Allegato “A” all’accordo d’uso e distribuzione dei codici TE

Sommario

A.1 Sito web TE 3

TE – Sito web 3

A.2 Librerie 3

BigTEMSWFInterfaces – Comunicazione con workflow 3

public interface ItrasportiEccezionaliExternalDataExchangeService 3

public class ParametersEventArgs 4

BigTEMSWFLib – Libreria di workflow 4

public class Content 5

public class ContentsCollection 5

public class SendMailActivity 5

public sealed class SmwPareriEntiTE 8

public sealed class SmwTestWorkflow 9

public sealed class SmwTrasportiEccezionali 9

public class StateActivityBase 11

public class StateMachineWorkflowActivityBase 13

A.3 Librerie 15

BigTEMSWFServices - Servizi di workflow 15

public abstract class MarchalByRefWFRuntime 15

public abstract class TrackingManager 16

public class WfAppDomainDispatcher 17

BigDBObjFuncs40– Libreria Servizi di Data Access e Sincronizzazione 18

ClientDBObject40– Libreria Servizi di Data Access e Sincronizzazione 18

GeneralUtils40 – Libreria Funzioni Generali 18

LibPareriEsterni- Libreria Integrazione con il TE Crossing 18

UrlAuth – Libreria Encrypting della QueryString 18

A.4 Ambiente 19

Elenco componenti di terze parti che compongono l’ambiente di lavoro per le quali verrà garantito il funzionamento di nuovi rilasci. 19

A.1 SITO WEB TE

TE – Sito web

Questo progetto contiene le pagine .aspx i controlli .ascx e i loro file associati. Viene ricompreso il Db degli utenti, mezzi, stato delle pratiche e rilasci.

A.2 LIBRERIE

Di seguito la descrizione delle diverse librerie utilizzate nel sistema per la gestione dei trasporti eccezionali.

BigTEMSWFInterfaces – Comunicazione con workflow

Questa libreria contiene l'interfaccia per comunicare con i workflow di TrasportiEccezionali e il tipo di argomenti usato per gli eventi.

public interface ItrasportiEccezionaliExternalDataExchangeService

L'interfaccia definisce gli eventi associati ai workflow e le funzioni esterne richiamate dagli stessi. Questi eventi e funzioni sono implementati nella classe TrasportiEccezionaliExternalDataExchangeService di WaMswf.

Events

event EventHandler<ParametersEventArgs> AutorizzazioneAccettata;	Evento che comunica al workflow che l'autorizzazione è stata accettata.
event EventHandler<ParametersEventArgs> AutorizzazioneRifiutata;	Evento che comunica al workflow che l'autorizzazione è stata rifiutata.
event EventHandler<ParametersEventArgs> DomandaProtocollata;	Evento che comunica al workflow che la domanda è stata protocollata.
event EventHandler<ParametersEventArgs> IstruttoriaAccettata;	Evento che comunica al workflow che l'istruttoria è stata accettata.
event EventHandler<ParametersEventArgs> IstruttoriaRifiutata;	Evento che comunica al workflow che l'istruttoria è stata rifiutata.
event EventHandler<ParametersEventArgs> PareреApprovato;	Evento che comunica al workflow che un ente esterno ha confermato il parere riguardo al percorso di sua competenza.

event EventHandler<ParametersEventArgs> ParereInserito;	Evento che comunica al workflow che un ente esterno ha inserito un parere riguardo al percorso di sua competenza.
event EventHandler<ParametersEventArgs> PercorsoAccettato;	Evento che comunica al workflow che il percorso è stato accettato.
event EventHandler<ParametersEventArgs> PercorsoRifiutato;	Evento che comunica al workflow che il percorso è stato rifiutato.
event EventHandler<ParametersEventArgs> SetParameters;	Evento che comunica al workflow il passaggio di nuovi parametri.

Methods

void ReportParereApprovato(Guid InstanceId, Dictionary<string, object> parameters);	Comunica che il workflow dei pareri è completato.

public class ParametersEventArgs

La classe definisce il tipo di argiomenti degli eventi associati ai workflow. Il dizionario dei parametri contiene i valori da assegnare alle proprietà pubilche, utilizzando come chiave il nomi case sensitive delle proprietà stesse.

Fields

public Dictionary<string, object> Parameters;	Elenco dei parametri per il workflow.

Methods

public ParametersEventArgs(Guid InstanceId);	Costruttore della classe

BigTEMSWFLib – Libreria di workflow

Questa libreria contiene la definizione dei workflow Trasporti Eccezionali e Pareri Enti. Inoltre Contiene le activity personalizzate utilizzate nei workflow. E implementa la parametrizzazione delle proprietà per I settaggi del workflow.

public class Content

I content vengono utilizzati per estrarre dei dati dall'oggetto Pratica passato ai workflow in formato XML, e possono essere inseriti nel testo per poi essere sostituiti dai relativi valori quando il campo viene interpretato dal workflow (es: SendMailActivity.Body).

Methods

public Content(string name, Type type, string xPath);	Costruttore di classe.
public object GetValue(string xmlString);	Ritorna il valore del content.

Properties

public string Name { get; set; }	Nome del content.
public Type Type { get; set; }	Il tipo del content.
public string XPath { get; set; }	Il XPATH per avere il valore dal flusso.

public class ContentsCollection

Un dizionario di Content, in grado di restituire il valore di un content per nome. La StateMachineWorkflowActivityBase dichiara la proprietà Contents di tipo ContentsCollection per contenere tutti i content associati ad un workflow.

Methods

public ContentsCollection();	Costruttore di classe.
public Content GetContent(string name);	Ritorna il content con il nome specificato.

public class SendMailActivity

L'activity espone tutte le proprieta' necessarie per generare in dinamico una mail, ed implementa il metodo SendMail da richiamare quando si decide di inviarla. La StateMachineWorkflowActivityBase dichiara la funzione Exec_SendMailActivity utilizzata per questo scopo: la funzione ricerca una SendMailActivity a partire dal nome della CodeActivity che la richiama, che quindi deve chiamarsi "Exec_[NomeSendMailActivity]".

Fields

public static DependencyProperty BodyProperty;	DipendencyProperty per la proprietà Body.
public static DependencyProperty FromProperty;	DipendencyProperty per la proprietà From.
public static DependencyProperty IsBodyHTMLProperty;	DipendencyProperty per la proprietà IsBodyHTML.
public static DependencyProperty SmtpDomainProperty;	DipendencyProperty per la proprietà SmtpDomain.
public static DependencyProperty SmtpPasswdProperty;	DipendencyProperty per la proprietà SmtpPasswd.
public static DependencyProperty SmtpPortProperty;	DipendencyProperty per la proprietà SmtpPort.
public static DependencyProperty SmtpServerProperty;	DipendencyProperty per la proprietà SmtpServer.
public static DependencyProperty SmtpUserProperty;	DipendencyProperty per la proprietà SmtpUser.
public static DependencyProperty SubjectProperty;	DipendencyProperty per la proprietà Subject.
public static DependencyProperty UsersConnectionStringProperty;	DipendencyProperty per la proprietà UsersConnectionString.
public static DependencyProperty UsersFilterProperty;	DipendencyProperty per la proprietà UsersFilter.
public static DependencyProperty UsersProperty;	DipendencyProperty per la proprietà Users.
public static DependencyProperty UsersTableNameProperty;	DipendencyProperty per la proprietà UsersTableName.

Methods

public SendMailActivity();	Costruttore di classe.
public SmtpClient GetSmtpClient();	Ritorna il riferimento a SMTP.

public NetworkCredential GetSmtpCredentials();	Ritorna le credenziali di conessione a SMTP.
public DataTable GetUsersDataTable(string connectionString, string tableName, string filter);	Ritorna I dati degli utenti destinatari della email.
public MailAddressCollection GetUsersMailAddressCollection();	Ritorna gli indirizzi degli utenti destinatari della email.
public string ReplaceContentsValues(string sourceStr);	Esegue il replace dei contents nel corpo della email.
public string ReplacePropertiesValues(string sourceStr);	Esegue il replace delle proprietà nel corpo della email.
public bool SendMail();	Invia la email.

Properties

public string Body { get; set; }	Corpo della email.
public string From { get; set; }	Mittente della email.
public bool IsBodyHTML { get; set; }	True se il corpo è di tipo HTML.
public string SmtpDomain { get; set; }	Dominio dell'utente SMTP.
public string SmtpPasswd { get; set; }	Password dell'utente SMTP.
public int SmtpPort { get; set; }	Porta di SMTP.
public string SmtpServer { get; set; }	Server SMTP.
public string SmtpUser { get; set; }	Username SMTP.
public string Subject { get; set; }	Oggetto della email.
public string Users { get; set; }	Elenco primary key degli utenti a chi inviare la email (in formato: "|pk1|pk2|pk3|").

public string UsersConnectionString { get; set; }	Connectionstring al database degli utenti.
public string UsersFilter { get; set; }	Filtro per la tabella degli utenti destinatari.
public string UsersTableName { get; set; }	Nome della tabella degli utenti nel database.

public sealed class SmwPareriEntiTE

E' il workflow dei pareri di enti esterni. Viene invocato automaticamente dal workflow di Trasporti Eccezionali (uno per ogni ente), il quale a sua volta rimane in attesa finche' tutti i sottoworkflow non terminino l'esecuzione.

Fields

public Dictionary<string, object> ReportParereApprovatoParameters;	Parametri del workflow dei pareri.

Methods

public SmwPareriEntiTE();	Costruttore di classe.
public override void Exec_SendMail(object sender, EventArgs e);	Invia la mail di una SendMailActivity. NB: la CodeActivity sender dovrebbe chiamarsi "Exec_[NomeSendMailActivity]"

Properties

public string DataProtocolloIn { get; set; }	Data di protocollo in entrata della pratica.
public int PK_Pratica { get; set; }	PK della pratica per la cui sono richiesti I pareri.
public string ProtocolloIn { get; set; }	Numero di protocollo della pratica.
public string sCodicePratica { get; set; }	Codice della pratica.
public string sCognomeInseritore { get; set; }	Cognome dell'inseritore della pratica.

public string sDesTipoPratica { get; set; }	Tipo della pratica.
public string sNomeInseritore { get; set; }	Nome dell'inseritore.
public string sRagioneSociale { get; set; }	Ragione sociale del richiedente.

public sealed class SmwTestWorkflow

Un workflow utilizzato per testare le funzionalità dei workflow in produzione, da usare solo in fase di sviluppo e test.

Methods

public SmwTestWorkflow();	Costruttore di classe.

public sealed class SmwTrasportiEccezionali

Workflow che definisce gli stati di una pratica di Trasporti Eccezionali.

Fields

protected Guid InstanceIdValue;	Identificatore del workflow.

Methods

public SmwTrasportiEccezionali();	Costruttore di classe.
public override void Exec_SendMail(object sender, EventArgs e);	Invia la mail di una SendMailActivity. NB: la CodeActivity sender dovrebbe chiamarsi "Exec_[NomeSendMailActivity]"

Properties

public string ApprovaPercorsoInseritoComment { get; set; }	Commento dell'approvazione del percorso inserito.

public StateActivityBase.StateStatus ApprovaPercorsoInseritoStatus { get; set; }	Stato dell'approvazione del percorso.
public string[] arrayGestori { get; set; }	Elenco di enti esterni a cui chiedere il parere.
public string[] arrFileNames { get; set; }	Nomi degli allegati della pratica.
public byte[][] arrFiles { get; set; }	I byte dei file allegati alla pratica.
public string CodicePratica { get; set; }	Codice della Pratica
public string CognomeInseritore { get; set; }	Cognome dell'inseritore.
public string DataProtocolloIn { get; set; }	Data del protocollo in entrata.
public string DataProtocolloOut { get; set; }	Data del protocollo dell'autorizzazione.
public string DesTipoPratica { get; set; }	Tipo di pratica.
public string FirmaComment { get; set; }	Commento dell'autorizzazione.
public StateActivityBase.StateStatus FirmaStatus { get; set; }	Stato dell'attività di firma.
public string GUIDCardIn { get; set; }	Id del protocollo in entrata.
public string GUIDCardOut { get; set; }	Id del protocollo in uscita.
public Guid InstanceId { get; }	Identificatore del workflow.
public string IstruttoriaPraticaComment { get; set; }	Commento dell'istruttoria.
public StateActivityBase.StateStatus IstruttoriaPraticaStatus { get; set; }	Stato dell'istruttoria.
public int ModalitaPresentazione { get; set; }	Modalità di presentazione della pratica.
public string NomeInseritore { get; set; }	Nome dell'inseritore della pratica.

public int PK_Inseritore { get; set; }	PK dell'inseritore della pratica.
public int PK_Pratica { get; set; }	PK della pratica.
public string PraticaSostituita { get; set; }	Identifica la pratica che è stata sostituita.
public string ProtocolloIn { get; set; }	Numero di protocollo in entrata della pratica.
public string ProtocolloOut { get; set; }	Numero di protocollo in uscita della pratica.
public string RagioneSociale { get; set; }	Ragione sociale del richidente.
public string startWS { get; set; }	Indica lo stato di protocollazione.
public int TipoPratica { get; set; }	PK del tipo di pratica richiesta.
public int TipoPresentazione { get; set; }	PK del tipo di presentazione della pratica.
public string WSResult { get; set; }	Risultato della chiamata al web service di protocollazione.

public class StateActivityBase

Attività che estende le funzionalità delle StateActivity, aggiungendo le proprietà e funzioni necessarie per gestire gli utenti assegnati ed altri dati che identificano lo stato di un workflow di Trasporti Eccezionali (codice pratica, la prossima attivita richiesta, altro).

Fields

public static DependencyProperty ActionRequiredProperty;	DependencyProperty per la proprietà ActionRequired.
public static DependencyProperty AssignedOnProperty;	DependencyProperty per la proprietà AssignedOn.
public static DependencyProperty ResponseByProperty;	DependencyProperty per la proprietà ResponseBy.
public static DependencyProperty SeverityProperty;	DependencyProperty per la proprietà Severity.

public static DependencyProperty SubjectProperty;	DependencyProperty per la proprietà Subject.
public static DependencyProperty UsersConnectionStringProperty;	DependencyProperty per la proprietà UsersConnectionString.
public static DependencyProperty UsersFilterProperty;	DependencyProperty per la proprietà UsersFilter.
public static DependencyProperty UsersProperty;	DependencyProperty per la proprietà Users.
public static DependencyProperty UsersTableNameProperty;	DependencyProperty per la proprietà UsersTableName.

Methods

static StateActivityBase();	Costruttore per le DependencyProperty
public StateActivityBase();	Costruttore di classe.
public static string GetUsers(string connectionString, string tableName, string filter);	Ritorna l'elenco degli utenti filtrati per FILTER.

Properties

public string ActionRequired { get; set; }	Prossima azione richiesta.
public DateTime AssignedOn { get; set; }	Data di assegnazione.
public string ResponseBy { get; set; }	Utente che ha eseguito l'azione.
public StateSeverity Severity { get; set; }	L'importanza dell'attività.
public string Subject { get; set; }	Descrizione dell'attivita.
public string Users { get; set; }	Elenco primary key degli utenti (in formato: "|pk1|pk2|pk3|").

public string UsersConnectionString { get; set; }	Connessione SQL al DB degli utenti.
public string UsersFilter { get; set; }	Filtro WHERE degli utenti a cui e' assegnata l'attività.
public string UsersTableName { get; set; }	Tabella di utenti in DB.

public class StateMachineWorkflowActivityBase

Classe base che definisce proprietà e funzioni necessarie al funzionamento di tutti i workflow. Non contiene, ne deve contenere, alcuna attività, altrimenti l'editor di visual studio va in errore generando due attività con lo stesso nome (una per il workflow di base padre e una per il workflow che lo eredita).

Methods

public StateMachineWorkflowActivityBase();	Costruttore di classe.
public virtual void Exec_SendMail(object sender, EventArgs e);	Invia una email.
public virtual List<SqlParameter> GetSqlCommandParametersFromProperties(string sqlCmd);	Converte le proprietà dell'attività in parametri da utilizzare nelle query SQL.
public virtual string GetUsers(string connectionString, string tableName, string filter);	Ritorna gli utenti in base al filtro.
public virtual DataTable GetUsersDataTable(string connectionString, string tableName, string filter);	Ritorna informazione aggiuntiva sugli utenti in base al filtro.
public virtual string ReplaceContentsValues(string sourceStr);	Esegue il replace dei contents su una stringa.
public virtual string ReplacePropertiesValues(string sourceStr);	Esegue il replace delle proprietà su una stringa.

Properties

public string ActionRequired { get; }	Azione da eseguire per parte dell'utente.
public DateTime AssignedOn { get; }	Data di assegnazione.

public ContentsCollection Contents { get; set; }	Elenco delle varibili di tipo content.
protected StateActivity CurrentState { get; }	Stato corrente dell'attività.
public string DefaultUsersCSV { get; set; }	Elenco di utenti che vedranno tutti gli stati (usata per I test).
public string ResourcesConnectionString { get; set; }	Connessione SQL al DB degli utenti.
public string ResponseBy { get; set; }	Utente che ha eseguito l'azione.
public StateActivityBase.StateSeverity Severity { get; }	Importanza dell'attività.
public string SmtpDomain { get; set; }	Domino server SMTP.
public string SmtpPasswd { get; set; }	Password utente SMTP.
public int SmtpPort { get; set; }	Porta SMTP.
public string SmtpServer { get; set; }	Indirizzo server SMTP.
public string SmtpUser { get; set; }	Utente per le credenziali del SMTP.
public string StateDescription { get; }	Descrizione dello stato corrente dell'attività.
public string StateName { get; }	Nome dello stato corrente.
public string Subject { get; }	Oggetto dell'email.
public string Users { get; }	Utenti destinatari della email.
public string UsersTableName { get; set; }	Tabella degli utenti in DB.
public Guid WFInstanceId { get; }	Id del workflow.
public string WorkflowDescription { get; }	Descrizione del workflow.

public string WorkflowName { get; }	Nome del workflow.
public string WorkflowType { get; }	Tipo del workflow.
public string XMLString { get; set; }	XML della pratica.

A.3 LIBRERIE

BigTEMSWFServices - Servizi di workflow

Questa libreria generica, indipendente dal progetto di trasporti eccezionali, è utilizzata per la gestione del versioning. Mette a disposizione un dispatcher per gestire in parallelo più runtime, utilizzando contemporaneamente workflow di versioni diverse e mantenendo le librerie dei vecchi workflow salvate in DB. Di seguito l'elenco delle classi con i rispettivi metodi:

public abstract class MarchalByRefWFRuntime

La classe astratta, utilizzata come proxy tra l'applicativo e i diversi runtime, deve essere implementata dagli oggetti passati al dispatcher (vedi TEMarchalByRefWFRuntime di WaMswf).

Methods

protected MarchalByRefWFRuntime();	Costruttore di classe da richiamare nella classe che eredita.
public Guid CreateAndStartWorkflow(Type workflowType, Dictionary<string, object> namedArgumentValues);	Crea e fa partire una nuova istanza di workflow.
public T GetService<T>();	Ritorna servizi all'interno del runtime (exception se il servizio non e' serializzabile).
public abstract void InitWFRuntime(Dictionary<string, object> wfSettings);	Inizializza il runtime per il servizio creato.
public void SetState(Guid instanceId, string targetStateName);	Cambia lo stato del workflow.
public void StartRuntime();	Inizializza il runtime per il servizio creato.
public void StopRuntime();	Ferma il runtime per il servizio creato.

public void Terminate(Guid instanceId, string error);	Termina (in maniera forzata) l'esecuzione di un workflow.

Properties

protected WorkflowRuntime WFRuntime { get; }	Ritorna il runtime.

public abstract class TrackingManager

La classe contiene le funzioni di tracking per interrogare lo stato dei workflow, implementa la gestione dei tracking profiles, e deve essere ereditata da un oggetto che definisce il tracking profile utilizzato (vedi TETrackingManager di WaMswf).

Methods

public TrackingManager(string sqlServicesConnectionString);	Costruttore di classe.
public void CheckSqlTrackingProfiles(Assembly wfAssembly);	Verifica l'esistenza in database dei tracking profile per i workflow dichiarati nella wfAssembly, creando quelli che mancano.
public abstract TrackingProfile GetTrackingProfile(Type wfType);	Genera un nuovo tracking profile per il tipo di workflow specificato.
public Version GetVersion(Assembly assembly);	Restituisce la version dell'assembly.
public Version GetVersion(Type wfType);	Restituisce la versione dell'assembly di appartenenza del wfType.
public List<TrackingExtract> GetWorkflowExtracts(Type workflowType);	Restituisce i TrackingExtras utilizzati per generare il TracklingProfile.
public DataTable GetWorkflowInstanceData(int userPK, WorkflowStatus? workflowStatus, TrackingWorkflowEvent[] trackingWorkflowEvents, string propertiesCSV, Type[] workflowTypes, Assembly assembly);	Restituisce l'elenco di workflow in esecuzione con l'elenco delle proprietà (propertiesCSV) relative allo stato corrente.

public Type[] GetWorkflowTypes(Assembly myAssembly);	Restituisce i Tipi di workflow dichiarati nell'assembly.
public IList<ActivityTrackingRecord> TrackActivityEvents(Guid InstanceId);	Restituisce la lista di ActivityTrackingEvents, dichiarati nel TrackingProfile, per l'istanza di workflow specificata.
public IList<SqlTrackingWorkflowInstance> TrackRunningWorkflows(Type[] workflowTypes);	Restituisce la lista di istanze di workflow per il tipo specificato
public IList<WorkflowTrackingRecord> TrackWorkflowEvents(Guid InstanceId);	Restituisce la lista di WorkflowTrackingEvents, dichiarati nel TrackingProfile, per l'istanza di workflow specificata.
public bool TryUpdateTrackingProfile(TrackingProfile profile, Type workflowType);	Salva sul DB il tracking profile se nessun'istanza con la stessa versione e' gia associata al tipo di workflow specificato.

Properties

public string SqlServicesConnectionString { get; set; }	Stringa di connessione al database di persistence e tracking.

public class WfAppDomainDispatcher

La classe implementa la logica della gestione di più runtime in parallelo, gestisce il salvataggio ed il caricamento delle librerie di versioni differenti, ed espone le funzioni per richiamare il runtime corretto in base all'id di un workflow già istanziato o al tipo di un workflow da creare. Utilizza come proxy un oggetto di tipo MarchalByRefWFRuntime.

Methods

public WfAppDomainDispatcher(string sqlServicesConnectionString, string versioningAppDomainsBaseDir, Dictionary<string, object> wfSettings, bool logWfAppDomainDispatcher);	Costruttore di classe
public T GetRuntime(Guid instanceId);	Ritorna il runtime che ospita il workflow specificato.

private T GetRuntime(string assemblyFullName);	Ritorna il runtime che ospita i workflow dell'assembly specificata.
public T GetRuntime(Type wfType);	Ritorna il runtime che ospita i workflow dell'assembly di appartenenza del tipo specificato.
public bool UnloadAppDomains();	Elimina dalla memoria i domini che ospitano i runtime dei workflow.
public void WriteLog(string message, params object[] details);	Aggiunge un messaggio al log automatico della classe.

Properties

public bool AllowLog { get; set; }	Abilita il log automatico della classe.

BigDBObjFuncs40– Libreria Servizi di Data Access e Sincronizzazione

Libreria utilizzata per il collegamento al DB delle pratiche e sincronizzare quello centralizzato delle anagrfiche tramite webservice.

ClientDBObject40– Libreria Servizi di Data Access e Sincronizzazione

Libreria di interfaccia per la gestione delle entity.

GeneralUtils40 – Libreria Funzioni Generali

Libreria helper per diverse funzioni del sistema.

LibPareriEsterni- Libreria Integrazione con il TE Crossing

Libreria contente l'interface per collegarsi con diversi sistemi esterni per la gestione dei pareri.

UrlAuth – Libreria Encrypting della QueryString

Libreria per garantire la sicurezza e criptare le URL delle chiamate.

A.4 AMBIENTE

Elenco componenti di terze parti che compongono l'ambiente di lavoro per le quali verrà garantito il funzionamento di nuovi rilasci.

- Infragistics Ver. 11.1 - 13.1
- Active Reports 12
- Microsoft .NET Framework 4.8
- Telerik 2014 e 2020
- Microsoft SQL Server 2008 o superiore.