# FusionPro Server Web Service API version 9.3.12

This document is a functional description of the FusionPro Server Web Service, in C# format.

## *Objects*

- Template

  Information about a template (design).
  - string Name // The name of the template.
  - string Description // User-defined description (optional).
  - string ThumbnailURL // The URL of the thumbnail image.
  - string Message // Message String.

- TemplateCollectionResponse

  Collection of Arrays related to Templates.
  - Template[] Templates // Array of Templates.
  - string[] TemplateNames // String Array of TemplateNames.
  - string[] Groups // String Array of GroupNames.
  - string Message // Message String.

- TemplateRequest

  Template information request.
  - string GroupName // Name of the group containing the template, or empty to denote the top-level group
  - string TemplateName // Name of the template.
  - FileType Type //Type of File
  - string FileName //Could be just the file name or fully qualified file name.
  - Language Language // Language to retrieve for prompts and descriptions.
  - public bool AsWSDL //True to return a WSDL (SOAP) file representing the data fields, false to return the XML generated in the HTML Form Definition dialog in FusionPro

- TemplateResponse

  Template information response.

  - string TemplatePathURLFormData //Template File Path or URL.
  - byte[] TemplateFileInBytes //TemplateFile in Bytes.
  - Stream TemplateFileStream //TemplateFile as a Stream.
  - string Message // Message String.

- TemplateFieldResponse

  TemplateField Add Information Response

  - string[] FieldNames //String Array of FieldNames.
  - FieldInfo[] FieldInfo //Array of FieldInfo Objects.
  - string Message //Message String.

- TemplateAddRequest

  Template Add Information Request.

  - string FilePath //Could be just the file name or fully qualified file name.
  - byte[] TemplateZipFile //Template ZipFile as an Array of Bytes.
  - string GroupName //Template GroupName.
  - string Description //Template Description.
  - string TemplateName //Template Name.
  - bool Overwrite //Overwrite if Template already exists.
  - string Message //Message String.

- TemplateDuplicateRequest

  Request for Duplication of an Existing Template.

  - public string GroupName //Template GroupName.
  - string TemplateName //Template Name.
  - string NewGroupName //New Template GroupName.
  - string NewTemplateName //New Template Name.

- TemplateAddResponse

  Response Information for Adding a Template.

  - string InternalTemplateName //Template name for InternalUse
  - string ActualTemplateName //The Actual Template Name
  - string Message //Message Stringpublic string GroupName

- FieldInfo

  Information about a data field.
  - string Name // The name of the data field.
  - bool IsGraphic // True if marked as a Graphic field in the Data Definition; else a Text field.
  - FieldType Type; // Field type as defined in the HTML form definition, if any.

- DataField

  A key-value pair representing an input data field.
  - string FieldName // Name of the data field.
  - string Value // Value of the data field.
  - bool IsGraphic // True if marked as a Graphic field in the Data Definition; else a Text field.

- DataRecord

  A record of input data.
  - DataField[] Fields // Array of input data.

- Preview

  A single-record Preview composition request.
  - string GroupName // Name of the group containing the template, or empty to denote the top-level group.
  - string TemplateName // Name of the template.
  - DataField[] Data // The input data.
  - bool HighRes // True to return a high-resolution image, false to return a low-res preview image.
  - bool Override // True to set high priority for this composition; false for normal priority.
  - int TimeoutSeconds // Number of seconds to wait, zero for no wait, or negative to wait indefinitely.
  - int Priority; // 0 = normal, positive number = high priority, negative number = low priority
  - int PageNumber; // Output page number to return as a JPEG; one-based, defaults to 0 to generate all pages and return page 1.

- JPEG_Options

  JPEG Output settings.
  - bool JPEGEnabled // defaults to false
  - int StartPage // one-based, defaults to 1 (0 or 1 mean first page)
  - int NumberOfPages// 0 (default): do one page; non-zero: add the page # to the JPG file name; -1: do all pages after StartPage
  - int QualityPercent // defaults to 100
  - int ResolutionDPI // defaults to 72
  - int ScalingPercent // defaults to 100
  - int MaxWidth // pixels, defaults to 0 to use original size
  - int MaxHeight // pixels, defaults to 0 to use original size
  - string PDFBoundingBox // defaults to "CropBox"
  - string BaseFileName // defaults to use PDF name with ".jpg" extension; pages numbers are added before extension

- JobOptions

  Job settings (CFG file).  Defaults to settings in the CFG file uploaded with the template.
  - OutputFormat OutputFormat
  - bool SuppressBackground

- int StartRecordNumber
- int RecordsToCompose // Zero to compose all
- bool UseImposition
- JPEG_Options JPEGOptions
- DataField[] NamedSettings // Miscellaneous job (CFG) settings.
- string sMessage // Message String

- CompositionRequest

  A multi-record composition session (job).

  - string GroupName // Name of the group containing the template, or empty to denote the top-level group.
  - string TemplateName // Name of the template.
  - DataRecord[] DataRecords // The input data records.
  - JobOptions Options // Job (CFG) settings to override from the uploaded template.
  - int Priority; // 0 = normal, positive number = high priority, negative number = low priority
  - bool OnDemand // Defines whether this is an "On Demand" composition or not.
  - OnDemandComponents OnDemandComp // The necessary component information to run an OnDemand Composition.

- CompositionStatus

  Information about a composition's (job's) status.

  - JobStatus JobStatus // Job status.
  - int PercentComplete // from 0 to 100 - valid only for multi-record compositions
  - int ReturnCode // FusionPro return code - valid only if Status is Done. Zero denotes success.
  - int TotalRecords // total number of records composed
  - int TotalPages // total number of pages composed, in all records
  - string WorkingFolder; // working folder for current composition
  - string sMessage // Message String

- CompositionInfo

  Information about a composition session.

  - string AssignedServer = ""; // URI of Server Assigned to Job
  - string CFGFile = ""; // CFGFile.
  - string GroupName // Name of the group containing the template, or empty to denote the top-level group.
  - int ID // Composition identifier.
  - JobOptions Options// Job (CFG) settings to override from the uploaded template.
  - int Priority // 0 = normal, positive number = high priority, negative number = low priority
  - int ProcessID = 0; // from 0 to 100 - valid only for multi-record compositions
  - CompositionStatus Status // Job status
  - string TemplateName // Name of the template.
  - string Message // Message String.
  - bool OnDemand // Defines whether this is an "On Demand" composition or not.
  - OnDemandComponents OnDemandComp // The necessary component information to run an OnDemand Composition.

- StartCompositionRequest

  Information about a composition start request.

  - string WorkingFolder; // working folder for current composition
  - int CompositionID // Composition identifier.
  - bool requeue // for requeue requests.

  - int Priority // 0 = normal, positive number = high priority, negative number = low priority

- AddCompositionComponentRequest

  Information about a component being added.

  int CompositionID // Composition identifier.
  - FileType Type; // Type of File.
  - bool KeepFileName; // Keep the file name passed.
  - string FileContentsAsString // Pass the file in as a String.
  - byte[] FileContentsInBytes // Pass the file in as an array of bytes.
  - FPCompositionComponentType ComponentType //Sets how the file is passed in.

- AddCompositionComponentResponse

  Any messages about a component being added.
  - string Message // Message String.

- CompositionFilePathURLRequest

  Information used for a Composition request.
  - int CompositionID // Composition identifier.
  - FileType Type // Type of File.
  - int PageOrChunkNumber // Page or Chunk Number.
  - string FileName // FileName.

- CompositionFilePathURLResponse

  Response Composition File information for a requested Composition.
  - String Output // The File Path or URL returned.
  - byte [] OutputFileInBytes // The file returned as a byte array.
  - string Message // Message String.

- OnDemandComponents

  On Demand Composition (job) components.
  - string InstanceFile // Instance/Data File.
  - string ConfigFile // CFG File.
  - string OutputFile // Output File.
  - int CompositionID // Composition Identifier.

- CompositionSessionRequest

  Information about a composition session.
  - string GroupName// Group Name.
  - string TemplateName// Template Name.
  - bool ActiveOnly // ActiveOnly sessions.
  - int NumRecent// Number of recent sessions to return.

## *Constants*

```
enum FileType
{
        Template, // Collected Zip file
        Thumbnail, // Template thumbnail (hint) graphic
        InputData, // Either flat (delimited) or tagged markup
        Format, // DIF (Layout)
        DataDefinition, // DEF (Fields, Rules, Resources, etc.)
```

```
  HTMLFormDefinition, // Web Collect XML file
        FontAssets, // Collected fonts (Assets.dat)
        JobSettings, // CFG (Job settings)
        Output, // Composed output, in any valid format
        JPEGOutput, // Composed PDF output, rasterized to JPEG
        Log, // Composition log (.msg) file
        XMLLog, // XML Metadata log
}

enum OutputFormat
{
        PostScript,
        PDF,
        AFP,
        VDX,
        Digimaster_PS,
        PPML,
        HP_PPML,
        JLYT,
        MultiFile_VPS,
        SingleFile_VPS,
        VIPP,
        AFP,
}

enum JobStatus
{
  None, // the composition has not been started or has been deleted
  InProcess // the composition is still in progress
  Queueing, // the composition is pending, waiting to be run
  Cancelled, // the composition was cancelled
  Failed, // the composition could not be run
  Success, // the composition succeeded and output is ready
        // can also be a FusionPro Server error code
}

enum FieldType
{
   None,
   SingleLine,
   MultiLine,
   Gender,
   YesNo,
   PickList,
   Child,
   GraphicUpload,
   GraphicPicklist,
   Separator,
}

enum FPCompositionComponentType
{
   FromString, // File component passed in as a string
   FromFile, // File component passed in as a path/url
   FromBytes // File component passed in as a byte array
}
```

## *Methods*

### Group and Template info

- ListGroups: TemplateCollectionResponse ListGroups()
  Lists the groups.  May include an empty name.
  - Returns: TemplateCollectionResponse object
      - Groups: Array of group names.

- ListTemplates: TemplateCollectionResponse ListTemplates(TemplateRequest req)
  Lists the templates in a group.  See also GetTemplates.
  - TemplateRequest object
    - GroupName: The name of the group whose templates to retrieve.  Can be empty to return the top-level list of templates.
  - Returns: TemplateCollectionResponse object
      - TemplateNames: Array of template names, or null if the specified group is not found.
      - Message: An Empty String if no errors occurred.

- GetTemplateInfo: Template GetTemplateInfo(TemplateRequest req)
  Returns information about a template.
  - TemplateRequest object
    - GroupName: The name of the group containing the Template.  Can be empty to return a template in the top-level group.
  - TemplateName: The name of the template.Returns: Template object.
        - Message: An Empty String if no errors occurred.

- GetTemplateJobOptions: JobOptions GetTemplateJobOptions(TemplateRequest req)
  Returns the default job options (CFG file settings) uploaded with a template.
  - TemplateRequest object
    - GroupName: The name of the group containing the Template.  Can be empty to return a template in the top-level group.
  - TemplateName: The name of the template.
  - Returns: a JobOptions object
      - Message: An Empty String if no errors occurred.

- GetTemplates: TemplateCollectionResponse GetTemplates(TemplateRequest req)
  Lists the templates in a group, with all the information about each template.  Can be used in place of ListTemplates and GetTemplateInfo.
  - TemplateRequest object
    - GroupName: The name of the group whose templates to retrieve.  Can be empty to return the top-level list of emplates.
  - Returns: TemplateCollectionResponse object
      - Array of Template objects.
      - Message: An Empty String if no errors occurred.

- GetTemplateFileURL: TemplateResponseGetTemplateFileURL(TemplateRequest req)

  Returns the URL of a file associated with the template.

  - TemplateResponse object
    - GroupName: The name of the group containing the Template. Can be empty to return a template in the top-level group.
    - TemplateName: The Name of the Template.
    - Type: The type of file to retrieve. Must be one of:
      - Template, // Collected Zip file
      - Thumbnail, // Template thumbnail (hint) graphic
      - InputData, // Either flat (delimited) or tagged markup
      - Format, // DIF (Layout)
      - DataDefinition, // DEF (Fields, Rules, Resources, etc.)
      - HTMLFormDefinition, // Web Collect XML file
      - FontAssets, // Collected fonts (Assets.dat)
  - Returns: TemplateResponse object
      - TemplatePathURL..
      - Message: An Empty String if no errors occurred.

- GetTemplateFilePath: TemplateResponse GetTemplateFilePath(TemplateRequest req)

  Returns the path of a file associated with the template. Same as GetTemplateFileURL, but returns a TemplateResponse object with a file path instead of a URL located in the TemplatePathURL property. Useful only within a LAN environment.

- GetTemplateFileBytes: TemplateResponse GetTemplateFileBytes(TemplateRequest req)

  Retrieves a file associated with the template, as a stream of data. Same as GetTemplateFileURL and GetTemplateFilePath, but returns a TemplateResponse object with a stream of data located in the TemplateFileStream property.

- GetTemplateResourceURL: TemplateResponse GetTemplateResourceURL(TemplateRequest req)

  Returns the URL of a generic resource file associated with the Template.

  - TemplateRequest object
  - GroupName: The name of the group containing the Template. Can be empty to return a template in the top-level group.
  - TemplateName: The Name of the Template.
  - FileName: The name of file to retrieve.
  - ErrorMessage: Returns a description of any error that occurred.
  - Returns: TemplateResponse
      - TemplatePathURL.
      - Message: An Empty String if no errors occurred.

- GetTemplateResourceFilePath: TemplateResponse GetTemplateResourceFilePath(TemplateRequest req)

  Returns the path of a generic resource file associated with the Template. Same as GetTemplateResourceURL, but returns a TemplateResponse object with a file path in the TemplatePathURL member instead of a URL. Useful only within a LAN environment.

- GetTemplateResourceBytes: TemplateResponse GetTemplateResourceBytes(TemplateRequest req)

  Retrieves a generic resource file associated with the template, as a stream of data. Same as GetTemplateResourceURL and GetTemplateResourceFilePath, but returns a TemplateResponse object with a stream of data located in the TemplateFileStream member.

- GetTemplateFormData: TemplateResponse GetTemplateFormData(TemplateRequest req)

  Gets XML markup representing a list of data fields for the template.
  - TemplateRequest object
    - GroupName: The name of the group containing the Template. Can be empty to return a template in the top-level group.
    - TemplateName: The Name of the Template.
    - AsWSDL: True to return a WSDL (SOAP) file representing the data fields, false to return the XML generated in the HTML Form Definition dialog in FusionPro.
  - Returns: TemplateResponse object
    - TemplatePathURLFormData: The XML..
    - Message: An Empty String if no errors occurred.

- GetTemplateFieldNames: TemplateFieldResponse GetTemplateFieldNames(FieldInfoRequest req)

  Gets a list of data fields for the template.
  - FieldInfoRequest object
    - GroupName: The name of the group containing the Template. Can be empty to return a template in the top-level group.
    - TemplateName: The Name of the Template.
  - Returns: TemplateFieldResponse object
    - FieldNames Array with the names of the field, or null if an error occurred..
    - Message: An Empty String if no errors occurred.

- GetTemplateFields: TemplateFieldResponse GetTemplateFields(FieldInfoRequest req)

  Gets information about data fields in the template.
  - FieldInfoRequest object
    - GroupName: The name of the group containing the Template. Can be empty to return a template in the top-level group.
    - TemplateName: The Name of the Template.
  - Returns: TemplateFieldResponse object
    - FieldInfo Array of FieldInfo, or null if an error occurred.
    - Message: An Empty String if no errors occurred.

## Group and Template management

- AddTemplateFromFile: TemplateAddResponse AddTemplateFromFile(TemplateAddRequest req)

  Adds a template to the repository from a local file. Useful only within a LAN environment, or if the file has already been uploaded to the Web Service machine via a separate Web application.
  - TemplateAddRequest object
    - FilePath: Full path and file name of the template Zip file.
    - GroupName: The name of the group to which to add the Template. Can be empty to denote the top-level group.
    - Description: User-defined description (optional).
    - TemplateName: The Name for the Template. If empty, the template file name (without the extension) is used.
    - Overwrite: True to overwrite an existing template with the same name; if False and a template with the same name exists, the newly added template will be given a different name.
  - Returns: TemplateAddResponse Object
    - Message: An empty string if the template was successfully added, or a description of any error that occurred.
    - InternalTemplateName: The name of the template file, without the extension.

        - ActualTemplateName: The ID of the newly added template.

- AddTemplateFromBytes: TemplateAddResponse AddTemplateFromBytes(TemplateAddRequest req)
  Adds a template to the repository. Same as AddTemplateFromFile, but takes a stream of data in the TemplateZipFile member of the TemplateAddRequest object instead of a local file name. May be subject to file upload size limits.

- RenameTemplate: TemplateResponse RenameTemplate(TemplateDuplicateRequest req)
  Renames a template in the repository.
    - TemplateResponse object
        - GroupName: The name of the group containing the Template. Can be empty to denote the top-level group.
        - OldTemplateName: The existing Name of the Template.
        - NewTemplateName: The new Name for the Template.
    - Returns: TemplateResponse
        - Message: An empty string if the template was successfully added, or a description of any error that occurred.

- DuplicateTemplate: TemplateAddResponse DuplicateTemplate(TemplateDuplicateRequest req)
  Makes a copy (duplicate) of a template.
    - TemplateDuplicateRequest object
        - GroupName: The name of the group containing the Template. Can be empty to denote the top-level group.
        - TemplateName: The Name of the Template to duplicate.
        - NewGroupName: The group into which to add the duplicate Template. Can be empty to denote the top-level group.
        - NewTemplateName: The new Name for the Template.
    - Returns: TemplateAddResponse object
        - Message: An empty string if the template was successfully duplicated, or a description of any error that occurred.
        - InternalTemplateName: Returns the name of the template file, without the extension.
        - ActualTemplateName: Returns the ID of the newly added template.

- DeleteTemplate: TemplateResponse DeleteTemplate(TemplateRequest req)
  Removes a template, and any composition sessions and generated output.
    - TemplateRequest object
        - GroupName: The name of the group containing the Template. Can be empty to denote the top-level group.
        - TemplateName: The Name of the Template to remove.
    - Returns:
            - An empty string if the template was successfully duplicated, or a description of any error that occurred.

- DeleteGroup: TemplateResponse DeleteGroup(TemplateRequest req)
  Removes an group from the repository. The group must be empty (i.e. containing no Templates).
    - TemplateRequest object
        - GroupName: The name of the group to delete. Cannot be empty.
    - Returns: TemplateResponse object
            - An empty string if the group was successfully duplicated, or a description of any error that occurred.

## Single record (preview) composition

- GetPreviewJPEG_URL: CompositionFilePathURLResponse GetPreviewJPEG_URL(Preview Request)
  Runs a single-record preview composition and creates JPEG output, returned as a URL.
  - Request: A Preview object.
  - CompositionID: On input, optional: if non-zero, specifies an existing session; if zero, creates a new composition session and returns the composition (session) ID.
  - ErrorMessage: Returns a description of any error that occurred.
  - Returns: A CompositionFilePathURLResponse object which has the URL of the image as "Output", if it was created. May be empty if an error occurred.

- GetPreviewJPEG_Path: CompositionFilePathURLResponse GetPreviewJPEG_Path(Preview Request)
  Runs a single-record preview composition and creates JPEG output, returned as a file path. Same as GetPreviewJPEG_URL, but returns a A CompositionFilePathURLResponse object with a file path instead of a URL as "Output". Useful only within a LAN environment.

- GetPreviewJPEG_Bytes: CompositionFilePathURLResponse GetPreviewJPEG_Bytes(Preview Request)
  Runs a single-record preview composition and creates JPEG output, returned as bytes. Same as GetPreviewJPEG_URL and GetPreviewJPEG_Path, but returns CompositionFilePathURLResponse object with a stream of data as "OutputFileInBytes".

## Multiple record (batch) composition

- CreateCompositionSession: CreateCompositionResponse CreateCompositionSession(Composition Request)
  Creates a batch composition session. Does not actually start the composition; see StartCompositionFromSession.
  - Request: A Composition object.
  - Returns a CreateCompositionResponse object with a "Message" which has the description of any error that occurred.
  - Returns: A CreateCompositionResponse object which contains a unique composition session ID (token). The "Message" will have the description of any error that occurred.

- StartCompositionFromSession: StartCompositionResponse StartCompositionFromSession(StartCompositionRequest Request)
  Starts a batch composition. CreateCompositionSession must have already been called.
  - CompositionID: A StartCompositionResponse object which contains a composition (session) ID, returned from CreateCompositionSession.
  - Returns: A StartCompositionResponse object which contains an empty string if composition was successfully started, or a description of any error that occurred.

- StartComposition: StartCompositionResponse StartComposition(Composition Request)
  Creates and starts a batch composition. (Same as calling CreateCompositionSession and StartCompositionFromSession, with no additions to or overrides of template files.)
  - Request: A Composition object.
  - ErrorMessage: A StartCompositionResponse object which contains a description of any error that occurred.
  - Returns: A StartCompositionResponse object
    - CompositionID: Unique composition session ID (token).

- AddCompositionFile: AddCompositionComponentResponse AddCompositionFile(CompositionFilePathURLResponse request)

  Adds a file specific to a composition session, from a local file. If the same type of file exists in the template, the new file overrides it for this session only. Useful only within a LAN environment, or if the file has already been uploaded to the Web Service machine via a separate Web application.

  - AddCompositionComponentRequest
    - CompositionID: The composition ID.
    - Type: The type of file being added. Must be one of:
      - InputData, // Either flat (delimited) or tagged markup
      - Format, // DIF (Layout)
      - DataDefinition, // DEF (Fields, Rules, Resources, etc.)
      - JobSettings, // CFG (Job settings)
      - FontAssets, // Collected fonts (Assets.dat)
    - RemoteFilePath : The full path of the file. Used for FromFile ComponentType.
    - FileContentsInBytes // For the ComponentType FromBytes, file passed in as a byte array. (Valid for both text and binary files. May be subject to file upload size limits.)
    - KeepFileName // Keep the filename
    - FileContentsAsString // For the FromString, file passed in as a string (Valid only for text (non-binary) files. May be subject to file upload size limits.)
    - ComponentType // FromString, FromFile, FromBytes defines. what is passed in
  - Returns: A AddCompositionComponentResponse object which contains an empty string if the file was successfully added, or a description of any error that occurred.

- AddCompositionResource: AddCompositionComponentResponseAddCompositionResource (AddCompositionComponentRequest request)

  Adds a generic resource file to the composition session. If a file with the same name exists in the template, the new file overrides it for this session only. Useful only within a LAN environment or if the file has already been uploaded to the Web Service machine via a separate Web application.

  - AddCompositionComponentRequest
    - CompositionID: The composition ID.
    - Type: The type of file being added. Must be one of:
      - InputData, // Either flat (delimited) or tagged markup
      - Format, // DIF (Layout)
      - DataDefinition, // DEF (Fields, Rules, Resources, etc.)
      - JobSettings, // CFG (Job settings)
      - FontAssets, // Collected fonts (Assets.dat)
    - RemoteFilePath : The full path of the file. Used for FromFile ComponentType.
    - FileContentsInBytes // For the ComponentType FromBytes, file passed in as a byte array. (Valid for both text and binary files. May be subject to file upload size limits.)
    - KeepFileName // Keep the filename
    - FileContentsAsString // For the FromString, file passed in as a string (Valid only for text (non-binary) files. May be subject to file upload size limits.)
    - ComponentType // FromString, FromFile, FromBytes defines. what is passed in
  - Returns: A AddCompositionComponentResponse object which contains an empty string if the file was successfully added, or a description of any error that occurred.

- CheckCompositionStatus: CompositionStatus CheckCompositionStatus(CompositionFilePathURLRequest request)

  Checks the status of a previously submitted batch composition.

  - CompositionFilePathURLRequest
    - CompositionID: The composition ID.

    - Returns: A CompositionStatus Object which contains all of the information available for the requested Composition.

- GetCompositionFileURL: string GetCompositionFileURL(CompositionFilePathURLRequest request)
  Returns the URL of a file associated with the composition session or its template.
    - CompositionID: The composition ID.
    - Type: The type of file to retrieve.  Must be one of:
        - Template, // Collected Zip file
        - Thumbnail, // Template thumbnail (hint) graphic
        - InputData, // Either flat (delimited) or tagged markup
        - Format, // DIF (Layout)
        - DataDefinition, // DEF (Fields, Rules, Resources, etc.)
        - HTMLFormDefinition, // Web Collect XML file
        - JobSettings, // CFG (Job settings)
        - Output, // Composed output, in any valid format
        - JPEGOutput, // Composed PDF output, rasterized to JPEG
        - Log, // Composition log (.msg) file
        - XMLLog, // XML Metadata log
    - ErrorMessage: Returns a description of any error that occurred.
    - Returns: The URL.

- GetCompositionFilePath: CompositionFilePathURLResponse GetCompositionFilePath(CompositionFilePathURLRequest request)
  Returns the path of a file associated with the composition session or its template.  Same as GetCompositionFileURL, but returns a file path instead of a URL.  Useful only within a LAN environment.

- GetCompositionFileBytes: CompositionFilePathURLResponse GetCompositionFileBytes(CompositionFilePathURLRequest request)
  Retrieves a file associated with the composition session or its template, as a stream of data.  Same as GetCompositionFileURL and GetCompositionFilePath, but returns a stream of data.

- GetCompositionResourceURL: CompositionFilePathURLResponse GetCompositionResourceURL(CompositionFilePathURLRequest request)
  Returns a CompositionFilePathURLResponse object with the URL of a generic resource file associated with the composition session or its template.
    - CompositionFilePathURLRequest
        - CompositionID: The composition ID.
        - FileName: The name of file to retrieve.
        - 
    - Returns: A CompositionFilePathURLResponse object which contains the URL in “Output”.

- GetCompositionResourceFilePath: CompositionFilePathURLResponse GetCompositionResourceFilePath(CompositionFilePathURLRequest request)
  Returns the path of a generic resource file associated with the composition session or its template. Same as GetCompositionResourceURL, but returns a CompositionFilePathURLResponse object which contains the file path instead of a URL in “Output”. Useful only within a LAN environment.

- GetCompositionResourceBytes: CompositionFilePathURLResponse GetCompositionResourceBytes(CompositionFilePathURLRequest request)

Retrieves a generic resource file associated with the composition session or its template, as a stream of data. Same as GetCompositionResourceURL and GetCompositionResourceFilePath, but returns a CompositionFilePathURLResponse object which contains the stream of data in "OutputFileInBytes".

- CancelComposition: CompositionFilePathURLResponse CancelComposition(CompositionFilePathURLRequest request)
  Cancels an active composition session.
  - CompositionFilePathURLRequest
    - CompositionID: The composition ID.
  - Returns: A CompositionFilePathURLResponse object with an empty "Message" string if the composition was successfully cancelled, or a description of any error that occurred.

- EndCompositionSession: CompositionFilePathURLResponse EndCompositionSession(CompositionFilePathURLRequest request)
  Ends a composition session, deleting all sesssion files except the generated output.
  - CompositionFilePathURLRequest
    - CompositionID: The composition ID.
  - Returns: A CompositionFilePathURLResponse object with a n empty "Message" string if the session was successfully ended, or a description of any error that occurred.

- DeleteCompositionSession: CompositionFilePathURLResponse DeleteCompositionSession(CompositionFilePathURLRequest request)
  Removes a composition session, and any generated output.
  - CompositionFilePathURLRequest
    - CompositionID: The composition ID.
  - Returns: A CompositionFilePathURLResponse object with a n empty "Message" string if the session was successfully deleted, or a description of any error that occurred.

- ListCompositionSessions: int[] ListCompositionSessions(CompositionSessionRequest
- request)
  Lists the composition sessions for a template.
  - CompositionSessionRequest
    - GroupName: The name of the group containing the Template. Can be empty to return a template in the top-level group.
    - TemplateName: The Name of the Template.
    - ActiveOnly: True to return only queueing, active, and recently finished jobs.
    - NumRecent: The number of recently finished jobs to return if ActiveOnly is true.
  - Returns: Array of composition session IDs.

## GetCompositionSessions

## CompositionInfo[] GetCompositionSessions(CompositionSessionRequest request)

Retrives information about the composition sessions for a template.
Input:

- CompositionSessionRequest
  - GroupName: The name of the group containing the Template. Can be empty to return a template in the top-level group.

- TemplateName: The Name of the Template. Can be empty to return compositions from all templates.
- ActiveOnly: True to return only queueing, active, and recently finished jobs.
- NumRecent: The number of recently finished jobs to return if ActiveOnly is true.

Returns:

- An array of CompositionInfo objects. May be subject to message size limits.

Notes: The input parameters listed above are used together to form the filter. As noted, an empty field will return the results for All jobs for the unspecified parameter.

## Setup/diagnostics

These functions can be used to examine the configuration of the Web Service.

- GetAssemblyVersion: string GetAssemblyVersion()
  Returns the assembly version of the Web Service, such as "7.1".

- GetProductVersion: string GetProductVersion()
  Returns the full product version of the Web Service, such as "7.1P1c".

- IsCompatibleVersion: bool IsCompatibleVersion(int ClientVersionMajor, int ClientVersionMinor)
  Used to determine whether the Web Service is compatible with a specific version of the FusionPro composition engine.
  - ClientVersionMajor: Major version number.
  - ClientVersionMinor: Minor version number.
  - Returns: True if the Web Service is compatible, false if not.

- GetUser: string GetUser()
  Returns the full user name (domain\user) of the account under which the Web Service is running.

- GetConfigurationError: string GetConfigurationError()
  Returns an error message if the Web Service is not configured correctly (for certain error conditions). Returns an empty string if no configuration errors are detected; note that this does not necessarily mean that everything is set up correctly to be able to compose output.

## Layout Modification System

public **bool** **Format_AddGraphicFrame**(out **string** *ErrorMessage*, **int** *CompositionID*, **FPWebServicesDemo.FusionProAPI.GraphicFrame** *graphicFrame*)
  Member of **FPWebServicesDemo.FusionProAPI**.**FPQueueWCFServiceClient**

public bool **Format_AddTextFrame**(out string *ErrorMessage*, int *CompositionID*, FPWebServicesDemo.FusionProAPI.TextFrame *textFrame*)
Member of FPWebServicesDemo.FusionProAPI.FPQueueWCFServiceClient

public bool **Format_DeleteGraphicFrame**(out string *ErrorMessage*, int *CompositionID*, string *name*)
Member of FPWebServicesDemo.FusionProAPI.FPQueueWCFServiceClient

public bool **Format_DeleteTextFrame**(out string *ErrorMessage*, int *CompositionID*, string *name*)
Member of FPWebServicesDemo.FusionProAPI.FPQueueWCFServiceClient

public string[] **Format_GetAllGraphicFrameNames**(out string *ErrorMessage*, int *CompositionID*)
Member of FPWebServicesDemo.FusionProAPI.FPQueueWCFServiceClient

public FPWebServicesDemo.FusionProAPI.GraphicFrame[] **Format_GetAllGraphicFrames**(out string *ErrorMessage*, int *CompositionID*)
Member of FPWebServicesDemo.FusionProAPI.FPQueueWCFServiceClient

public string[] **Format_GetAllTextFrameNames**(out string *ErrorMessage*, int *CompositionID*)
Member of FPWebServicesDemo.FusionProAPI.FPQueueWCFServiceClient

public FPWebServicesDemo.FusionProAPI.TextFrame[] **Format_GetAllTextFrames**(out string *ErrorMessage*, int *CompositionID*)
Member of FPWebServicesDemo.FusionProAPI.FPQueueWCFServiceClient

public FPWebServicesDemo.FusionProAPI.GraphicFrame **Format_GetGraphicFrame**(out string *ErrorMessage*, int *CompositionID*, string *name*)
Member of FPWebServicesDemo.FusionProAPI.FPQueueWCFServiceClient

public string[] **Format_GetGraphicFrameNamesOnPage**(out string *ErrorMessage*, int *CompositionID*, int *Page*)
Member of FPWebServicesDemo.FusionProAPI.FPQueueWCFServiceClient

public FPWebServicesDemo.FusionProAPI.GraphicFrame[] **Format_GetGraphicFramesOnPage**(out string *ErrorMessage*, int *CompositionID*, int *Page*)
Member of FPWebServicesDemo.FusionProAPI.FPQueueWCFServiceClient

public int **Format_GetNumberOfBodyPages**(out string *ErrorMessage*, int *CompositionID*)
Member of FPWebServicesDemo.FusionProAPI.FPQueueWCFServiceClient

public **FPWebServicesDemo.FusionProAPI.TextFrame** **Format_GetTextFrame**(out **string** *ErrorMessage*, **int** *CompositionID*, **string** *name*)
  Member of **FPWebServicesDemo.FusionProAPI**.**FPQueueWCFServiceClient**

public **string[]** **Format_GetTextFrameNamesOnPage**(out **string** *ErrorMessage*, **int** *CompositionID*, **int** *Page*)
  Member of **FPWebServicesDemo.FusionProAPI**.**FPQueueWCFServiceClient**

public **FPWebServicesDemo.FusionProAPI.TextFrame[]** **Format_GetTextFramesOnPage**(out **string** *ErrorMessage*, **int** *CompositionID*, **int** *Page*)
  Member of **FPWebServicesDemo.FusionProAPI**.**FPQueueWCFServiceClient**

public **bool** **Format_UpdateGraphicFrame**(out **string** *ErrorMessage*, **int** *CompositionID*, **FPWebServicesDemo.FusionProAPI.GraphicFrame** *graphicFrame*)
  Member of **FPWebServicesDemo.FusionProAPI**.**FPQueueWCFServiceClient**