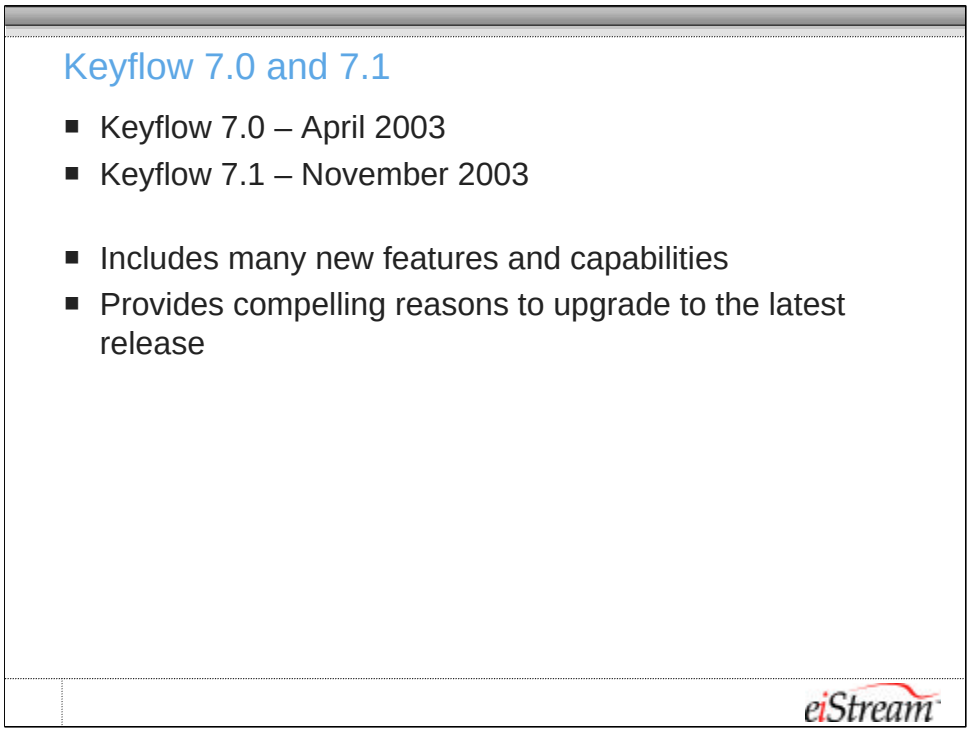
The following presentation is divided into two parts.

First we'll look back at the latest releases of Keyflow … versions 7.0 and 7.1. We will review the major new features and improvements made to the product, providing compelling reasons, for those customers who have not yet done so, to upgrade to the latest release.

Then, we'll look ahead, presenting the strategy going forward … the timeframe and features that are planned for future Keyflow releases.

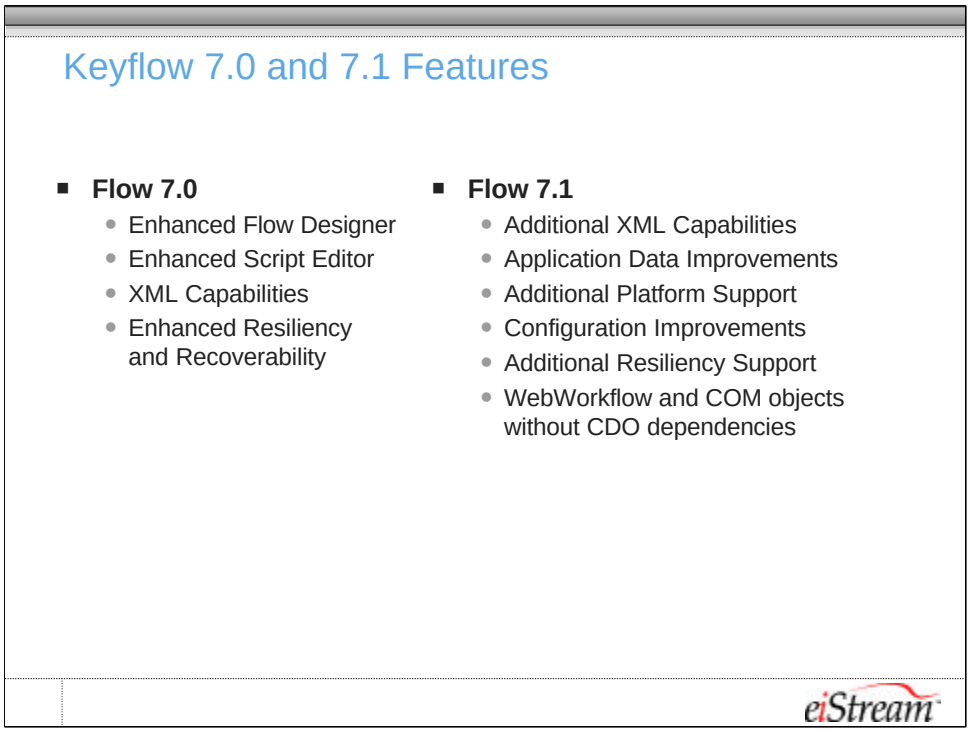


Looking back …

Version 7.0 was released in April of 2003.

Version 7.1 was released in November of 2003.

Each of these releases included many new features and capabilities.



Those features and improvements, as shown here, span such areas as:

•Graphical UI improvements,

•Extensive XML capabilities,

•Resiliency and recoverability issues,

•Application data improvements,

•Platform support,

•Configuration Improvements,

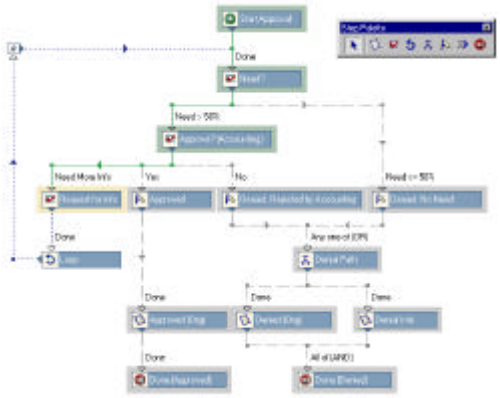
•And improvements regarding dependencies on CDO.

We'll take a look at each of these items separately …

We'll start first with the Flow 7.0 Features

In Flow 7.0, the Flow Designer underwent a significant facelift, making it more aesthetically appealing, and adding many new features that provide more comprehensive means of viewing templates and flows, and that make it easier to navigate and get at information.

Flow Designer Graphical Elements



- Modernized color schemes
- Redesigned line styles
- Redesigned icons

eiStream

Let's start with the numerous graphical improvements included in Flow 7.0. These include:

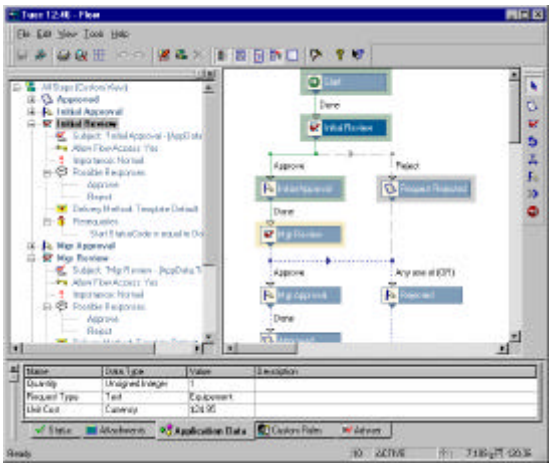
•Modernized color schemes for step graphics that appear in the flow composition pane, when designing templates or viewing running flows.

•Redesigned colors and line styles that connect steps in the flow composition pane, particularly to distinguish different paths taken, or not taken, when viewing a running flow.

•Redesigned icons that are used on the toolbar, in the step palette, in the flow composition pane, and in other various places within the Flow Designer.

These graphical improvements dramatically improve the appearance of the Flow Designer.

Step List Pane, Tree View



- Separate pane
- Tree view of all step properties
- Selection tracking between Step List and Flow Composition area

eiStream

A new Step List windowpane is added in Flow 7.0, where all of the steps, and their properties, can be viewed in one hierarchical tree display. This makes viewing information about the flow, navigating to various locations in large flows and modifying the information, easier and more flexible.

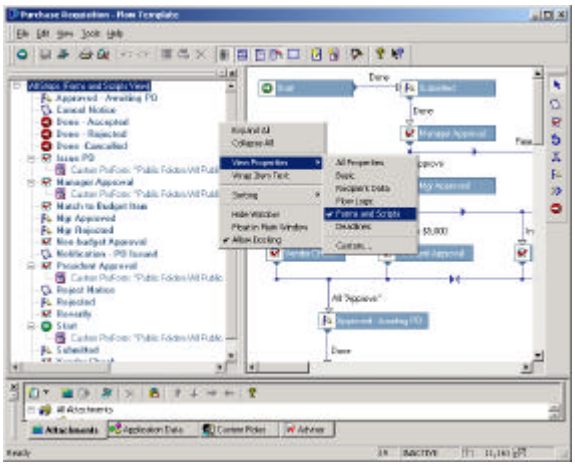
The Step List Pane includes the following features:

•Expanding and collapsing individual steps, or all steps, to view or hide particular information in various parts of the hierarchical tree display.

•Editing step properties, by accessing the step properties dialog directly from any step or property in the Step List.

•Tracking selection between the Step List and the flow composition area, so that selecting a step in either the Step List or in the composition area, scrolls to, and selects, the corresponding step in the other pane.

Step List Pane, Views



- Views – allow you to include only a subset of properties for each step

eiStream

"Views" in the Step List can be used to include only a subset of step properties within each step. As can be seen here, by the selections available in the drop down menus, a view selection can include:

– All properties of the step

– Basic properties (which provides a view of Subjects, Recipients, due dates and alerts, for FYI and Response steps)

– Recipient Data (which shows recipients, due dates, delivery methods, importance settings, and flow access settings)

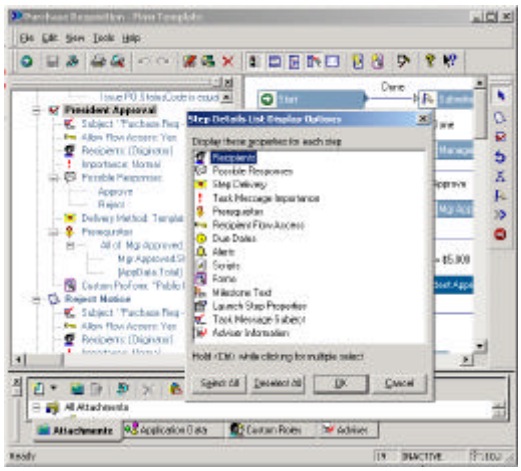
– Flow Logic (which shows the possible responses, and the prerequisite conditions, for each step)

– Forms and Scripts (This is the example shown in the image on this slide – which shows all of the links to separately stored forms and scripts)

– Another "view" selection, is … Deadlines (which shows due dates and alerts)

– And the last one is a Custom view … (next slide)

Step List Pane, Custom View

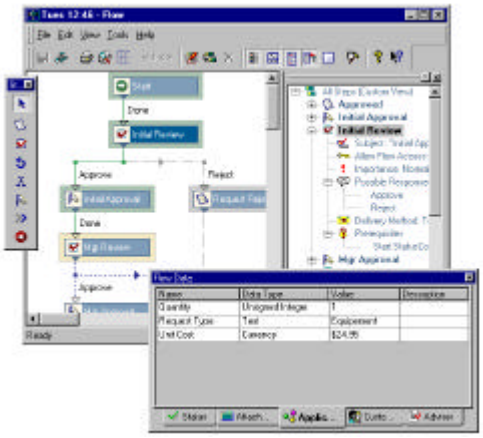


- Allows you to individually select which properties to view

The Custom view allows you to individually select which properties to include in the view.

(pause … to allow audience to peruse the selection list … then on to the next slide)

Floating and Docking Windows



- Free floating
- Or, docked along any border
  - Flow Data pane
  - Step List pane
  - Menus
  - Toolbar
  - Step Palette

eiStream

In Flow 7.0, windowpanes and toolbars are fully set up as floating or docking windows. Each windowpane can be free floating. Or, each windowpane can be docked along any edge of the Flow Designer window. This includes:

•the Flow Data pane (showing Attachments, Application Data, Custom Roles, and the Adviser)

•the Step List pane

•the Menus

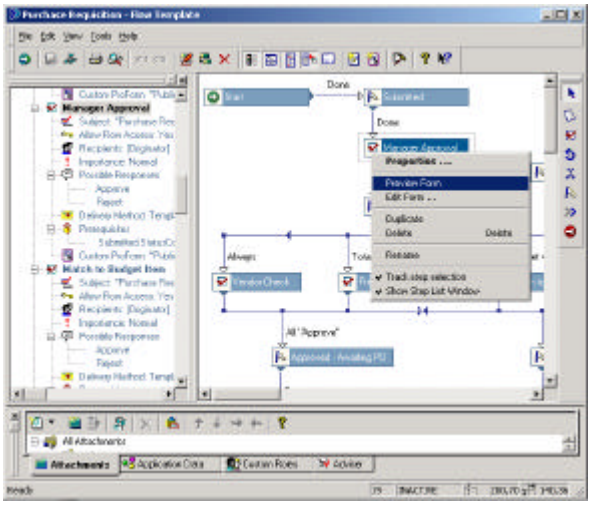
•the Toolbar

•and the Step Palette

This gives tremendous flexibility in how one organizes their design environment, and their design tools.

In addition, new customization facilities are included, to change any toolbar, or to create new custom toolbars with buttons organized in any way desired. These custom toolbars can also be free floating, or docked within the main window.

Form Preview (ProForms)



- Preview Form, for any step

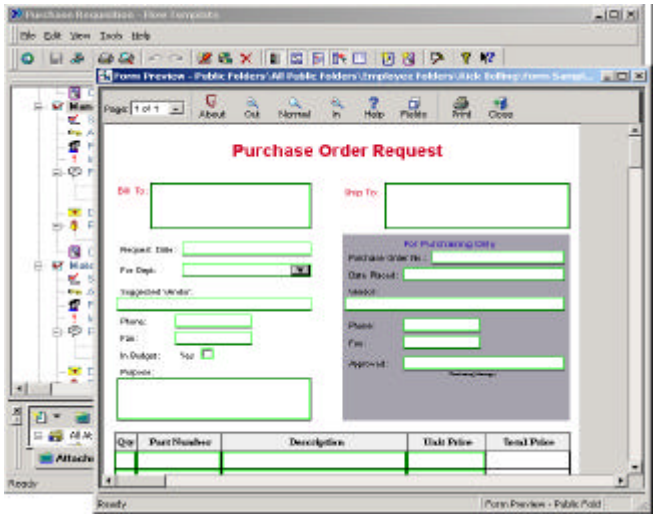
eiStream

Prior to version 7, the only way to view a ProForm, as the task recipient would actually see it, was to initiate a sample flow and observe the form in the context of a running flow. In most circumstances, this could be awkward.

With version 7.0, the Flow Designer introduces the ability to preview the ProForm associated with a step directly.

The context menu for a step allows you to Preview the form …. (next slide)

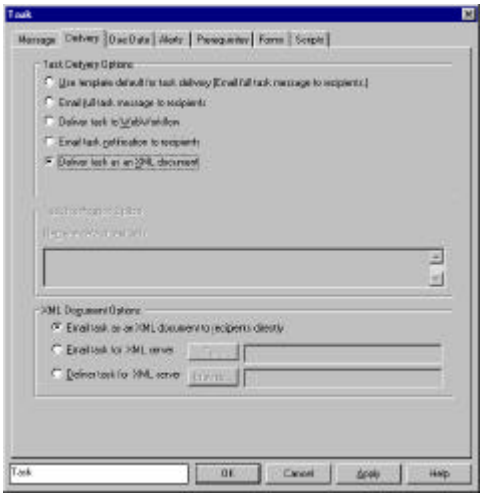
Form Preview (ProForms)



eiStream

Here is a Form Preview for a step. All elements of the form are active, including form event scripts, so all aspects of the form can be tested.

Delivery Options

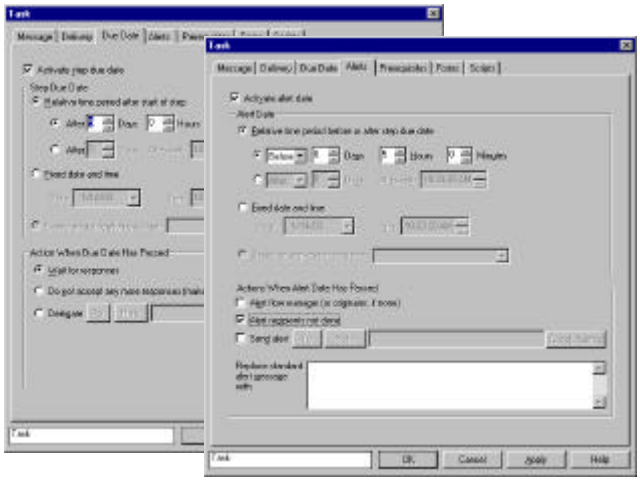


- New tab in Step Properties dialog
- New XML delivery options

eiStream

With the addition of new delivery options in Flow7.0, the step properties dialog has been reorganized to accommodate the more comprehensive selections. The delivery options are presented in a separate tab, to allow better presentation of those selections. The old selections, including full thick client task delivery, and webworkflow delivery options are retained … but notice also the new XML delivery options … these will be explained later.

Due Dates and Alerts



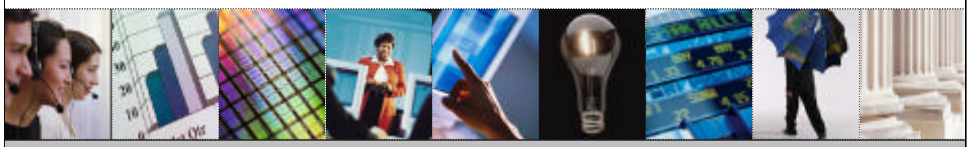
- Separate tabs
- Easier to understand

eiStream

With Flow 7.0, the Due Date and Alert options are separated onto different step dialog tabs, and given a new layout, to make it much easier to understand.

eiStream™

Streamlining Business Processes™

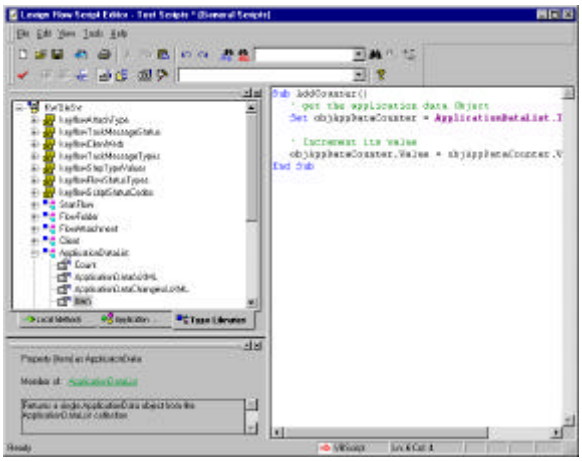


Keyflow 7.0

Script Editor Improvements

In Flow 7.0, the Script Editor has been revamped, and dramatically improved …

Script Editor Improvements



- Object Windowpane
- Object Details windowpane
- Drag and Drop
- Floating and Docking windows

eiStream

This includes:

•An Object windowpane (shown here in the upper left), which allows flexible browsing of local script methods, application data, custom roles, and type libraries.

•An Object Details windowpane (shown here in the lower left), which shows descriptions of any selected item in the Object window.

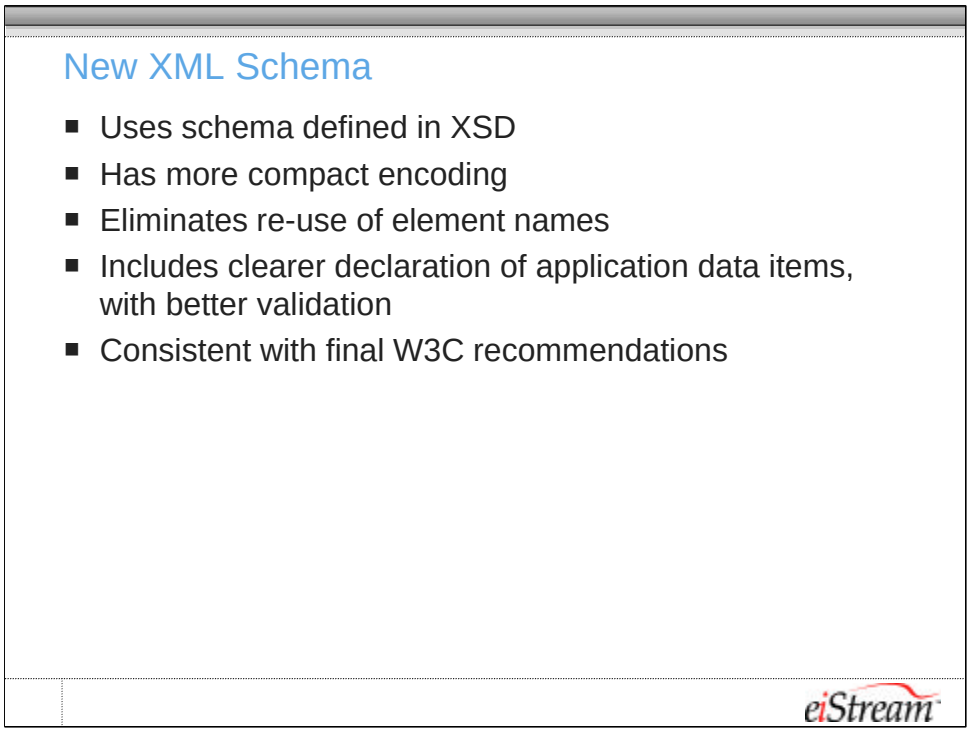
•A Drag and drop ability, from the object window into script code, to add properties or methods.

•The ability to float or dock windowpanes, menus, and toolbars. The editing environment can be flexibly customized to organize the components, where *you* want them.

•A richer set of editing functions.

The growing popularity of XML sets the stage for Keyflow to provide better customization features, as well as integrate more easily and flexibly into large and complex business processes. XML provides the means for data to flow through the integration points – into Flow and out of Flow.

The new 7.0 XML features provide the mechanisms that allow the designer of a workflow to deliver task messages as XML to both Flow clients, and the XML-aware world outside of Flow that includes custom XML clients, agents, and servers. It also provides mechanisms that allow sending XML replies or requests back to Flow for processing.



With version 7.0, Lexign Flow introduces a new XML schema, updated significantly from the original schema introduced in earlier versions of Flow.

Some of the advantages of the new schema:

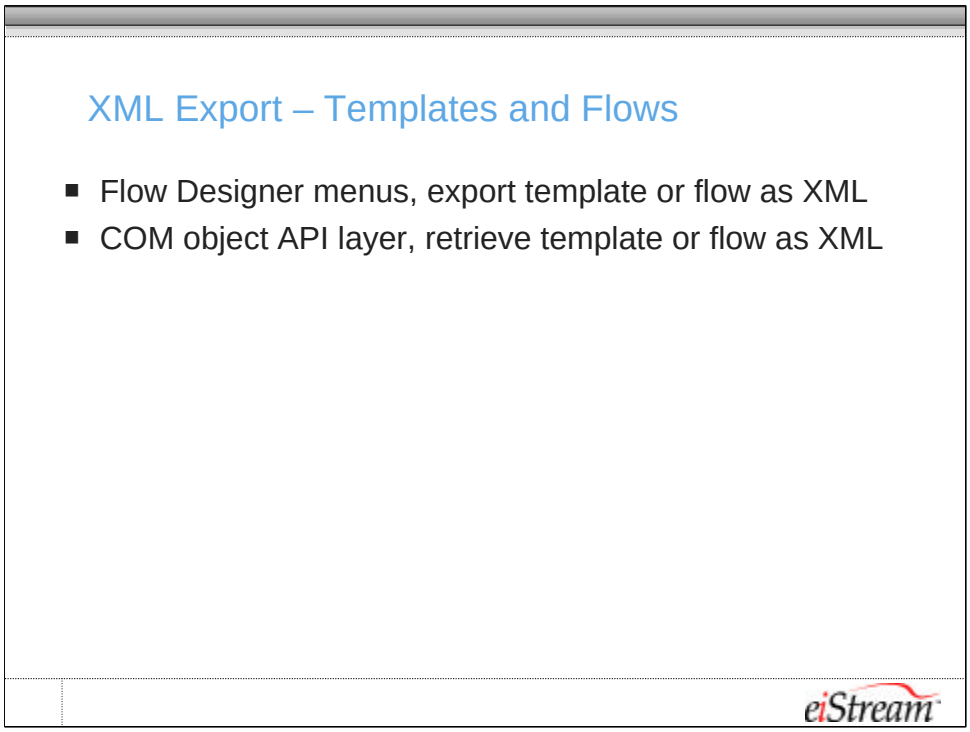
•It uses a schema defined in the XML Schema Definition language (or XSD).

•It has a more compact encoding.

•It eliminates re-use of element names.

•It includes clearer declaration of application data items, so that an XML DOM can validate them.

•And, it is consistent with final W3C recommendations, especially related to data types.



Using menu choices in the Flow Designer, or by using the COM object API layer, you can export or retrieve templates and running flows as XML.

This has a number of uses:

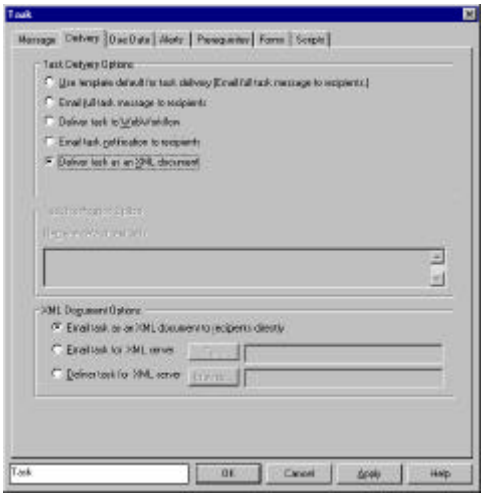
•The ability to archive templates in document management tools.

•The ability to identify what has changed between two versions of a template.

•The ability to save a template (with its full definition of all properties) for documentation purposes (for example for ISO9000 record keeping).

•The ability to read a template or flow under program control to access any element of information within the template or flow, and not be constrained to only elements exposed via the COM object layer.

New Task Delivery Options (XML)



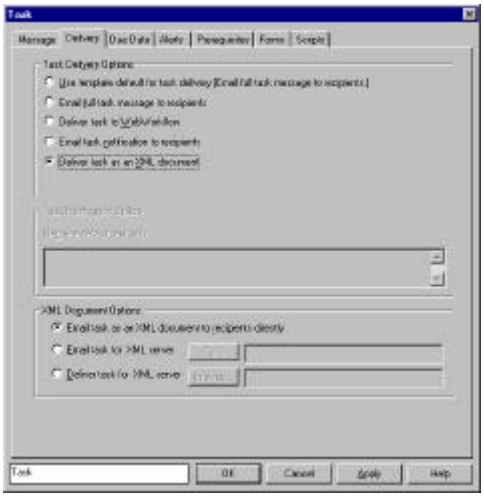
- 3 New XML Task Delivery Options

eiStream

Version 7.0 introduces three new delivery options under the heading “Deliver Task as an XML document”, all involving messages based on the new XML schema. In all cases, the expectation is that these message delivery options will be used for new applications, and new types of application integration – they are not intended to be transparently compatible with the previous task messages and older applications.

However, note that existing task message delivery options continue to be supported for full backward compatibility, and it continues to be possible to select the delivery options on a per-step basis. This means that new or existing business solutions can mix the use of older-style task messages (in most cases for interacting with users using existing forms) with the new delivery options (often for integrating with other application components).

New Task Delivery Options (XML)



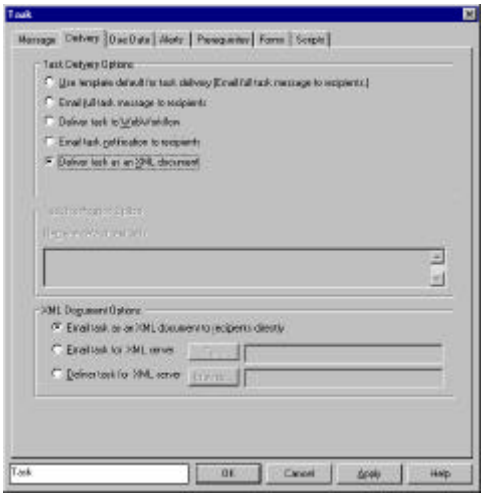
- 3 New XML Task Delivery Options

eiStream

The new delivery choices are:

•**Email task as an XML document to recipients directly**. This option emails the task message as XML to a recipient's inbox, and would usually be combined with the Message Class Option on the Forms tab, so that the message received by the recipient has a custom message class. This delivery choice is most valuable for a customer, partner or OEM wanting to develop an entirely custom client. For instance, it is possible to develop an Outlook custom form that uses the Microsoft XML DOM to read the content of the XML message, present the information to the user, and generate a Reply message in XML. Such a client would support on-line or off-line operations, while not requiring the installation of any Keyflow code on every desktop. Note that any custom client will operate directly on the XML.

New Task Delivery Options (XML)



- 3 New XML Task Delivery Options

eiStream

•**Email task for XML server.** This delivery choice is designed for use by mail-based agents, including application integration agents. The message itself is identical to the XML messages delivered directly to the user, with one exception: the designer of the flow specifies the email address to which the message is delivered, separately from (or in addition to) the intended recipient(s) of the message (normally human users). This permits the development of a variety of servers, including servers that act similar to mail gateways, in that a single server accepts messages on behalf of many different users and delivers those messages to the user.

•**Deliver task for XML server.** Like the XML message mailed to a server, this message type is intended for use by a server or agent that is acting as an integration agent, or is acting on behalf of many users. Unlike the emailed message, this option delivers the XML message to a file system directory (sometimes referred to as a “file drop”). As with the XML message mailed to a server, the designer of the flow specifies an intended recipient (human or automated) independently of the directory into which the file will be dropped, which allows a single server to act on behalf of many users or agents.

XML Task Delivery

```
<?xml version="1.0" ?>
<Task SchemaVersion="2.0" Type="ResponseTask"
      xmlns="http://www.lexign.com/schemas/flow"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <FlowProps>
      <Title Access="Read" Required="1">S37 Check</Title>
      <FlowManager Access="Read" Required="1">Sally Smith
      [SMTP:Sally.Smith@x.com];</FlowManager>
      <RunFolder Access="Read" Required="1">All Public Folders\Flows</RunFolder>
      <DoneFolder Access="Read" Required="1">All Public Folders\Done
      Flows</DoneFolder>
      <ServerAddress Access="Read">Flow Server
      [SMTP:FlowServer@x.com];</ServerAddress>
      <TemplateTitle Access="Read" Required="1">S37-2 John Smith</TemplateTitle>
      <Originator Access="Read" Required="1">John Smith
      [SMTP:John.Smith@x.com];</Originator>
      <FlowGuid Access="Read" Required="1">{16637CD4-0DE3-11D7-A75D-
      00C04FA0D0D7}</FlowGuid>
      <FlowID Access="Read"
      Required="1">A6611CD9BC800AA002FC45A0900BBC500300</FlowID>
  </FlowProps>
  <StepProps>
      <StepID Access="Read" Required="1">2</StepID>
      <StepInstanceID Access="Read" Required="1">1</StepInstanceID>
      <DeliveryFlags Access="Read">68</DeliveryFlags>
      <SendAddressList Access="Read">A J Booth [SMTP:Booth@x.com];
      </SendAddressList>
      <Subject Access="Read">S37 – Review and Approve</Subject>
      <Memo Access="Read" DisplayValuePlainText="Task memo ...\n">
      {\\rtf1\\ansi\\ansicpg1252\\deff0\\deflang1033{\\fonttbl{\\f0\\fswiss\\fpr
      q2\\fcharset0 MS SansSerif;}}\r\n\\viewkind4\\uc1\\pard\\f0\\fs20 Task
      memo ...\\par\r\n}\r\n</Memo>
      <Importance Access="Read">1</Importance>
      <FlowAccess Access="Read">1</FlowAccess>
      <PossibleResponses Access="Read">Approve\nReject\n\n</PossibleResponses>
      <DistributionList Access="Read">[Role:Approver]; </DistributionList>
```

- XML Task
  - Flow properties
  - Step properties
  - Application Data
  - Addressing info on how to send a reply
- Reply can be sent back to Flow Server, as an XML message

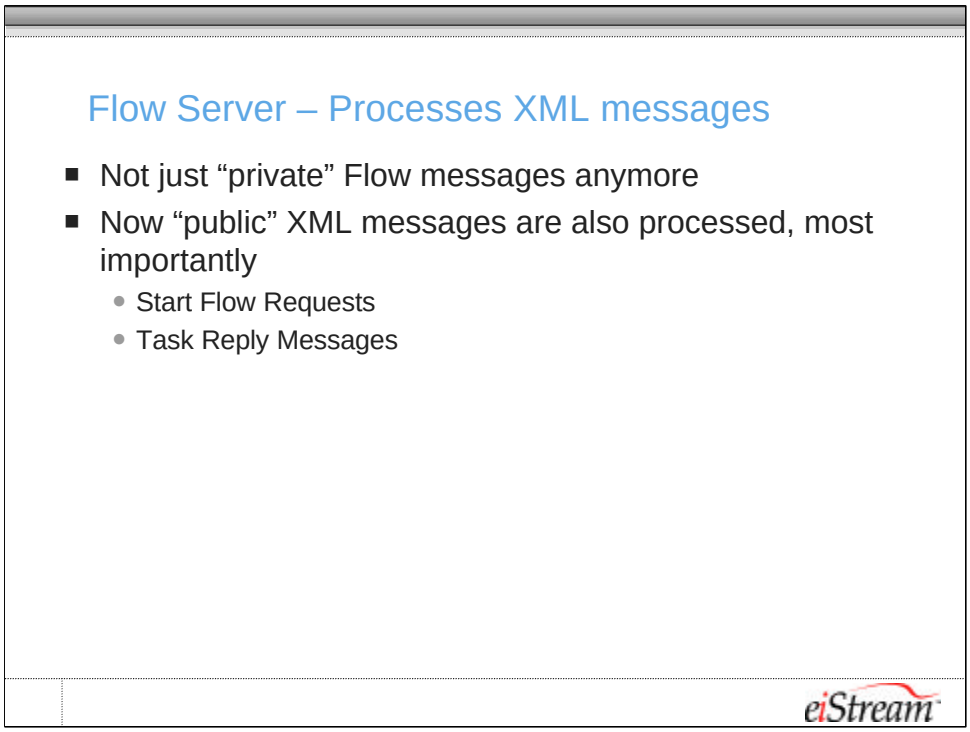
eiStream

Let's talk, for a moment, about the task XML which is delivered.

It contains all pertinent information about the flow, the step, application data and custom roles … and very importantly, addressing information on how to send a reply back to the Flow Server.

The agent or custom client which processes the XML, replies by simply sending a message back to the Flow Server's inbox, with the body of that message containing XML that describes the response.

The agent or custom client does not need any Keyflow code installed … it is pure XML processing, to process the task, and to reply.



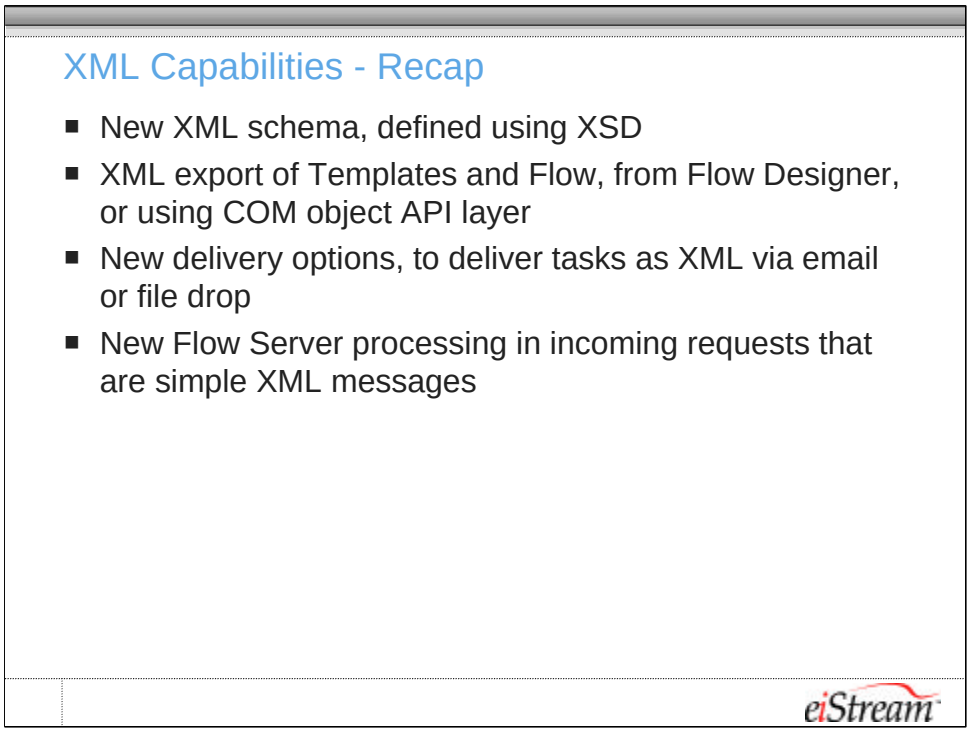
Prior to Flow 7.0, the Flow Server only processed private messages (with specific message classes) arriving in the Server’s Inbox. Those private messages were only generated and sent by a Keyflow client, or other client using Flow’s COM object library. This could have been the Lexign Flow Client, Web Workflow, or some other custom agent that utilized the COM object library to access a task message.

Now as of Flow 7.0, the Server also processes “public” XML messages. A “public” XML message is any message sent to the Flow Server’s Inbox using a standard message class IPM.NOTE (or subclass) that contains, in its body, XML that corresponds to one of the expected Lexign Flow schemas.

•For a start flow request … the XML content identifies pertinent information to name the new flow, to identify who the originator is, and to assign values to flow application data. It also can identify the template by reference – that is, there is an element within the XML schema that identifies the folder path and name of a Template message stored in Exchange, that is to be used for starting the new Flow.

•For a reply messages … they would ordinarily be generated and sent to a Flow Server either:

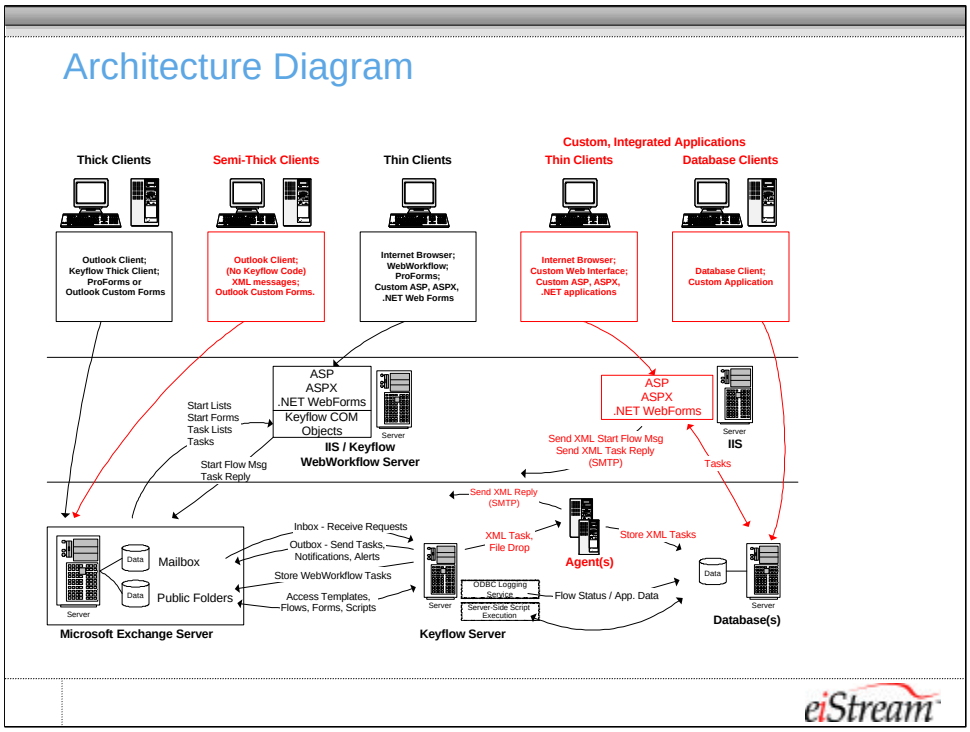
- •By a custom client, such as an Outlook custom form, which is utilizing an XML task message sent directly to a recipient
- •Or, by a server or agent that is utilizing XML task message sent to the server via either email or as a file drop.



In summary, the new 7.0 XML features provide the mechanisms that allow the designer of a workflow to deliver task messages as XML to both Flow clients, and the XML-aware world outside of Flow that includes custom XML clients, agents, and servers. It also provides mechanisms that allow sending XML replies or requests back to Flow for processing.

These XML capabilities open up numerous customization and integration possibilities, providing flexible mechanisms to utilize the power of the Keyflow workflow engine in many new ways.

To show some of these possibilities, I'd like to show a Keyflow architecture diagram …



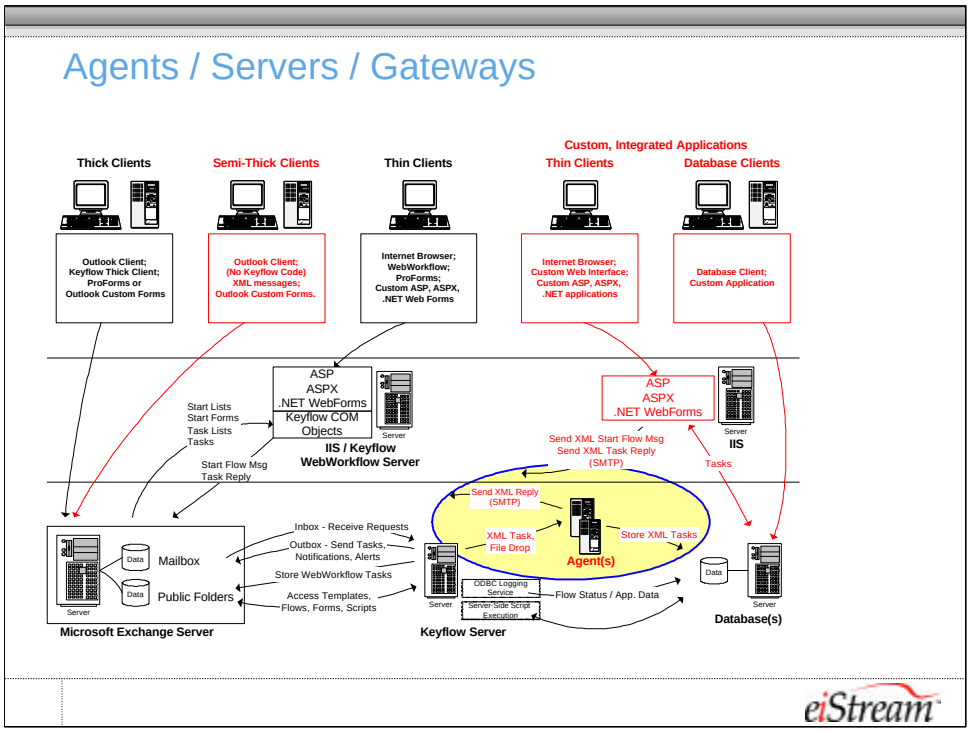
This diagram represent a 3-tiered architecture.

•The bottom tier is the server layer, including the Microsoft Exchange Server, the Keyflow Server, and Databases.

•The middle tier is the IIS and WebWorkflow Servers.

•The top tier includes the clients.

In this diagram, everything in black represents the Keyflow architecture **prior** to version 7.0. Everything in red represent the added possibilities available in version 7.0, utilizing the XML capabilities of the product.



Let's first look at the bottom server tier, in the area circled.

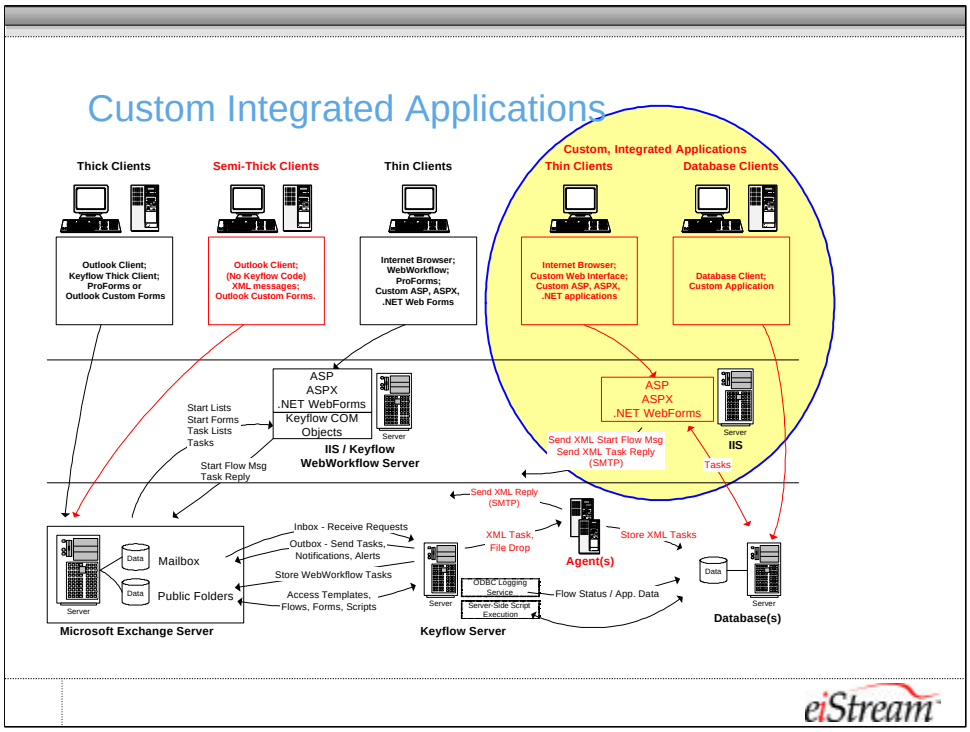
Agents, prior to version 7, were generally built as clients, monitoring tasks delivered to an inbox. Now, in Keyflow 7, they can be more flexibly built, not requiring any Exchange session, and not requiring any Keyflow code.

•The Flow Server can deliver a Task as an XML file-drop into a file system directory.

•An Agent can monitor that directory, and process and reply via an XML message (a normal email msg with XML in the body) sent to the Flow Server (via a standard SMTP or SendMail facility).

•Also, an agent can act as a gateway, storing Tasks in private storage for custom handling.

This last point "Storing XML Tasks" is one of the most intriguing. In the diagram shown here, the Agent has an arrow pointing down to a Database where tasks are stored. This allows for complete custom management of tasks. This would replace, for example, Keyflow storing tasks for WebWorkflow, in public folders in Exchange. This also allows for the development of custom client interactions, as shown … (next slide)



(as shown…) here. The circled components represent

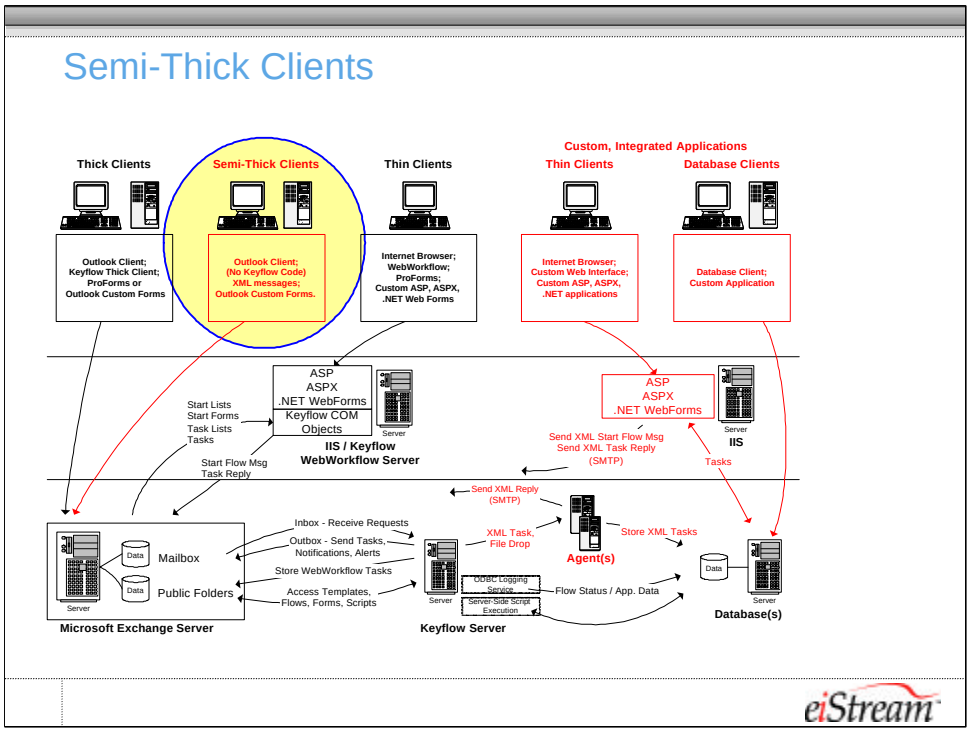
•Tasks stored and handled privately, and integrated with a custom application.

•The entire application is implemented without any Keyflow installed code.

•The client interaction is via a custom web (thin-client) interface, or possibly as a custom database oriented client.

•Communication back to the Flow Server is accomplished via XML messages sent to the Flow server (via standard SMTP or SendMail facilities).

The possibilities here are extensive. Once the tasks are stored and integrated external to Keyflow, they can be managed in any special way desired.



The last architectural area to discuss … is what can be called “Semi-Thick” clients.

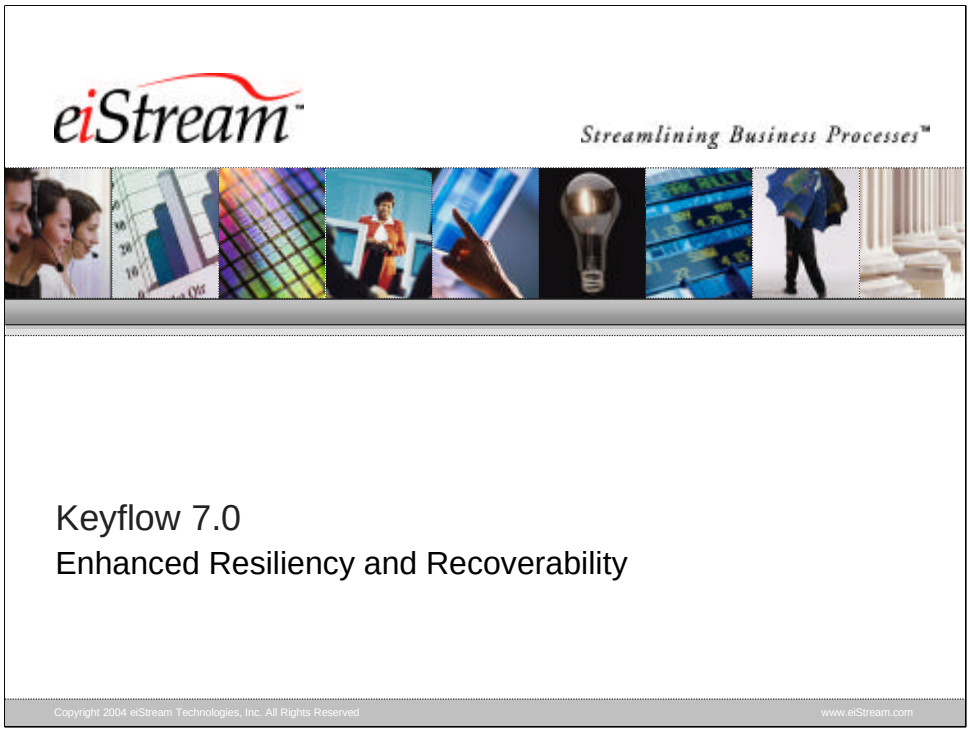
•The Flow Server delivers Tasks as XML, directly to participant’s Inbox, with a private Message Class.

•Participants view tasks via Outlook Custom Forms.

•The Outlook Custom Forms are designed and written to process the XML.

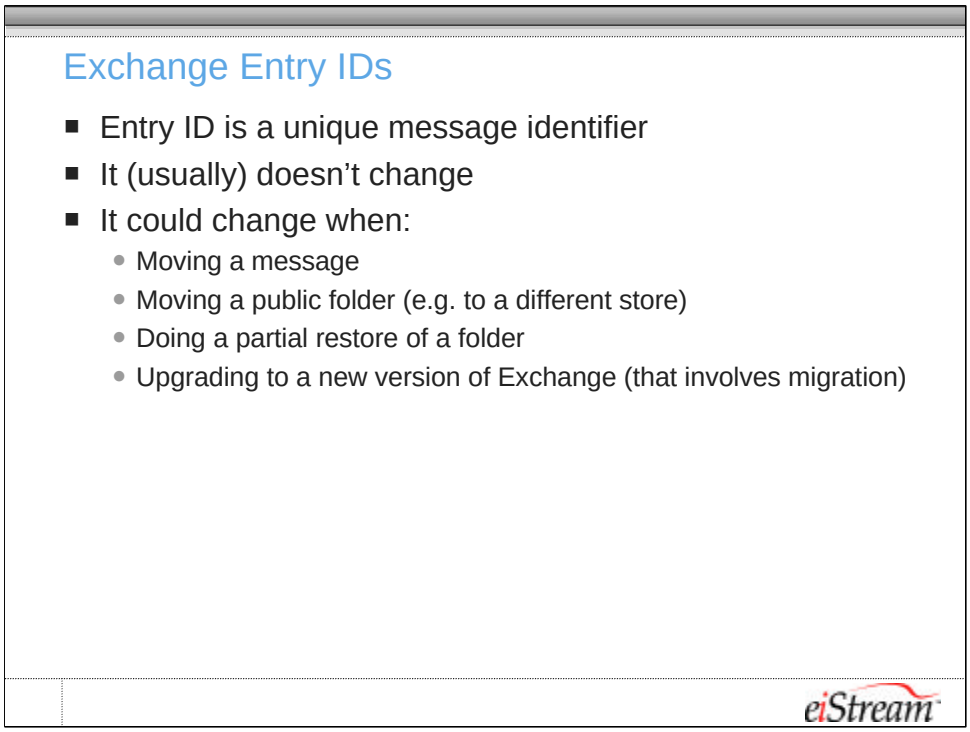
•Replies are sent as XML messages back to the Flow Server.

•No Keyflow client code is installed (and this is why it is called a “Semi-thick Client” … although it is thick, in the sense of being an Outlook client to Exchange, it does not require any Keyflow installation).



We'll now move on to the next feature enhancement of Flow 7.0 … that of improved resiliency and recoverability.

To understand the issues surrounding the work done in Keyflow 7.0, we first need to understand the concept of Exchange Entry ID's, and how they are used by Keyflow.



An Exchange Entry ID is a unique identifier assigned by Exchange for every message.

It is **usually** fixed, and does not change … but the key word here is … "usually".

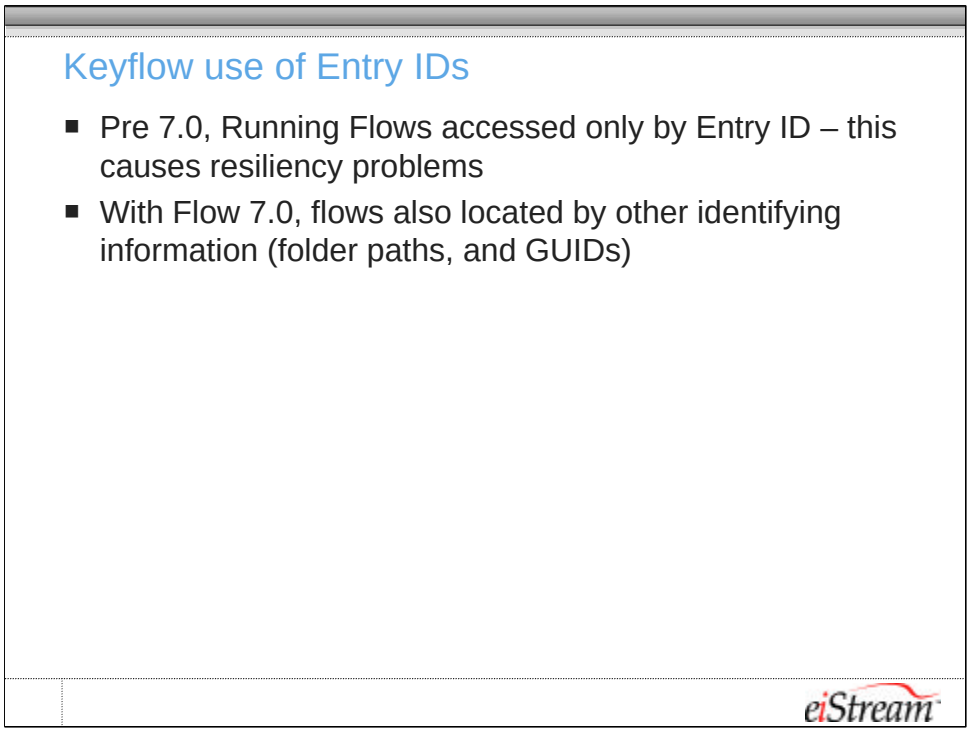
There are some circumstances when it **can** change … such as:

•When you accidentally move a message out of and back into a public folder.

•When you move a public folder to a different Exchange store or to a different server.

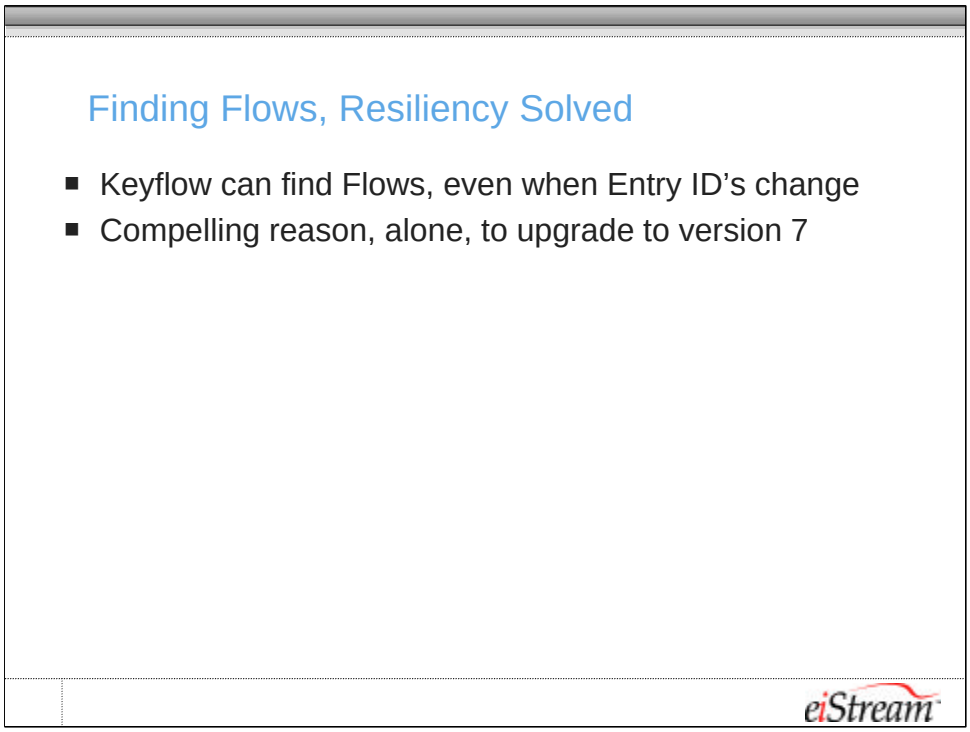
•When you do a partial restore of a folder from a backup.

•Or, when you upgrade or migrate to a new **version** of Exchange or to a new Exchange machine, in any way other than an in-place upgrade, or what's called a "fork-lift" upgrade.



Prior to version 7.0, Running Flows were located using **only** Exchange Entry IDs. Accessing Exchange messages by Entry ID has a significant performance advantage, because it represents the fastest way to access objects in Exchange. At the same time, it presents a resiliency problem, because of those situation that cause the EntryID of an object to change.

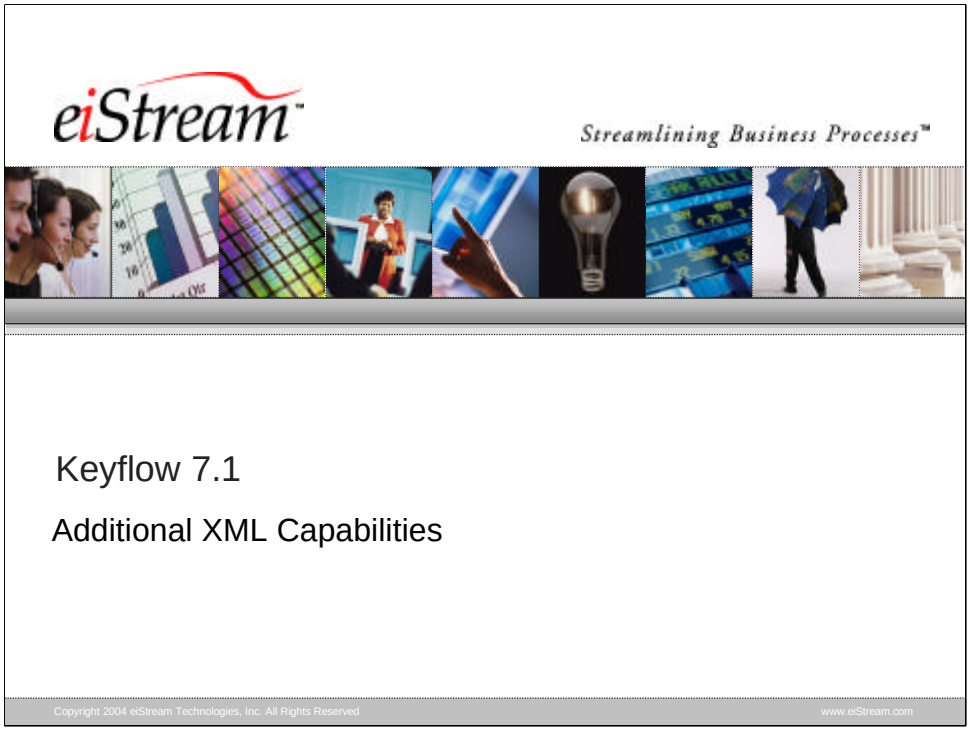
With Flow 7.0, when the Server is processing messages, it first attempts to locate the running flow by the Entry ID, and if that fails, it attempts to locate the running flow using other identifying information, including folder paths and a unique (persistent) identifyer assigned by Keyflow itself (a GUID). Note that all incoming messages to the Flow Server now contain enough information (Entry ID, GUID, and Exchange folder paths) to locate the running flow by either method.



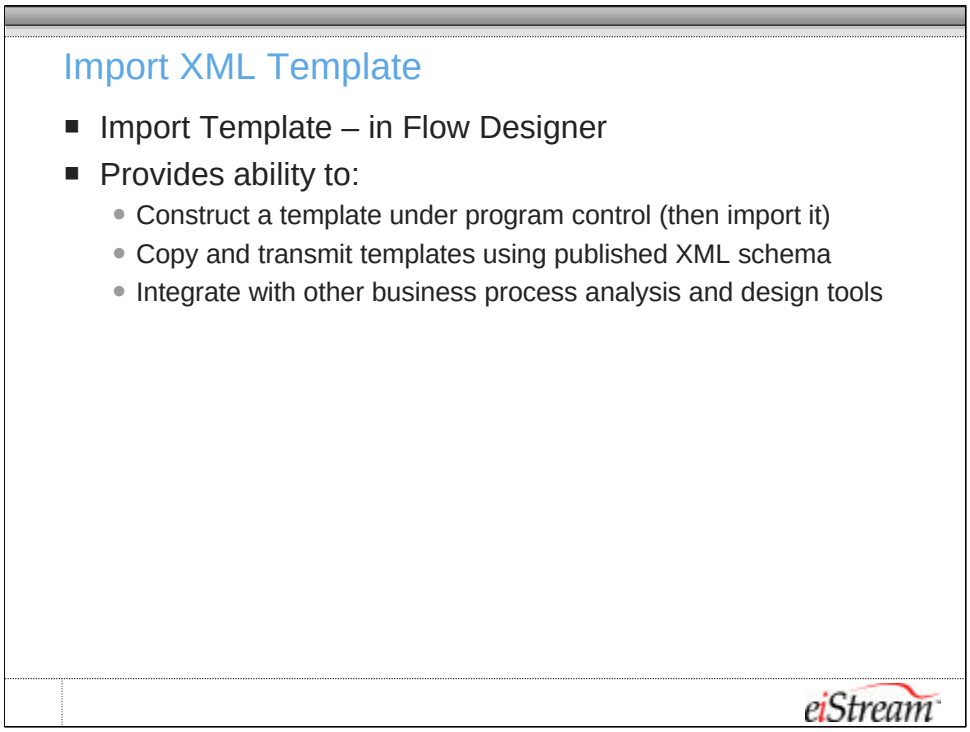
So, as of Version 7.0, the resiliency issues surrounding properly finding running flows is solved.

If there is no other reason to upgrade to the latest version 7 release – if you don't care for the improved interface, if you don't care about the extensive new XML capabilities, if you don't have need for **any** of the new features – this one issue is a compelling reason to move toward the new Keyflow release.

And one other point. If you have any Microsoft Exchange upgrade being planned, and since those upgrades usually involve migrating public folders to new servers on new machines, we strongly recommend that you be running Keyflow 7.1, before proceeding with that upgrade.



That completes the discussion of Keyflow version 7.0. We now move on to version 7.1 … and to start, we'll look at some additional XML capabilities added in 7.1.



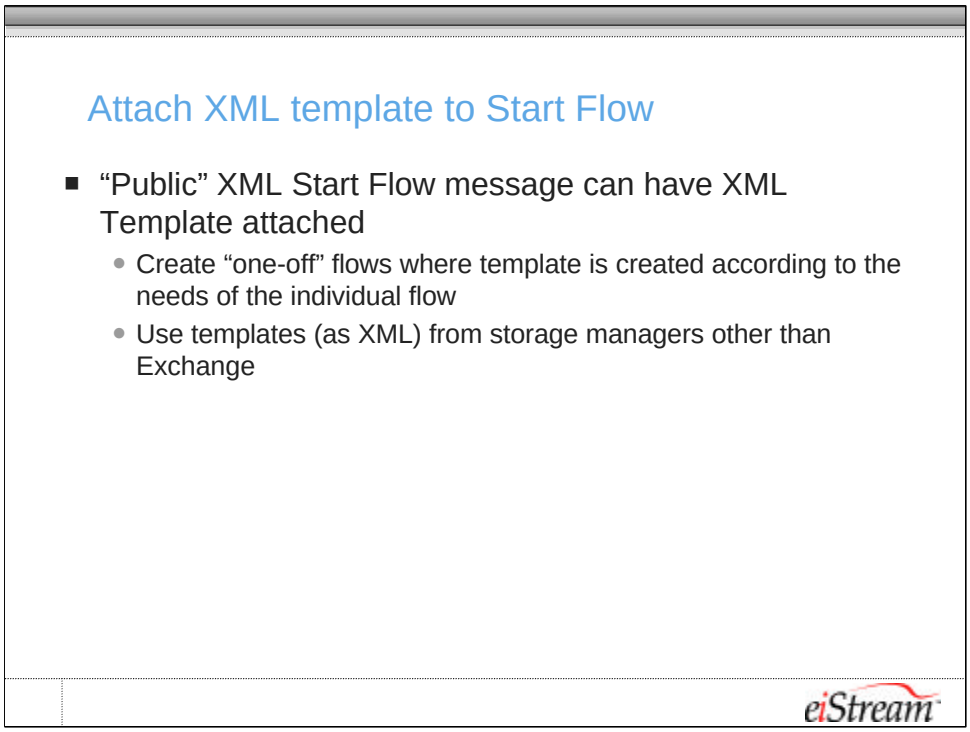
Flow 7.1 adds a menu item to the Flow Designer's Import menu, to import templates previously saved or constructed in XML.

This provides such things as:

•The ability to construct a template under program control.

•The ability to copy and transmit templates from one Lexign Flow system to another, using a published XML schema as the intermediate format.

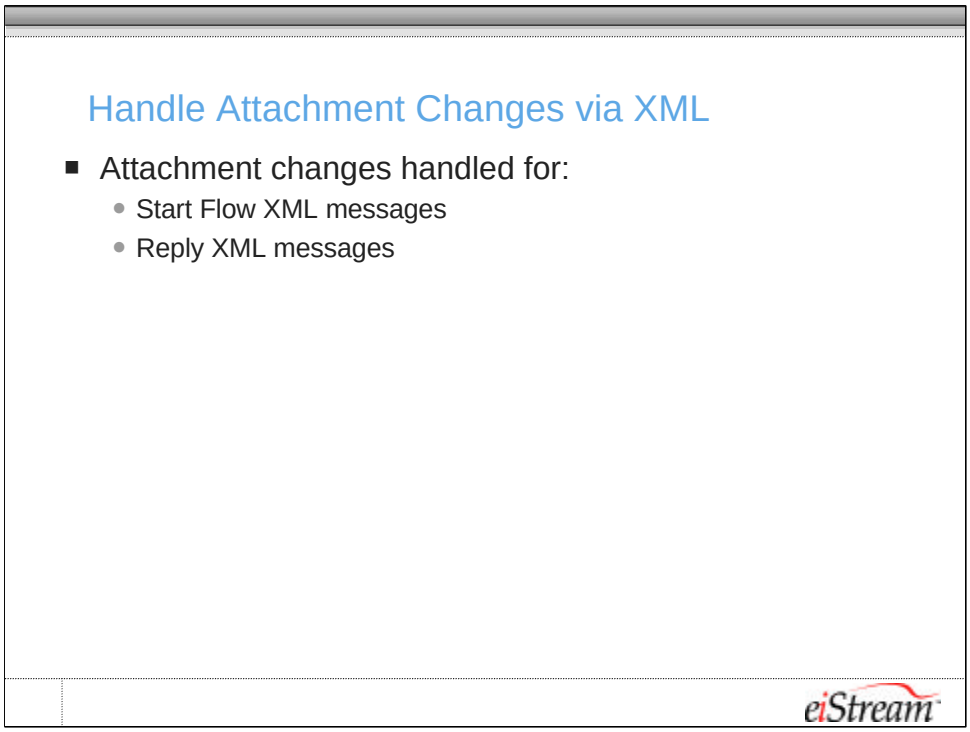
•And, the ability to integrate with other business process analysis and design tools.



Flow 7.1 also includes the ability to attach an XML specified template to a “public” XML start message, which is sent to the Flow Server directly through the mail system. An XML based start message, as introduced in Flow 7.0, consists of a message body containing the XML representing the start message. Rather than identifying the template (as in 7.0) by referencing a particular template by Exchange public folder path and message name, the template (in Flow 7.1) can also be specified by adding the XML template to the start message as a named attachment.

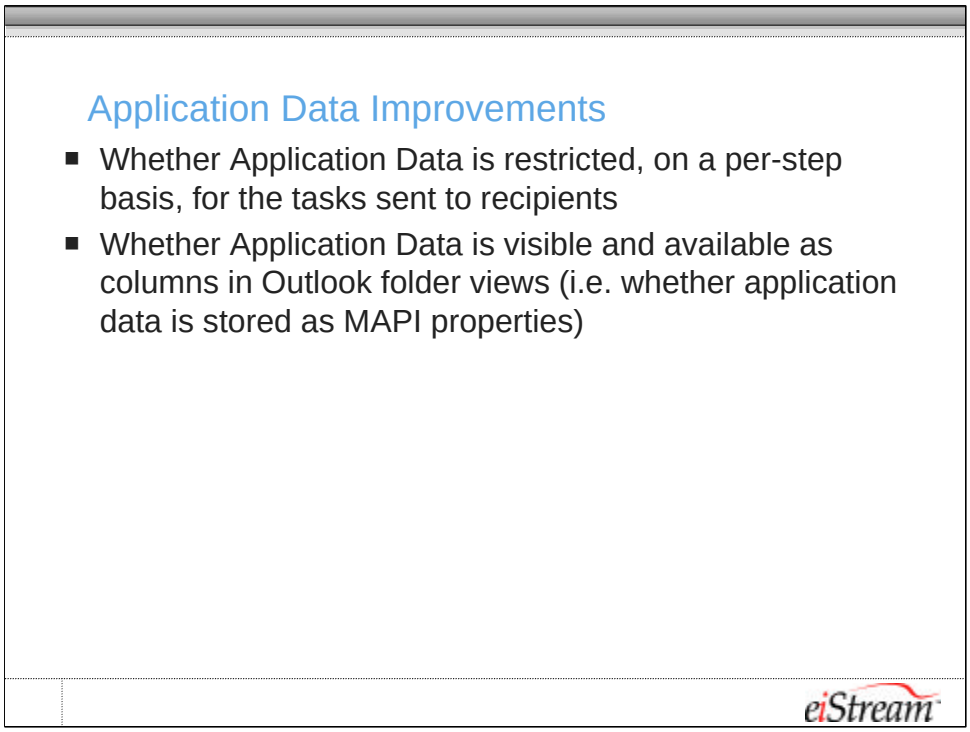
The ability to attach an XML template to a start message makes it possible for you to create “one-off” flows in which the template describing the flow definition is created under program control according to the needs of the individual flow.

It also makes it possible for you to store a template in a storage manager other than Exchange, and supply the content of that template when starting the flow.



The original XML messaging work done in Flow 7.0 did not include support for returning attachment changes as part of a task reply, or for start flow messages. The “public” XML messaging was completed in Flow 7.1, and now includes the ability to handle all attachment changes – attachments which have been added, modified, or deleted.

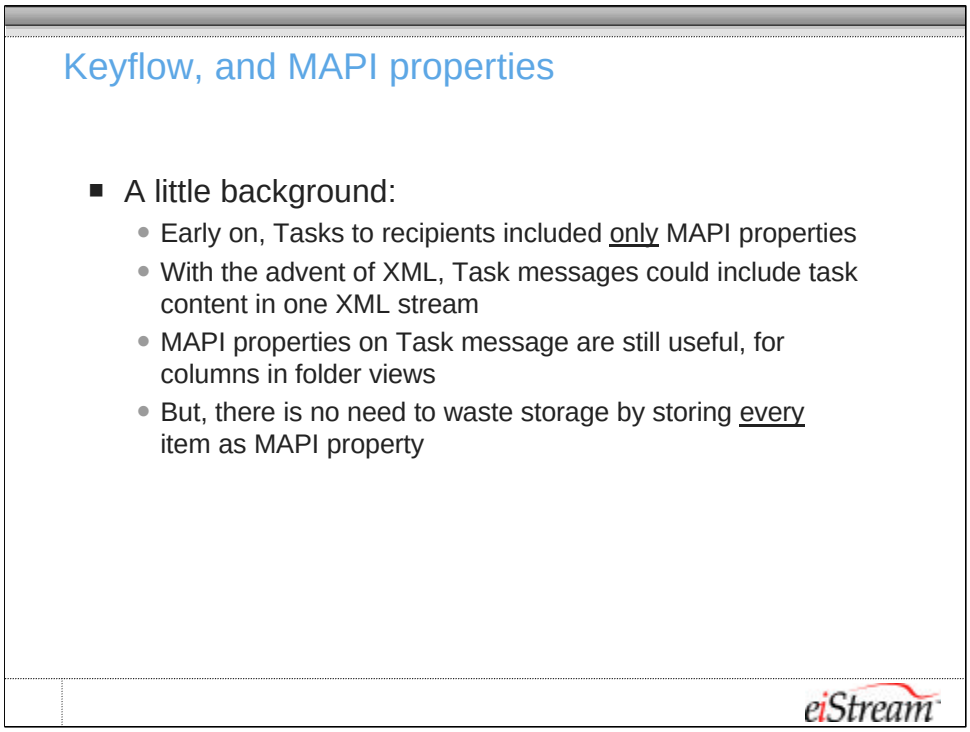
There are a number of enhancements to Keyflow 7.1 regarding Application Data …



The improvements are in two areas:

•First, whether application data is restricted, on a per-step basis, on the task messages which are sent to recipients. Many steps don't require some of the application data at all, to be part of the task sent to participants.

•And second, whether any application data is additionally visible (or not) as MAPI properties (in other words, if they are available – or not - as viewable columns in Outlook folder views),



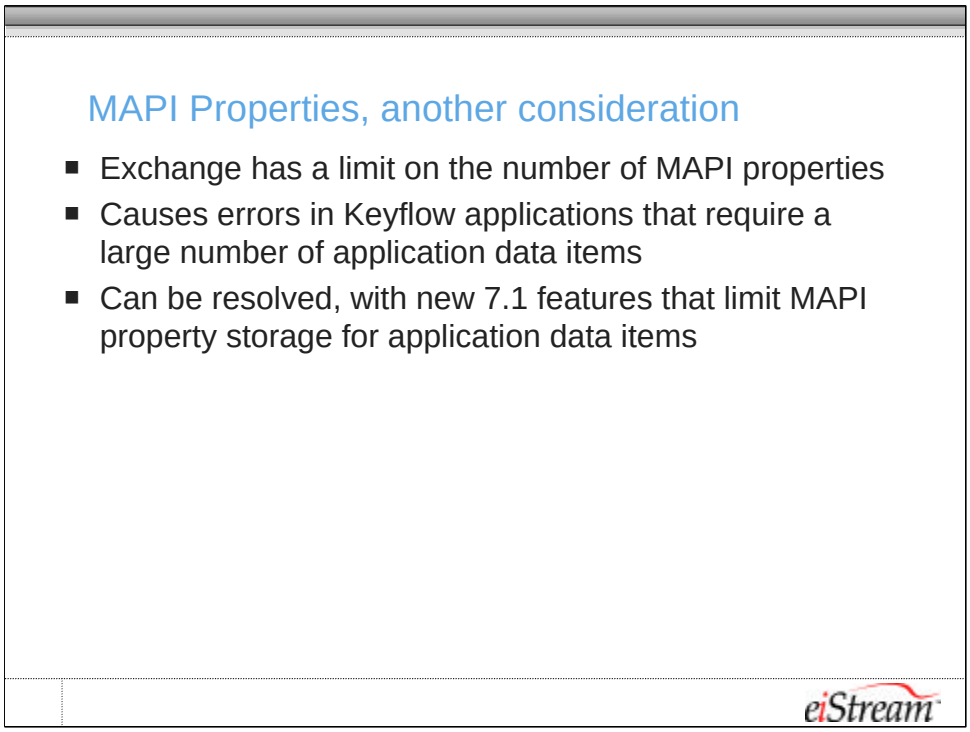
First some background on Keyflow's use of MAPI properties ...

•Task messages, from the early development of Keyflow, have been MAPI based messages, meaning that ALL information being communicated to the flow participant were stored in the message, each individual piece of task information stored as a separate MAPI property. This included individual MAPI properties for subject, memo, possible responses, importance, flow ID's, Flow Server addresses, and also … application data (one MAPI property for each application data item). ALL application data items for a flow have been sent as part of every task message.

•With the advent of XML, we are able to instead send Task messages, where the task content (all of those individual data elements that make up the task message) can be packaged in one XML stream, rather than as separate MAPI properties on the message.

•However, we still want the flow designer to be able to choose which of the application data items additionally get stored (in other words, replicated) as MAPI properties, to make them available for Exchange Folder Views, as columns in an Inbox, or other Exchange folders.

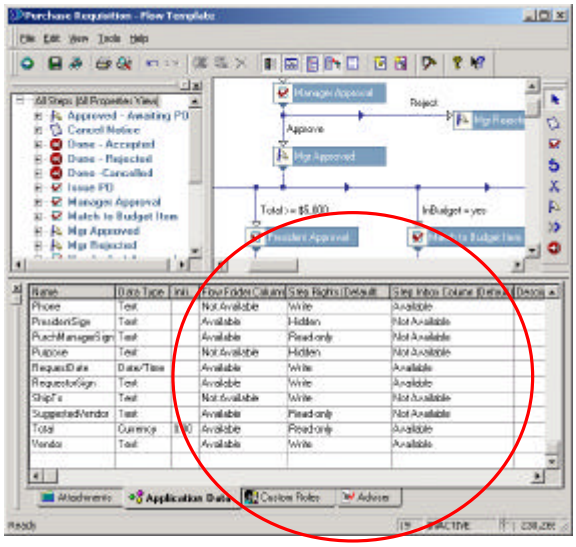
•In converting to the use of XML, we still need some application data MAPI properties, because many applications rely on those custom views, and rely on having particular MAPI properties (of the application data) available. The application data that is desired for this use is usually a very small subset of all application data sent in the message.



There is one more issue to consider … one which a number of customers have encountered.

Exchange has a limit on how many MAPI properties can be stored on a message. The limits are somewhat unclear, because it's not an explicit limit on the number of MAPI properties, but dependent on the size of the data (cumulative) stored across all of those MAPI properties. From experience, the limit is somewhere around 400 application data items which can be defined in a flow, before there are errors. This has caused problems in some Keyflow applications that require a large number of application data items, and can now be resolved with the new 7.1 features that limit MAPI property storage for application data items.

New Application Data Options



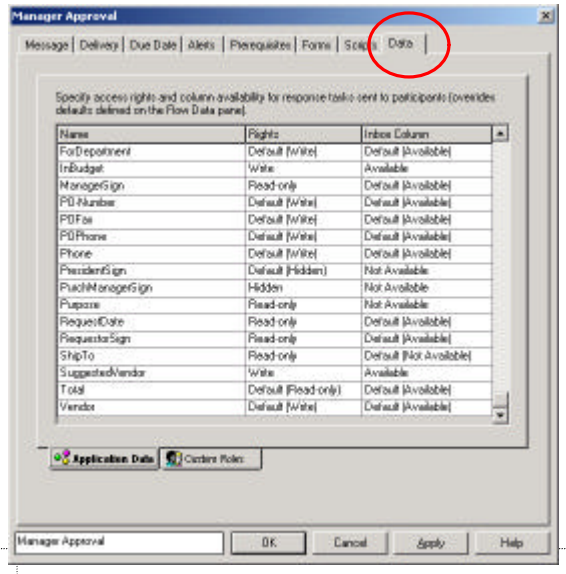
- Three new options for each application data item

Flow 7.1 included three new options for each application data item, available in the Data Pane of the Flow Designer.

•The first is labeled “Flow Folder Column”, and specifies whether each application data item, when viewing running flows, is “Available”, or “Not Available” as a column in Outlook folder views. This allows you to limit the MAPI property storage on running flow objects to certain application data items that you want available.

•The other two new options are default settings, which can be overridden on a per-step basis, within each step’s properties dialog. They identify the default setting that indicates whether the application data item is “hidden” within steps (in other words, not-included in the tasks sent to participants), or whether it has “Read-only” or “Write” access … and additionally whether any application data item that is not hidden, is also “available” or “not available” as a column in Outlook inbox views or webworkflow task list views where task messages are being delivered.

New Step Options



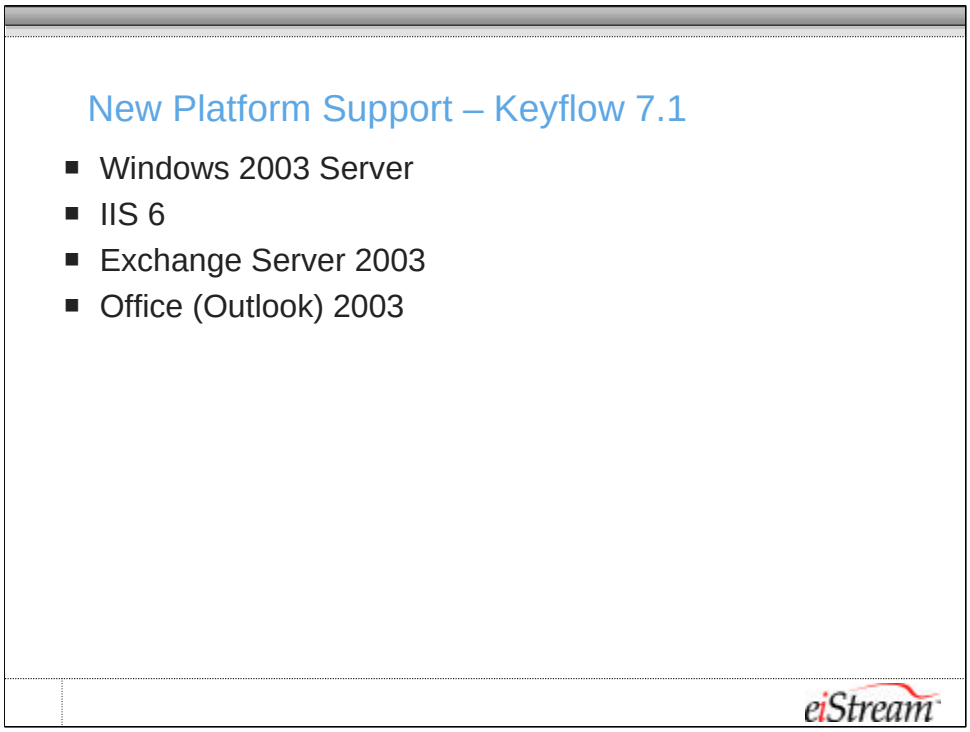
- Step Properties Dialog, new “Data” tab
- Specify which Application Data is included or hidden
- Specify whether Application Data is available as column in Inbox views

eiStream

The options can be overridden on a per-step basis, within the step properties dialog of each step. Notice the new tab in this dialog … the “Data” tab.

For each application data item, the setting can be left using the default setting, or be overridden, to make the item “hidden”, or included with “read-only” or “write” access, … and to make the item available as a column in Inbox views or WebWorkflow task list views. This allows you to limit the MAPI property storage for task messages sent to participants.

Flow 7.1 also contained some platform and configuration enhancements …



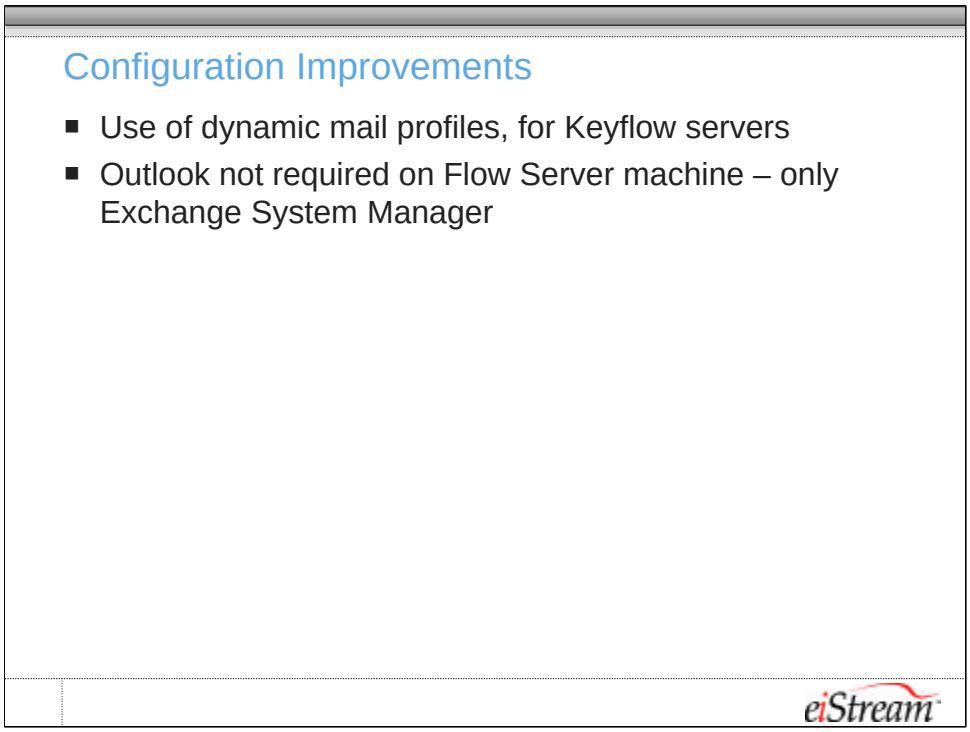
The following platforms are certified:

•Windows 2003 Server

•IIS 6

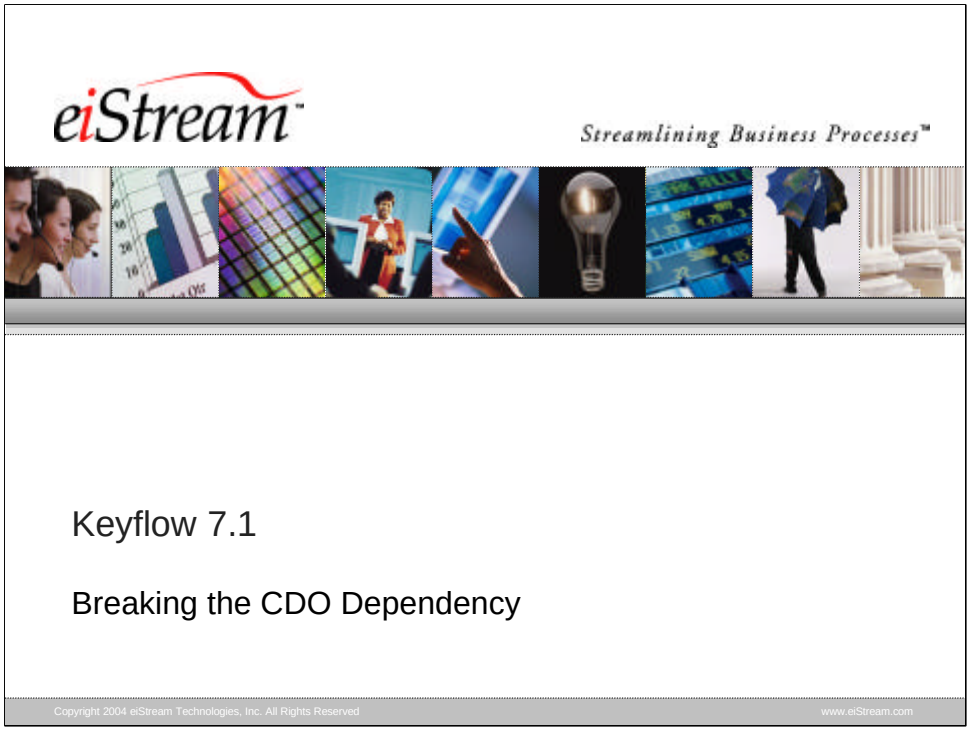
•Exchange Server 2003

•And, Outlook 2003

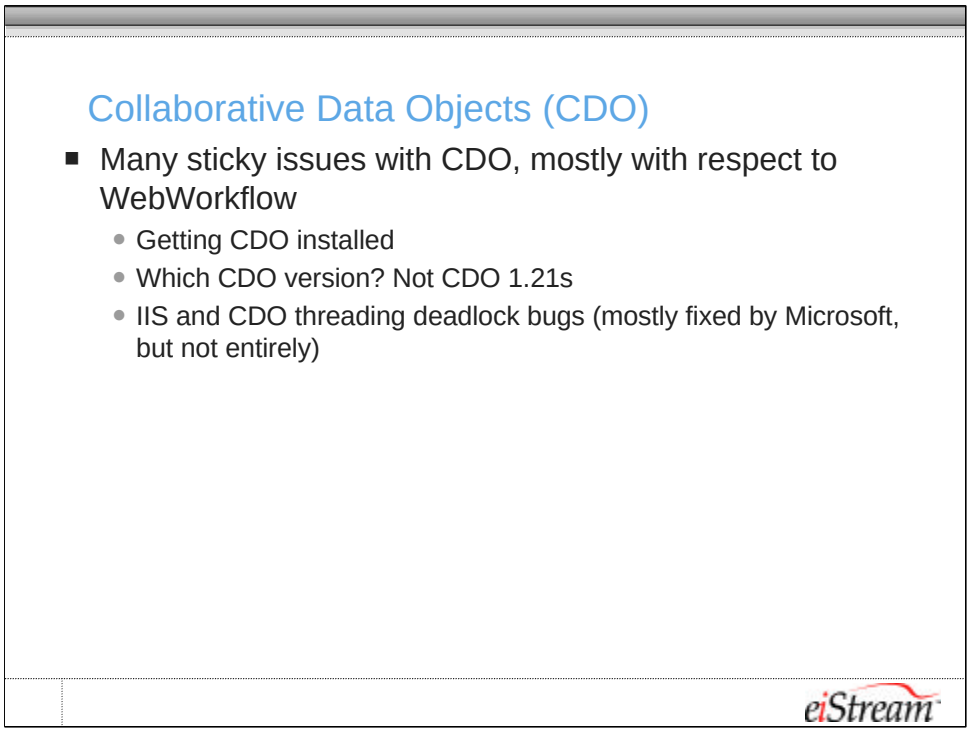


Keyflow 7.1 makes use of dynamic mail profiles, for the Flow Configuration / Setup utilities, and for the Flow Server. This means that, on both the Flow Server machine, and the WebWorkflow Server machine, Outlook does not need to be installed, and mail profiles do not need to be created, before configuring and using the Flow Server and WebWorkflow. The Exchange System Manager is still recommended on these machines.

This simplifies the installation and configuration procedures, and eliminates the confusion regarding installing Outlook, and the Exchange System manager … in the right order.



And finally, the last item to be covered for the Keyflow 7.1 release … concerns CDO …

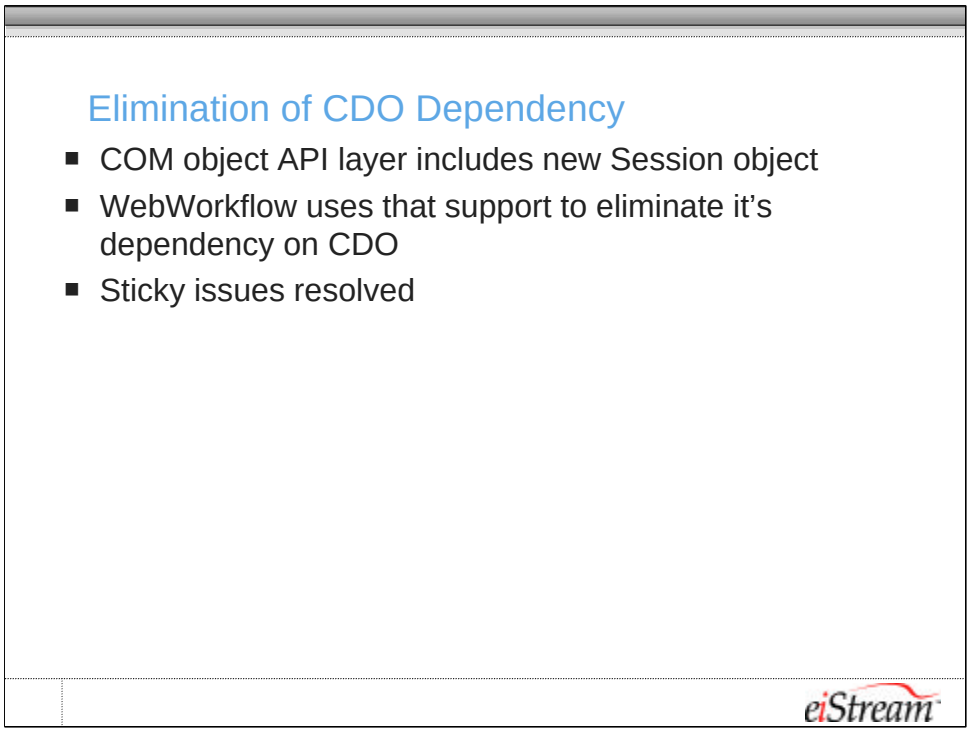


Keyflow's use of CDO, within WebWorkflow, has involved many sticky issues.

•First, it's difficult to even get installed, since it is not installed by default in a Microsoft Outlook installation.

•Second, you have to be very careful which version you have … particularly related to a special high security version (1.21s) … that does not work correctly with Keyflow WebWorkflow.

•And finally, there have been some serious IIS and CDO threading deadlock bugs … now mostly fixed by Microsoft, but … unfortunately … not entirely fixed.



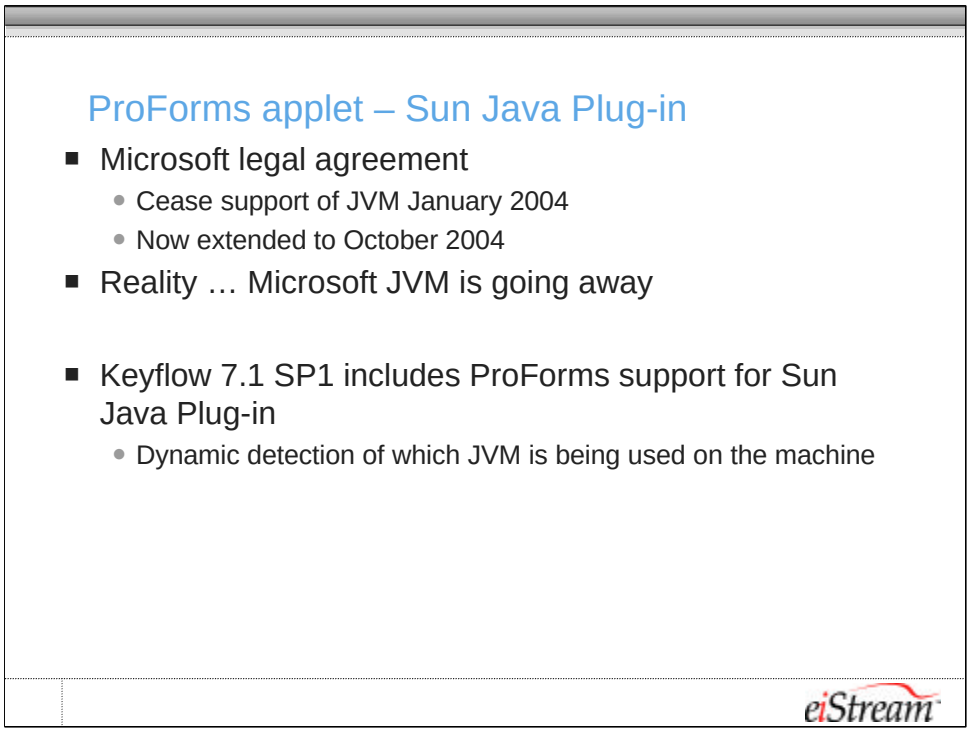
Keyflow 7.1 eliminates the dependency on CDO.

The Keyflow COM object API layer includes new Session objects to manage a MAPI session to Exchange. CDO is no longer needed.

WebWorkflow (whose ASP pages are built using the Keyflow COM object layer), utilizes this new support, and thus eliminates **it's** dependency on CDO.

The sticky CDO issues are resolved.

As one additional note on Keyflow 7.1 … I'd like to mention the release of Service Pack 1 … and discuss the issues of the Microsoft JVM vs. the Sun Java Plug-In …



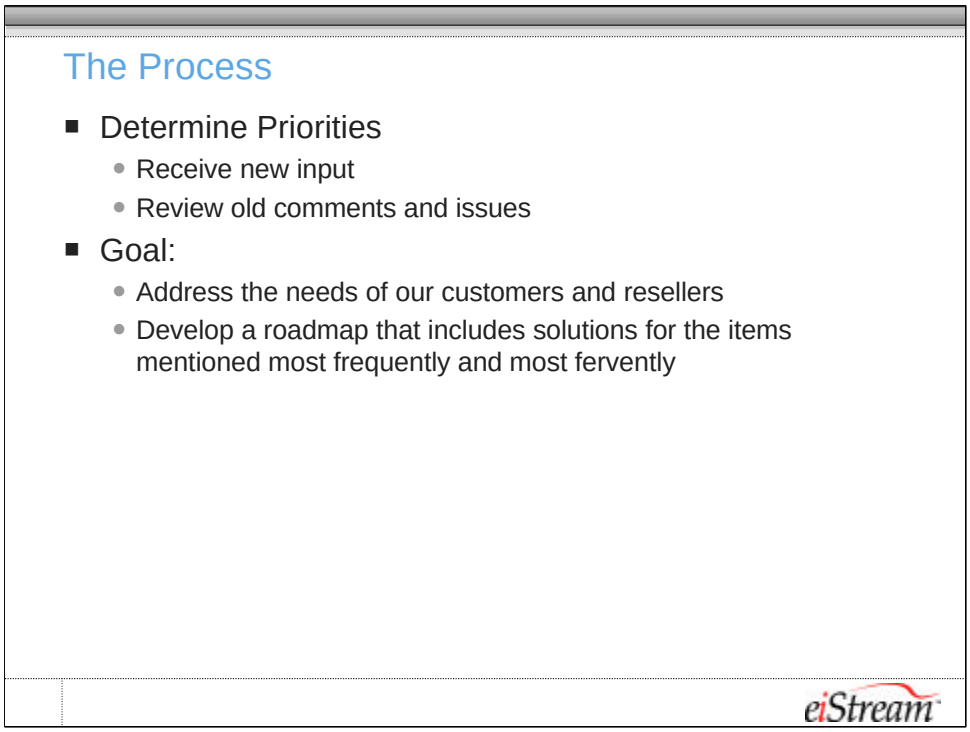
The reality is … the Microsoft JVM is going away. From legal agreements, support was to have ended in January of 2004. That has now been extended to October 2004.

The ProForms applet has … in the past … only worked with the Microsoft JVM … but as of Keyflow 7.1 SP1, it now runs using either the Microsoft JVM, or the Sun Java Plug-in. Both the Keyflow thick client, and a WebWorkflow thin client, automatically detect which JVM is being used on the machine, and operate in either environment.

This allows you to plan your own transition, at your own pace, away from the Microsoft JVM on your client machines, to the Sun Java plug-in.

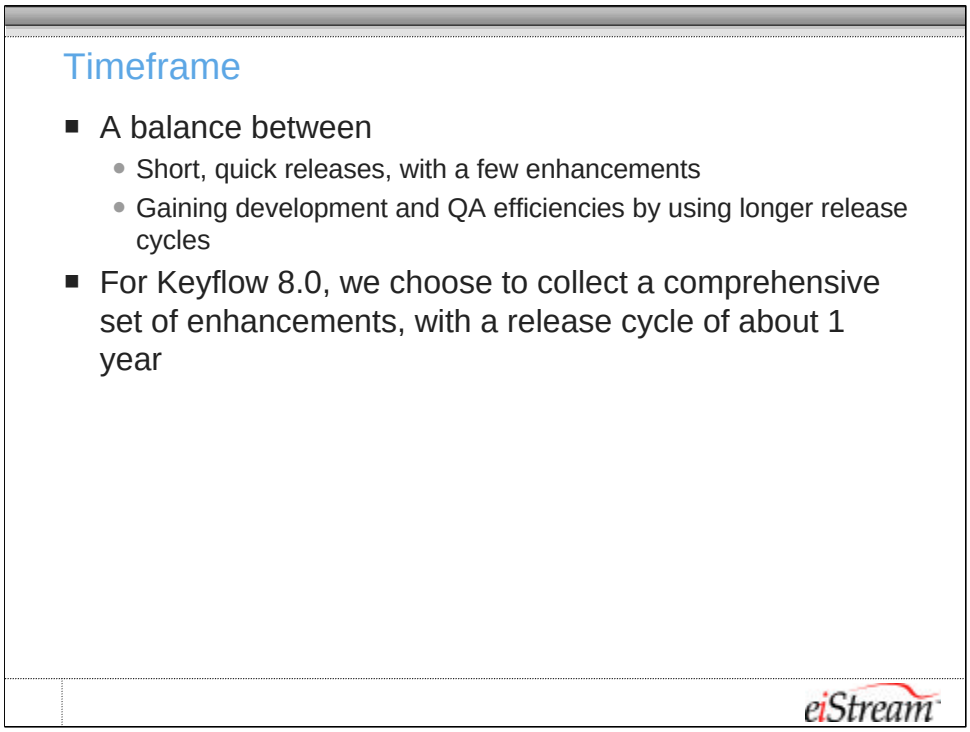
We believe the enhancements and important improvements in Keyflow 7.0 and 7.1 offer some compelling reasons to upgrade …

But, we've now spent enough time talking about the present. Let's move on to the future. The Keyflow Roadmap.



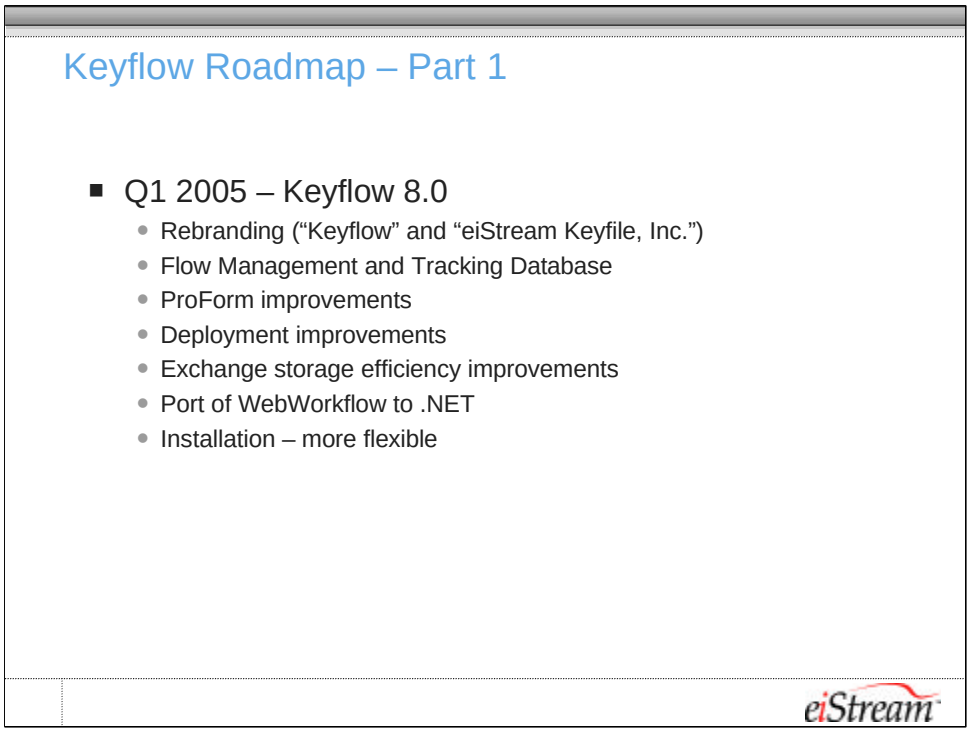
We have just gone through a process to determine priorities on many, many Keyflow enhancements, improvements, and issues. We did this by analyzing new input over the last couple of months, and reviewing many comments and issues that we had collected in the past.

Our goal? … to address the needs as expressed by our customers and resellers, and to develop a roadmap that includes solutions for the top priority items … those items mentioned most frequently and most fervently.



In developing software releases, there is always a balance between trying to move enhancements quickly to market through **short** release cycles, and gaining development and quality assurance efficiencies by using **longer** release cycles.

The plan for the next release of Keyflow, Keyflow 8.0, is to collect a comprehensive set of features and enhancements that address many of the top priority items, and by so doing set a release cycle that is about 1 year long.



That results in the following, as the first part of the Keyflow Roadmap:

A Keyflow 8.0 release in Q1 2005 with the following items …

•A Re-branded product, from Lexign Flow, back to the “Keyflow” product name with “eiStream Keyfile, Inc.”

•A revamped Flow management and tracking database feature (replacing the current ODBC logging and reporting facility)

•Fixing some ProForms issues

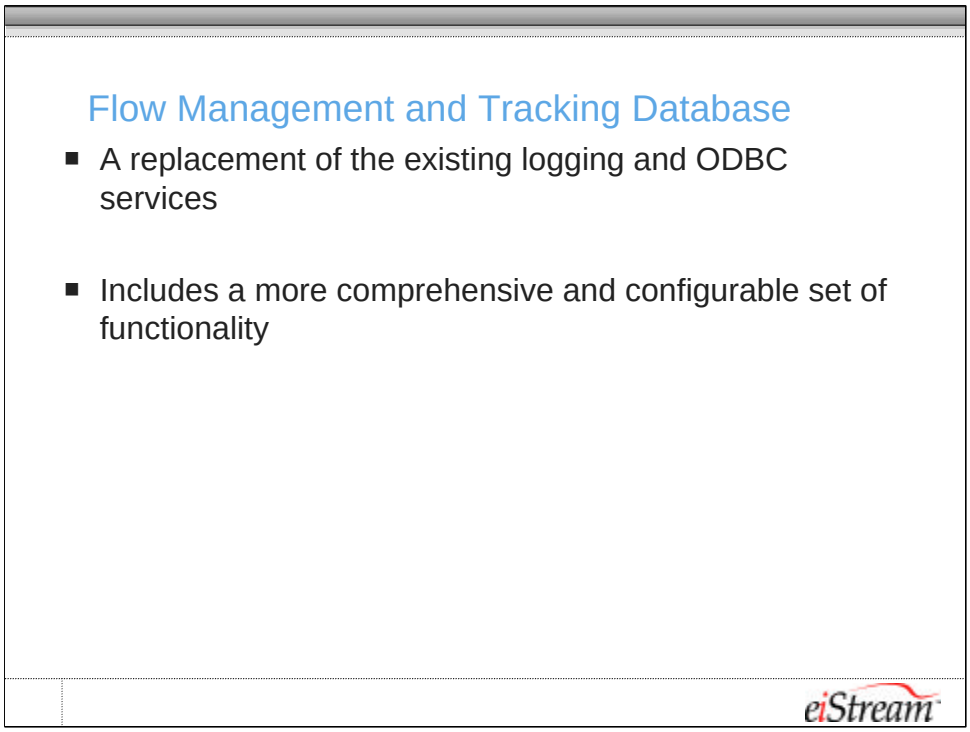
•Solving some Deployment issues (particularly relinking forms, scripts, and subflows when moving a template to a deployment system)

•Making some Exchange storage efficiency improvements, looking at issues regarding storage loads and Exchange throughput

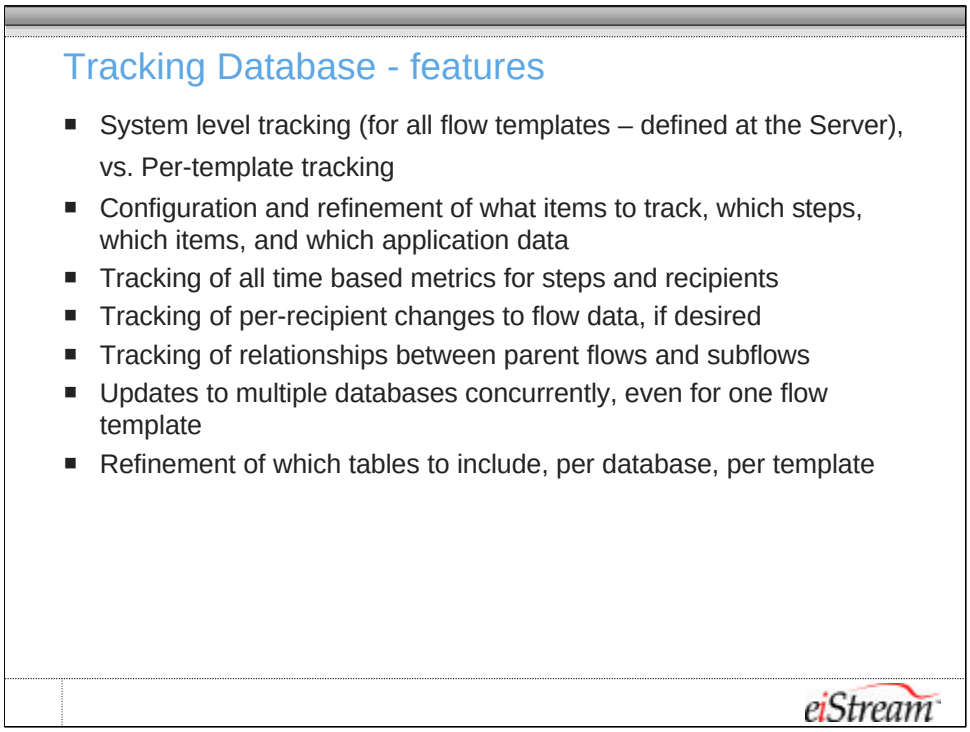
•A port of WebWorkflow to .NET

•And some new Installation work

We’ll look at each of these in a bit more detail …

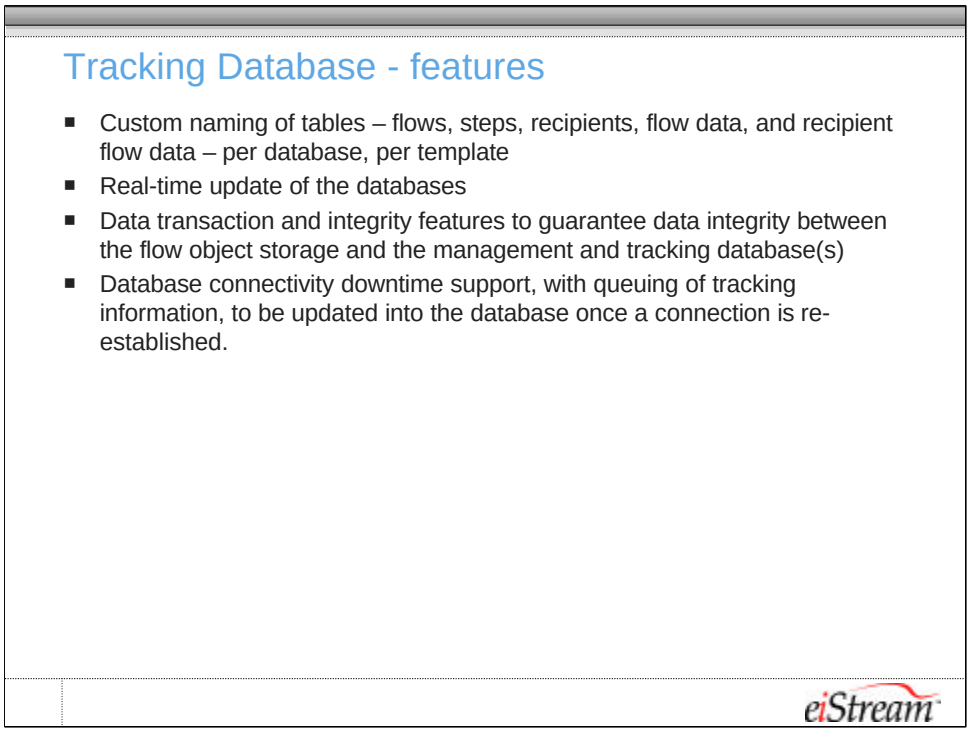


First, the Flow Management and Tracking Database … this is the big ticket item for this release, and provides a replacement of the existing logging and ODBC services, and a more comprehensive and configurable set of functionality.



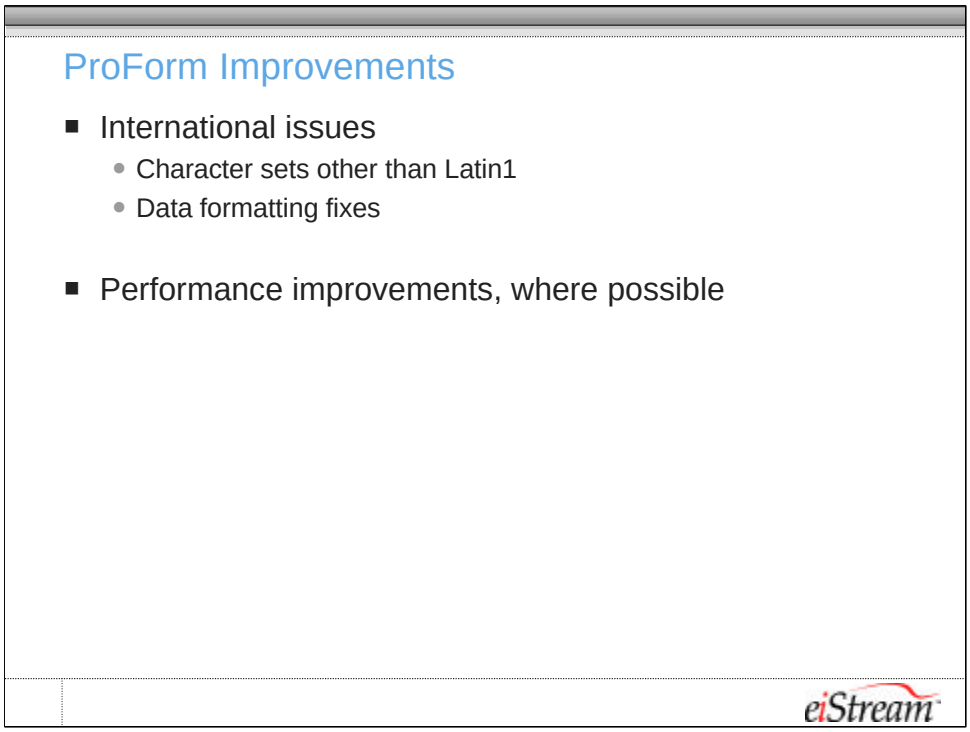
Some of the features include:

(read the slide)



(read the slide)

As you can see, the new Flow Management and Tracking Database features will provide a complete and comprehensive facility to manage and track all workflow information.

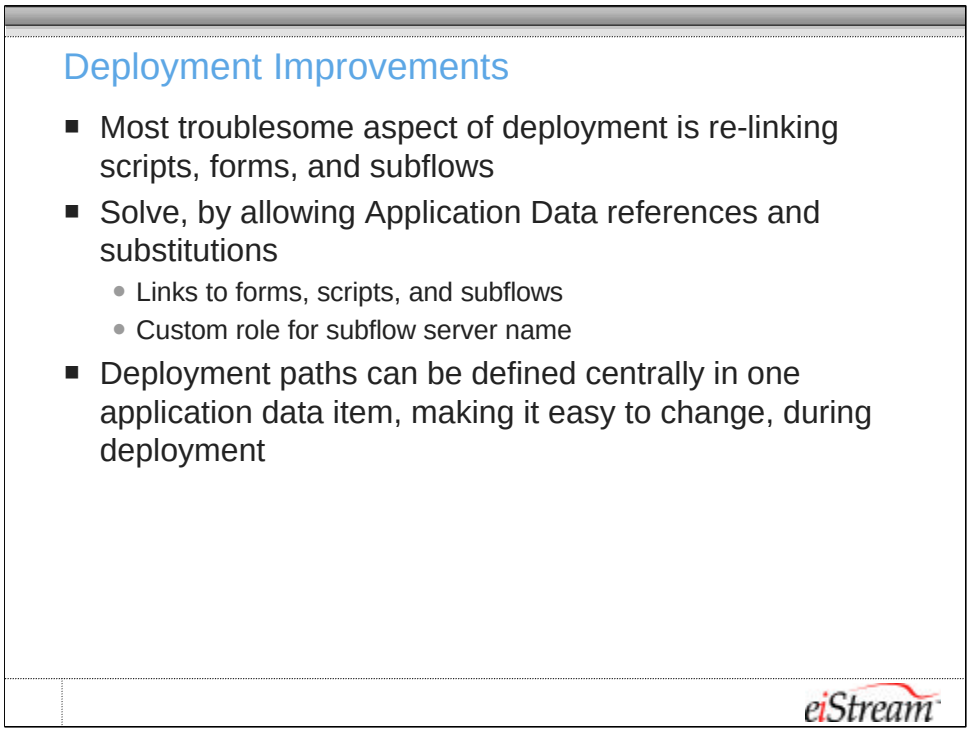


A number of ProForm improvements will be made.

Currently ProForms operates only with Latin1 character sets. This will be expanded.

Also, related to international issues, certain data formatting issues will be fixed, related to number fields, currency fields, and date time fields.

There have also been some performance issues, and these will be looked at … keeping in mind that there are some limitations inherent in the java applet environment itself, related to JVM and browser JavaScript interactions.



Regarding Deployment Improvements …

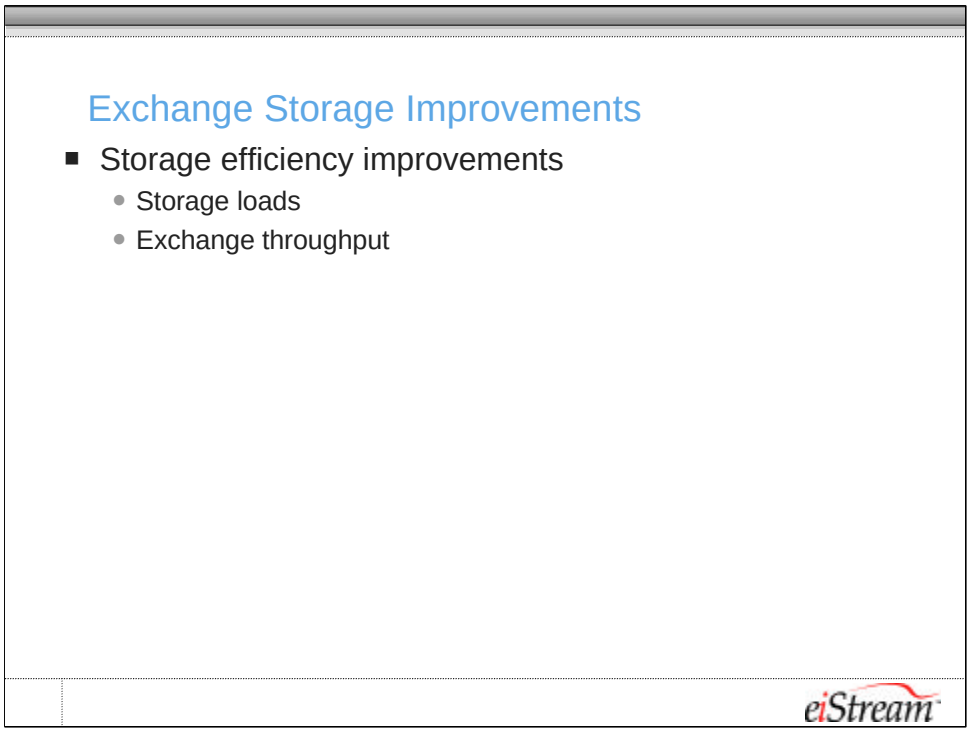
The most troublesome aspect, currently, of deployment – is that of re-linking the references to scripts, forms, and subflows … those references include public folder paths to where those items are stored.

This issue can be solved by expanding on facilities already present in the product – that of application data references and dynamic substitutions when the flow is running.

•We will allow application data references when defining links to forms, scripts, and subflows.

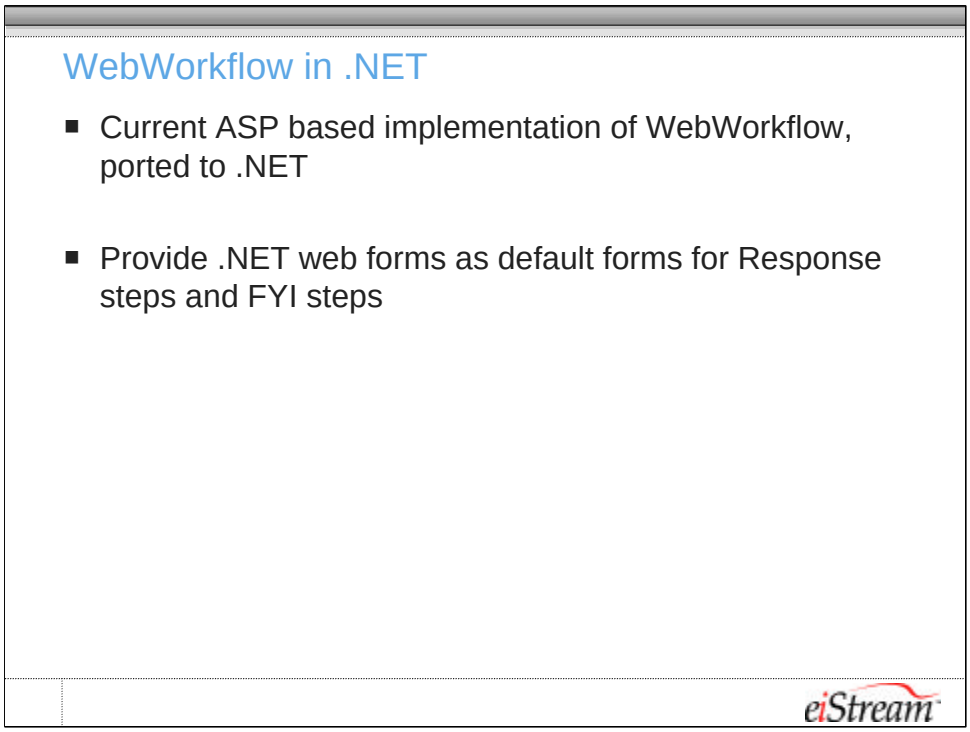
•And, we will allow use of a Custom Role when defining the Flow Server name to be used to launch a subflow.

This allows the flow template author to centralize platform and deployment data (within a single application data item and custom role), which can be changed quickly when a flow template is being deployed to a production environment.



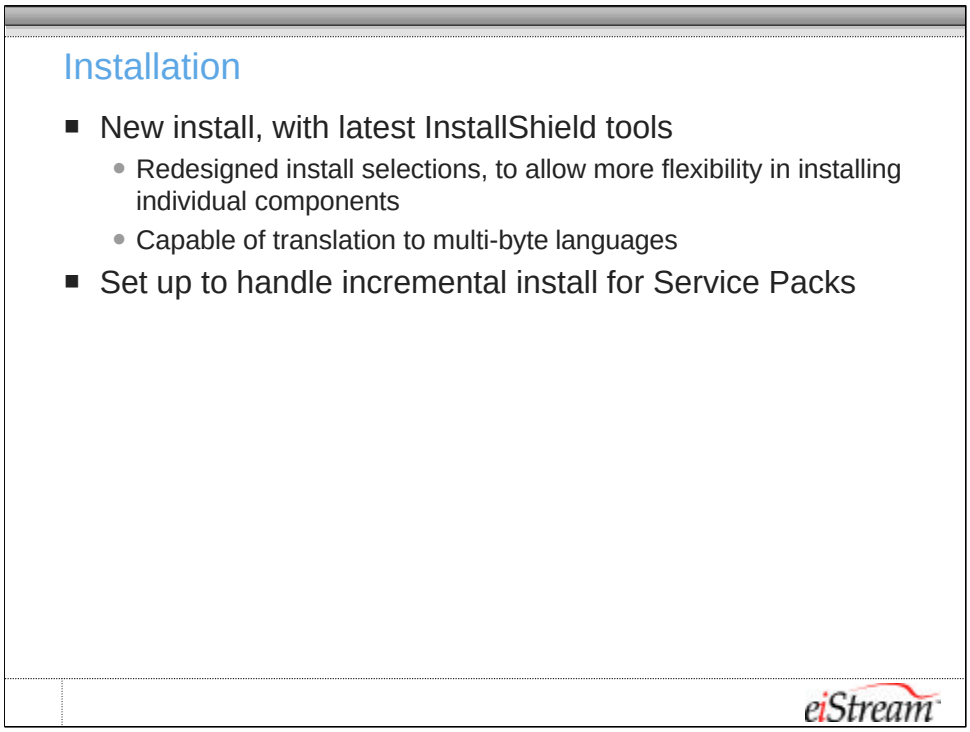
Keyflow, using Exchange as it's database storage repository, relies heavily on the integrity of that platform. Keyflow has made significant improvements in version 7.0 and 7.1 regarding data resiliency and recoverability.

One additional area that we would like to address, is looking at certain storage efficiencies. We will make some improvements regarding limiting the storage size and load that is placed on Exchange, and looking at any other data throughput issues that can improve out interaction with Exchange.



As part of Keyflow 8.0, we will provide WebWorkflow, with the current ASP based implementation ported to .NET. Also planned, are two .NET web forms that can be used as default forms for Response steps and FYI steps.

For those workflow applications that require more customized web interfaces, the installed .NET implementation can be modified. Having the full benefit of the .NET environment allows use of the latest Microsoft technology and tools.

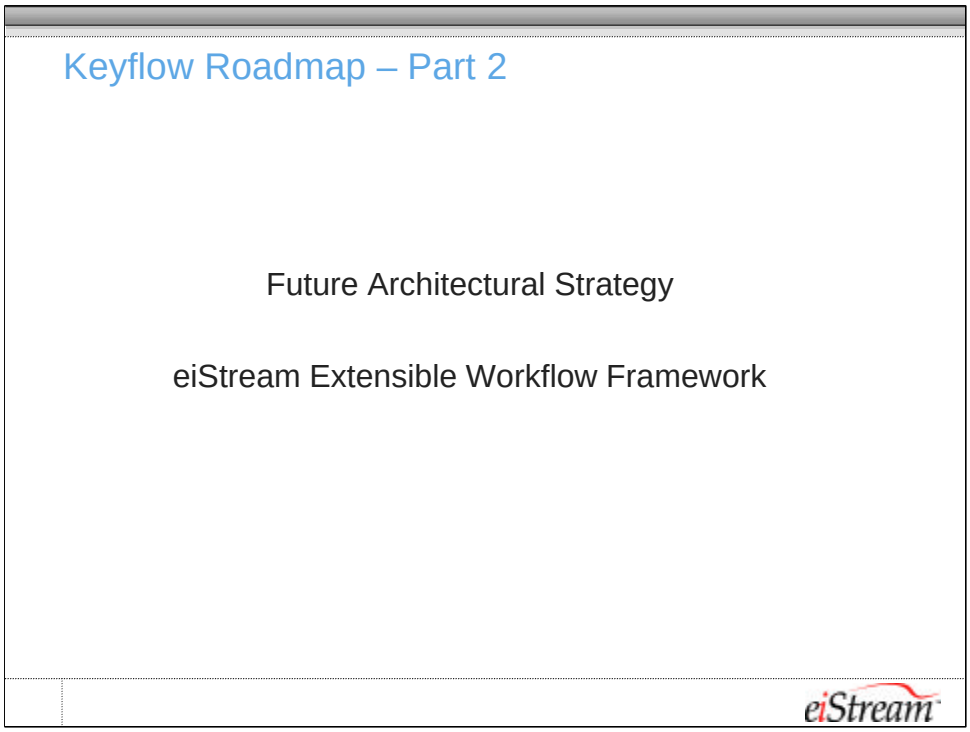


Regarding the Keyflow installation, we will build a new install, with the latest Install Shield tools:

•We will redesign the install selections, to allow more flexibility in installing individual components on different machines. This begins to become more important as more components, such a the Flow management and tracking database services, and the .NET WebWorkflow components, are added.

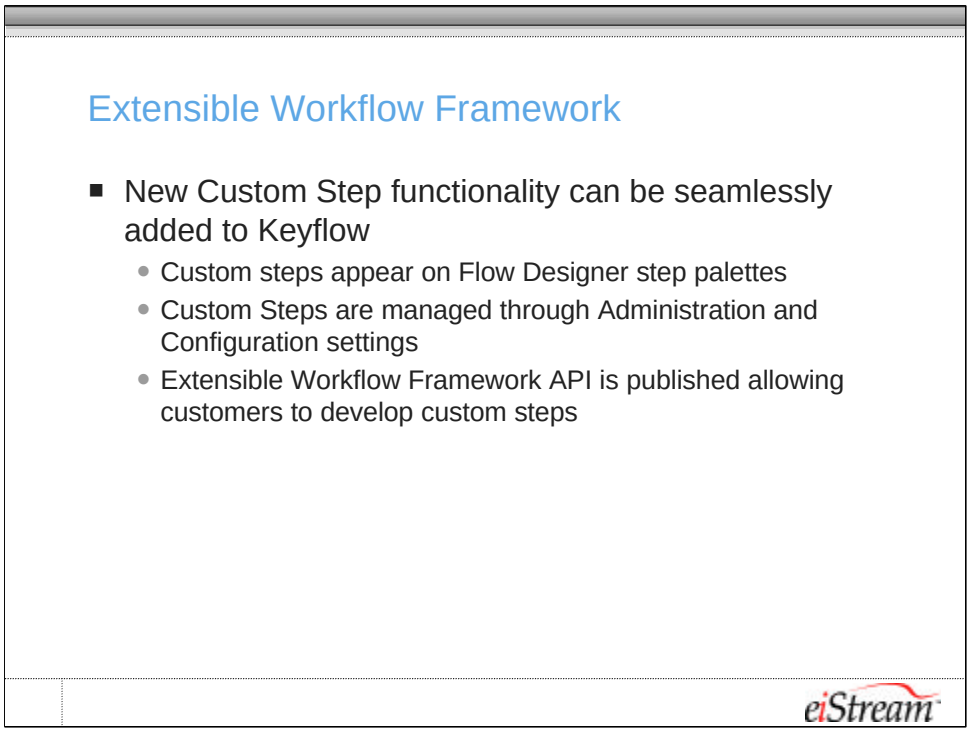
•We will provide an installation that can be translated to many other languages – the current install, because of its basis on an old version of InstallShield, has many limitations.

•We will also provide better streamlined service pack installations, rather than issuing full installations when incremental ones are better.



As a “Part 2” to this Roadmap discussion, looking at the point releases beyond release 8.0, we would like to introduce and explain a key future architectural strategy for Keyflow.

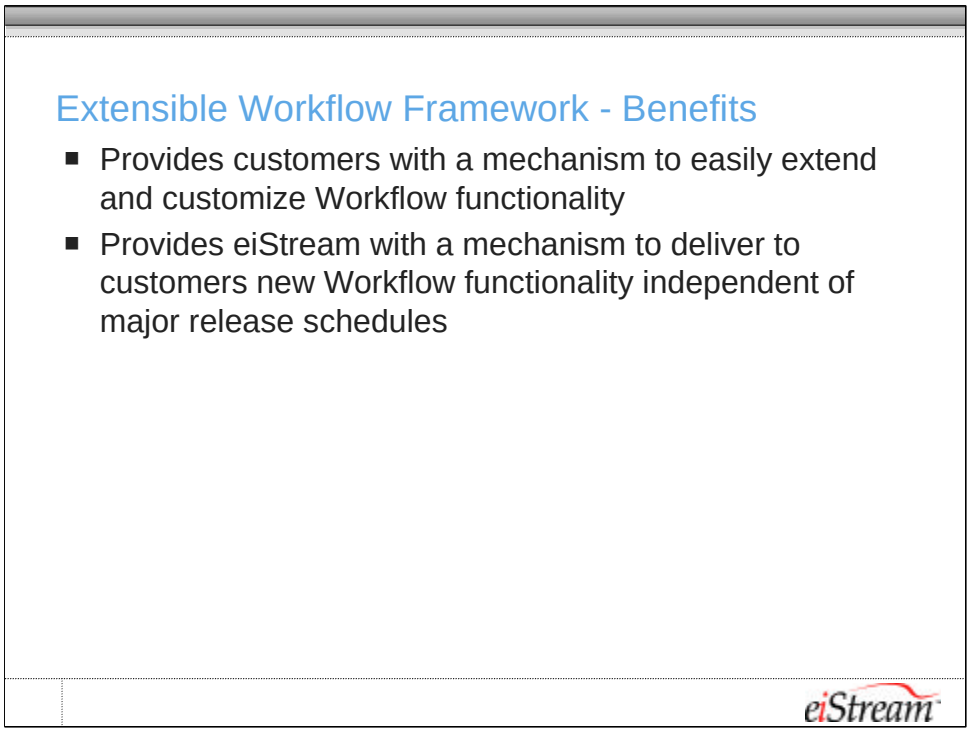
eiStream, to leverage the synergy of its workflow products, has developed an Extensible Workflow Framework, designed to become part of all of its products. This capability is part of the eiStream WMS product to be released in March of this year, and is planned for the WMS-J product this fall. Keyflow can greatly benefit from this technology, utilizing shared development efforts.



The Extensible Workflow Framework provides new custom step functionality, with new steps seamlessly added to Keyflow.

Custom steps will appear on Flow Designer step palettes, and will be managed through Administration and Configuration settings.

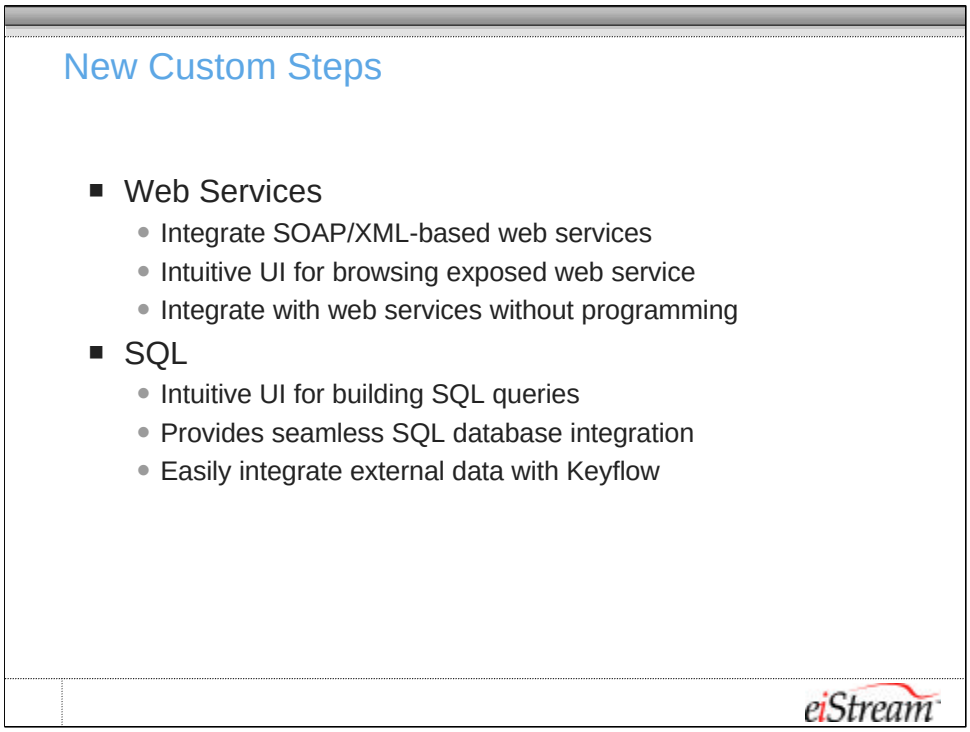
The Extensible Workflow Framework API will be published allowing customers to develop custom steps.



This provides customers and partners with a mechanism to easily extend and customize workflow functionality.

It also provides eiStream with a mechanism to deliver to customers new workflow functionality independent of major release schedules.

For Keyflow, some of this new workflow functionality will already be developed for other eiStream products, and will be able to be utilized with minimal effort by the Keyflow development team, to make it available to Keyflow customers.



Two such custom steps, which have already been developed for the eiStream WMS product, will be very valuable also, in Keyflow. These will be Out-of-the-box steps provided by Keyflow and managed within the Extensible Workflow Framework

First is a Web Services step.

•It provides the ability to integrate Keyflow within a SOAP/XML-based Web Services Infrastructure, with workflow data passed to and from the web service.

•It includes an intuitive User Interface for browsing the structure of the exposed web services.

•This allows Keyflow applications to be integrated with Web Services without programming.

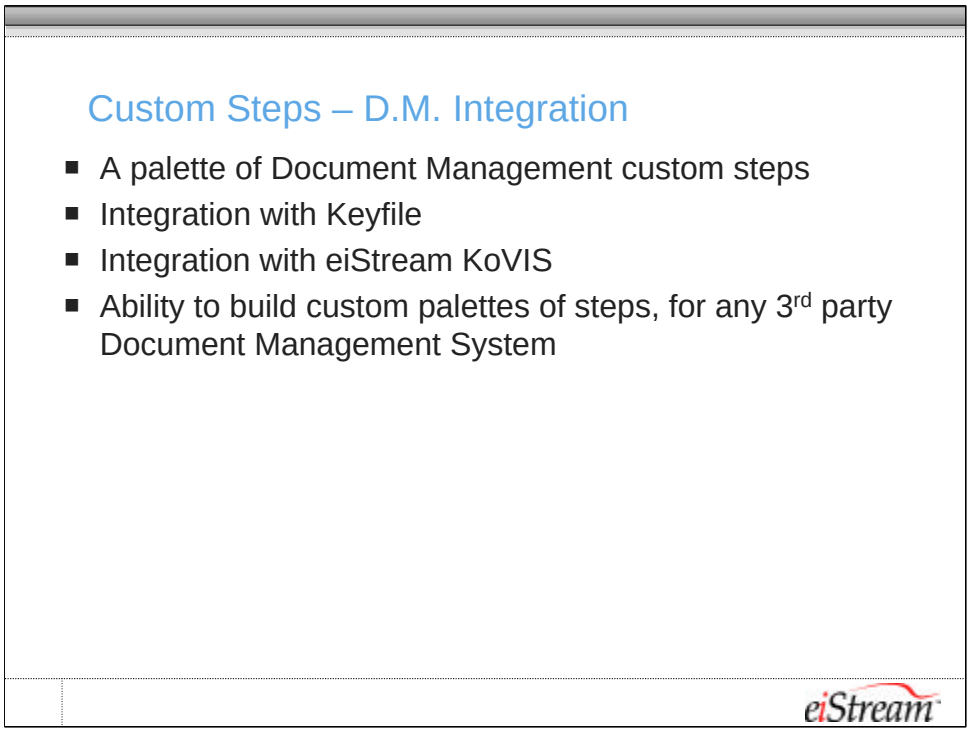
•This also allows Keyflow applications to be incorporated within a Corporate infrastructure that includes a Services-Oriented Architecture

The second custom step, is a SQL step.

•It includes an intuitive User Interface for building queries and stored procedures.

•This provides seamless SQL database integration to build SQL queries, without the need to develop custom code

•You will be able to easily integrate external data with Keyflow processes.



Once the Extensible Workflow Framework is built into Keyflow, it can be utilized to provide Document Management Integration features.

Palettes of custom steps will be built that provide tightly coupled functions that store and/or retrieve data and workflow attachments to and from a document management system. This can be done for Keyfile integration, eiStream KoVIS integration, as well as any other 3rd party Document Management system.

The Extensible Workflow Framework can be the basis for very interesting and compelling workflow functionality. We will be working toward integrating this technology into Keyflow in subsequent releases.


eiStream™
Streamlining Business Processes™