

## **Magic 9**

Magic eDeveloper ha sido diseñado para desarrollar aplicaciones Empresariales usando un paradigma de desarrollo basadas en Internet/Intranet.

## **Paradigma de desarrollo basado en el Browser**

La plataforma de Magic eDeveloper le permite crear aplicaciones donde el browser se utiliza como el cliente. El lado del cliente de este paradigma provee de toda la lógica de un Runtime manteniendo los controles HTML asociados a sus datos, tales como captura y manipulación de eventos. El cliente tiene una comunicación mínima con el servidor.

El servidor en este paradigma hace el procesamiento del Web, maneja la relación de tareas Magic, permite activar concurrentemente los programas y continua la ejecución de las tareas entre peticiones (request).

## **Componentes Magic**

Magic eDeveloper le permite definir objetos de aplicación como componentes que pueden ser compartidos con otras aplicaciones Magic. El exportar componentes permite compartir recursos entre aplicaciones.

La utilidad “Component Interface Builder” permite construir un archivo de Interfase de Componentes de los objetos mostrados en los repositorios de desarrollo de la aplicación.

Magic Versión 9 le permite ejecutar simultáneamente más de una aplicación. Esto se realiza definiendo la Aplicación como un componente. Un componente en este caso sería un archivo de aplicación Magic con una interfase. Usando esta interfase, otra aplicación Magic puede llamar a un objeto en la primera aplicación, y ejecutarla o usarla como si fuera parte de la aplicación huésped.

El uso de componentes provee los siguientes elementos de la Programación Orientada a Objetos:

- Encapsulación – La aplicación huésped es independiente a como el componente implementa su lógica.
- Polimorfismo – Un objeto puede revelar diferentes interfases a diferentes aplicaciones huésped, y proveyendo así diferentes capacidades.
- Herencia – Cuando se define un modelo en un componente, y se usa el modelo en la aplicación huésped, todas las propiedades del modelo son heredadas al objeto huésped.

## **Engine controlado por Eventos**

Magic eDeveloper ha creado manipuladores para implementar diferentes tipos de eventos. Así uno ya no se encuentra limitado por el flujo de operación de la tarea.

## ***Manejo de Eventos***

Magic le permite definir una lógica de respuesta a eventos implícitos y/o explícitos que pueden ocurrir en cualquier momento durante la ejecución de la tarea.

## ***Tipos de Eventos***

Los eventos en Magic están divididos en dos grupos:

- 1) Eventos Predefinidos:
  - a) Internos
  - b) De Sistema
- 2) Eventos definidos por el Usuario:
  - a) Temporizados
  - b) Por Expresión

## ***Eventos de Operación***

Los Eventos de Operación, son aquellos que suceden por el flujo de operaciones del programa en ejecución.

- 1) Eventos de Operación
  - a) A Nivel del Ciclo de Ejecución
  - b) A Nivel Control
- 2) Eventos de Error

## ***Administración de Datos***

La administración de datos (Data Management) le permite organizar los datos tal y como están almacenados en la base de datos física. Las nuevas características en la administración de datos son::

- Transacciones diferidas: retardan la implementación de las sentencias de Manejo de Datos (DM) hasta el momento en que se salven los cambios (commit) hechos a los registros de la vista de datos. Magic almacena cada sentencia DM en un caché. Cuando está listo para guardar los cambios (commit) a la base de datos, todos los cambios son implementados “de golpe”.
- Integridad referencial: El programador Magic podrá definir la relación entre las tablas. Esta característica será usada para obtener más información acerca de la tabla mientras se recibe su definición, para crear ‘constraints’ en la base de datos mientras se crea la tabla desde Magic (en bases de datos SQL), y para APG en la tabla.
- Uso funcional de la condición del Link: la liga será ejecutada solo si la condición evaluada es verdadera (True). La condición opcionalmente participa en el mecanismo de recálculo (recompute mechanism).

- Cláusula Where de Magic: donde el programador, utilizando únicamente sentencias Magic, podrá realizar consultas avanzadas en SQL.

## Manipulación de Errores

La manipulación de errores le permite modificar el comportamiento por omisión de la aplicación en ejecución. Los desarrolladores pueden escribir manipuladores para respuestas específicas a diferentes errores que pueden presentarse durante la ejecución.

Magic provee de una lista de los errores conocidos y esperados que pueden ser manipulados; y además le permite manejar otros errores específicos al DBMS.

### ***Estrategias de Comportamiento en Errores***

Magic provee de dos estrategias predefinidas (dentro de las Propiedades de la tarea):

***Abortar (Abort Strategy):*** Cuando ocurre un error, Magic ejecuta un 'rollback' a la transacción actual. En una transacción Física o Diferida, remueve la vista de datos actual, y aborta la tarea en la cual ocurrió el error. Esta estrategia no ejecuta el rollback en la transacción actual cuando el usuario puede corregir el error, tal como en el bloqueo (locking) o al registrarse incorrectamente.

***Recuperar (Recover Strategy):*** Cuando ocurre el error, Magic tratará de mantener la vista actual de datos, tratará de permanecer en la tarea, y permitirá al usuario recuperar el error. Esta estrategia le permite al usuario continuar con la ejecución de la aplicación.

## Procesamiento de Múltiples Peticiones

En Magic eDeveloper, el servidor Magic de aplicación en background puede procesar múltiples peticiones (requests) al mismo tiempo. Cada proceso corre en un diferente contexto de Runtime y no interactúa con otros procesos. Los servidores de aplicación Online, tanto en Toolkit como en Runtime mantienen el uso de Peticiones y procesos simples.

Este es por supuesto, un cambio mayor en la arquitectura, que le permite a un desarrollador tener un simple proceso Magic que permitirá correr muchos hilos en los cuales, cada uno tendrá la funcionalidad completa de un engine de Magic.

Esto significa que un proceso en un engine de Magic Versión 9 puede reemplazar varios engines de Magic Versión 8.

Habiendo reemplazado los múltiples procesos de Magic por múltiples hilos, se ahorran muchos de los recursos de la máquina y el performance alcanzado es mucho mayor.

## Mejoras al GUI

Un panel de navegación con múltiples carpetas, un control “table” mejorado y el uso de comentarios para los objetos, todos forman parte de una interfase dinámica fácil de utilizar.

En Magic Versión 9 el control Table ha sido ampliamente mejorado:

- El control Table ahora puede ser colocado con configuraciones de posición (placement) completas; o sea, puede ser aumentada o decrementada con el cambio del tamaño de la forma, además habilitando las columnas y su contenido para ser posicionado también.
- Permite modificar el ancho de las columnas en modo Runtime. Ordena automáticamente cuando al título de la columna se le hace un click.
- Permite marcar múltiples renglones y la ejecución de cualquier lógica definida por el usuario para todos los registros marcados, o el uso de operaciones internas como borrar (Delete).
- Permite manipular las columnas de la tabla individualmente, configurando propiedades como color, ancho, y visibilidad.
- Magic 9 SP1 agregó dos propiedades a los objeto Columna: Top Border & Right Border; los cuales permiten hacer modificaciones a los bordes de las celdas en la tabla.

Al Toolkit de Magic Versión 9 se le han realizado mejoras mayores:

- Primero, sus elementos de interfase han sido enriquecidos con controles mejorados como combo ‘boxes’ y ‘list boxes’ con funcionalidad de catálogo (selection list).
- El nuevo control Table con sus nuevas características.
- Su interfase ha sido actualizada para proveer una vista y una sensación tipo Windows.
- Provee de una utilidad navegadora que da una vista consistente de la aplicación entera conforme usted trabaja. Por ejemplo, puede dividir las entradas en cada repositorio dentro de fólderes, puede crear marcadores para varias posiciones en la aplicación para un acceso rápido, puede ver y trabajar con los resultados de la Referencia Cruzada durante su trabajo, etc...
- Una nueva hoja de propiedades flotante ha sido introducida dentro de las partes del Toolkit.

## Mejoras para Productividad

Las siguientes mejoras para productividad han sido incluidas en Magic eDeveloper:

- **Numeric column incremental update:** La propiedad 'Update Style' en las propiedades del Campo (Field Properties) pueden tener los siguientes valores (es el similar, implementado a los modelos, del 'Update How: Normal/Incremental'):
  - Absolute* – un valor fijo permitido por las versiones previas de Magic (FLD1=X)
  - Differential* – valores diferenciales (FLD1=FLD1+X)
- **Transacciones diferidas:** retardan la implementación de las sentencias de Manejo de Datos (DM) hasta el momento en que se salven los cambios (commit) hechos a los registros de la vista de datos. Magic almacena cada sentencia DM en un caché. Cuando está listo para guardar los cambios (commit) a la base de datos, todos los cambios son implementados “de golpe”.
  - Hay cuatro modos de transacción:
    - Diferido - Deferred,
    - Diferido Anidado - Nested Deferred,
    - Físico - Physical, y
    - Como el Padre - As parent.
- **Statement invocation order:** La propiedad 'Sync Data Before Call' en la operación Call puede tener los siguientes valores:
  - No – igual que en las versiones previas
  - Yes – el engine de Magic implementa los cambios al registro antes de ejecutar la operación Call
- **Bloqueo de Recursos (Resource locking):** La habilidad para crear una entidad imaginaria (un recurso) que puede ser controlado por exactamente un usuario en cualquier momento. Solo el usuario que bloqueo el recurso puede desbloquearlo. Para esto hay dos nuevas funciones: LOCK y UNLOCK.
- **SQL Range:** El SQL Where de Magic será definido usando un conjunto de expresiones Magic (a diferencia del 'SQL Where range' existente, el cual usa texto de formato libre). El “Magic SQL Where range” solo puede incluir esas expresiones que el engine sabe traducir a SQL (en la sintaxis apropiada para cada base de datos SQL).
- **Referencias Cruzadas Mejoradas:**
  - Referencias Cruzadas de más objetos como: Pantallas de ayuda, Derechos, Menús, Modelos, Eventos, Componentes.
  - La Referencia Cruzada será a través de los repositorios (y no solo en el repositorio de Programas)
  - Accesar a un programa para encontrar la posición exacta de una referencia en un objeto.
  - Imprimir los resultados de la Referencia Cruzada
  - El programador puede hacer una referencia cruzada de un objeto desde un conjunto de resultados
- **Uso de un Programa Principal (Main Program):** permitirá que el programador defina variables, eventos, y Manipuladores de eventos que estarán disponibles para su

uso en toda la aplicación. El Programa Principal será ejecutado tan pronto Magic entre a una nueva aplicación en modo Runtime, y terminará justo antes de que Magic salga de la aplicación en modo Runtime. Esto puede ser utilizado tanto para programas de inicio y de finalización.

- **Controles de datos:** asignar directamente una tabla de la base de datos a un control (List o Combo Box), para que obtenga de este sus datos correspondientes. Usando esta característica el desarrollador puede conectar un control a la vista de datos de una tabla Magic y permitirá un valor para ser seleccionado directamente de las tablas de las bases de datos. Esta característica le permite distinguir entre el valor mostrado y el valor actual del campo. El control de datos participa en el mecanismo de re-cálculo, lo que implica que una nueva lista es construida una vez que el rango usado ha sido cambiado.
- **La implementación de Modelos:** Un modelo es un conjunto de propiedades heredadas para un tipo específico de objeto. Magic le permite definir modelos para los siguientes objetos:
  - Campos (Fields)
  - Controles (Controls)
  - Formas (Forms)
  - Pantallas de ayuda (Help Screens)

Cuando las propiedades de un modelo son actualizadas, los valores para cada objeto asociado se reflejan en todos los valores de sus propiedades excepto para aquellos que han sido definidos individualmente. Los Modelos de objetos sirven como una plantilla para un objeto. Cada objeto es automáticamente asignado a un conjunto de propiedades por default. Todo objeto asociado al Modelo tendrá las mismas propiedades. Usted puede cancelar la herencia de propiedades, simplemente cambiando el valor de la propiedad directamente en el objeto.

- **Archivo Plano de Aplicación Magic (MAF/MFF):** El archivo de aplicación Magic (Flat Magic Application File - MAF) le permite ejecutar un archivo de Aplicación independiente de la base de datos. La aplicación, la cual es creada como un archivo binario, puede ser almacenada en cualquier lugar en la computadora del usuario y es utilizada para ejecución (deployment) únicamente. Los archivos MAF fueron creados para ser usados en un ambiente multiplataforma. Por ejemplo, puede transferir un archivo MAF de Windows NT a UNIX sin tener que importar o exportar la aplicación.

## Mejoras al Toolkit

Las siguientes mejoras al Toolkit han sido agregadas a Magic Versión 9:

- **Date Time:** Magic puede soportar el almacenamiento de tipos Date-Time tal y como se definen en la mayoría de las bases de datos. Diferentes funciones que soporten este tipo de almacenamiento estarán disponibles.

- **Engine basado en ANSI:** los engines Toolkit y Runtime de Magic mantendrán todos los lenguajes extranjeros en el estándar ANSI y no en OEM como se venía haciendo hasta el momento.
- **Select Parameter:** Permitirá la declaración explícita de que una variable virtual es un parámetro. Cuando se realiza una llamada a un programa en modo Toolkit, Magic mostrará la lista de los parámetros definidos y permitirá al desarrollador seleccionar cualquiera de ellos.
- **If-Else Block:** Un bloque ELSE ha sido agregado
- **Comentarios:** El desarrollador podrá agregar comentarios a cada entrada en cualquier repositorio (En el menú *Edit/Comment* o la tecla F10)
- **Valor de retorno de la tarea:** Una tarea tendrá una nueva propiedad llamada Return Value y será asignada con una variable o una expresión. Después de terminar un programa, este valor será regresado. Una nueva función llamada CALLPROG() será agregada, a través de la cual un programa podrá ser llamado definiendo una expresión. Esta función será evaluada como el valor de retorno del programa llamado.
- **Debugger:** Este permite:
  - Crear, borrar, condicionar y deshabilitar los marcadores (breakpoints)
  - Ver marcadores (breakpoint Watch) dependiendo de un cambio en el valor de una cierta variable
  - Actualizar variables durante la Ejecución (Runtime)
  - Ver el flujo de Magic en la tabla Toolkit flow
  - Ver valores de expresiones y variables
  - Ver el "Call stack," el cual es el árbol de invocación del Runtime

## Uso de tecnologías Emergentes

### Web Services

La Versión 9.30 de Magic eDeveloper incluye una poderosa herramienta para proveer Web Services, mediante la exposición de nuestros métodos, generados originalmente en lenguaje Magic, utilizando los estándares WSDL (Web Services Descriptive Language) y SOAP (Simple Object Access Protocol), para contestar peticiones hechas desde Internet/Intranet.

Además, Magic eDeveloper 9.30 permite utilizar Web Services ya existentes (sean de terceros o generados por el propio eDeveloper), de una manera muy sencilla, a través de un asistente; y así, integrarlos en nuestra aplicación, sea esta en C/S o en Internet/Intranet.

## **EJB's**

Los Enterprise Java Beans (EJB) es una tecnología emergente que parte del conjunto de especificaciones J2EE.

J2EE es un grupo de *especificaciones* diseñadas por Sun que permiten la creación de aplicaciones empresariales, esto seria: acceso a base de datos (JDBC), utilización de directorios distribuidos (JNDI), acceso a métodos remotos (RMI/CORBA), funciones de correo electrónico (JavaMail), aplicaciones Web(JSP y Servlets)...etc.

Aquí es *importante notar que J2EE es solo una especificación*, esto permite que diversos productos sean diseñados alrededor de estas especificaciones; algunos ejemplos son SunReference, JBoss, WebSphere y Weblogic ;

La arquitectura de los Enterprise Java Beans permite una manera de re-utilizar componentes de software que pueden ser manipulados en herramientas de desarrollo("Builder Tools"). Estos "Beans" pueden ser tan sencillos como un botón, o complejos como el acceso a una base de datos; una característica primordial de un JavaBean son los métodos (funciones).

Magic eDeveloper nos permite generar EJB's. De esta manera, clientes operando en un ambiente J2EE pueden ver y activar programas Magic, como si fuesen métodos EJB's.

## **XML**

XML, es el estandar de Extensible Markup Language. XML no es más que un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados.

XML fue creado al amparo del Word Wide Web Consortium (W3C) organismo que vela por el desarrollo de WWW partiendo de las amplias especificaciones de SGML. Su desarrollo se comenzó en 1996 y la primera versión salió a la luz el 10 de febrero de 1998. La primera definición que apareció fue: *Sistema para definir, validar y compartir formatos de documentos en la web.*

Respecto a sus objetivos son:

- XML puede ser directamente utilizable sobre Internet.
- XML debe soportar una amplia variedad de aplicaciones.
- XML debe ser compatible con SGML.
- Debe ser fácil la escritura de programas que procesen documentos XML.
- El número de características opcionales en XML debe ser absolutamente mínima, idealmente cero.
- Los documentos XML deben ser legibles por humanos y razonablemente claros.
- El diseño de XML debe ser preparado rápidamente.
- El diseño de XML debe ser formal y conciso.

- Los documentos XML deben ser fácilmente creables.
- La concisión en las marcas XML es de mínima importancia.

Esta especificación, junto con los estándares asociados (Unicode e ISO/IEC 10646 para caracteres, Internet RFC 1766 para identificación de lenguajes, ISO 639 para códigos de nombres de lenguajes, e ISO 3166 para códigos de nombres de países), proporciona toda la información necesaria para entender la Versión 1.0 de XML y construir programas de computadora que los procesen.

Es una arquitectura más abierta y extensible. No se necesitan versiones para que puedan funcionar en futuros navegadores. Los identificadores pueden crearse de manera simple y ser adaptados en el acto en Internet/Intranet por medio de un validador de documentos (parser). La extensibilidad y flexibilidad de este lenguaje nos permitirá agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.

Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido web en XML consigue que la estructura de la información resulte más accesible. Se desarrollarán de manera extensible las búsquedas personalizables y subjetivas para robots y agentes inteligentes. También conllevará que los clientes web puedan ser más autónomos para desarrollar tareas que actualmente se ejecutan en el servidor.

El concepto de "hipertexto" se desarrollará ampliamente (permitirá denominación independiente de la ubicación, enlaces bidireccionales, enlaces que pueden especificarse y gestionarse desde fuera del documento, hiperenlaces múltiples, enlaces agrupados, atributos para los enlaces, etc. Creado a través del Lenguaje de enlaces extensible (XLL).

## **Integración con XML**

Magic eDeveloper 9.30 provee los elementos para poder realizar una validación y obtención de datos (parsing) provenientes de un archivo XML, de una manera simple.

A través de funciones de las funciones XML de Magic, podemos acceder fácilmente a los documentos XML e integrarlos a nuestras aplicaciones. Con estas funciones podemos:

1. Recibir datos de varios objetos XML
2. Verificar la existencia de objetos XML
3. Localizar objetos XML en el documento
4. Contar las ocurrencias de un objeto XML en específico

## **XML Component Generator (XCG)**

Magic eDeveloper 9.30 le permite manipular cualquier archivo XML para el cual exista un Esquema XML. El utilizar la herramienta XML Component Generator, nos permite crear una aplicación componente que pueda acceder y manipular la información del archivo XML.