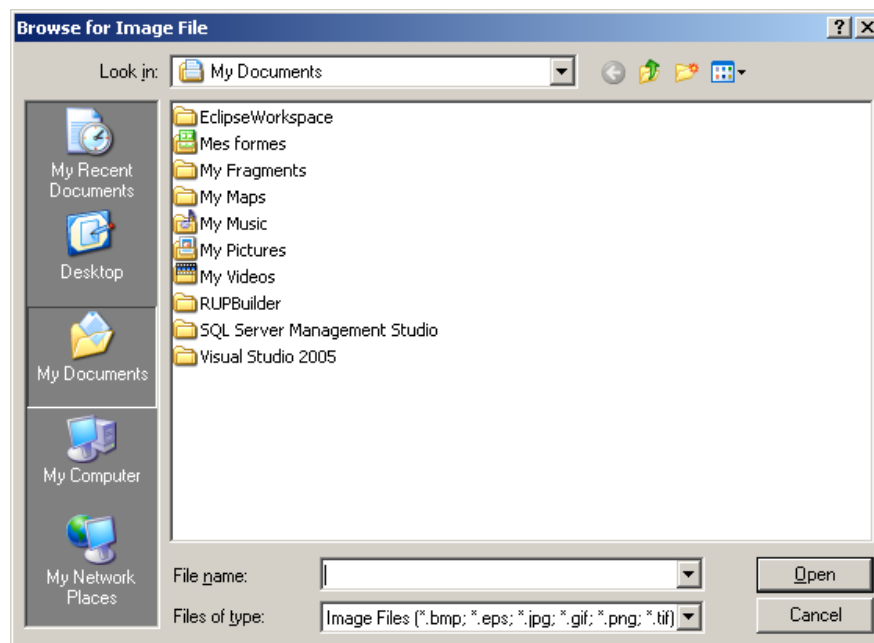


Figure 309 - Tutoriel 3: formulaire, explorateur de fichier (image)

Une fois l'image trouvée, on clique sur "Open". L'image s'affiche dans la fenêtre de design du formulaire:



Figure 310 - Tutoriel 3: formulaire, image

Il faut encore aller dans l'onglet "Object" puis "Draw" et cocher la case "Embed Image Data". En effet, cette option permet d'attacher l'image au formulaire. Sans cette dernière, l'image ne s'affichera pas une fois le formulaire déployé:

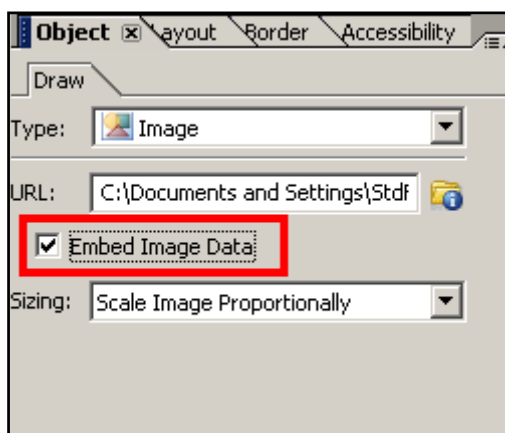


Figure 311 - Tutoriel 3: formulaire, liaison de l'image au formulaire

On ajoute ensuite un champ de texte (Text) pour le titre. Il suffit ensuite de redimensionner le champ et de jouer avec la taille de la police pour obtenir le résultat voulu.

Les champs de saisie pour le nom, prénom et matricule ont ces propriétés en commun:

Onglets	Propriétés	Valeurs
Object → Field	Appearance	Solid Box
Object → Value	Type	Read Only

Le champ de saisie de la date de naissance a les propriétés suivantes:

Onglets	Propriétés	Valeurs
Object → Field	Appearance	Solid Box
	Display Pattern	DD.MM.YYYY
	Edit Pattern	DD.MM.YYYY
Object → Value	Type	Read Only

7.6.14.4 Partie centrale

On passe ensuite à la partie centrale du formulaire:

Code motif

Motifs absences

1100 = Vacances	1200 = Maladie inférieure à 3 jours	1400 = Service militaire
1105 = Vacances 12 jour	1210 = Maladie supérieure à 3 jours	1410 = Ecole de recrue
1120 = Congé naissance	1230 = Visite médicale	1420 = Protection civile
1130 = Congé non payé	1235 = Viti tiers canton	1430 = Cours 2+5
1140 = Congé spécial	1240 = Traitements accutents	1500 = Compensation HS
1150 = Congé maternité	1250 = Urgences médicales	1510 = Compensation HV
1155 = Congé d'adoption	1300 = Accident professionnel	1600 = Affaires publiques
1160 = Congé flexité	1310 = Accident non-professionnel	1900 = Divers
	1320 = Visite médicale après accident	

Date de début **Heure de début**

Date de fin **Heure de fin**

Heures d'absence **Calculer**

Envoyer

Figure 312 - Tutoriel 3: formulaire, partie centrale

Le champ du type d'absence (Code motif) a les paramètres suivants:

Onglets	Propriétés	Valeurs
Object → Field	Appearance	Solid Box
	Comb of ... characters	4
Object → Value	Type	Read Only

A droite du Code motif, on retrouve une image qui liste tous les types d'absence. Etant donné que nous avons une liste déroulante qui nous aide à choisir le bon type, cette liste ne devrait apparaître que si on imprime le formulaire et qu'on souhaite le remplir à la main. On a donc les propriétés suivantes pour cette image:

Onglets	Propriétés	Valeurs
Object → Draw	Embed Image Data	(case cochée)
	Presence	Visible (Print Only)

Pour les champs de date de début et de fin, nous avons les paramètres suivants:

Onglets	Propriétés	Valeurs
Object → Field	Appearance	Solid Box
	Display Pattern	DD.MM.YYYY
	Edit Pattern	DD.MM.YYYY
Object → Binding	Name	dateStart (pour la date de début)
	Name	dateEnd (pour la date de fin)

Nous allons aussi leur rajouter du code qui permettra de les faire afficher la date du jour par défaut lorsque le formulaire sera affiché. Lors des 2 ajouts, on fera appel à l'événement "docReady" et au langage "FormCalc". On a:

Date de début (dateStart):

```
if (dateStart.isNull) then
    dateStart.formattedValue = Num2Date(date(), "DD.MM.YYYY")
endif
```

Date de fin (dateEnd):


```
if (dateEnd.isNull) then
    dateEnd.formattedValue = Num2Date(date(), "DD.MM.YYYY")
endif
```

Pour les champs d'heures de début et de fin de l'absence, nous avons les paramètres suivants:

Onglets	Propriétés	Valeurs
Object → Field	Appearance	Solid Box
	Display Pattern	HH:MM
	Edit Pattern	HH:MM
Object → Binding	Name	hourStart (pour l'heure de début)
	Name	hourEnd (pour l'heure de fin)

Pour le champ correspondant au nombre d'heures d'absence on a:

Onglets	Propriétés	Valeurs
Object → Field	Appearance	Solid Box
Object → Binding	Name	hoursAbs

On ajoute ensuite un bouton simple ( Button) qui permettra de calculer le nombre d'heures de l'absence et de les saisir dans le champ correspondant. Pour l'instant, il est préférable de laisser à l'utilisateur le choix de saisir le chiffre lui-même ou de laisser le formulaire l'estimer étant donné que le calcul que nous allons proposer est extrêmement simple et ne prend pas en compte certaines contraintes, comme par exemple le fait d'être absent la veille d'un jour férié (ce qui correspond à une journée d'absence, et non deux). De plus, un bouton n'a pas d'utilité si on imprime un formulaire, par conséquent nous voulons qu'il ne soit visible que lorsqu'il est utilisé sur un ordinateur.

Les propriétés du bouton sont donc:

Onglets	Propriétés	Valeurs
Object → Field	Control Type	Regular
	Presence	Visible (Screen only)

Nous ajoutons aussi le code suivant pour l'évènement "mouseUp" et le langage "FormCalc":

```
if (HasValue(hourStart) AND HasValue(hourEnd)) then
    hoursAbs =
        ((Date2Num(dateEnd.formattedValue, "DD.MM.YYYY") -
          (IsoTime2Num(hourEnd) -
           IsoTime2Num(hourStart)) / 1000 / 60 / 60
        )
    endif
```

On ajoute ensuite le bouton qui permettra d'envoyer les données du formulaire vers SAP. On utilise ici un bouton "Submit" dans la catégorie "WebDynProNative" de la librairie:

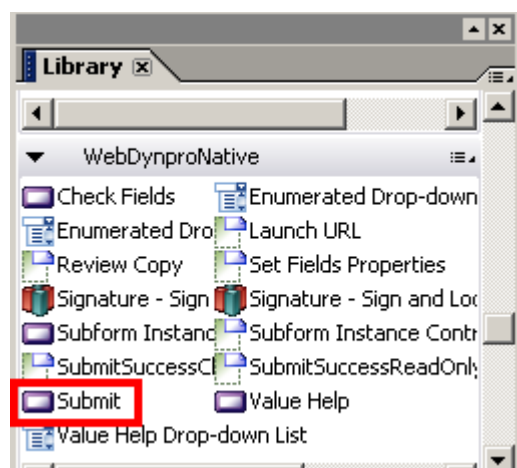


Figure 313 - Tutoriel 3: formulaire, bouton Submit (natif)

Remarque: Attention à ne pas confondre avec la catégorie "WebDynProActiveX". Pour rappel, notre formulaire a comme propriété pour le type d'affichage "Native". Il faut pas conséquent utiliser les contrôles appropriés sans quoi le formulaire risque de ne pas fonctionner correctement ou alors demander l'installation d'un contrôle ActiveX qu'on cherche justement à ne pas installer dans ce tutoriel.

Tout comme le bouton précédant, le bouton "Submit" ne doit pas être visible à l'impression. On met donc sa présence en "Visible (Screen Only)" dans ses propriétés.

7.6.14.5 Partie inférieure

On passe maintenant à la partie inférieure du formulaire:

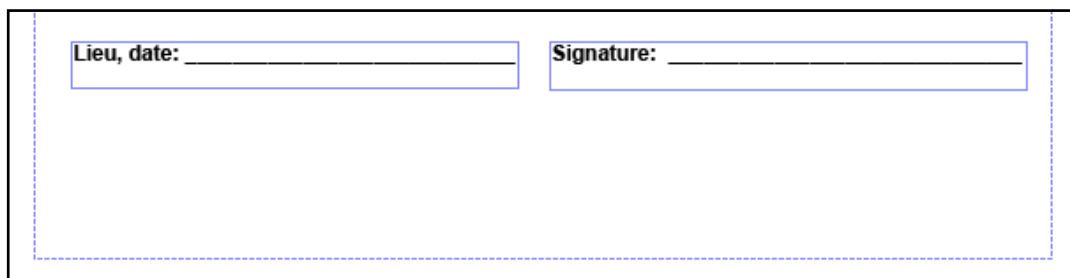



Figure 314 - Tutoriel 3: formulaire, partie inférieure

Ici c'est très simple, nous ajoutons 2 champs texte ( Text). Ils correspondent à ce qu'un lecteur voit s'il imprime et remplit son formulaire manuellement. Ces champs ont aucune utilité lors d'une utilisation par ordinateur, par conséquent nous les rendons visible qu'à l'impression en modifiant leur propriété "Présence" et en choisissant "Visible (Print Only)".

7.6.14.6 Ordre de tabulation

Un dernier élément que nous pouvons configurer est l'ordre de tabulation, c'est-à-dire dans quel ordre un champ de saisi est sélectionné. Il y a des chances que l'ordre ne soit pas logique dès le départ, par conséquent nous pouvons le corriger en allant dans View→Tab Order. Il suffit alors de cliquer sur les différents champs, dans l'ordre désiré:

CONTROLE DES ABSENCES
DEPARTEMENT DES FINANCES, DES INSTITUTIONS ET DE LA SECURITE
SERVICE CANTONAL DE L'INFORMATIQUE

4 Nom

5 Matricule

6 Prénom

7 Date de naissance

8 Code motif

9 Motifs

1100 = Vacances	1200 = Maladie inférieure à 3 jours	1400 = Service militaire
1105 = Vacances 1/2 jour	1210 = Maladie supérieure à 3 jours	1410 = Ecole de recrue
1120 = Congé naissance	1220 = Visite médicale	1420 = Protection civile
1130 = Congé non payé	1230 = VM hors canton	1430 = Cours J+5
1140 = Congé spécial	1240 = Traitements récurrents	1500 = Compensation HS
1150 = Congé maternité	1250 = Urgences médicales	1510 = Compensation HR
1155 = Congé d'adoption	1300 = Accident professionnel	1600 = Affaires publiques
1160 = Congé flexité	1310 = Accident non-professionnel	1900 = Divers
	1320 = Visite médicale après accident	

10 Date de début

11 Heure de début

12 Date de fin

13 Heure de fin

14 Durée d'absence

15 Calculer

Figure 315 - Tutoriel 3: formulaire, ordre de tabulation

Une fois l'ordre défini, il faut encore retourner dans View → Tab Order afin de désactiver l'éditeur de l'ordre.

Le formulaire est à présent terminé.

7.6.15 Déploiement, test, résultat

Il ne reste plus qu'à retourner dans la structure du projet et effectuer les mêmes étapes de déploiement que nous avons vu lors du tutoriel précédent. A savoir:

- Créer une archive
- Déployer l'archive
- Accéder au Web Dynpro Content Administrator (via <http://mshsap15.hevs.ch>)
- Créer les connexions JCO
- Revenir dans SAP NetWeaver Developer Studio
- Créer une application
- Lancer l'application

Si tout s'est bien passé, l'application s'affiche dans Internet Explorer:

Figure 316 - Tutoriel 3: test de l'application

On peut alors tester l'application en saisissant les données suivantes:

- **Matricule:** 11199130 (Cliquer sur "Rechercher" et le formulaire est pré-rempli avec les informations de l'employé)
- **Catégorie d'absence:** Catégorie A
- **Type d'absence:** 0230: Doctor's Appointment

Le formulaire est alors pré-rempli avec les informations disponibles. On constate par ailleurs qu'il n'affiche pas les champs pour lesquels la visibilité n'est active que lors de

l'impression:

The screenshot shows a web browser window titled 'Rapport_Tuto3_Application - Microsoft Internet Explorer'. The address bar shows a URL starting with 'http://mshsap15:50000/webdynpro/dispatcher/local/Rapport_Tuto3/Rapport_Tuto3_Application?SAPtestId=27'. The form is titled 'Rapport de formulaire' and contains the following fields and controls:

- N° Matricule:** 11199130 (with a 'Rechercher' button)
- Catégorie d'absence:** Radio buttons for 'Toutes', 'Catégorie A' (selected), 'Catégorie B', and 'Catégorie C'.
- Type d'absence:** Dropdown menu showing '0230: Doctor's appointment'.
- Formulaire toolbar:** Includes icons for print, save, back, forward, zoom (73.2%), and search.
- Header:** 'Sion, le 09.05.2008' and the logo of 'CANTON DU VALAIS KANTON WALLIS'.
- Main Title:** 'CONTROLE DES ABSENCES'.
- Subtitle:** 'DEPARTEMENT DES FINANCES, DES INSTITUTIONS ET DE LA SECURITE SERVICE CANTONAL DE L'INFORMATIQUE'.
- Form Fields:**
 - Nom:** Jacquin
 - Prénom:** David
 - Date de naissance:** 02.02.1982
 - Matricule:** 11199130
 - Code motif:** 0 2 3 0 (each digit in its own box)
 - Date de début:** 09.05.2008
 - Heure de début:** (empty box)
 - Date de fin:** 09.05.2008
 - Heure de fin:** (empty box)
 - Heures d'absence:** (empty box) and a 'Calculer' button.

Figure 317 - Tutoriel 3: test de l'application, pré-remplissage

On saisit ensuite les informations suivantes pour l'absence:

- **Heure de début:** 09:30
- **Heure de fin:** 11:30
- **Heures d'absences:** on clique sur le bouton, qui saisit lui-même 2

7.6.16 Vérification

On clique ensuite sur le bouton d'Envoi pour envoyer les données vers SAP. Pour vérifier, on lance SAP logon et on se connecte à la D07, mandant 100:

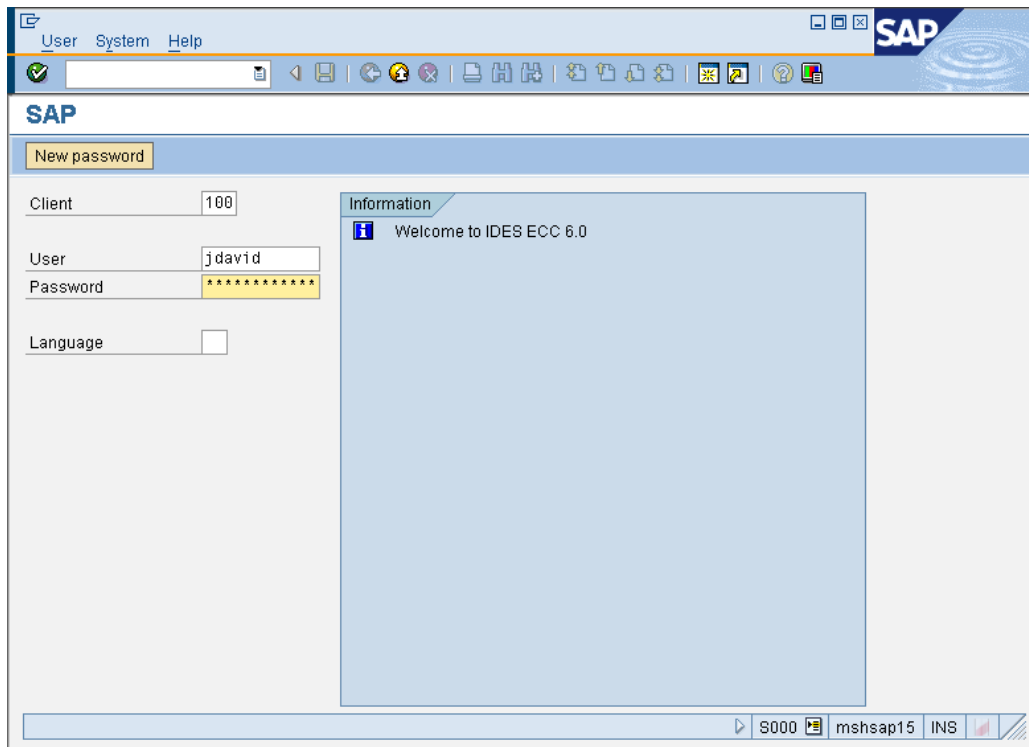


Figure 318 - Tutoriel 3: test de l'application, connexion à SAP

On utilise ensuite le code transaction PA61 pour afficher la gestion du temps de travail.
On arrive sur la fenêtre suivante:

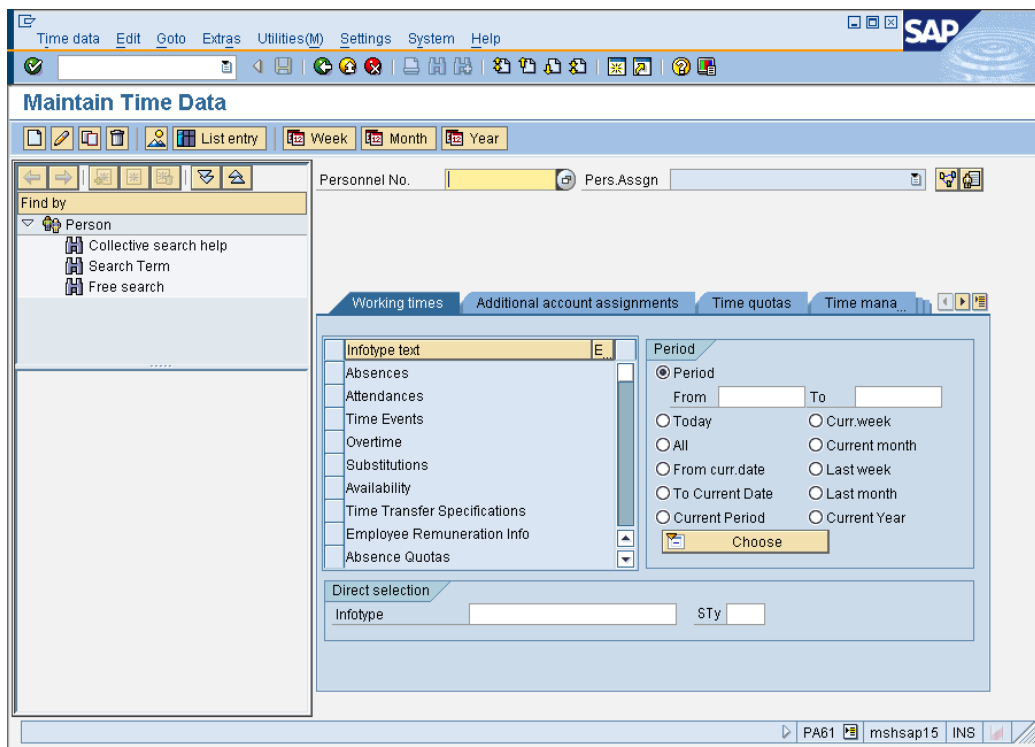



Figure 319 - Tutoriel 3: test de l'application, fenêtre de gestion du temps de travail dans SAP

On saisit le numéro de l'employé (11199130) et on appuie sur enter. Un résumé de ses informations s'affiche alors:

The screenshot shows the SAP 'Maintain Time Data' window. The 'Personnel No.' field is highlighted with a red box, containing the value '11199130'. Below it, the 'Name' field contains 'David Jacquin'. The 'EE group' is '1' and 'Active'. The 'Pers.area' is 'CABB' and 'Caliber A Bicycle Compa'. The 'EE subgroup' is 'X0' and 'Salaried employee'. The 'Cost Center' is '4711' and 'Administration'. The interface also shows tabs for 'Working times', 'Additional account assignments', 'Time quotas', and 'Time mana...'. The 'Infotype text' table is visible, with 'Absences' selected. The 'Period' section shows 'From' and 'To' fields, and a 'Choose' button.

Figure 320 - Tutoriel 3: test de l'application, saisie du matricule de l'employé dans SAP

On clique ensuite sur "Absences" dans la table "Infotype text" et ensuite sur le bouton "List Entry" ( List entry). On constate alors que son absence à bien été enregistrée:

Maintain Absences

Choose [Icons] New page

Find by: Person

- Collective search help
- Search Term
- Free search

Pers. No. 11199130 Pers. Assgn 11199130

Pers. No. 11199130 Name David Jacquin

Pers. area CABB Caliber A Bicycle Compa Cost Ctr 4711 Administration

EE subgrp X0 Salaried employee WS rule NORM

From	To	Abs.	Att./abs. ty.	Start	End	Hours	Days	Used
27.05.2008	27.05.2008	0230	Doctor's app	08:20	09:20	1.00	0.13	1.00
09.05.2008	09.05.2008	0230	Doctor's app	09:30	11:30	2.00	0.27	2.00
16.04.2008	16.04.2008	0230	Doctor's app	10:00	11:30	1.50	0.20	1.50

Choose 01.01.1800 - 31.12.9999 Entry 1 of 3

PA61 mshsap15 INS

Figure 321 - Tutoriel 3: test de l'application: affichage des absences de l'employé dans SAP

Ce troisième tutoriel est à présent terminé.

7.6.17 Commentaires

Un élément qui me dérange particulièrement avec ce formulaire est que les boutons radio et la liste déroulante ne se trouvent pas dans le formulaire. A cause d'un bug étrange abordé plus loin dans le rapport, j'ai été obligé de sortir ces éléments du formulaire si je voulais pouvoir rester dans les temps.

En revanche, on voit ici que l'option qui permet de configurer la visibilité d'un élément du formulaire en fonction du mode d'affichage (écran ou papier) est certes simple, mais extrêmement efficace. On voit clairement qu'il n'y a pas la nécessité de maintenir 2 versions de ce formulaire en même temps: il suffit de réaliser correctement le design d'un seul formulaire.

7.7 Tutoriel 4: Formulaire hors-ligne (Offline Form)

7.7.1 Objectif

Nous allons à présent aborder la dernière partie relative à l'implémentation d'un formulaire: celle d'un formulaire hors-ligne. Cette partie a 2 buts: permettre à l'utilisateur de remplir un formulaire quand il le souhaite et améliorer le formulaire précédent en le rendant un peu plus indépendant: en effet, nous allons ajouter une liste déroulante dans le formulaire. Celle-ci permettra de sélectionner un type d'absence.

7.7.2 Description

Une fois l'application lancée, l'utilisateur pourra saisir un matricule, ce qui permettra de préparer le formulaire d'absence. En cliquant sur un lien, l'utilisateur pourra télécharger ce formulaire et le compléter quand il le souhaite, avec ou sans connexion Internet. Une fois qu'il est rempli, l'utilisateur retourne alors sur l'application et utilise la fonction d'upload de cette dernière. Une fois le formulaire envoyé, une page affiche un résumé des valeurs saisies et demande à l'utilisateur s'il souhaite revenir en arrière ou valider les données.

Voici à quoi ressemblera l'application à la fin:

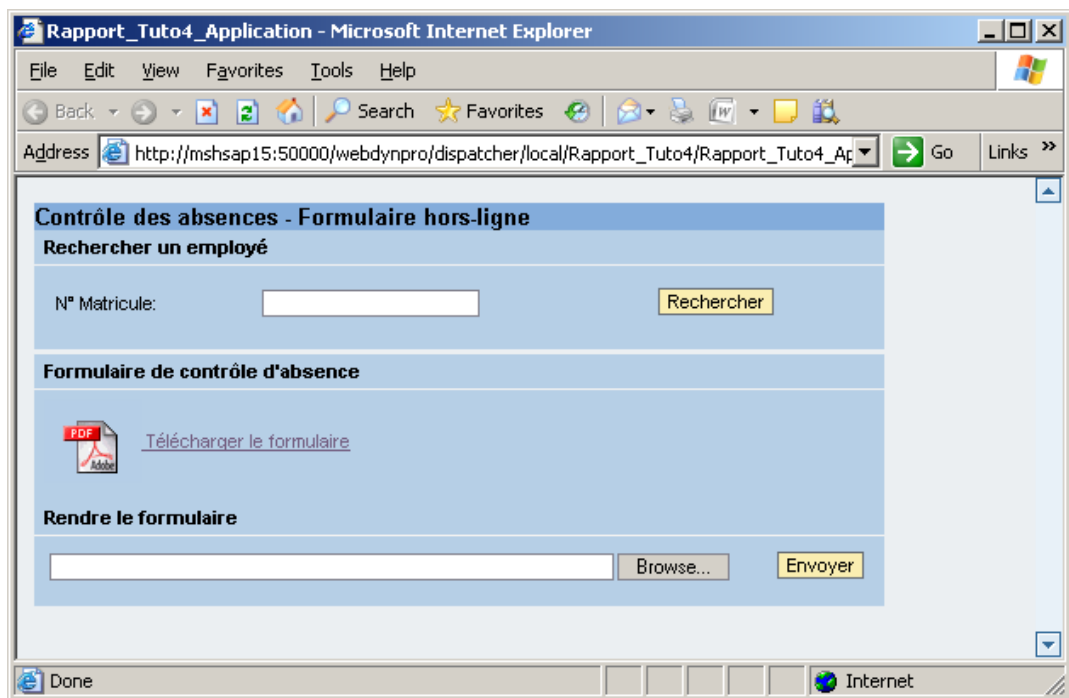


Figure 322 - Tutoriel 4: aperçu du résultat

7.7.3 Pré-requis

Avant de se lancer dans le développement, il faut d'abord créer le formulaire qui sera utilisé. Pour ce faire, on fait appel à une vue temporaire qui contiendra juste un

formulaire, un type simple permettant une énumération, ainsi que le contexte nécessaire.

7.7.3.1 Création d'un type simple

Pour le type simple, il suffit juste d'en créer un. Pas besoin de le paramétrer. En voici un exemple dans notre structure du projet:

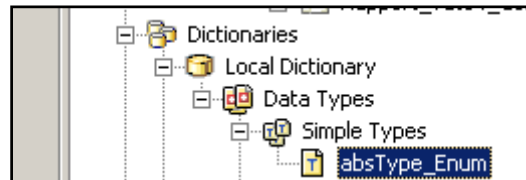


Figure 323 - Tutoriel 4: création d'un type simple

7.7.3.2 Création du Custom Controller et de son contexte

On ajoute ensuite un Custom Controller pour lequel on crée le contexte suivant et les types indiqués:

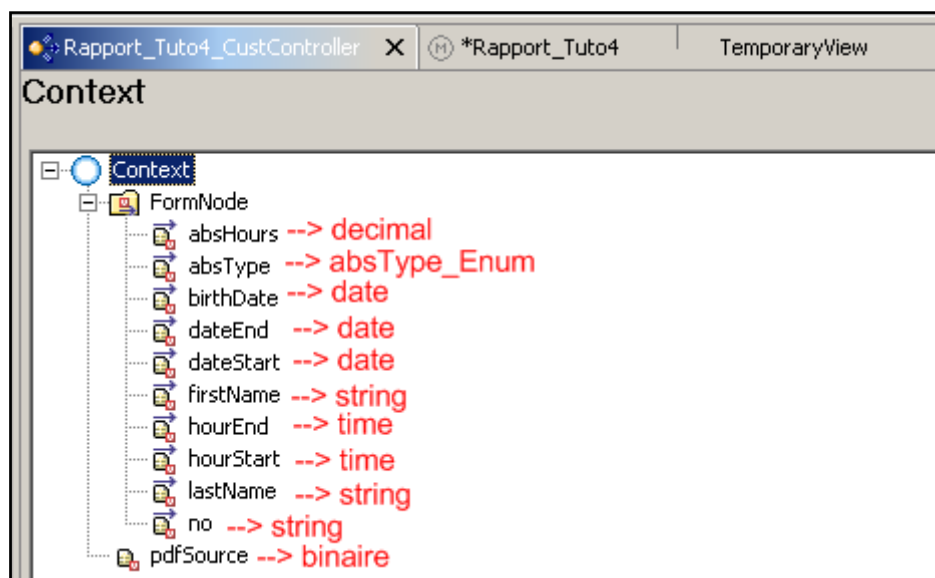


Figure 324 - Tutoriel 4: nœud FormNode (avec types) dans le contexte du Custom Controller

7.7.3.3 Implémentation de wdDoInit() (Custom Controller)

Toujours pour le Custom Controller, on accède à sa méthode wdDoInit() et on fait en sorte que le nœud "FormNode" ne soit pas vide en ajoutant le code suivant:

```
//Création d'un élément dans le nœud pour qu'il soit non-vide
IPublicRapport_Tuto4_CustController.IFormNodeElement
formNodeInput = wdContext.createFormNodeElement();

wdContext.nodeFormNode().bind(formNodeInput);
```

7.7.3.4 Mapping vue temporaire → Custom Controller

On accède ensuite au "Data Modeler" et on effectue un mapping de "FormNode" entre la vue temporaire et le contrôleur. On ne mappe cependant pas l'attribut "pdfSource": on en crée un indépendant (toujours de type binaire). On obtient alors:

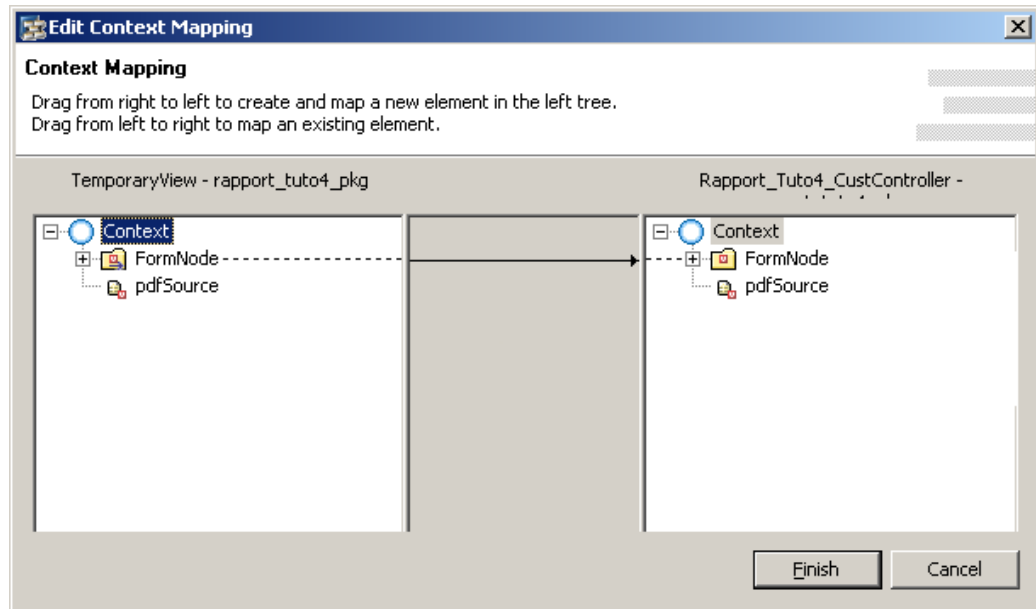


Figure 325 - Tutoriel 4: mapping vue temporaire → Custom Controller

7.7.3.5 Création du formulaire

Ensuite, on se rend dans la vue temporaire. On ajoute juste un formulaire interactif utilisant "FormNode" comme source de données ("dataSource"), l'attribut "pdfSource" pour la propriété "pdfSource" et "ActiveX" pour "displayType" (on n'utilise pas le mode natif pour ce tutoriel car les listes déroulantes qu'il utilise n'ont pas le comportement désiré).

Une fois le formulaire paramétré, on passe à son design. Attention, il doit être entièrement réalisé dès maintenant. En effet, nous ne reviendrons pas dessus plus tard.

Visuellement parlant, il est identique au formulaire d'absence du tutoriel 3 avec cependant les exceptions suivantes:

- Le bouton d'envoi ("Submit") est retiré
- Tout ce qui est visible à l'impression (liste des types d'absence, signature,...) est retiré
- Le champ contenant le code du type d'absence est créé différemment:
 - o Dans la partie "Web Dynpro ActiveX", on utilise une "Enumerated Drop-Down list (no select)":

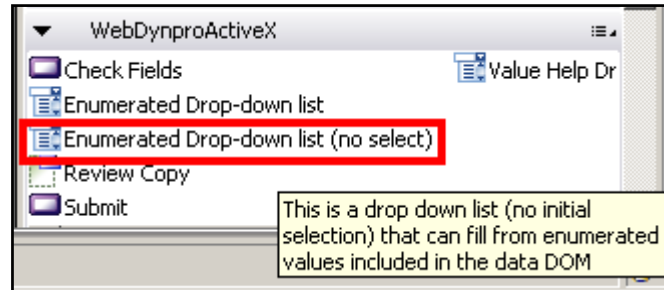


Figure 326 - Tutoriel 4: enumerated drop-down list (no select)(ActiveX)

- o On glisse ensuite cet élément dans le formulaire, là où on retrouve le code du type d'absence
- o Enfin, on effectue un binding en effectuant un drag & drop du type d'absence vers la liste déroulante:

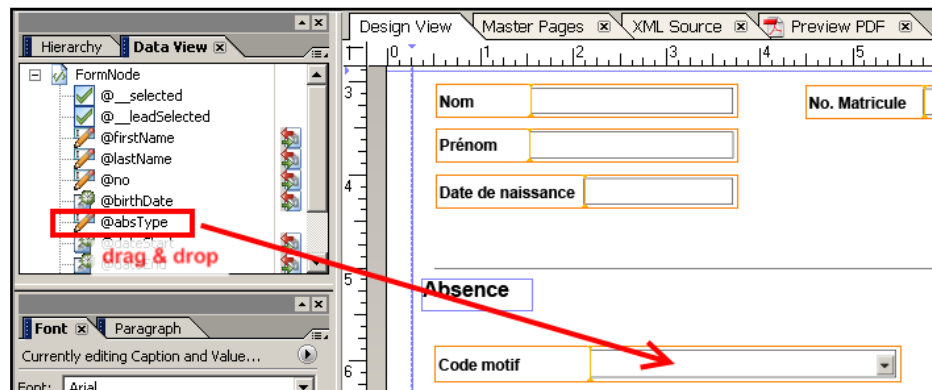


Figure 327 - Tutoriel 4: drag & drop du type d'absence

- La fenêtre suivante apparaît alors. On clique sur le dernier bouton radio et on valide:

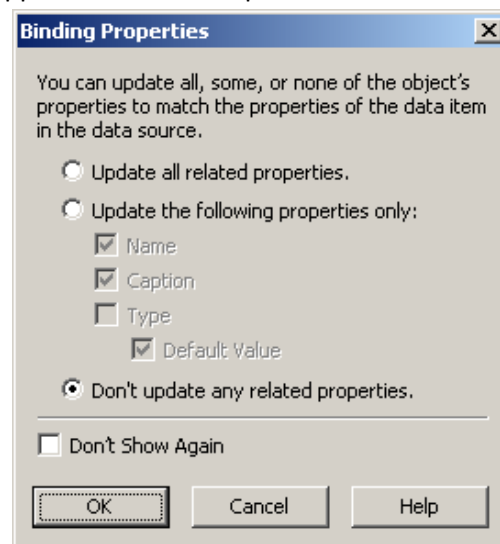


Figure 328 - Tutoriel 4: options de binding

Pour les calculs et propriétés des autres champs, on reprend simplement le tutoriel précédent (attention au nom des différents champs de saisie qu'il faudra renommer soit dans leurs propriétés soit dans les calculs).

7.7.3.6 Exemple de formulaire


Une fois le design terminé, on obtient un formulaire similaire à celui-ci:

The screenshot shows a web form prototype for 'Hes·SO VALAIS WALLIS'. The header includes the university's name in French and German, and the date 'Sierre, le 13.05.2008'. The main title is 'CONTROLE DES ABSENCES FORMULAIRE PROTOTYPE'. The form contains several input fields: 'Nom', 'Prénom', 'Date de naissance', and 'No. Matricule'. Below these is a section titled 'Absence' which includes a 'Code motif' dropdown menu, and two rows of date and time inputs: 'Date de début' and 'Date de fin' (both pre-filled with '13.05.2008'), and 'Heure de début' and 'Heure de fin'. At the bottom, there is a 'Heures d'absence' input field and a 'Calculer' button.

Figure 329 - Tutoriel 4: aperçu du formulaire

7.7.3.7 Template du formulaire

L'étape suivante consiste à récupérer le fichier template du formulaire. Il s'agit d'un fichier .xdp que l'on retrouve dans l'un des répertoires du projet et qui permet de générer le formulaire que l'on vient de créer.

Avant toute chose, on ferme l'Adobe LiveCycle Designer et on sauvegarde le projet (). Ensuite, à l'aide de l'explorateur Windows, on se rend dans le répertoire du projet. Dans mon cas, le chemin est:

C:\Documents and Settings\StdF\Documents\SAP\workspace\Rapport_Tuto4

Si votre espace de travail pour SAP NetWeaver Developer Studio n'est pas au même endroit, le plus simple reste d'effectuer une recherche automatique du répertoire portant le nom du projet.

Une fois dans le répertoire du projet, on se rend dans:

src\configuration\Components\rapport_tuto4_pkg.Rapport_Tuto4_Component

Le dernier répertoire du chemin correspond à la concaténation des noms du package et du component de l'application. On y trouve alors le fichier .xdp:

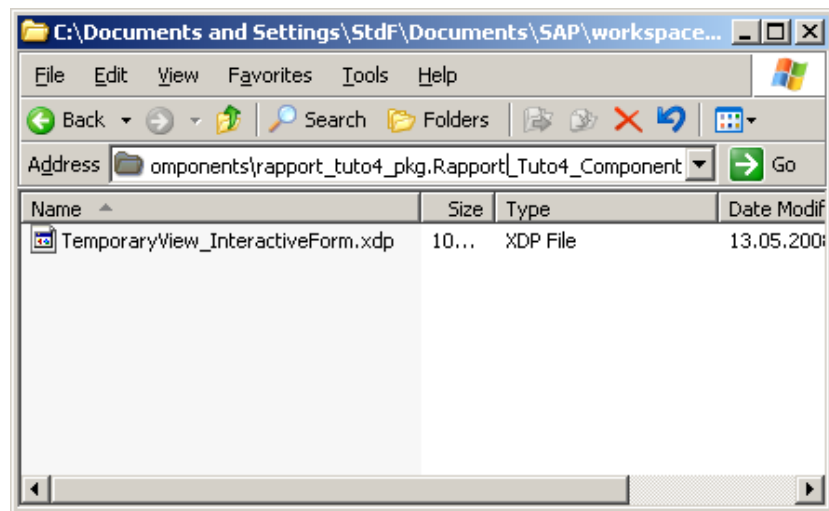


Figure 330 - Tutoriel 4: fichier .xdp dans l'explorateur Windows (dossier source)

On copie ensuite ce fichier dans le répertoire:

\src\mimes\Components\rapport_tuto4_pkg.Rapport_Tuto4_Component

Ce répertoire permet par la suite d'accéder aux différents fichiers qui interviennent dans l'application. Nous allons d'ailleurs y ajouter un fichier pdf.gif qui correspond à une image de l'icône d'un fichier pdf et que l'on va rajouter juste à côté du lien de téléchargement dans l'application.

Le répertoire se présente donc ainsi:

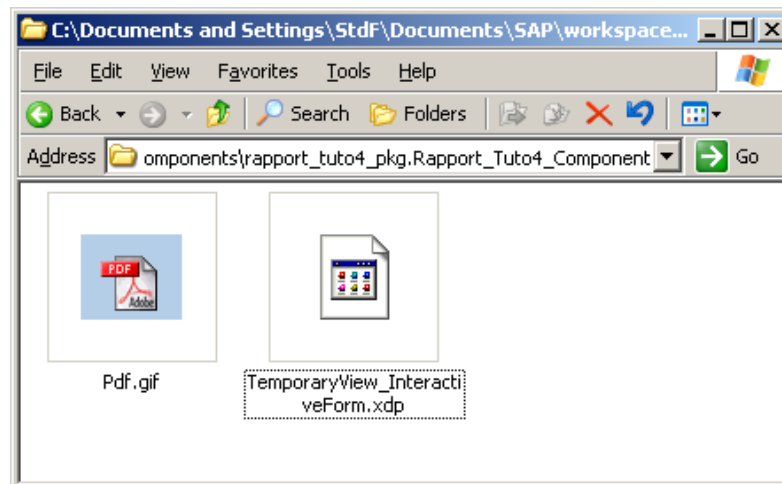


Figure 331 - Tutoriel 4: contenu du répertoire de destination

On retourne ensuite dans SAP NetWeaver Developer Studio et on ferme puis rouvre le projet afin que celui-ci prennent en compte les nouveaux fichiers. Pour vérifier, il suffit d'aller à l'emplacement indiqué ci-dessous:

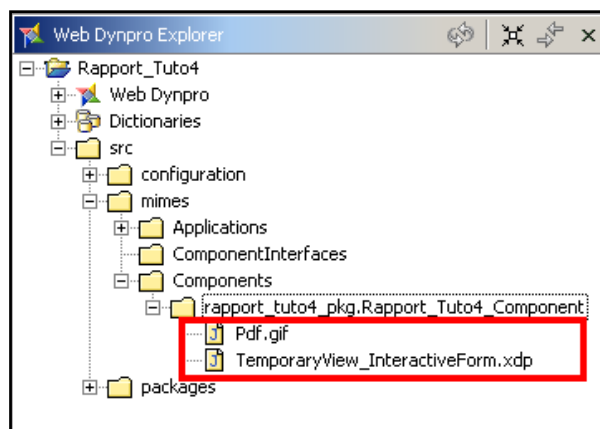


Figure 332 - Tutoriel 4: fichiers disponibles dans Web Dynpro Explorer

Maintenant que le template du formulaire et l'icône sont disponibles, nous pouvons passer au développement de l'application.

7.7.4 Importation des BAPIs

La première chose à faire consiste à créer un modèle qui importe les mêmes BAPIs/RFC que lors du tutoriel précédant, c'est-à-dire:

- BAPI_PERSDATA_GETDETAILEDINFO
- BAPI_ABSENCE_CREATE
- BAPI_EMPLOYEE_ENQUEUE
- BAPI_EMPLOYEE_DEQUEUE
- BAPI_TRANSACTION_COMMIT
- RFC_ABSENCETYPES_READ

Le modèle se présente donc ainsi:

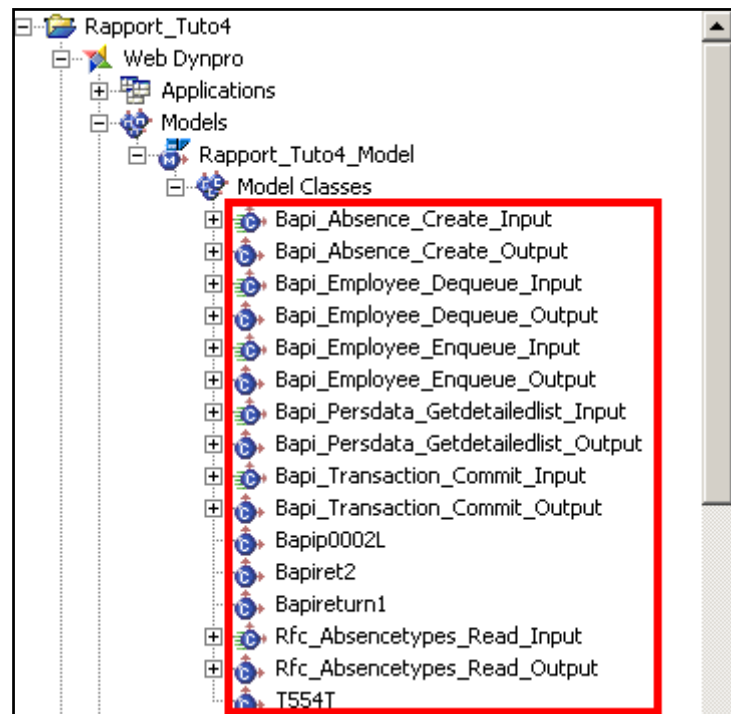


Figure 333 - Tutoriel 4: modèle importé

7.7.5 Usage du modèle

On déclare ensuite l'utilisation de ce modèle via "Used Models" → "Add":

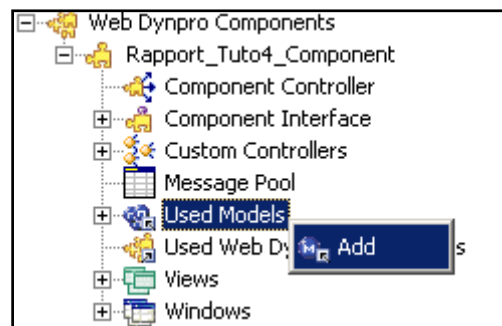


Figure 334 - Tutoriel 4: utilisation d'un modèle

7.7.6 Mapping Custom Controller → Modèle

On ouvre le "Data Modeler" et on crée un mapping entre le Custom Controller et le modèle. Toutes les BAPIs/RFC dont le nom se termine par "_Input" sont mappées dans leur entièreté. Une fois le mapping effectué, on obtient:

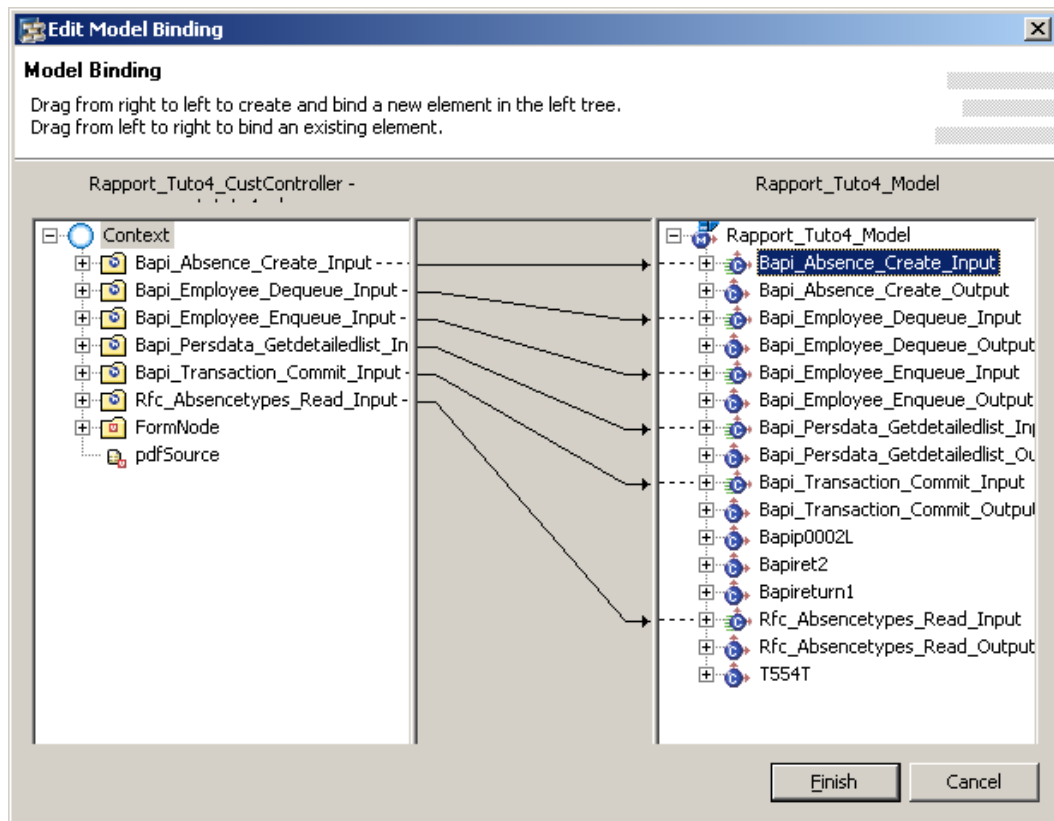


Figure 335 - Tutoriel 4: mapping Custom Controller → modèle

7.7.7 Création et implémentation des méthodes du Custom Controller

Toujours au niveau du Custom Controller, on crée 2 méthodes sans paramètre:

- **execute_GetPersData():** qui aura pour tâche d'exécuter la BAPI qui retourne les informations d'un employé et de les assigner aux différents attributs du nœud "FormNode" destiné au formulaire.
- **execute_SaveAbs():** qui récupère les attributs du nœud du formulaire une fois rempli et renvoyé et qui exécute ensuite les BAPIs relatives à la création d'une absence.

Nous allons les implémenter immédiatement. Nous allons aussi en profiter pour nous compléter la méthode wdDoInit() qui devra initialiser les différents nœuds pour qu'ils soient non vides. Elle s'occupera aussi de retrouver tous les types d'absence, ceci dans le but de configurer l'attribut qui sera lié à la liste déroulante.

7.7.7.1 wdDoInit() du Custom Controller

```
public void wdDoInit()
{
    //@@begin wdDoInit()
    //@=====
    //Création d'un élément dans le noeud pour qu'il soit non-vide
    IPublicRapport_Tuto4_CustController.IFormNodeElement
    formNodeInput = wdContext.createFormNodeElement();
}
```

```
wdContext.nodeFormNode().bind(formNodeInput);

//=====
//Exécution de la BAPI qui retourne les types d'absence
Rfc_Absencetypes_Read_Input inputAbsTypes = new
    Rfc_Absencetypes_Read_Input();

wdContext.nodeRfc_Absencetypes_Read_Input().bind(inputAbsTypes);

try {
    wdContext.currentRfc_Absencetypes_Read_InputElement()
        .modelObject().execute();
}
catch (Exception e){
    e.printStackTrace();
}

//Récupération des types d'absences pour les mettre dans le
simple //type créé
int size = wdContext.nodeAbsencetext_Output().size();

//on définit une variable qui permettra de manipuler l'attribut
//utilisant un type personnalisé.
IWDAttributeInfo customAbsTypeInfo = wdContext.nodeFormNode()
    .getNodeInfo().getAttribute("absType");

ISimpleTypeModifiable customAbsTypeText =
    customAbsTypeInfo.getModifiableSimpleType();

IModifiableSimpleValueSet customAbsTypeValueSet =
    customAbsTypeText.getSVServices().getModifiableSimpleValueS
et();

//on définit 2 variables de type string:
// > type: code du motif d'absence
// > text: descriptif du motif d'absence
String type, text;

//on lance une boucle qui parcourt les types d'absence
//retournés par la RFC, en tenant compte de l'intervalle
//qui correspond à la catégorie.
for (int i = 0; i < size; i++) {
    wdContext.nodeAbsencetext_Output().setLeadSelection(i);
    text = wdContext.currentAbsencetext_OutputElement().getAtext();
    type = wdContext.currentAbsencetext_OutputElement().getAwart();

    //on ajoute dans l'attribut absType la combinaison type +
    texte
    customAbsTypeValueSet.put(type, type + ": " + text);
}

//=====
//Création d'un noeud vide pour toutes les autres BAPIS
Bapi_Absence_Create_Input inputAbsCreate = new
    Bapi_Absence_Create_Input();
wdContext.nodeBapi_Absence_Create_Input().bind(inputAbsCreate);
```

```
Bapi_Persdata_Getdetailedlist_Input inputPersData = new
    Bapi_Persdata_Getdetailedlist_Input();
wdContext.nodeBapi_Persdata_Getdetailedlist_Input()
    .bind(inputPersData);

Bapi_Employee_Enqueue_Input inputEnqueue = new
    Bapi_Employee_Enqueue_Input();
wdContext.nodeBapi_Employee_Enqueue_Input().bind(inputEnqueue);

Bapi_Employee_Dequeue_Input inputDequeue = new
    Bapi_Employee_Dequeue_Input();
wdContext.nodeBapi_Employee_Dequeue_Input().bind(inputDequeue);

Bapi_Transaction_Commit_Input inputCommit = new
    Bapi_Transaction_Commit_Input();
wdContext.nodeBapi_Transaction_Commit_Input().bind(inputCommit);
//@@end
}
```

7.7.8 execute_GetPersData() du Custom Controller

```
public void execute_GetPersData( )
{
    //@@begin execute_GetPersData()
    try {
        wdContext.currentBapi_Persdata_Getdetailedlist_InputElement
        ()
            .setEmployeeNumber(wdContext.currentFormNodeElement()
            .getNo());

        wdContext.currentBapi_Persdata_Getdetailedlist_InputElement
        ()
            .modelObject().execute();

        wdContext.nodeOutput_Persdata().invalidate();

        //on assigne les valeurs retournées au noeud du formulaire
        wdContext.currentFormNodeElement()
            .setNo(wdContext.currentPersonaldata_OutputElement()
            .getEmployeeNo());

        wdContext.currentFormNodeElement()
            .setFirstName(wdContext
            .currentPersonaldata_OutputElement()
            .getFirstname());

        wdContext.currentFormNodeElement()
            .setLastName(wdContext
            .currentPersonaldata_OutputElement()
            .getLastname());

        wdContext.currentFormNodeElement()
            .setBirthDate(wdContext
            .currentPersonaldata_OutputElement()
```

```

        .getDateOfBirth();
    } catch (Exception e) {
        e.printStackTrace();
    }
    //@@end
}

```

7.7.8.1 execute_SaveAbs(): du Custom Controller

```

public void execute_SaveAbs( )
{
    //@@begin execute_SaveAbs()
    //Récupération de l'ID de l'employé pour qu'il soit utilisé par
    //les BAPIs de création d'absence, de verrouillage et de
    //déverrouillage.
    String employeeId = wdContext.currentFormNodeElement().getNo();
    wdContext.currentBapi_Employee_Enqueue_InputElement()
        .setNumber(employeeId);

    wdContext.currentBapi_Employee_Dequeue_InputElement()
        .setNumber(employeeId);

    wdContext.currentBapi_Absence_Create_InputElement()
        .setEmployeeNumber(employeeId);

    wdContext.currentBapi_Absence_Create_InputElement()
        .setAbsencetype(wdContext.currentFormNodeElement()
            .getAbsType());

    wdContext.currentBapi_Absence_Create_InputElement()
        .setAbsencehours(wdContext.currentFormNodeElement()
            .getAbsHours());

    wdContext.currentBapi_Absence_Create_InputElement()
        .setEnd(wdContext.currentFormNodeElement()
            .getHourEnd());

    wdContext.currentBapi_Absence_Create_InputElement()
        .setStart(wdContext.currentFormNodeElement()
            .getHourStart());

    wdContext.currentBapi_Absence_Create_InputElement()
        .setValiditybegin(wdContext.currentFormNodeElement()
            .getDateStart());

    wdContext.currentBapi_Absence_Create_InputElement()
        .setValidityend(wdContext.currentFormNodeElement()
            .getDateEnd());

    /*enregistrement de l'absence: verrouillage, validation,
    création, validation
    déverrouillage, validation*/

    try {
        wdContext.currentBapi_Employee_Enqueue_InputElement()
            .modelObject().execute();
    }
}

```

```
wdContext.currentBapi_Transaction_Commit_InputElement()  
    .modelObject().execute();  
  
wdContext.currentBapi_Absence_Create_InputElement()  
    .modelObject().execute();  
  
wdContext.currentBapi_Transaction_Commit_InputElement()  
    .modelObject().execute();  
  
wdContext.nodeOutput_AbsCreate().invalidate();  
  
wdContext.currentBapi_Employee_Dequeue_InputElement()  
    .modelObject().execute();  
  
wdContext.currentBapi_Transaction_Commit_InputElement()  
    .modelObject().execute();  
  
} catch (Exception e) {  
    e.printStackTrace();  
}  
//@@end  
}
```

7.7.9 Création des vues

On passe ensuite à la création des vues. Il nous en faut 3:

- **DownloadView:** qui permet de chercher les informations d'un employé et de télécharger le formulaire
- **UploadView:** qui permet de rendre le formulaire
- **ResultView:** qui affiche les données du le formulaire et permet de les sauvegarder dans SAP ou de revenir en arrière.

Pour ce faire, on accède au diagramme de navigation de l'application en double cliquant sur le nom du composant dans la branche "Windows":

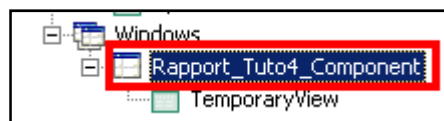


Figure 336 - Tutoriel 4: lancement du diagramme de navigation

On crée ensuite à l'aide des outils sur la gauche un diagramme similaire à celui-ci:

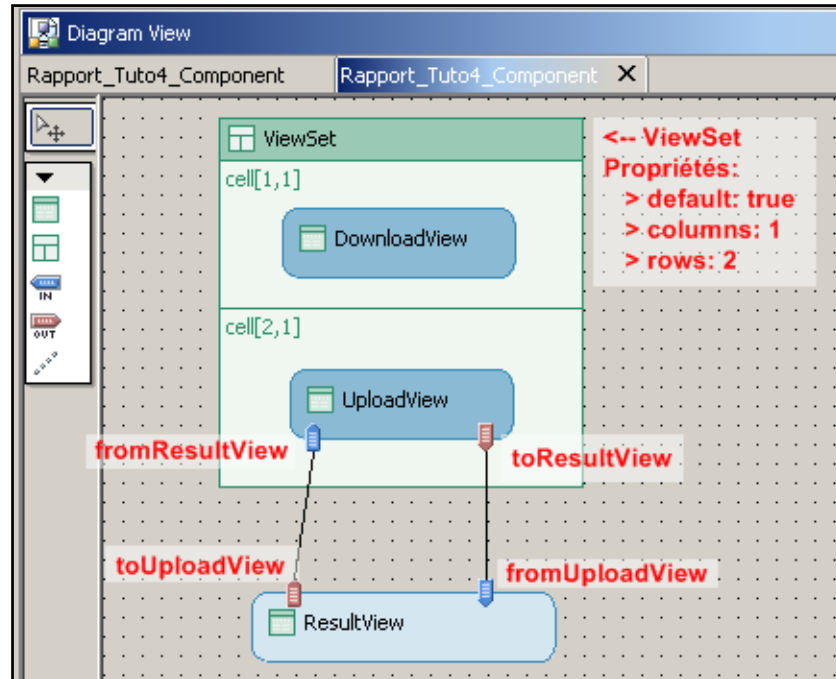



Figure 337 - Tutoriel 4: configuration de la navigation

7.7.10 Développement de la vue DownloadView

Développons à présent la vue DownloadView. Voici brièvement ce dont elle a besoin:

- **Contexte:**
 - o mapping avec le nœud FormNode du Custom Controller
 - o un attribut pdfSource de type binaire (pas celui du controller)
 - o un attribut pdfRessource de type Resource. Pour le trouver, on clique sur le bouton  dans la propriété "type" et on va dans
 - o Local Dictionary → com.sap.ide.webdynpro.ulelementdefinitions
 - o On sélectionne "Resource"
 - o Cela donne, pour le type:

Properties	
Property	Value
calculated	false
name	pdfRessource
readOnly	false
structureElement	
type	com.sap.ide.webdynpro.ulelementdefinitions.Resource

Figure 338 - Tutoriel 4: type Resource

- o Le contexte de DownloadView se présente ainsi:

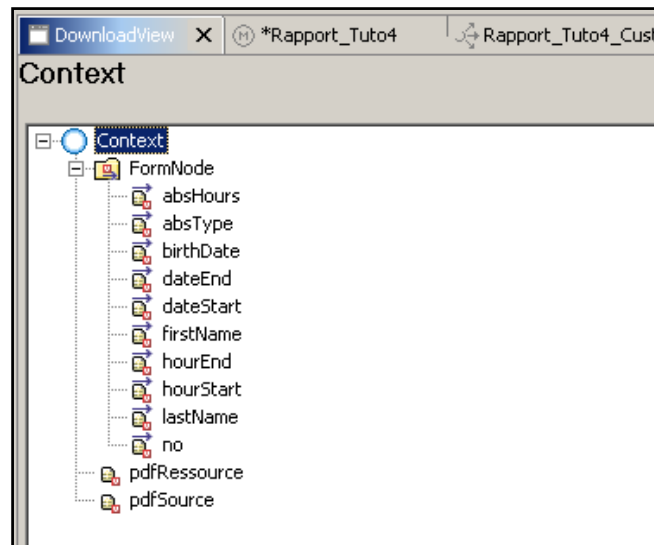


Figure 339 - Tutoriel 4: contexte de DownloadView

- **Méthodes:**

- o Une méthode setForm(), sans paramètre. Celle-ci aura pour but de créer un formulaire à partir du template que nous avons copié avant, lui assigner un nom de fichier et le lier aux données du nœud "FormNode".
- o Voici son implémentation:

```
public void setForm( )
{
    //@@begin setForm()
    try {
        //template.xdp is kept in src\mimes\Components\

```

```
.createResource(pdf.getPDF(), "Absence_OfflineForm_Fr",
WDWebResourceType.PDF);

wdContext.currentContextElement()
.setPdfResource(pdfResource);

} catch (Exception e) {
    e.printStackTrace();
}
//@@end
}
```

- **Action:**

- o Une action "Search" qui permettra de lancer la recherche d'un employé.
- o Son code:

```
public void onActionSearch(com.sap.tc.webdynpro.progmodel
    .api.IWDCustomEvent wdEvent )
{
    //@@begin onActionSearch(ServerEvent)
    IWDMessagesManager manager = wdComponentAPI.getMessageManager();

    wdThis.wdGetRapport_Tuto4_CustControllerController()
        .execute_GetPersData();

    setForm();

    manager.reportSuccess("Le formulaire a été préparé avec
succès.");

    //@@end
}
```

7.7.10.1 Design de DownloadView

La vue a besoin au minimum:

- D'un champ de saisie (InputField) lié à l'attribut correspondant au numéro de l'employé dans le nœud "FormNode"
- D'un bouton lié à l'action "Search" (propriété "onAction")
- D'un élément "FileDownload" avec au minimum les propriétés suivantes:

Properties	
Property	Value
[-] Element Properties [UIElement]	
behaviour	allowSave
data	pdfSource
enabled	true
id	FileDownload
imageAlt	
imageFirst	true
imageHeight	
imageSource	Pdf.gif
imageWidth	
resource	pdfRessource
size	standard
target	self
text	Télécharger le formulaire
textDirection	inherit
tooltip	<>
type	navigation
visible	visible
wrapping	false
[-] Layout Data [GridData]	

Figure 340 - Tutoriel 4: propriétés de l'élément FileDownload

Voici un exemple de design complet:

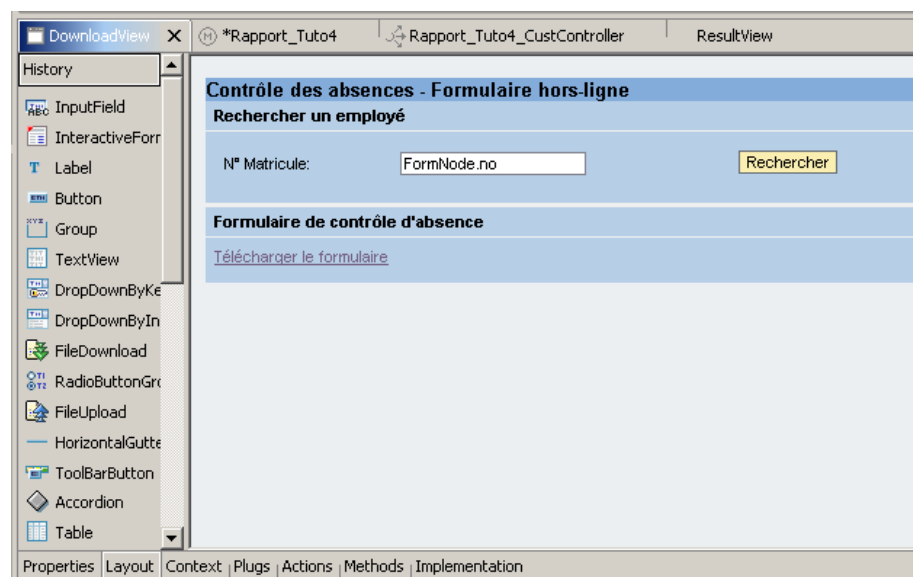


Figure 341 - Tutoriel 4: exemple de design de la vue DownloadView

On remarque que l'image pdf.gif n'apparaît pas. C'est normal, ce sera le cas lors du déploiement.

7.7.11 Développement de la vue UploadView

On passe ensuite à la vue "UploadView". Elle a besoin de:

- **Un lien vers le Custom Controller:**

- o Possibilité de le créer à partir de l'onglet Properties de la vue:

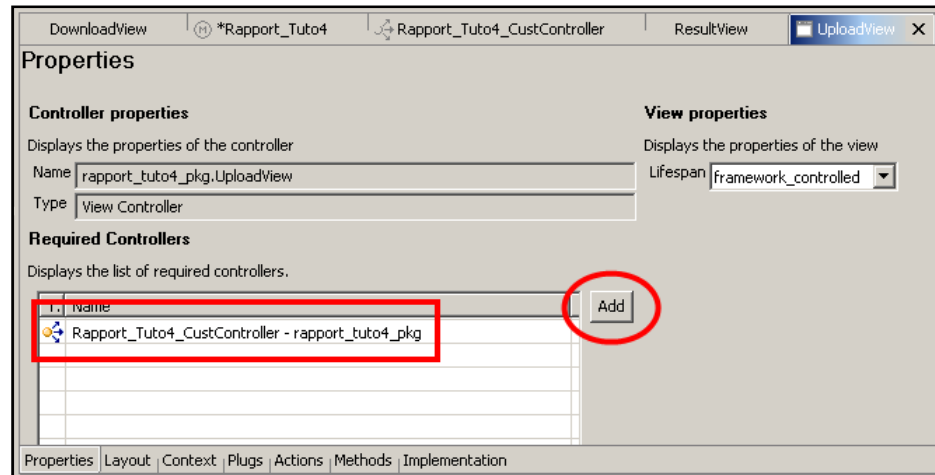


Figure 342 - Tutoriel 4: ajout du Custom Controller dans les propriétés de UploadView

- **Contexte:**
 - o Un attribut pdfSource de type binaire (pas celui du contexte du contrôleur)
- **Méthode:**
 - o On ne crée pas de méthode, par contre on ajoute une ligne de code à wdDoInit(). Celle-ci sera utilisée par le champ d'upload de l'application:

```
public void wdDoInit()
{
    //@@begin wdDoInit()
    wdContext.getNodeInfo().getAttribute("pdfSource")
        .getModifiableSimpleType();
    //@@end
}
```

- **Actions**
 - o Une action "Send" pour l'upload du formulaire et qui utilise le Fire Plug "ToResultView".
 - o Code de l'action:

```
public void onActionSend(com.sap.tc.webdynpro.progmodel
    .api.IWDCustomEvent wdEvent )
{
    //@@begin onActionSend(ServerEvent)
    wdThis.wdGetRapport_Tuto4_CustControllerController()
        .wdGetContext().currentContextElement()
        .setPdfSource(wdContext.currentContextElement()
            .getPdfSource());

    wdThis.wdFirePlugToResultView();
    //@@end
}
```

7.7.11.1 Design de UploadView

Il faut au minimum 2 choses:

- Un élément "FileUpload" avec la propriété suivante:
 - o data: pdfSource
 - o Un bouton lié à l'action "Send" dans la propriété "onAction".

Voici un exemple de design:

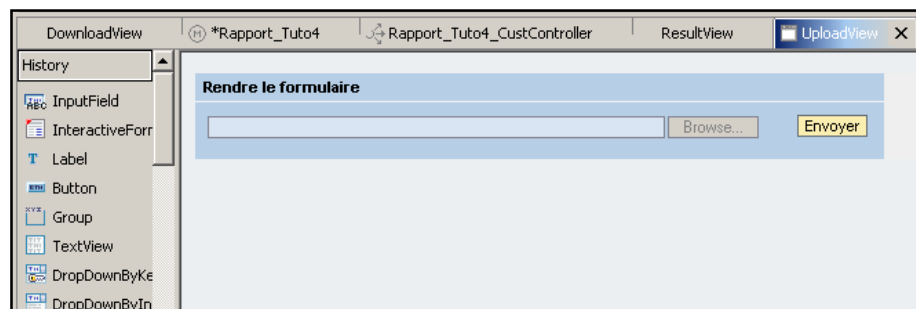


Figure 343 - Tutoriel 4: exemple de design de UploadView

7.7.12 Développement de la vue ResultView

Pour finir, on passe à la vue "ResultView", qui contient le résumé du contenu du formulaire et la possibilité de revenir à la page précédente ou d'envoyer les données vers SAP. Elle a besoin de:

- **Contexte:**
 - o Mapping du nœud du formulaire "FormNode"
 - o Mapping de l'attribut pdfSource
 - o Le contexte se présente alors ainsi:

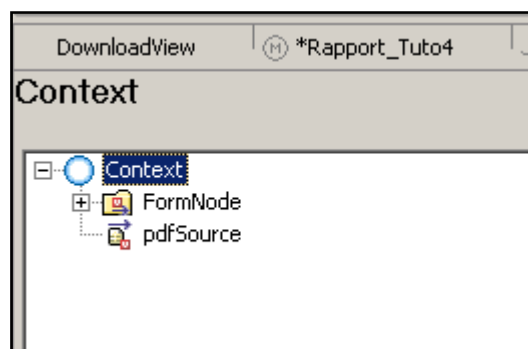


Figure 344 - Tutoriel 4: contexte de ResultView

- **Actions:**

- o Une action "Back" pour revenir à la page précédente et qui utilise le Fire plug "ToUploadView"
- Code: (automatiquement ajouté)
 - o Une action "Save" qui permet d'exécuter la méthode du contrôleur chargée de créer une absence.
 - o Code de l'action "Save":

```
public void onActionSave(com.sap.tc.webdynpro.progmodel.api
    .IWDCustomEvent wdEvent )
{
    //@@begin onActionSave(ServerEvent)
    IWDMessageManager manager = wdComponentAPI.getMessageManager();
    wdThis.wdGetRapport_Tuto4_CustControllerController()
        .execute_SaveAbs();

    manager.reportSuccess( "Données envoyées." );
    //@@end
}
```

7.7.12.1 Design de ResultView

On a besoin au minimum:

- D'un bouton pour revenir à la page précédente (onAction: Back)
- D'un bouton pour envoyer les données vers SAP (onAction: Save)
- Un label et un textview par attribut que l'on souhaite afficher dans le résumé.

Voici un exemple de design:



Figure 345 - Tutoriel 4: exemple de design de ResultView

7.7.13 Déploiement, test, résultat

L'étape suivante est celle du déploiement. Comme lors du tutoriel précédent, on réalise les choses suivantes:

- Créer une archive
- Déployer l'archive

- Accéder au Web Dynpro Content Administrator (via <http://mshsap15.hevs.ch:50000>)
- Créer les connexions JCO
- Revenir dans SAP NetWeaver Developer Studio
- Créer une application
- Lancer l'application

Une fois l'application lancée, on peut tester le résultat. Tout d'abord on saisit le numéro de l'employé dans le champ prévu à cet effet:

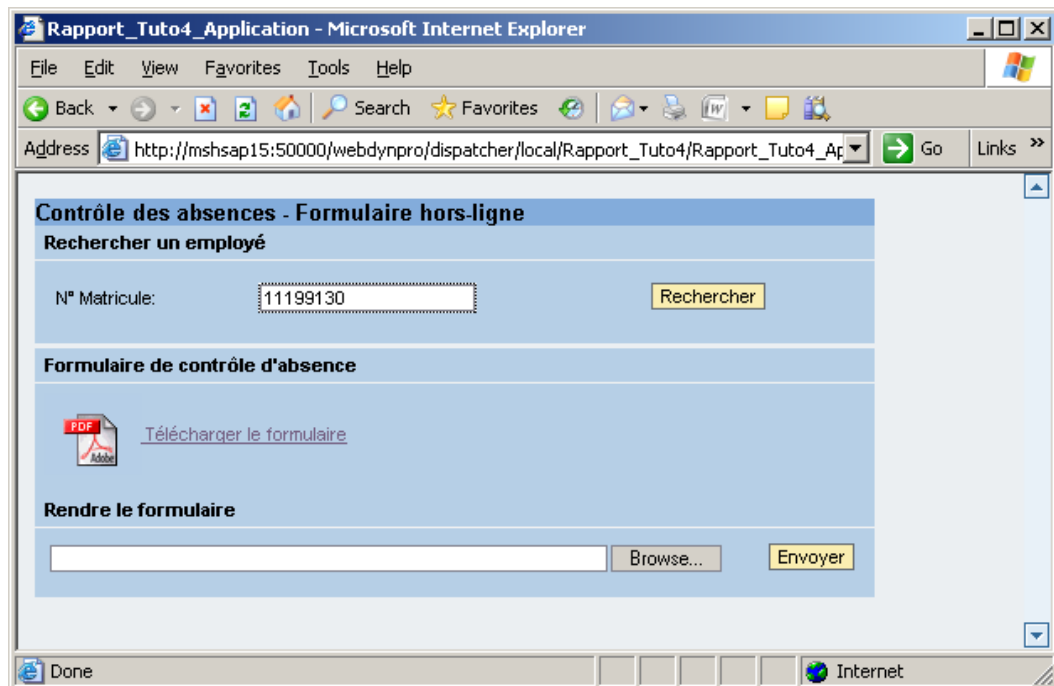


Figure 346 - Tutoriel 4: test de l'application, saisie d'un matricule

On clique ensuite sur le bouton "Rechercher". Un message apparaît en bas de l'écran quand le formulaire est prêt:

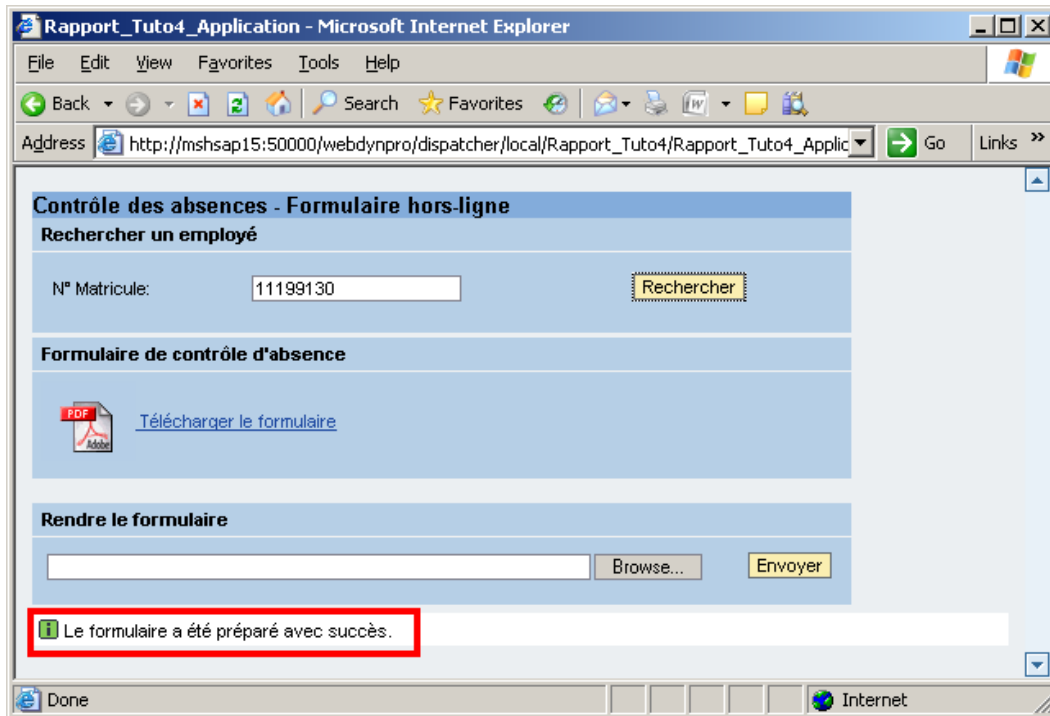


Figure 347 - Tutoriel 4: test de l'application, message

On clique alors sur le lien pour télécharger le formulaire. La fenêtre classique de téléchargement d'Internet Explorer apparaît alors:

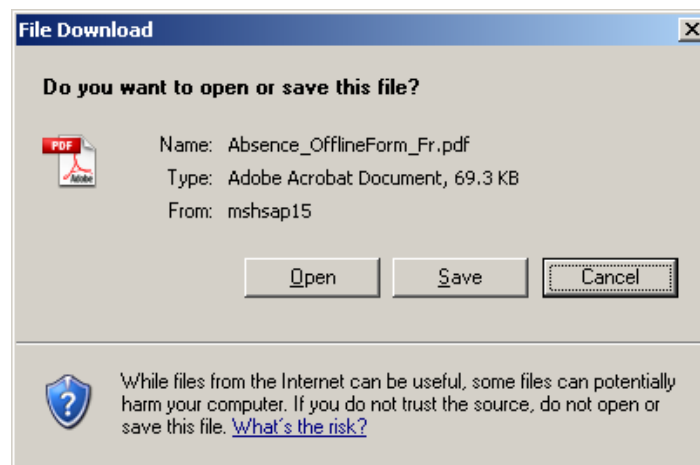


Figure 348 - Tutoriel 4: test de l'application: téléchargement du pdf

On sauvegarde le formulaire sur le bureau et on l'ouvre ensuite dans Acrobat Reader. On constate qu'il a été pré-rempli avec les informations de l'employé:

Absence_OfflineForm_Fr.pdf - Adobe Reader

File Edit View Document Tools Window Help

Find

Hes-SO VALAIS WALLIS
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences
Western Switzerland

Sierre, le 13.05.2008

CONTROLE DES ABSENCES
FORMULAIRE PROTOTYPE

Nom No. Matricule

Prénom

Date de naissance

Absence

Code motif

Date de début Heure de début

Date de fin Heure de fin

Heures d'absence

Figure 349 - Tutoriel 4: test de l'application, ouverture du formulaire dans Adobe Reader

On saisit ensuite les informations relatives à l'absence, on clique sur le bouton de sauvegarde en haut à gauche et on ferme Adobe Reader. On retourne ensuite dans l'application que nous venons de déployer, on upload le formulaire et on clique sur le bouton "Envoyer":

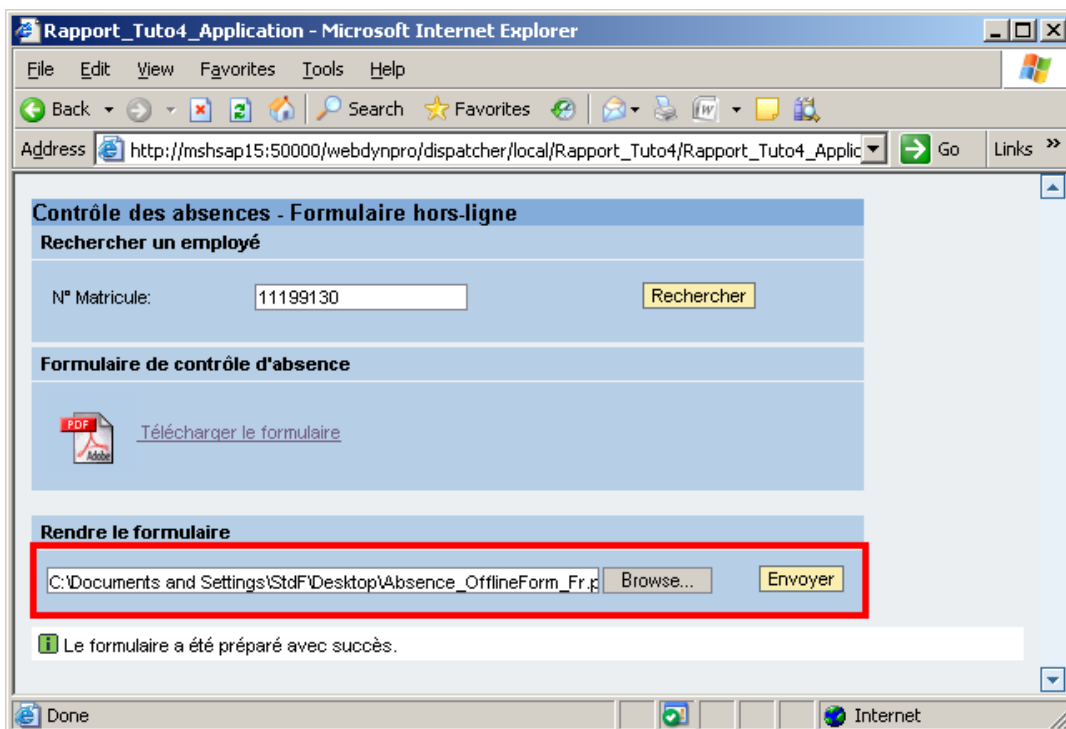


Figure 350 - Tutoriel 4: test de l'application, upload du formulaire

On arrive alors sur la fenêtre qui résume le contenu du formulaire. Il suffit alors de cliquer sur le bouton "Valider":

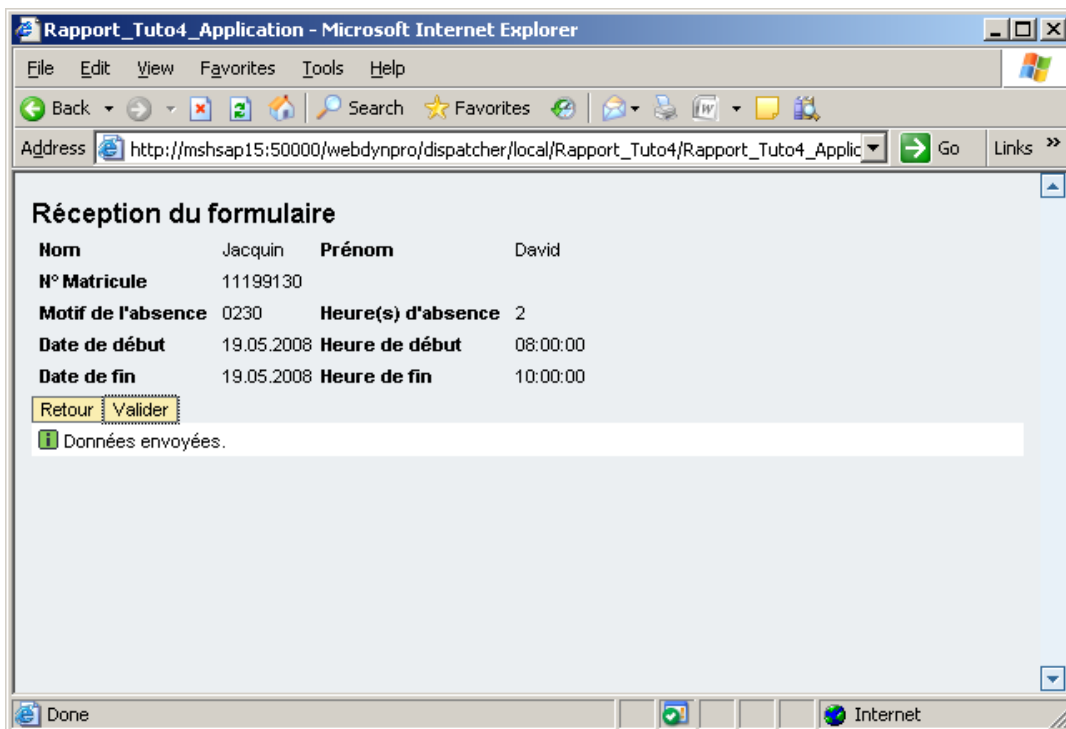


Figure 351 - Tutoriel 4: test de l'application, résumé des données saisies

Le formulaire est ainsi envoyé vers SAP

7.7.14 Vérification

Si on consulte ensuite la table des absences de l'employé 11199130 via le code transaction PA61, on constate que les données du formulaire ont bien été reçues:

From	To	Abs.	Att (abs. ty.)	Start	End	Hours	Days	Lead
19.05.2008	19.05.2008	0230	Doctor's app	08:00	10:00	2,00	0,27	2,00
13.05.2008	13.05.2008	0230	Doctor's app	08:00	11:00	3,00	0,40	3,00

Figure 352 - Tutoriel 4: test de l'application, vérification de la saisie dans SAP

Ceci conclut ce 4^{ème} tutoriel ainsi que la partie relative à l'implémentation des formulaires dans un contexte Web Dynpro.

7.7.15 Commentaires

Globalement, je suis assez satisfait de l'application réalisée dans ce quatrième tutoriel. On peut utiliser le formulaire hors ligne et quand on le souhaite, la navigation est assez complète et il y a même une page qui résume les données saisies et qui fait office de demande de confirmation.

Il y a cependant plusieurs améliorations possibles:

- cacher le lien de téléchargement tant que le matricule de l'employé n'est pas saisi
- contrôler les données saisies et informer l'utilisateur en cas d'erreur (ex: saisie d'une absence un jour férié, durée de l'absence incompatible avec le type d'absence, etc..)
- dans le formulaire, revoir le calcul des heures pour qu'il prenne en compte l'emploi du temps réel de l'employé.

7.8 Pour aller plus loin

Les bases du développement d'un formulaire interactif ont ainsi été abordées. Evidemment, l'utilisation des fichiers .pdf avec SAP ne se limite pas à ce que nous venons de voir, c'est-à-dire l'accès via une page web.

7.8.1 Standalone Forms

Bien que cela dépasse le cadre fixé à la base, il est intéressant de mentionner que l'utilisation d'un formulaire interactif peut se faire sans Web Dynpro, c'est-à-dire que des données peuvent être reçues et envoyées directement depuis le formulaire. C'est ce

qu'on appelle parfois une "Standalone Form". Pour ce faire, il faut créer un web service, dont un document WSDL (Web Service Definition Language), qui utilise diverses BAPIs. Ensuite, lorsque l'on crée le formulaire, on définit une connexion à ce web service et on relie son utilisation au fait de cliquer sur un bouton.

Malgré le fait que la liberté d'utilisation offerte par les "Standalone Forms" soit alléchante, il y a étonnamment peu de documentation à leur sujet. En effet, même si mes recherches n'ont pas été très approfondies, je n'ai trouvé qu'un blog indiquant comment les créer. Et bien que je ne dépasse pas la première étape (la D07 ne semble pas disposer d'un "Web Service Browser" ou du moins il est introuvable), le reste semble être relativement aisé à réaliser.

Ca peut donc être un thème très intéressant pour élargir ses connaissances sur le développement des formulaires interactifs.

7.8.2 Intégration des formulaires dans les processus

Un autre aspect des formulaires interactifs que j'aurais aimé explorer davantage est celui de leur utilisation lors de la définition d'un processus. Comment on intègre les formulaires, comment on définit un processus, etc... Je suis persuadé que ce thème offre énormément d'éléments intéressants, que ce soit au niveau des connaissances ou des avantages qu'ils peuvent apporter à une grande entreprises par exemple. Evidemment, avant de s'attaquer à cette partie, il est nécessaire de comprendre ce qui a été vu lors de ce travail.

7.8.3 Intégration dans Campus Management

L'un des formulaires qui a été abandonné était destiné à être utilisé avec Campus Management. L'idée était de pouvoir l'intégrer sur cette plateforme utilisée par l'école et de simplifier en même temps le traitement du formulaire proposé. Ainsi, en se connectant au portail, un étudiant aurait pu par exemple télécharger ce formulaire dans sa langue maternelle et saisir uniquement les informations pertinentes, les autres étant déjà remplies.

Il faut savoir aussi que l'ajout d'un formulaire sur le portail est plutôt aisé. En effet, durant ce travail j'ai demandé à Michael Müller, un collègue effectuant un travail de diplôme qui touche Campus Management, d'ajouter une application Web Dynpro contenant un formulaire interactif. Les deux fois, l'application était immédiatement opérationnelle. Voici ce à quoi ça peut ressembler:

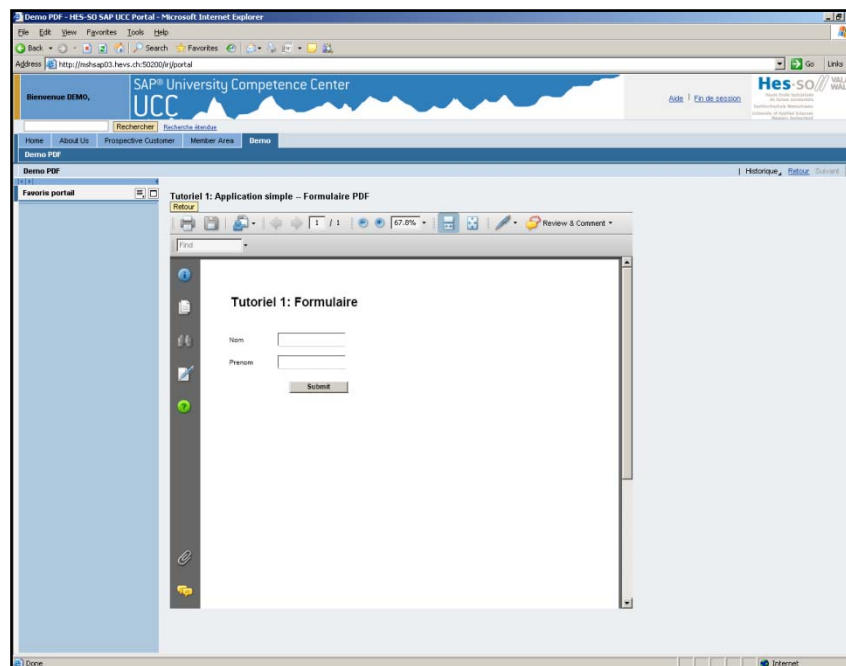


Figure 353 - Campus Management + Application Web Dynpro avec une Interactive Form

En utilisant Campus Management, il existe très certainement la possibilité d'identifier automatiquement l'utilisateur et donc de pré-remplir immédiatement certains champs du formulaire au lieu demander à chaque fois qu'il saisisse son matricule, comme nous l'avons vu dans les tutoriels.

Une fois de plus, je pense que c'est un thème qui mérite d'être approfondi dans la mesure où on fait collaborer 2 technologies qui ont une philosophie commune: offrir un contenu personnalisé.

7.9 Opinion sur l'implémentation

Dans l'ensemble, j'ai trouvé cette phase d'implémentation très intéressante. En effet, on passe un certain temps à comprendre comment les choses marchent, à tester, re-tester, effectuer des recherches, etc.. La progression est plutôt lente mais une fois qu'on a compris certains principes, on consacre un temps beaucoup plus réduit sur le développement de certaines fonctionnalités. La différence est tellement flagrante que lors de la planification de la phase d'implémentation, j'ai largement surestimé le temps que j'y consacrerai.

Un autre élément très appréciable est la quantité d'informations disponibles rien que sur le SDN. Les tutoriels, même s'ils ne marchent pas toujours pour diverses raisons, aident à acquérir certains mécanismes et à comprendre des points clés du développement.

La plupart du temps, on tombe aussi sur des messages postés sur les forums et qui proviennent de personnes qui ont été confrontés à des difficultés plus ou moins similaires. A partir de là, les différentes propositions de solution que l'on trouve dans le

sujet représentent à chaque fois de nouvelles pistes explorables, que ce soit pour le problème actuelle ou un autre que l'on a réglé d'une manière différente. De manière générale, si on sait taper les bons mots-clés, le SDN fournit toujours un élément de réponse, voire la réponse elle-même. Personnellement, je ne connais aucune autre plateforme d'échange qui concentre autant d'informations au même endroit.

7.10 Problèmes rencontrés

Il est un peu difficile de dire ce qui a posé problème durant l'implémentation étant donné que dans la majorité des cas mes problèmes consistaient à ne pas savoir comment coder quelque chose. Il serait assez inutile de lister toutes les solutions trouvées vu qu'elles sont pratiquement toujours similaires: identifier le problème, puis recherche sur le SDN ou dans un tutoriel existant.

Cependant, il y a 2 choses sur lesquels j'ai passé énormément de temps:

- **Comprendre les nœuds:**

Cette notion n'a pas été simple à maîtriser. En effet, j'ai mis beaucoup de temps à comprendre qu'il fallait absolument ajouter un élément, même vide, à un nœud afin d'éviter des erreurs au niveau des formulaires interactifs notamment. La première solution trouvée consistait à modifier la cardinalité d'un nœud. Cependant, cette solution a cessé de fonctionner à partir du moment où il fallait faire appel aux BAPIs. Je me suis donc tourné vers la solution qui passait par l'ajout de ligne de codes, ce qui s'est donc traduit par un temps de compréhension et de maîtrise supplémentaire.

- **Les groupes de bouton radio dans les formulaires:**

A la base, le tutoriel 3 devait indiquer comment effectuer un filtrage des types d'absence directement dans le formulaire. Malheureusement, après de nombreux tests et recherches, je n'arrêtais pas de tomber sur le même problème: la sélection d'une catégorie d'absence renommait la catégorie précédemment sélectionnée. C'est donc ce qui m'a poussé à remonter cette fonctionnalité de filtrage au niveau de l'application où là elle fonctionne parfaitement.

A ce jour, je n'ai toujours pas trouvé une solution à ce problème, mais je reste persuadé que ce que j'ai voulu faire est tout à fait possible.

8 Conclusion

8.1 Avis Personnel

Ce travail de diplôme s'est bien déroulé dans l'ensemble: je pense que les exigences du cahier des charges ont été respectées et que les connaissances relatives au thème ont été acquises et plus ou moins maîtrisées. J'ai aussi le sentiment de disposer des bases nécessaires pour pouvoir passer à la suite, notamment intégrer des formulaires dans des processus.

Durant le projet de groupe 676, j'avais déjà réalisé une petite application Web Dynpro basée sur un tutoriel très simple. A l'époque, je me rappelle qu'il y avait un certain nombre de choses que je ne comprenais pas: pourquoi fait-on cela ainsi? A quoi ça sert? J'avais en fait du mal à placer ce que je faisais dans un contexte plus général. C'était pareil au début de ce travail, j'avais du mal à réaliser où certains tutoriels voulaient en venir et finalement, au fil des semaines, la philosophie m'est parue de plus en plus claire.

Je pense que l'une des faiblesses de ce travail de diplôme est la planification, sur laquelle j'estime ne pas avoir assez travaillé. Bien qu'il soit normal que ce qu'il se passe ne correspondent pas à ce qui a été planifié, il y a certaines tâches flagrantes que je n'ai tout simplement pas pensé à inclure, la plus importante étant la préparation de la D07. Cette erreur est d'autant plus grave que la configuration du serveur a pris un temps non négligeable.

Je ressens aussi une certaine frustration au niveau des applications Web Dynpro que j'ai réalisées car elles peuvent être améliorées sur plusieurs points. Malheureusement, cela n'a pas été une priorité par rapport à l'un des objectifs de ce travail de diplôme qui consistait avant tout à montrer les possibilités de la technologie.

Malgré ça, je conserve un avis extrêmement positif sur les SAP Interactive Forms by Adobe et je pense vraiment que de plus en plus de grandes entreprises y verront un avantage non négligeable. Cette solution est à la fois relativement simple à mettre en place, très puissante et correspond parfaitement à la récente philosophie d'ouverture introduite par SAP NetWeaver.

8.2 Remerciements

Je souhaiterais remercier le Dr. Werner Maier pour m'avoir suivi et orienté durant ce travail de diplôme, pour sa disponibilité et aussi pour les diverses fournitures de bureau estampillées SAP qu'il m'a gracieusement offertes. Merci aussi à Ivan Schnyder et à Olivier Tosi qui m'ont été d'un grand secours à plusieurs reprises. Je remercie aussi Michael Müller pour l'aperçu de Campus Management. Enfin, merci à l'HES-SO Valais pour avoir mis à ma disposition un cadre de travail et un matériel irréprochables.

9 Attestation

Je déclare, par ce document, que j'ai effectué le travail de diplôme ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées.

Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de diplôme, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail .

Sierre, le 19 mai 2008

David Jacquin

10 Sources

10.1 Documents

Note: tous les documents .pdf cités ci-dessous sont disponibles sur le CD.

10.1.1 Informations

- **SAP AG And Adobe Systems, Enabling Collaboration with Interactive Forms**, SAP AG, 2003
Fichier: SAP_AG_and_Adobe_Systems.pdf
- **Adobe Fast Facts**, Adobe, Adobe Systems, janvier 2008
Fichier: Adobe_fast_facts.pdf
- **SAP NetWeaver ABAP and Adobe Forms**, Frakling Herbas, Thomas Jung, 2006
Fichier: SAP_NW_ABAP_Adobe_Forms.pdf
- **SAP INTERACTIVE FORMS BY ADOBE Extending the Power of SAP® Solutions by Automating Business Forms**, SAP AG, octobre 2006
Fichier: bwp_interactive_forms_adobe.pdf
- **Interoperability between SAP NetWeaver Portal and Microsoft SharePoint Technologies**, Thomas Meigen, Tilo Böttcher, décembre 2007
Fichier: Interoperability between SAP NW & MS SPT.pdf
- **PDF Reference Sixth Edition**, Adobe, version 1.7, novembre 2006
Fichier: pdf_reference_1-7.pdf
- **SAP Interactive Forms by Adobe Integration and Optimization of Forms-Based Processes**, Rob Mennen, 2007
Fichier: SAP_IF_Presentation.pdf
- **SAP Smart Form Overview Brief**
Fichier: Smartforms_overview.pdf

10.1.2 Installation, Configuration

- **SAP Interactive Forms by Adobe, Adobe Document Services**, SAP AG, septembre 2007
Fichier: ADS_Config_GuideWebAS7.0.pdf
- **Creating a form based process (Adobe)**, SAP AG, juillet 2006
Fichier: creating_a_form_based_process.pdf

- **How to... troubleshoot the render exception**, SAP AG, janvier 2006
Fichier: Troubleshoot_Render_Exception.pdf
- **Installing a credential for Adobe Document Services**
Fichier: Installing_ADS_Credentials.pdf

10.1.3 Développement

- **Application of Context Programming and Data Binding**, SAP AG, 2004
Fichier: context_and_data_binding.pdf
- **Yet another EVS Valuehelp - Showing Displaytexts for Keys**, Bertram Ganz
Fichier: extended_value_selector.pdf
- **FormCalc UserReference**, Adobe, juillet 2007
Fichier: FormCalc.pdf
- **View – Programming UI and Navigation**, SAP AG, octobre 2005
Fichier: view_programming_ui_and_nav.pdf

10.1.4 Tutoriels

- **BAPI Step-by-step**, Renjith Kumar. P, 26 Janvier 2007
Fichier: BAPI_Step-by-Step.pdf
- **Handling Transactions with BAPIs in Web Dynpro**, SAP AG, 2004
Fichier: handling_transactions_with_bapis_in_WD.pdf
- **Developing an HR Homepage Using Web Dynpro, Part I**, V. Ramakrishna, 9 juin 2005
Fichier: HR_homepage_tutorial_p1.pdf
- **Developing an HR Homepage Using Web Dynpro, Part II**, V. Ramakrishna, 9 juin 2005
Fichier: HR_homepage_tutorial_p2.pdf
- **How To... Create Online and Offline Forms in WebDynpro for Java**, SAG AG, janvier 2006
Fichier: offline_form.pdf
- **Creating you first Web Dynpro Application**, SAP AG, 2004
Fichier: First_WD_Application.pdf

- **Accessing ABAP Functions in Web Dynpro Java**, SAP NetWeaver Product Management, 8 décembre 2007
Fichier: ABAP_Functions_WD_Java.pdf
- **Offline interactive form based on download/upload functionality**
Fichier: offline_form_using_dl_ul.pdf

10.2 Sujets Forum SDN

- **Is it possible to run ADS on 64-bit windows machine and be supported by SAP**, posté le 13 avril 2007
Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=375643>
- **Inputfield is getting disabled for input**, posté le 2 avril 2005
Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=33968>
- **add model node element**, posté le 19 février 2007
Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=324266>
- **not getting value from DropDownByKey**, posté le 14 août 2006
Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=196865&tstart=0>
- **Dynamically adding values to DropDownByKey**, posté le 27 juillet 2006
Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=186324>
- **webdynpro populating dropdown values programmatically**, posté le 16 août 2006
Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=198280&messageID=2212592>
- **Change default pdf file name**, posté le 7 février 2008
Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=637981>
- **Problem in interactive form**, posté le 1er juin 2007
Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=428068>
- **Bound data node Node of InteractiveForm must not be empty!**, posté le 30 janvier 2007
Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=722880>
- **Download generated PDF**, posté le 15 janvier 2008
Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=744322>
- **Read adobe form element**, posté le 28 mars 2007
Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=358477>

- **Adobe Interactive Forms in Web Dynpro**, posté le 2 novembre 2005

Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=68500>

- **Radio Buttons**, posté le 27 mars 2007

Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=358031>

- **bapi and RFC main differences**, posté le 15 février 2008

Adresse: <https://www.sdn.sap.com/irj/sdn/thread?threadID=744277>

10.3 Blogs SDN

- **My struggles with the Adobe Document Services in WAS6.40**, Achim Hauck, posté le 4 février 2004

Adresse: <https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/1232>

- **Purely Offline Adobe Scenario for BAPI call**, Lisha Peter, posté le 16 octobre 2007

Adresse: <https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/7588>

10.4 SAP Notes

- Note 736902 - Adobe Credentials
- Note 811342 - Time Out exception when rendering to Adobe document service
- Note 848539 - Changes on ADS SP-12
- Note 836174 - ISR/Adobe/Web Dynpro Java: Support for troubleshooting
- Note 925741 - Adobe document services (ADS) with non-supported platforms
- Note 1066586 - Web Services Security - blank in the Visual Administrator

10.5 Sites divers

- **SAP:**
<http://www.sap.com>
- **Adobe:**
<http://www.adobe.com>
- **SAP Interactive Forms by Adobe, SDN:**
<http://www.sdn.sap.com/irj/sdn/adobe>
- **SAP Service Marketplace:**
<http://service.sap.com>
- **Centre de compétence SAP de la HES-SO Valais:**
<http://ccsap.hevs.ch>
- **HES-SO Valais:**
<http://www.hevs.ch>

A. Donnée du travail de diplôme

HEVs

SI	TV	EE	IC	EST
X	X	X	X	

Données du travail de diplôme Daten der Diplomarbeit

FO.0.2.02.07.DB
DD/18/05/2006

Filière / Studiengang : Informatique de gestion

Confidentiel / Vertraulich ☐

Etudiant / Student (nom+ numéro téléphone) David Jacquin	Année scolaire / Schuljahr 2007/08	No TD / Nr. DA
Proposé par / vorgeschlagen von Prof. Dr. Werner Maier		Lieu d'exécution / Ausführungsort Salle TD, Sierre Expert / Experte

Titre / Titel:

Adobe Interactive Form (Authoring) & SAP ECC

Description / Beschreibung:

<http://www.sap.com/solutions/solutionextensions/interactiveforms/index.epx>

SAP Interactive Forms by Adobe

SAP Interactive Forms enables organizations in any industry to extend the power of SAP and other software to improve customer communications and become more efficient. The tool helps you reach more users – internally and externally – by helping you create online and offline interactive forms in Adobe Portable Document Format (PDF).

With SAP Interactive Forms, your company can extend the reach of enterprise applications – such as product life-cycle management, enterprise resource planning, supply chain management, and customer relationship management – to a much broader set of users, while easily pulling vital data needed from SAP software.

SAP Interactive Forms can help you:

- Provide customers and employees with personalized forms generated directly from SAP applications and solutions
- Capture data in completed forms that can flow directly back to SAP software – eliminating the need for error prone, manual data input
- Customize forms to meet the specific needs of your business or industry

Objectifs / Ziele:

- • Installing Adobe Interactive Forms
- • Establishing link between ECC 6.0 (with IS CM) and Adobe Acrobat
- • Prototype of interaction between Adobe and ECC 6.0
- • 1 Form for "Etat du Valais"
- • 1 Form for Campus Management

Signature ou visa / Unterschrift oder Visum

Resp. de la filière Informatique de gestion

Professeur/Dozent: Dr. Maier, Werner

Etudiant/Student: David Jacquin

Délais / Termine

Attribution du thème / Ausgabe des Auftrags:
18.02.08

Remise du rapport / Abgabe des Schlussberichts:

19.05.08

Exposition publique / Ausstellung Diplomarbeiten:

Rapport reçu le / Schlussbericht erhalten am Visa du secrétariat / Visum des Sekretariats:

B. Cahier des charges



Travail de diplôme
David Jacquin
Février-Mai 2008



SAP Interactive Forms by Adobe

Cahier des charges

Description

Comme son nom l'indique, SAP Interactive Forms by Adobe permet de réaliser des formulaires interactifs. Tout en remplaçant leurs équivalents sur papier, ces formulaires peuvent être à la fois paramétrés selon les besoins d'une entreprise et pré-remplis avec des informations valides extraites d'un système SAP. En plus de cela, il est bien évidemment possible d'imprimer ces formulaires comme des fichiers .pdf classiques, mais aussi de modifier ou de saisir de nouvelles données qui seront ensuite renvoyées et traitées dans le système SAP.

Objectif

Le principal objectif de ce travail de diplôme est dans un premier temps de découvrir, comprendre, puis présenter SAP Interactive Forms by Adobe, et, dans un deuxième temps, de mettre en pratique les connaissances acquises afin de réaliser un prototype d'interaction entre un système SAP ECC 6.0 et Adobe Acrobat. Deux Interactive Forms, l'une pour l'Etat du Valais, l'autre pour Campus Management seront aussi réalisées en faisant appel aux différents outils étudiés. Les différentes phases de ce travail seront documentées et commentées dans le rapport final.

Remarque: suite à la séance du 20.02.2008 avec le Dr. Maier, le nombre d'Interactive Forms pour l'Etat du Valais demandé dans la donnée du travail de diplôme passe de 2 à 1.

Phase de préparation

Installer et configurer les différents outils nécessaires à la réalisation de ce travail

Phase d'analyse

Adobe Interactive Forms

- Présentation du partenariat entre SAP et Adobe
- Présentation d'Adobe Interactive Forms
- Fonctionnement
- Exemple
- Eventuelles alternative à Adobe Interactive Forms

Architecture utilisée

- Présentation des outils et systèmes utilisés avec Adobe Interactive Forms
 - SAP NetWeaver Developer Studio, SAP ECC 6.0, etc..

Phase d'implémentation

Etablir un premier lien entre ECC 6.0 et Adobe Acrobat

Créer un prototype d'interaction entre ECC 6.0 et Adobe Acrobat

- Création d'un formulaire
- Pré-remplir certains champs avec des valeurs personnalisées venant d'ECC 6.0
- Renvoyer des valeurs vers ECC 6.0 à partir du formulaire
- Contrôle des données saisies
- Effectuer un calcul simple à partir des données du formulaire
- Réaliser un formulaire de saisie utilisable hors ligne

Forms pour l'Etat du Valais et Campus Management

- Créer les formulaires selon les modèles proposés et les besoins exprimés
- Les réaliser en se basant sur le prototype d'interaction
- Tenter d'ajouter des éléments qui facilitent la saisie (ex: liste déroulantes, ..)

Phase de test

Tester les formulaires à chaque étape (création, interaction, saisie, etc...)

Documentation supplémentaire

Pour les différentes étapes de l'implémentation créer un guide ou un tutoriel vidéo

Indiquer les problèmes rencontrés et expliquer les solutions apportées

Systèmes et logiciels utilisés

Windows 2003 Server

SAP GUI 7.10 (avec Patch-level 6)

SAP ECC 6.0

SAP NetWeaver Developer Studio 7.0.09

Cahier des charges remis à Sierre, le 22.02.2008

Signatures

Responsable de la filière
Informatique de gestion

Bruno Montani

Professeur

Dr. Werner Maier

Etudiant

David Jacquin

C. Planification



Hes·SO VALAIS WALLIS
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences
Western Switzerland



SAP Interactive Forms by Adobe

Travail de diplôme (Fevrier-Mai 2008)

David Jacquin

Planning

Taches	Début	Fin	Durée	Réel
Démarrage				
Phase de démarrage/préparation				
Séance d'attribution des TD	18.02.08	18.02.08	0.5	
Première séance avec Dr. Maier	20.02.08	20.02.08	0.5	
Préparation du poste de travail				
Installation de SAP GUI + Patch	18.02.08	18.02.08	0.5	
Installation de SAP NetWeaver Developer Studio	18.02.08	18.02.08	0.5	
Installations diverses	18.02.08	18.02.08	2	
Configuration	18.02.08	18.02.08	2	
Plan de travail				
Cahier des charges	18.02.08	19.02.08	4	
Planification	19.02.08	19.02.08	6	
Total phase de démarrage			16	47.5
Phase d'analyse/Documentation				
Adobe Interactive Forms				
Recherches informations	20.02.08	21.02.08	20	
Partenariat SAP/Adobe	21.02.08	21.02.08	5	
Fonctionnement	21.02.08	26.02.08	20	
Exemple	26.02.08	26.02.08	4	
Comparatifs (ex: Smartforms)	26.02.08	26.02.08	6	
Architecture utilisée				
Vue d'ensemble	26.02.08	27.02.08	5	
SAP NetWeaver Developer Studio	27.02.08	27.02.08	10	
SAP ECC 6.0	27.02.08	28.02.08	6	
Campus Management	28.02.08	29.02.08	8	
Total phase d'analyse/Documentation			84	45
Phase d'implémentation				
Premier lien entre ECC 6.0 et Acrobat	29.02.08	04.03.08	20	
Prototype d'interaction entre ECC 6.0 et Acrobat				
Création d'un formulaire	04.03.08	06.03.08	15	
Pré-remplissage avec valeurs existantes	06.03.08	12.03.08	30	
Renvoyer les valeurs (formulaire -> ECC6)	12.03.08	18.03.08	30	
Contrôle des données saisies	18.03.08	20.03.08	20	
Effectuer un calcul simple	20.03.08	24.03.08	15	
Formulaire hors ligne	24.03.08	27.03.08	25	
Form pour l'Etat du Valais				
Création selon modèle/besoins	27.03.08	01.04.08	20	
Realisation basée sur le prototype	01.04.08	07.04.08	30	
Ajout d'éléments qui facilitent la saisie	07.04.08	09.04.08	20	
Form pour Campus Management				

Création selon modèle/besoins	09.04.08	14.04.08	20	
Realisation basée sur le prototype	14.04.08	15.04.08	15	
Ajout d'éléments qui facilitent la saisie	15.04.08	17.04.08	15	
Total phase d'implémentation			275	172
Tests				
Prototype				
Tests	17.04.08	21.04.08	15	
Corrections des bugs	21.04.08	24.04.08	25	
Formulaires Etats du Valais et Campus Management				
Tests	24.04.08	25.04.08	10	
Corrections des bugs	25.04.08	28.04.08	15	
Total phase de tests			65	25
Finalisation				
Rapport final				
Redaction (y compris annexes)	25.02.08	12.05.08	85	
Mise en page	12.05.08	14.05.08	12	
Corrections	14.05.08	14.05.08	5	
Préparation rendu (selon directives)	14.05.08	15.05.08	5	
CD				
Interface	15.05.08	16.05.08	12	
Elaboration de l'arborescence	16.05.08	16.05.08	3	
Création des CDs demandés	16.05.08	16.05.08	1	
Total phase de finalisation			123	148.5
Divers				
Séances hebdomadaires avec le Dr. Maier	18.02.08	16.05.08	12	
Séance avec l'Etat du Valais	16.04.08	16.04.08	1	
Autre séances non fixées	18.02.08	16.05.08	5	
Communications (ex: e-mail)	18.02.08	16.05.08	4	
Rapports hebdomadaires	18.02.08	16.05.08	15	
Total divers			37	34
Total en heures:			600	472
Remarques				
Calendrier de rencontres avec le professeur: chaque vendredi, de 13h à 14h				
Suivi du module 626 le mercredi après-midi. Ces heures ne sont évidemment pas comptabilisées dans ce planning				

D. Suivi des heures

- Semaine 1:

HES-SO Valais		Weekly Report							
Title		Adobe Interactive Forms (Authoring) & SAP ECC							
Name		David Jacquin							
Week		18/02/2008 - 24/02/2008							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Analysis & Planification				0.50					0.50
Installation			1.00	1.00					2.00
Configuration									0.00
Programmation									0.00
Research / Lecture			7.00	2.50	7.00	2.00			18.50
Redaction					1.00	7.50		1.00	9.50
Course		1.00							1.00
Total		1.00	8.00	4.00	8.00	9.50	0.00	1.00	31.50
Working Report Details									
Date	Note								
18.02.2008	Lundi: Attribution TD à 17h								
19.02.2008	Mardi: Compréhension du sujet, recherche d'informations, lecture anciens TDs, installation de SAPGUI et de WebDeveloperStudio, lecture parties administratives du TD								
20.02.2008	Mercredi: 11h30: RDV avec W. Maier. Après-midi: cours 626								
21.02.2008	Jeudi: recherches, documentation, début cahier des charges								
22.02.2008	Vendredi: cahier des charges, planning								
24.02.2008	Dimanche: quelques recherches sur le web								
Date	Signature Professor								
Comment									

- Semaine 2:

HE-SO Valais		Weekly Report							
Title		Adobe Interactive Forms (Authoring) & SAP ECC							
Name		David Jacquin							
Week		25/02/2008 - 02/03/2008							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Analysis & Planification		0.50	0.50		0.50				1.50
Installation				2.00	4.00				6.00
Configuration					4.00	4.00			8.00
Programmation									0.00
Research / Lecture		3.50	4.50	2.00		4.50	1.50		16.00
Redaction		4.00	3.00						7.00
Course									0.00
Total		8.00	8.00	4.00	8.50	8.50	1.50	0.00	38.50
Working Report Details									
Date	Note								
25.02.2008	Lundi: séance avec Dr. Maier, recherche + redaction sur SAP Interactive Forms								
26.02.2008	Mardi: documentation/recherche sur SAP et Adobe en général. Séance avec Ivan Schnyder à 13h.								
27.02.2008	Mercredi: recherches sur les outils, après-midi: cours 626								
28.02.2008	Jeudi: tentatives d'installation d'Adobe Livecycle designer (installation, désinstallation, test)								
29.02.2008	Vendredi: début de prise en main d'Adobe Livecycle designer, quelques tests, recherches erreurs de déploiement. Visite d'Ivan Schnyder dans l'après-midi.								
01.03.2008	Samedi: recherche web sur erreur de déploiement (time out)								
Date	Signature Professor								
Comment									

• **Semaine 3:**

HES-SO Valais		Weekly Report							
Title		Adobe Interactive Forms (Authoring) & SAP ECC							
Name		David Jacquin							
Week		03/03/2008 - 09/03/2008							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Analysis & Planification		0.50				0.50			1.00
Installation				3.00	2.50				5.50
Configuration		4.00	4.00			3.00			11.00
Programmation		0.50	0.50						1.00
Research / Lecture		4.00	4.00	2.50	2.50	4.00			17.00
Redaction					3.50	2.00			5.50
Course									0.00
Total		9.00	8.50	5.50	8.50	9.50	0.00	0.00	41.00
Working Report Details									
Date	Note								
03.03.2008	Lundi: recherches/tests sur causes de l'Erreur de Time Out, séance avec Dr. Maier								
04.03.2008	Mardi: recherche/tests sur la même erreur, solution trouvée, début de son application								
05.03.2008	Mercredi: séances avec Ivan Schnyder, importation Support Packages. Après-midi: module 626								
06.03.2008	Jeudi: importation de support packages, rédaction (schémas, divers)								
07.03.2008	Vendredi: séance avec le Dr. Maier, importation Support Packages, rédaction (notamment début sur l'architecture, recherches sur les problèmes d'importation)								
Date	Signature Professor								
Comment									

• **Semaine 4:**

HES-SO Valais		Weekly Report							
Title		Adobe Interactive Forms (Authoring) & SAP ECC							
Name		David Jacquin							
Week		10/03/2008 - 14/03/2008							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Analysis & Planification									0.00
Installation						1.00			1.00
Configuration		1.50			1.00	2.50			5.00
Programmation									0.00
Research / Lecture		3.50	4.00	2.00	3.50	3.00			16.00
Redaction		3.50	4.00	2.00	3.50	3.00			16.00
Course									0.00
Total		8.50	8.00	4.00	8.00	9.50	0.00	0.00	38.00
Working Report Details									
Date	Note								
10.03.2008	Lundi: recherches et rédaction sur l'architecture, le format PDF, les alternatives. Matin: visite d'Ivan								
11.03.2008	Mardi: pareil								
12.03.2008	Mercredi: pareil le matin. Après-midi: module 676.								
13.03.2008	Jeudi: recherches et rédaction sur l'architecture, le format PDF, les alternatives.								
14.03.2008	Vendredi: installation de packages Java, installation du Reader Rights credential, rédaction guide,								
Date	Signature Professor								
Comment									

• **Semaine 5:**

HES-SO Valais		Weekly Report							
Title		Adobe Interactive Forms (Authoring) & SAP ECC							
Name		David Jacquin							
Week		17/03/2008 - 21/03/2008							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Analysis & Planification					0.50				0.50
Installation									0.00
Configuration		1.00	1.00		2.00				4.00
Programmation		6.00	6.50		5.00				17.50
Research / Lecture		1.00	1.00		1.50				3.50
Redaction									0.00
Course									0.00
Total		8.00	8.50	0.00	9.00	0.00	0.00	0.00	25.50
Working Report Details									
Date	Note								
17.03.2008	Lundi: essais tutoriels (essentiellement formulaires), recherches sur les erreurs, tests								
18.03.2008	Mardi: essais tutoriels (formulaires+form driven process), recherches sur les erreurs, tests								
19.03.2008	Mercredi: férié								
20.03.2008	Jeudi: tutoriel (surtout sur les BAPIs), recherche sur les erreurs, tests.								
	Après-midi: Séance avec le Dr. Maier								
21.03.2008	Vendredi: férié								
Date	Signature Professor								
Comment									

• **Semaine 6:**

HES-SO Valais		Weekly Report							
Title		Adobe Interactive Forms (Authoring) & SAP ECC							
Name		David Jacquin							
Week		31/03/2008 - 04/04/2008							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Analysis & Planification									0.00
Installation									0.00
Configuration			1.00						1.00
Programmation			6.00		3.00	7.00			16.00
Research / Lecture		2.00	2.00	5.00	2.00	2.50			13.50
Redaction					4.00				4.00
Course									0.00
Total		2.00	9.00	5.00	9.00	9.50	0.00	0.00	34.50
Working Report Details									
Date	Note								
31.03.2008	Lundi: malade, reprise du TD et recherches sur les prochaines étapes du prototype. Rattrapage des heures prévu.								
01.04.2008	Mardi: création prototype: liaison application <->PDF								
02.04.2008	Mercredi: matin: prototype: liaison application<->PDF + calculs. Après-midi: Module 626								
03.04.2008	Jeudi: matin: création d'une vidéo pour le tutoriel d'intégration d'un formulaire PDF dans une application web dynpro. Après midi: Recherches et implémentation partie RFC								
04.04.2008	Vendredi: prototype: partie RFC: création, test, debug								
Date		Signature Professor							
Comment									

• **Semaine 7:**

HES-SO Valais		Weekly Report							
Title		Adobe Interactive Forms (Authoring) & SAP ECC							
Name		David Jacquin							
Week		07/04/2008 - 11/04/2008							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Analysis & Planification		0.50							0.50
Installation									0.00
Configuration									0.00
Programmation		6.00	6.00	1.00	5.50				18.50
Research / Lecture		2.00	2.00	3.00	3.00				10.00
Redaction						8.00			8.00
Course									0.00
Total		8.50	8.00	4.00	8.50	8.00	0.00	0.00	37.00
Working Report Details									
Date	Note								
07.04-08.04	Lundi & Mardi: tentative de récupération de données depuis SAP via RFC. Lundi matin: séance avec le Dr. Maier.								
09.04.2008	Mercredi: recherches sur l'envoi de données vers SAP via RFC. Après-midi: module 626								
10.04.2008	Jeudi: tentative de renvoi de donnée vers SAP, test d'un tutoriel existant, tentative d'adaptation								
11.04.2008	Vendredi: rédaction d'une partie du rapport, début de préparation pour la présentation du 16.04								
Date		Signature Professor							
Comment									

• **Semaine 8:**

HES-SO Valais		Weekly Report							
Title		Adobe Interactive Forms (Authoring) & SAP ECC							
Name		David Jacquin							
Week		14/04/2008 - 18/04/2008							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Analysis & Planification		0.50							0.50
Installation			1.00						1.00
Configuration			1.00			1.00			2.00
Programmation		3.00	5.00	3.00	6.00	6.00			23.00
Research / Lecture		1.00	1.00		2.00	2.00			6.00
Redaction		4.00							4.00
Course				1.00					1.00
Total		8.50	8.00	4.00	8.00	9.00	0.00	0.00	37.50
Working Report Details									
Date	Note								
14.04.2008	Lundi: 11h: séance avec la Dr. Maier. Préparation fichier .ppt pour la présentation, réalisation de schémas, tests sur la reception de données depuis SAP								
15.04.2008	Mardi: préparation présentation (vérification salle+machine, création formulaire de démo, tests), quelques tests sur les calculs et l'importation de données								
16.04.2008	Mercredi: préparation présentation, présentation. Après-midi: module 626								
17.04.2008	Jeudi: recherches/tests envoi données vers SAP (via SE37, puis application web dynpro)								
18.04.2008	Vendredi: tentative d'adaptation application-> formulaire pour l'envoi de données vers SAP. Debug. Tests/comparaison/évaluation natif vs ActiveX								
Date		Signature Professor							
Comment									

• Semaine 9:

HES-SO Valais	Weekly Report							
Title	Adobe Interactive Forms (Authoring) & SAP ECC							
Name	David Jacquin							
Week	21.04.2008 - 25.04.2008							
	Mo	Tu	We	Th	Fr	Sa	So	Total
Analysis & Planification		0.50						0.50
Installation								0.00
Configuration								0.00
Programmation	6.50	5.50	2.00	2.00				16.00
Research / Lecture	2.00	2.00	2.00	1.00				7.00
Redaction				5.00	8.50			13.50
Course								0.00
Total	8.50	8.00	4.00	8.00	8.50	0.00	0.00	37.00
Working Report Details								
Date	Note							
21.04.2008	Lundi: développement prototype complet, traque de bugs							
22.04.2008	Mardi matin: séance avec le Dr. Maier. Développement prototype complet, traque de bugs, calculs							
23.04.2008	Mercredi matin: Développement prototype complet: recherches et test sur le problème des calculs							
24.04.2008	Jeudi: finition prototype, réalisation d'une video de démonstration, rédaction rapport final							
25.04.2008	Vendredi: rédaction rapport final							
Date	Signature Professor							
Comment								

• Semaine 10:

HES-SO Valais	Weekly Report							
Title	Adobe Interactive Forms (Authoring) & SAP ECC							
Name	David Jacquin							
Week								
	Mo	Tu	We	Th	Fr	Sa	So	Total
Analysis & Planification								0.00
Installation								0.00
Configuration								0.00
Programmation								0.00
Research / Lecture								0.00
Redaction	8.00	8.00	9.00	8.00	9.00	5.00		47.00
Working Report Details								0.00
Total	8.00	8.00	9.00	8.00	9.00	5.00	0.00	47.00
Working Report Details								
Date	Note							
28.04-03.05	Rédaction rapport final, création de tutoriels vidéo, création interface CD							
Date	Signature Professor							
Comment								

• **Semaine 11:**

HES-SO Valais		Weekly Report							
Title		Adobe Interactive Forms (Authoring) & SAP ECC							
Name		David Jacquin							
Week		05.05.2008-11.05.2008							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Analysis & Planification									0.00
Installation									0.00
Configuration									0.00
Programmation		3.00	3.00	6.00	6.50	4.50		6.00	29.00
Research / Lecture		1.00	1.00		1.00	1.00			4.00
Redaction		4.00	5.00		2.50	4.50			16.00
Working Report Details									0.00
Total		8.00	9.00	6.00	10.00	10.00	0.00	6.00	49.00
Working Report Details									
Date	Note								
05.05-09.05	rédaction rapport, tutoriels, réalisation formulaires de rendu, debug, tests, recherches								
11.05.2008	Dimanche: heures de travail prévues								
Date		Signature Professor							
Comment									

• **Semaine 12:**

HES-SO Valais		Weekly Report							
Title		Adobe Interactive Forms (Authoring) & SAP ECC							
Name		David Jacquin							
Week		12.05.2008 - 18.05.2008							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Analysis & Planification		0.50							0.50
Installation									0.00
Configuration									0.00
Programmation		9.50							9.50
Research / Lecture									0.00
Redaction			10.50	11.00	10.00	9.00	5.00		45.50
Working Report Details									0.00
Total		10.00	10.50	11.00	10.00	9.00	5.00	0.00	55.50
Working Report Details									
Date	Note								
12.05.2008	Lundi: création offline form. Visite du Dr. Maier le matin								
13.05-18.05	Mardi à Samedi: rédaction rapport, finalisation								
Date		Signature Professor							
Comment									

• Total:

HES-SO Valais	Report Summary					
Title	Adobe Interactive Forms (Authoring) & SAP ECC					
Student	David Jacquin					
Period	18/02/2008 - 19/05/2008					
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Analysis & Planification	0.50	1.50	1.00	0.00	0.50	0.00
Installation	2.00	6.00	5.50	1.00	0.00	0.00
Configuration	0.00	8.00	11.00	5.00	4.00	1.00
Programmation	0.00	0.00	1.00	0.00	17.50	16.00
Research / Lecture	18.50	16.00	17.00	16.00	3.50	13.50
Redaction	9.50	7.00	5.50	16.00	0.00	4.00
Course	1.00	0.00	0.00	0.00	0.00	0.00
Total	31.50	38.50	41.00	38.00	25.50	34.50

HES-SO Valais	Report Summary						
Title	Adobe Interactive Forms (Authoring) & SAP ECC						
Student	David Jacquin						
Period	18/02/2008 - 19/05/2008						
	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Total
Analysis & Planification	0.50	0.50	0.50	0.00	0.00	0.50	5.50
Installation	0.00	1.00	0.00	0.00	0.00	0.00	15.50
Configuration	0.00	2.00	0.00	0.00	0.00	0.00	31.00
Programmation	18.50	23.00	16.00	0.00	29.00	9.50	130.50
Research / Lecture	10.00	6.00	7.00	0.00	4.00	0.00	111.50
Redaction	8.00	4.00	13.50	47.00	16.00	45.50	176.00
Course	0.00	1.00	0.00	0.00	0.00	0.00	2.00
Total	37.00	37.50	37.00	47.00	49.00	55.50	472.00

E. Rapports hebdomadaires

Pour une meilleure lisibilité, ces rapports sont disponibles sur le CD.

Semaine 1

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <01> : 18.02. – 24.02.2008

Initial situation

- Début du travail de diplôme
- Attribution du thème

1 THE BOOT-NEW BUSINESS RUN C&P SAP

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <01> : 18.02. – 24.02.2008

Finished

- Plan de travail selon annexe
 - Cahier des charges
 - Planning
- Première prise de contact avec le professeur
- Documentation et lecture pour avoir une idée générale du sujet.
- Contact d'Olivier Tosi et Ivan Schnyder:
 - Informations de base (logins, adresses, ports, etc...) afin de pouvoir continuer obtenues

2 THE BOOT-NEW BUSINESS RUN C&P SAP

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <01> : 18.02. – 24.02.2008

Problem(s)

- Aucun problème rencontré

3 THE BOOT-NEW BUSINESS RUN C&P SAP

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <01> : 18.02. – 24.02.2008

Resolved problem(s)

- Pas de problème, pas de solution

4 THE BOOT-NEW BUSINESS RUN C&P SAP

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <01> : 18.02. – 24.02.2008

Planning for the next week

- Début de semaine
 - Mieux comprendre Interactive Forms
 - Rédiger documentation
 - Rechercher des informations sur Smartforms
- Courant de la semaine
 - Rencontre avec Ivan Schnyder pour parler de Campus Management notamment.
- Fin de semaine
 - Consulter les différents tutoriels sur www.adobe.com/interactiveforms et faire quelques essais avec WebDyn Pro

5 THE BOOT-NEW BUSINESS RUN C&P SAP

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <01> : 18.02. – 24.02.2008

Questions

- Pas de question pour le moment

6 THE BOOT-NEW BUSINESS RUN C&P SAP

Semaine 2

Initial situation

1

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «02» : 25.02. – 29.02.2008

- Cahier des charges/Planning terminés
- SAPGUI, SAPNetWeaver Developer Studio installés
- Logins ECC6 sur (D06, D07), SAP Marketplace, SDN actifs

THE HES-SO BUSINESS SCHOOL

Finished

2

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «02» : 25.02. – 29.02.2008

- Présentation rapide de SAP et Adobe terminée
- Présentation générale d'Interactive Forms terminée à ~60%
- SAP Interactive Forms installé avec NetWeaver Developer Studio
 - Quelques brefs essais de mise en page et de configuration de champs interactifs effectués

THE HES-SO BUSINESS SCHOOL

Problem(s)

3

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «02» : 25.02. – 29.02.2008

- (Pour plus tard) Déploiement depuis NetWeaver Developer Studio impossible: le système D07 doit disposer de l'Adobe Reader Rights Credential afin de pouvoir utiliser des interactive forms.
 - 2 solutions à tester:
 - Tenter d'utiliser les credentials destinées à SAP NetWeaver Trial & Sneak preview (expiration le 30 avril)
 - Via obtenir un credential permanent. (cf SAP Note 736902). Dans ce cas là, je ne dispose pas de l'autorisation de demander une licence.
 - D07: lors du déploiement: erreur de time out quand l'application à déployer contient un élément d'interactive Forms.
 - Cause inconnue pour l'instant.

THE HES-SO BUSINESS SCHOOL

Resolved problem(s)

4

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «02» : 25.02. – 29.02.2008

- Installation d'Adobe Livecycle Designer:
 - Solution: une fois que mon compte a obtenu l'autorisation d'effectuer des téléchargements à partir du SAP Marketplace, j'ai téléchargé Adobe Livecycle Designer 7.1 puis 8.0
 - Problème avec la version 7.1: une fois le Designer lancé, gros ralentissements de l'ordinateur (charge mémoire importante et en constante augmentation)
 - Solution: désinstallation de la version 7.1 puis installation de la version 8.0

THE HES-SO BUSINESS SCHOOL

Planning for the next week

5

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «02» : 25.02. – 29.02.2008

- Début de semaine:
 - Réussir à résoudre les erreurs liées au déploiement:
 - Autorisations
 - Adobe Reader Rights Credential
 - Créer une première version des formulaires demandés (disposition des champs, etc...)
- Courant de la semaine:
 - Réussir à déployer les tutoriels de base disponibles sur le SDN dans leur intégralité et sans erreur
- Fin de la semaine:
 - Idéalement, établir un premier lien entre ECC 6.0 et un formulaire.

THE HES-SO BUSINESS SCHOOL

Questions

6

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «02» : 25.02. – 29.02.2008

- Quelques questions sur les formulaires:
 - Sécurité? Signatures?
 - Autres questions techniques
- Posées durant la prochaine séance, lundi à 11h

THE HES-SO BUSINESS SCHOOL

Semaine 3

Initial situation

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <03> : 03.03. – 07.03.2008

- Implementation: erreur de Read Time Out lorsqu'on tente de déployer un formulaire
 - Cause inconnue en début de semaine
- Installation/configuration:
 - D07: SAP NetWeaver 2004s, SPS 6
 - Adobe Document Services SP 9

1 THE HES-SO BUSINESS SCHOOL SAP

Finished

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <03> : 03.03. – 07.03.2008

- Cause et solution théoriques à l'erreur de Read Time Out trouvées, pas encore testées
 - Erreur de "render" repérée et analysée: Adobe Document Services SPS incompatible avec un système d'exploitation 64 bits
 - Même si ce n'est pas ça, la solution sera utile pour plus tard.
- Support Packages (SPS 13) pour la D07 téléchargés, en cours d'installation
- Adobe Credentials (Reader rights) reçus, prévus pour installation après la mise à jour de la D07

2 THE HES-SO BUSINESS SCHOOL SAP

Problem(s)

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <03> : 03.03. – 07.03.2008

- Actuellement, impossibilité d'importer le Support Package du composant "Logistics & Accounting" (SAP_APPL), SAPIH60006.
 - Erreur dans la phase de DDIC_ACTIVATION (vue lors de la séance du vendredi 7 mars)
 - Impossibilité d'activer la table WVKOMP

3 THE HES-SO BUSINESS SCHOOL SAP

Resolved problem(s)

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <03> : 03.03. – 07.03.2008

- Diverses erreurs lors de l'importation de Support Packages:
 - Solutions:
 - Mise à jour du SPAM
 - Download/Installation d'un CRT
 - Installation d'un support package (PI_BASIS_1_700 via code transaction SAINT)

4 THE HES-SO BUSINESS SCHOOL SAP

Planning for the next week

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <03> : 03.03. – 07.03.2008

- Début de semaine:
 - résoudre le problème d'importation du Support Package pour SAP_APPL
 - Terminer l'importation de la partie ABAP
 - Commencer l'importation de la partie JAVA
- Milieu de la semaine:
 - Terminer l'importation de la partie JAVA
 - Reprendre les tentatives de déploiement des Interactive Forms
- Fin de semaine: selon l'avancement:
 - Si le déploiement marche, réaliser 1 ou 2 tutoriels et avancer sur un prototype ou formulaire pour l'état du Valais
 - En cas d'erreur: chercher les causes et trouver des solutions
- Durant les phases d'importation: rédiger de la documentation quand il faut attendre.

5 THE HES-SO BUSINESS SCHOOL SAP

Questions

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <03> : 03.03. – 07.03.2008

- Questions répondues lors de la séance du vendredi 7 mars.

6 THE HES-SO BUSINESS SCHOOL SAP

Semaine 4

Initial situation

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «54» : 10.03. – 14.03.2008

- Impossibilité de déployer des formulaires
- Mise à jour de la D07 bloquée.

1

Finished

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «54» : 10.03. – 14.03.2008

- Mise à jour du système sur la D07 "terminée":
 - Quelques packages pas encore installés et en principe pas essentiels pour pouvoir utiliser des formulaires interactifs. Ils seront installés ultérieurement.
- En raison des problèmes de mise à jour, avancement sur la partie écrite du travail de diplôme + recherches diverses, notamment:
 - Architecture
 - Format pdf
 - Autres possibilités
- Premier déploiement et déploiements successifs REUSIS !! →

2

Problem(s)

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «54» : 10.03. – 14.03.2008

- Aucun problème à l'heure actuelle. Rendez-vous la semaine prochaine pour de nouvelles erreurs !

3

Resolved problem(s)

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «54» : 10.03. – 14.03.2008

- Précédentes erreurs de déploiement disparues depuis la mise à jour:
 - Read Time Out
 - Erreur de "Render"
- Problème de la mise à jour de la D07 bloquée (problème d'activation d'une table):
 - Solution: restauration d'un backup, puis installation de certains packages, mais pas tous cette fois-ci.
- Problèmes d'authentification d'un Technical User (ADISUser) qui communique avec le web service Adobe Document Services
 - Mot de passe incorrect/invalid, il fallait juste le redéfinir.

4

Planning for the next week

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «54» : 10.03. – 14.03.2008

- Début de semaine:
 - Réaliser quelques tutoriels disponibles sur le SDN
- Milieu et fin de semaine:
 - Selon l'avancement, tenter une liaison SAP → Form, puis Form → SAP pour le prototype.
 - Rédiger un tutoriel basique montrant comment effectuer cette liaison (+ vidéo si assez de temps)

5

Questions

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «54» : 10.03. – 14.03.2008

- Rapport final, guides: faut-il inclure les différents mots de passe utilisés?

6

Semaine 5

Initial situation

1

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week <05> : 17.03. – 21.03.2008

- Erreur de Read Time Out résolue
- Déploiements possibles.

SAP

Finished

2

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week <05> : 17.03. – 21.03.2008

- Premier déploiement:
 - Pas d'autres erreurs
 - Droits liés aux Reader Rights Credential actifs
- Quelques tutoriels réalisés, testés
- Début d'un formulaire "Prototype"

SAP

Problem(s)

3

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week <05> : 17.03. – 21.03.2008

- Tutoriel "Form driven Process"
 - Erreur lors du renvoi, formulaire pas réceptionné.
- Tutoriel "BAPis with Web Dynpro"
 - Erreur lors de l'envoi de données vers SAP (table SBOOK)
 - Erreur de NullPointerException

Dans l'ensemble, les erreurs se produisent vers la fin de leurs scénarios.

SAP

Resolved problem(s)

4

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week <05> : 17.03. – 21.03.2008

- Lors de l'importation de "sample applications" relatives aux différents tutoriels, problème de packages Java introuvables
 - Cause: la plupart des répertoires ont un nom au format <Dossier>_2.0.0 au lieu de <Dossier> comme l'attend l'application
 - Solution: éditer les chemins vers ces dossiers dans Web Developer Studio ("Propriétés du projet").

SAP

Planning for the next week

5

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week <05> : 17.03. – 21.03.2008

- Prochaine semaine: lundi 31 mars -> vendredi 4 avril
 - Début de semaine:
 - Poursuivre le développement du formulaire prototype
 - Interactions, opérations simples
 - Milieu de la semaine:
 - Revenir sur le tutoriel qui utilise les BAPis et corriger l'erreur de NullPointerException
 - Fin de semaine:
 - Selon avancement, appliquer les éléments du tutoriel sur les BAPis au formulaire prototype

SAP

Questions

6

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week <05> : 17.03. – 21.03.2008

- Pas de question

SAP

Semaine 6

Hes·SO

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «06» : 31.03. – 04.04.2008

Initial situation

- Reprise du TD
- Quelques tutoriels effectués avant les vacances.

1 THE HES·SO BUSINESS RUN SAP

Hes·SO

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «06» : 31.03. – 04.04.2008

Finished

- Réalisation d'un prototype basé sur plusieurs tutoriels et qui:
 - Intègre un formulaire PDF simple
 - Permet de passer des valeurs de l'application vers le PDF et inversement
 - Permet d'effectuer un calcul simple
- Réalisation d'un tutoriel vidéo qui couvre la création du prototype (sauf le calcul, mais cela sera abordé dans un autre tutoriel)

2 THE HES·SO BUSINESS RUN SAP

Hes·SO

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «06» : 31.03. – 04.04.2008

Problem(s)

- Actuellement, réalisation de la partie RFC du prototype et qui permet de récupérer des informations venant de SAP. Le but est de saisir le numéro d'un employé et d'obtenir son nom et prénom en retour.
 - Problème dans le code ou le "Model Node" et qui empêche de récupérer des informations
 - Pas de problème avec les connexions Jira (testées et validées)
 - Pas d'erreur de compilation, ni de déploiement.

3 THE HES·SO BUSINESS RUN SAP

Hes·SO

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «06» : 31.03. – 04.04.2008

Resolved problem(s)

- Problème de l'obtention des informations depuis SAP
 - Solution: utilisation de BAPI déjà existantes (pour le prototype: BAPI_EMPLOYEE_GETDATA)
- Prototype avec intégration d'un PDF dans l'application: certains champs (inputfield toujours inactifs)
 - Solution: changement de la cardinalité dans les propriétés du Value Node ou ajout de lignes de code, les 2 solutions fonctionnent.

4 THE HES·SO BUSINESS RUN SAP

Hes·SO

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «06» : 31.03. – 04.04.2008

Planning for the next week

- Début et milieu de semaine: travailler sur les RFC
- Fin de semaine: selon avancement, travailler sur le rapport final, rédiger un tutoriel sur la partie RFC. (+vidéo)

5 THE HES·SO BUSINESS RUN SAP

Hes·SO

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «06» : 31.03. – 04.04.2008

Questions

- Pas de question en particulier, tout avance correctement.

6 THE HES·SO BUSINESS RUN SAP

Semaine 7

Initial situation

1

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «07» : 07.04.– 11.04.2008

- Liaison application Web Dynpro <=> Form effectué avec persistance des données.
- Calculs possibles.

Finished

2

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «07» : 07.04.– 11.04.2008

- Liaison Form <=> SAP
 - Dans l'application prototype: récupération des informations d'un employé en saisissant son numéro + pré-remplissage d'un formulaire avec les informations reçues.
 - Réception depuis SAP: terminée
 - Envoi vers SAP: quelques tentatives mais pas encore réussies.

Problem(s)

3

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «07» : 07.04.– 11.04.2008

- Envoi des valeurs d'une application ou formulaire vers SAP:
 - En cliquant sur le bouton d'envoi, il semblerait qu'une connexion soit établie et que quelque chose soit envoyé, mais les données n'apparaissent pas dans les informations relatives à l'employé (via SAP GUI)
 - Causes possibles à tester:
 - Les données sont peut-être sauvegardées dans une autre table.
 - Les données sont pas acceptées par le système
 - Erreur dans l'application: les données ne sont pas envoyées.

Resolved problem(s)

4

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «07» : 07.04.– 11.04.2008

- Problèmes de la réception des données depuis SAP
 - Dans l'application, les champs de recherches et de résultats étaient grisés et pas utilisables
 - Cause liée au "Model Node" vide.
 - Aucune des solutions proposées sur le SCN ou dans les différents tutoriels ne fonctionnent.
 - Solution: modification du code en tentant une "cuisine personnelle".
 - Avoir si cette modification pose problème pour la suite (risque faible à priori)

Planning for the next week

5

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «07» : 07.04.– 11.04.2008

- Lundi: préparer le formulaire de démonstration pour la présentation de mercredi.
- Jusqu'à mercredi ou jeudi: réussir à envoyer des données vers SAP
- Fin de semaine: avancer dans le rapport, commencer à créer des tutoriels sur les SAP Interactive Forms

Questions

6

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «07» : 07.04.– 11.04.2008

- Présentation de mercredi, besoin d'avis:
 - Démonstration: si ça peut intéresser: je peux incorporer la réception de données relatives à un employé et pré-remplir un formulaire avec ces données (nom, prénom...). Il suffit de saisir le matricule de la personne.
 - Pour être sûr de ne pas avoir de problème, je peux créer une vidéo de démonstration au cas où.
- Il se pourrait que l'ordinateur utilisé lors de la présentation ne dispose pas d'un plugin ActiveX nécessaire à l'utilisation des formulaires. Il faudrait que je sache de quel ordinateur il s'agit (et dans quelle salle il se trouve) afin de le contrôler et éventuellement d'installer le plugin au plus tard mardi soir.

Semaine 8

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «08» : 14.04. – 18.04.2008

Initial situation

- Reception de données depuis SAP possible
- Quelques tests infructueux sur l'envoi de données vers SAP

1

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «08» : 14.04. – 18.04.2008

Finished

- Envoi de données vers SAP réussi
- Présentation pour l'Etat du Valais effectuée.

2

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «08» : 14.04. – 18.04.2008

Problem(s)

- Pas de problème, ça avance.

3

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «08» : 14.04. – 18.04.2008

Resolved problem(s)

- Envoi de données vers SAP
 - En utilisant le code transaction SE37 et en testant la BAPI "BAP_LABSENCE_CREATE", il n'était pas possible de créer une absence pour un employé ("employee is not locked yet")
- Solution: utiliser 2 BAPIs supplémentaires:
 - Avant de créer l'absence, utiliser "BAP_EMPLOYEE_ENQUEUE" pour verrouiller l'employé
 - Créer l'absence avec "BAP_LABSENCE_CREATE"
 - Après la création, utiliser "BAP_EMPLOYEE_DEQUEUE" pour déverrouiller l'employé

4

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «08» : 14.04. – 18.04.2008

Planning for the next week

- Lundi: réaliser ou compléter un prototype pour qu'il soit complet, c'est-à-dire être capable à la fois de:
 - Recevoir des données à partir de SAP
 - Pré-remplir certains champs
 - Renvoyer les données saisies vers SAP
- Reste de la semaine: rédaction du rapport final
 - Plusieurs jours prévus car il y a beaucoup de choses à rédiger (explications, tutoriels, schémas,...)

5

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week «08» : 14.04. – 18.04.2008

Questions

- Pas de question en particulier.

6

Semaine 9

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «09» : 21.04. – 25.04.2008

Initial situation

- Envoi de données vers SAP possible

1

THE BEST-KRM-BUSINESSIDEAS FOR SAP

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «09» : 21.04. – 25.04.2008

Finished

- Prototype complet terminé:
 - Reception de données provenant de SAP
 - Pré-remplissage d'un formulaire
 - Calcul
 - Renvoi de données vers SAP

2

THE BEST-KRM-BUSINESSIDEAS FOR SAP

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «09» : 21.04. – 25.04.2008

Problem(s)

- Pas de problème en ce moment (en cours d'avancement dans la rédaction du rapport final)

3

THE BEST-KRM-BUSINESSIDEAS FOR SAP

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «09» : 21.04. – 25.04.2008

Resolved problem(s)

- Prototype:
 - Un seul noeud de données peut être rattaché à un formulaire interactif.
 - Première solution testée: créer un noeud général qui regroupe tous les noeuds → Beaucoup d'erreurs, ne semble pas fonctionner. Solution abandonnée.
 - Deuxième solution: créer un noeud hybride qui "mix" les différents attributs des autres noeuds et qui sont utilisés par le formulaire → Fonctionne.
 - Calcul des heures d'absence: au niveau du code, problème sur le calcul à cause du formatage des dates. Au lieu de retourner un résultat valide, ça retournait 0 voire rien du tout.
 - Modifier le code et le formatage pour qu'il soit compatible dans la formule du calcul mais aussi et surtout avec le format demandé par la API lorsque les données sont renvoyées vers SAP.

4

THE BEST-KRM-BUSINESSIDEAS FOR SAP

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «09» : 21.04. – 25.04.2008

Planning for the next week

- Toute la semaine: rédaction rapport final pour parler de tout ce qui a été fait jusqu'à présent et création de tutoriels vidéos

5

THE BEST-KRM-BUSINESSIDEAS FOR SAP

Hes·SO
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe
Week «09» : 21.04. – 25.04.2008

Questions

- Pas de question

6

THE BEST-KRM-BUSINESSIDEAS FOR SAP

Semaine 10

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <10> : 28.04. – 05.05.2008

Initial situation

- Prototype complet terminé
- Début de rédaction du rapport final

1

THE HES·SO BUSINESS BOX SAP

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <10> : 28.04. – 05.05.2008

Finished

- Contenu du rapport final terminé à ~70%

2

THE HES·SO BUSINESS BOX SAP

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <10> : 28.04. – 05.05.2008

Problem(s)

- Aucun problème, que de la rédaction

3

THE HES·SO BUSINESS BOX SAP

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <10> : 28.04. – 05.05.2008

Resolved problem(s)

- Rédaction rapport = pas de problème à résoudre

4

THE HES·SO BUSINESS BOX SAP

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <10> : 28.04. – 05.05.2008

Planning for the next week

- Lundi → Jeudi: créer les livrables:
 - Formulaire pour l'Etat du Valais (ajouter liste déroulante pour les types d'absence)
 - Formulaire pour Campus Management
- Vendredi, éventuellement samedi: rédaction rapport sur les livrables, tutoriels sur les parties spécifiques

5

THE HES·SO BUSINESS BOX SAP

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <10> : 28.04. – 05.05.2008

Questions

- Fixer une prochaine séance? De préférence en début de semaine.

6

THE HES·SO BUSINESS BOX SAP

Semaine 11

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <11> : 05.05 – 09.05.2008

Initial situation

- Rédaction rapport final
- Début de création du formulaire pour l'état du valais

1

THE HES-SO BUSINESS SCHOOL

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <11> : 05.05 – 09.05.2008

Finished

- Quelques parties du rapport et quelques tutoriels terminés
- Un formulaire terminé

2

THE HES-SO BUSINESS SCHOOL

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <11> : 05.05 – 09.05.2008

Problem(s)

- Doit encore tenter de créer un formulaire interactif "offline"
 - Quelques essais effectués
- Problème des listes déroulantes:
 - Incompatibilité des noeuds de données utilisés
 - Peu d'informations disponibles pour une solution
 - Peut être une solution qui consiste à créer un noeud personnalisé et à effectuer une sorte de mapping via code → à tester

3

THE HES-SO BUSINESS SCHOOL

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <11> : 05.05 – 09.05.2008

Resolved problem(s)

- Problèmes en cours de résolution.

4

THE HES-SO BUSINESS SCHOOL

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <11> : 05.05 – 09.05.2008

Planning for the next week

- Lundi soir au plus tard: terminer les formulaires
- Reste de la semaine: finalisation du travail de diplôme
- Tenter de rendre le TD vendredi après-midi pour éviter tout problème éventuel qui surviendrait au dernier moment.

5

THE HES-SO BUSINESS SCHOOL

Hes·SO **SAP**

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <11> : 05.05 – 09.05.2008

Questions

- Suite à l'e-mail de Jean-Pierre Folonier (09.05, 16h38), nouvelles directives?

6

THE HES-SO BUSINESS SCHOOL

Semaine 12

Hes·SO

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <12> : 12.05 – 18.05.2008

Initial situation

- Rapport final en cours de rédaction
- Offline Form à créer.

1

Hes·SO

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <12> : 12.05 – 18.05.2008

Finished

- Rapport final
- CD
- Offline Form
- Video flash pour Friedbert Eberle
- Screens pour Jean-Pierre Folonier

2

Hes·SO

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <12> : 12.05 – 18.05.2008

Problem(s)

- Pas de problèmes rencontrés, heureusement

3

Hes·SO

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <12> : 12.05 – 18.05.2008

Resolved problem(s)

- Pas de problème, pas de solution

4

Hes·SO

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <12> : 12.05 – 18.05.2008

Planning for the next week

- Lundi: rendre TD avant midi

5

Hes·SO

Name : David Jacquin
Title of Bachelor Thesis : SAP Interactive Forms by Adobe Week <12> : 12.05 – 18.05.2008

Questions

- Pas de question.

Merci !

6

F. Mots de passe utilisés

- **Campus Management:**
 - o Adresse: <http://mshsap03.hevs.ch:50200>
 - o Username: demo
 - o Password: diplom08
- **BAPIs/RFC – Utilisateur technique:**
 - o User: jdavid_tech
 - o Password: tdsap2008
- **Adobe Credential, Reader Rights credential:**
 - o Password: Sai5p9c51K
- **Déploiement:**
 - o Password: pwd4sdm

G. Formulaire d'absence pour l'Etat du Valais (modèle)

CONTROLE DES ABSENCES/PRESENCES

DEPARTEMENT DES FINANCES, DES INSTITUTIONS ET DE LA SECURITE
SERVICE CANTONAL DE L'INFORMATIQUE

Sion, le

Nom, prénom :

Nous vous prions de bien vouloir prendre note de l'absence ou de la présence pour le motif suivant :

Mois

Code d'absence/présence	Motif	jour			en	jour			en	Début	Fin	Solde au facteur
		le (du)	mois	année		le (du)	mois	année				
					en				en			
					en				en			
					en				en			
					en				en			
					en				en			
					en				en			
					en				en			
					en				en			
					en				en			
					en				en			

* Remarque

Le Chef de service:

Le (la) requérant(e):

Motifs absences

1100 = Vacances
1105 = Vacances 1/2 jour
1130 = Congé naissance
1135 = Congé non payé
1140 = Congé spécial
1150 = Congé maternité
1155 = Congé d'adoption
1160 = Congé famille

1200 = Maladie infectieuse à 3 jours
1210 = Maladie infectieuse à 5 jours
1220 = Maladie infectieuse
1230 = VM hors canton
1240 = Traitements récurrents
1250 = Urgence médicale
1300 = Accident professionnel
1310 = Accident non-professionnel
1320 = Visite médicale après accident

1400 = Service militaire
1410 = Etude de recon
1420 = Protection civile
1430 = Cours J+5
1500 = Compensation IIS
1510 = Compensation IV
1600 = Affaires publiques
1900 = Divers

Motifs présences

2100 = Service extérieur
2200 = Formation interne
2210 = Formation informatique
2220 = Formation externe
2230 = Cours prof. apprenti
2300 = Télétravail
2400 = Entretien collègue service

Vue du gestionnaire temps

CONTROLE DES ABSENCES/PRESENCES

DEPARTEMENT DES FINANCES, DES INSTITUTIONS ET DE LA SECURITE
SERVICE CANTONAL DE L'INFORMATIQUE

Sion, le

Nom, prénom :

Nous vous prions de bien vouloir prendre note de l'absence ou de la présence pour le motif suivant :

Mois

Code d'absence/présence	Motif	jour			en	jour			en	Début	Fin	Solde au facteur
		le (du)	mois	année		le (du)	mois	année				
					en				en			
					en				en			

Correction des mouvements		jour			en	jour			en	Début	Fin
le (du)	mois	année	le (du)	mois		année					
					en				en		
					en				en		

Comptabilisation		jour			en	jour			en	Début	Fin
IIS	IV	le (du)	mois	année		le (du)	mois	année			
					en				en		
					en				en		

Heures supplémentaires effectuées
(entre 1900 et 0500, samedi, dimanche, jour férié/obsid, au-delà de 10h30)

* Remarque

Le Chef de service:

Le (la) requérant(e):

Motifs absences

1100 = Vacances
1105 = Vacances 1/2 jour
1130 = Congé naissance
1135 = Congé non payé
1140 = Congé spécial
1150 = Congé maternité
1155 = Congé d'adoption
1160 = Congé famille

1200 = Maladie infectieuse à 3 jours
1210 = Maladie infectieuse à 5 jours
1220 = Maladie infectieuse
1230 = VM hors canton
1240 = Traitements récurrents
1250 = Urgence médicale
1300 = Accident professionnel
1310 = Accident non-professionnel
1320 = Visite médicale après accident

1400 = Service militaire
1410 = Etude de recon
1420 = Protection civile
1430 = Cours J+5
1500 = Compensation IIS
1510 = Compensation IV
1600 = Affaires publiques
1900 = Divers

Motifs présences

2100 = Service extérieur
2200 = Formation interne
2210 = Formation informatique
2220 = Formation externe
2230 = Cours prof. apprenti
2300 = Télétravail
2400 = Entretien collègue service

Vue du gestionnaire temps

H. Formulaire pour Campus Management

HES-SO Valais

SI	TV	EE	IG	EST
		X	X	X

**Demande de déplacement
de service militaire/protection civile**

FO 2.2.02.02.EF
DD/11/10/2007

IDENTITÉ

Nom
Grade
Domicile
Tél.

Prénom
Incorporation
Rue

AVS N :

DÉPLACEMENT DE SERVICE

A un autre moment dans la même année ☐

Dans une année ultérieure ☐

☐ Ecole de recrue

Du Au

☐ Ecole de sous-officier

Du Au

☐ Cours de répétition militaire

Du Au

☐ Cours de répétition PCi

Du Au

☐ Autre service d'avancement

Du Au

☐ Un congé de

Du Au

Date possible de remplacement (à remplir dans tous les cas)

ETUDES

Genre d'études

Semestre Automne du

au

Semestre Printemps du

au

SEMESTRE(S) SUIVI(S)

Semestre (s) suivi (s)

MOTIFS

☐ Examen de promotion

☐ Examens de modules

☐ Stage

☐ Examen théorique de diplôme

☐ Travail pratique de diplôme

☐ Cours bloc (obligatoire)

Autre motifs _____

Du

Au

Et du

Au

Lieu et date :

Signature du requérant (qui a pris connaissance
des dispositions régissant le déplacement de service)

Signature du Responsable de filière

APPRECIATION DE LA DEMANDE PAR L'ÉTABLISSEMENT

Proposition : _____

Le médiateur : _____

PROPOSITION DE L'OFFICE DE LIAISON