مرجع کامل

Multimedia Builder 4.9.7

The Complete Reference

Unleash Your Creativity

- محیط کاملا مبتنی بر اشیا
- برنامه نویسی سریع،آسان و درعین حال قدرتمند
- سازگاری کامل با فرمتهای صوتی و تصویری متداول
- امکان تولید محصول در عرض چند دقیقه
- قابلیت توسعه توسط Plug-In
- قابلیت ساخت برنامه های بزرگ چند رسانه ای
- قابلیت ارتباط و استفاده از برنامه های دیگر نظیرFlash,Photoshop و...
- تولید برنامه های چندرسانه ای در عرض چند دقیقه

تالیف : حامد تکمیل

مرجع کامل MMB

تالیف : حامد تکمیل

info@mmbstation.com

www.mmbstation.com

فهرست مطالب

مقدمه :

در طی ۶ سالی که با نرم افزار MMB کار می کنم، احساس می کنم که جزئی از زندگی من شده. چرا که تقریبا هر روز در حال استفاده از MMB هستم. این باعث شده که من رابطه ی بسیار خوبی با ان برقرار کنم و بتوانم مهارت ها و قابلیت های خودم را توسعه دهم. به همین منظور خود را وام دار و مدیون جامعه MMB می دانم و همین احساس مرا بر آن داشت تا اقدام به تهیه این کتاب نمایم

من برای نوشتن این کتاب در حدود ۱۱ ماه وقت صرف کردم و در این زمان از کمک بسیاری ازدوستان خوبم بهره بردم که جا دارد از آنها تشکر کنم. در ابتدا من به قصد چاپ این اثر اقدام به نوشتن مطالب کردم، اما به لحاظ مشکلات بسیار و فراز و نشیب های بسیاری که پشت سر گذاشتم ، تصمیم گرفتم تا آن را به صورت کتاب الکترونیکی تقدیم علاقه مندان نمایم. هم اکنون ۲ عنوان کتاب برای MMB موجود است که بنده سعی نموده ام که در این اثر مطالب بیشتری به نسبت آنها ارائه کنم تا غالب نیازهای علاقه مندان برطرف شود.

در این مرجع سعی شده است تا نیاز غالب مخاطبان برآورده شود و با ارائه مطالب ساده و پیشرفته در کنار هم به نیاز مخاطبان پاسخ گفته شده است. نوشته ای را که پیش روی خود دارید به عنوان مرجعی کامل برای برنامه مولتی مدیا بیلدر محسوب می شود. کمتر کسی است که در عرصه مولتی مدیا فعالیت داشته باشد و نام این برنامه را نشنیده باشد و یا با آن کارنکرده باشد، لذا با توجه به این مطلب و نیاز روز افزون مخاطبان (که میتوان پرسش های مکرر آنان در اینترنت را گواهی بر این مطلب دانست) بر ان شدم تا مرجعی کامل به زبان فارسی برای این برنامه مفید تهیه کنم. در این نوشته مطالب پیشرفته از قبیل برنامه نویسی به گونه ای کاملا ساده بیان شده است و پس از پایان مطالع مطلب از جانب مخاطبان می توان از آنان انتظار تولید هر گونه محصول چند رسانه ای را داشت.

این مرجع در ابتدا با معرفی برنامه MMB آغاز می شود و در ادامه برای آشنایی بیشتر خوانندگان خود آموزی قرار داده شده است. در ادامه به تدریج رابط کاربر برنامه مورد بررسی قرار گرفته است. پس از آن در فصلی مفصل مفاهیم برنامه نویسی و پیشرفته تر این برنامه بیان شده است. در این بخش مطالب به همراه مثال های متعددی ارائه شده است. در بخش بعدی برای پاسخگویی به نیاز کاربران حرفه ای طریقه استفاده از Plug-In های این برنامه به همراه آموزش چند Plug-In قدرتمند این برنامه آمده است. به جرات می توان گفت که تا کنون هیچ مطلبی در این رابطه به زبان فارسی برای مخاطبان ارائه نشده است. در نهایت بخش تولید نهایی و ضمیمه ارائه شده است که حاوی نکات تکمیلی و مطالبی جهت تولید بهینه محصولات می باشد.در این بخش برای تثبیت مطالب مشروحه در فصل های قبلی خود اموزی جامع جهت تولید یک محصول ارائه شده است.برای کامل شدن مرجع یک مجموعه کم یاب از مثال ها و Plug-in های این برنامه بر روی یک سی دی که حاصل ٦ سال جستجو می باشد جمع اوری شده است که به مخاطبان عرضه می شود. جهت رفع نیاز های احتمالی مخاطبان پایگاهی اینترنتی به آدرس www.mmbstation.com به دو زبان فارسی و انگلیسی بر روی شبکه جهانی قرار گرفته است.

مطالب ارائه شده در این مرجع حاصل ترجمه مطالب متعدد ارائه شده و در بخش هایی تالیفی می باشند. که درصد مطالب تالیفی بیشتر می باشد. از آنجاییکه شخصا در تهیه راهنمای اصلی (زبان انگلیسی) برنامه مشارکت داشته و جز تیم بتای توسعه این برنامه می باشم روز آمد بودن و کارآمدی این مرجع را تایید نموده و آن را تقدیم مخاطبان می کنم.

از دیگر مزیت های این نرم افزار می توان به سایر نرم افزار های تکمیلی که برای این نرم افزار تولید شده اند نام برد که از جمله آنان نرم افزار RealDraw می باشد که نرم افزاری برداری جهت تولید عناصر گرافیکی می باشد و به شهادت کاربرانش بسیار ساده و در عین حال قدرتمند می باشد. این برنامه با ارائه خروجی مختص برنامه MMB فرآیند تولید برنامه های چند رسانه ای را بسیار آسان کرده است. همچنین این برنامه کاملا با نرم افزار Photoshop ساز گار بوده که می تواند قدرتش را دو چندان کند. با توجه به این مطالب یک طراح برنامه های مولتی مدیا غالب نیاز های خود را در این مجموعه یافته و جلب آن نیز می شود.

در پایان از اینکه این مرجع را برگزیدید از شما تشکر می کنم. بی صبرانه منتظر نظرات، پیشنهادات و انتقادات شما هستم.

حامد تکمیل

زمستان ۱۳۸٤

تقدیم به :

فهمیه راهی

مژگان کرمی

محسن نجفی

محمد نجفی

محمد زارع

هادی تکمیل

و همه علاقه مندان به برنامه MMB

فصل اول

مقدمات

Multimedia Builder

- معرفی نرم افزار Multimedia Builder
- کار با MMB را تجربه کنید

در ایـن بخـش سـعی شـده اسـت تـا تـصویری کلـی از برنامـه Multimedia Builder ارائـه شـود.بـه همـین منظـور ابتـدا ویژگی های شاخص این برنامـه مـورد بررسـی قـرار مـی گیـرد و در ادامـه بـا بررسـی خـود آمـوزی رونـد آشـنایی بـا برنامـه تکمیل می شود.

معرفی نرم افزار Multimedia Builder :

MMB (مولتی میدیا بیلدر، به اختصار MMB) نرم افزاریست جهت طراحی و تولید برنامه های چند رسانه ای[1] که مبتنی بر سیستم عامل ویندوز می باشد.اگر تا به حال توسط زبان های برنامه نویسی دیگری اقدام به تولید نرم افزارهای چند رسانه ای کرده باشید،حتما می دانید که مراحل نسبتا پیچیده ای را باید بپیمایید تا به نتیجه مطلوب برسید.وحتما در حین کار آرزو کرده اید که کاش می شد راه حل تازه تر وآسان تری را یافت.اما باید بگویم که راه حل هم اکنون پیش روی شماست و آن چیزی نیست جز MMB،نرم افزاری کوچک با قابلیت های بزرگ که کلیه ابزارهای مورد نیاز تولید برنامه های چند رسانه ای را در اختیار شما قرار می دهد.با گذشت زمان و کار با این برنامه خواهید فهمید که به آن علاقه مند شدیدو علل آن هم در سادگی کار با MMBو قدرتمندی آن نهفته است.حال جهت آشنایی بیشتر به توضیح اجمالی توانایی ها وخصوصیت های MMB می پردازیم:

ـ یادگیری آسان وسریع:

شما به عنوان یک طراح برنامه های چند رسانه ای با استفاده از MMB وبدون صرف هزینه و وقت جهت دوره های آموزشی این توانایی را خواهید یافت که دست به تولید و توسعه برنا مه های چند رسانه ای مورد نیاز خود بزنید.

ـ برای خود CDهای AutoRun(خود اجرا)بسازید

ـ چه نوع برنامه های چند رسانه ای را می توان با استفاده از MMBتولید نمود؟

- CD های AutoRun
- خودآموزهای کامپیوتری چند رسانه ای
- کارت ها ،راهنما ها و بروشورهای محصولات
- CDهای صوتی به همراه برنامه های اجرای آنها
- برنامه های چند رسانه ای مورد نیاز شخصی یا اداری
- برنامه های کوچک کاربردی تحت ویندوز

– در فرآیند تولید یک محصول چند رسانه ای چه امکاناتی در اختیار طراح است؟

- محیط WYSIWYG (What You See Is What You Get)
- محیط طراحی مبتنی بر اشیا
- سازگاری در بارگذاری و استفاده از فرمت های متداول گرافیکی از قبیلGIF,SWF و...
- جلوه های ویژه برای تغییر تصاویر
- امکان اسکریپت نویسی برای کنترل جزئیات بیشتر

۱–به برنامه هایی گویند که از طریق پخش صوت و تصویر با کاربر ارتباط برقرار می کنند.

- تولید پنجره نهایی پروژه به شکل دلخواه در زمان طراحی واجرا
- طراحی اشیا در لایه های مختلف
- Plug-In های متنوع و قدرتمند جهت تولید برنامه های قویتر

در نهایت با استفاده از MMB می توانید تولیدات خود را به صورت یک فایل اجرای مستقل (exe.*) در آورده و مورد استفاده قرار دهید.

چه کسانی باید این کتاب را بخوانند؟

کتاب حاضر برای کسانی است که هیچ تجربه ای در ساختن نرم افزارهای چند رسانه ای ندارند.البته کاربران حرفه ای نیز می توانند به جمع خوانندگان این کتاب بپیوندند ، چرا که راهی جدید رابرای تولید برنامه های مورد نظر خود خواهند یافت.این کتاب به عنوان مرجعی برای برنامه MMB به حساب می آید.این کتاب ابتدا بر روی اصول اولیه متمرکز می شود و سپس به سراغ مباحث پیشرفته از جمله Plug-In ها می رود .حداقل معلوماتی که برای شروع کار با MMB مورد نیاز است ، آشنایی با سیستم عامل ویندوزمی باشد

کار با MMB را تجربه کنید!

مطمئنا هنوز MMB برای شما ناشناخته است،به همین منظور خود آموزتولیدیک برنامه ساده را در ادامه قرار داده ایم تا به کنکاش و کسب تجربه بپردازید،مسلما قضاوت شما پس از انجام این خود آموز درباره MMB نظر نهایی شما در باره MMB خواهد بود.خب آماده شوید تا عملا کار باMMB را تجربه کنید.دراین خود آموز فرا می گیرید که چگونه برنامه Slideshow راکه برای مشاهده عکس ها می باشد،بنویسید.

- قبل از هر چیز لازم است تا برنامه MMB را نصب نمایید.چنانچه اینطور نیست می توانید برای نصب آن به CD همراه کتاب ویا سایت www.mediachance.com مراجعه نمایید.

ـ در آغازبرنامهMMB را اجرا نمایید.

۱.ابتدا توسط گزینه Add Page از منوی Page دو صفحه ایجاد نمایید.

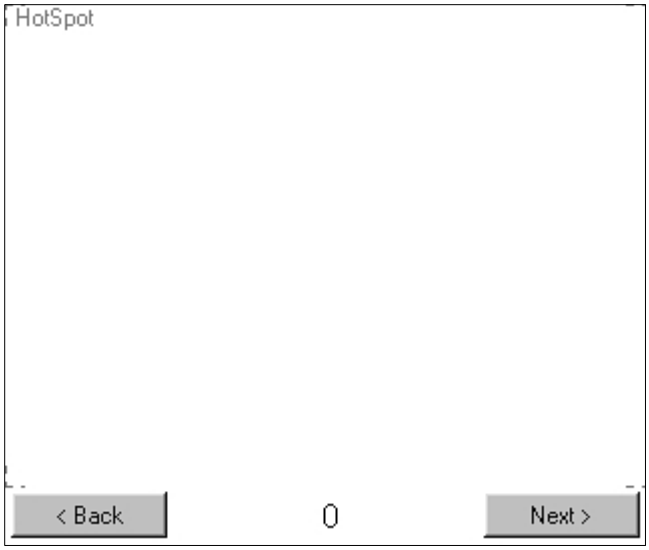
۲.از منوی Projectگزینه Project Settings را برگزینید.سپس از قسمت Window Size برای تعیین عرض و ارتفاع پنجره به ترتیب برای عرض (Width)عدد ۳۲۰ وبرای ارتفاع(Height) عدد ۲۷۰را وارد نمایید.در قسمت Window Title نیز عبارت My first project in MMB را وارد نمایید و نهایتا دکمه OK را جهت تایید تنظیمات فشار دهید.برای ذخیره پروژه از منوی Fileگزینه Save را انتخاب نمایید و پروژه را با نام MyProject ذخیره نمایید.

۳.در اولین صفحه توسط منوی Object->Createگزینه Rectangular HotSpot را برگزینید و توسط موس محدوده این شی را در صفحه تعیین نمایید.در حالی که شی HotSpot هنوز در حالت انتخاب قرار دارد (با کلیک بر روی شی،شی به حالت انتخاب در می آید)،برای تعیین موقعیت قرار گیری و اندازه شی از منوی Viewگزینه Dimensions را انتخاب کنید.در کادری که نمایان می شود به ترتیب از بالا به پایین اعداد ۳۲۰،۰و۲٤۰ را وارد نمایید و Enter را فشار دهید.سپس کادر Dimensions را ببندید.

٤.از منوی Object->Create توسط گزینه Text Button دو دکمه در صفحه ایجاد کنید.به ترتیب با ۲ بار کلیک بر روی هر دکمه به قسمت خصوصیات آنها رفته و در قسمت Text هر کدام جهت تعیین عنوان دکمه عبارت های Back < و> Next را وارد کنید.

۵.از منوی Object شی Text را برگزینید و آن را در صفحه ایجاد نمایید.با ۲ بار کلیک بر روی شی متن(Text) به قسمت خصوصیات آن بروید و در قسمت ویرایشگر آن عدد صفر را تایپ نمایید.و سپس با زدن کلید OK از کادر خصوصیات خارج شوید.

٦.اکنـون هماننـد تـصویر ، موقعیـت دو دکمـه و سایراشـیا را تنظـیم کنیـد.لازم بـه ذکـر اسـت کـه اشـیا در MMB در هنگـام طراحی هم توسط صفحه کلید و هم توسط موس قابل جا به جایی هستند.



شکل ۱-۱

۷.پـس از اینکـه موقعیـت اشـیا را تنظـیم نمودیـد از منـوی Edit گزینـه ی Select All را انتخـاب کنیـد تـا تمـام اشـیا موجـود در صـفحه بـه حالـت انتخـاب در بیاینـد.در حـالی کـه تمـام اشـیا در حـال انتخـاب هـستند کلیـد هـای Ctrl+C را جهت کپی برداری از اشیا فـشار دهیـد.سـپس بـه صـفحه دومـی کـه ایجـاد کـرده ایـد برویـد(از پنـل Pages در پـایین رابـط کاربربرنامه MMB مـی توانیـد بـرای نقـل مکـان بـین صـفحات اسـتفاده نماییـد.) در حـالی کـه در صـفحه دوم قـرار داریـد کلیـد هـای Ctrl+V را جهـت چـسباندن اشـیا بـه درون صـفحه فـشار دهیـد.بـا ایـن عمـل ظـاهر صـفحه دوم نیـز هماننـد صفحه اول می شود.

۸..در پنل Page دو بار بـر روی آیکـن صـفحه اول کلیـک نماییـد تـا بـه قـسمت خـصوصیات شـی منتقـل شـوید .در کـادر خـصوصیات صـفحه در قـسمت Page Transition از لیـست موجـود گزینـه Alpha Blendingرا انتخـاب کنیـد.توسـط قـسمت Page Transition مـی تـوان جلـوه هـای بـصری جـالبی را بـه هنگـام تعـویض صـفحات در موقـع اجـرای برنامـه ایجاد نمود.

۹.در پنجـره خـصوصیت صـفحه از قـسمت اسـکریپت (Script)دکمـه Action On Page Start را فـشار دهیـد تـا ویرایشگر اسکریپت ظاهر شود.پس از آن درقسمت ورود متن ویرایشگر، کد زیر را تایپ نمایید:

```
If (i$>'') Then
  LoadText("Text","i$")
```

```
End
If (i$='') Then
  i=i+1
  i$=CHAR(i)
  name$='<SrcDir>\images\\' + i$ + '.jpg'
  ReplaceImage("HotSpot","name$")
  LoadText("Text","i$")
End
```

سپس با زدن OK از ویرایشگر خارج شوید و درادامه پنجره خصوصیت را با زدن کلید OK ببندید.از اینکه مفهوم کدهای نوشته شده در این خود آموز را نمی فهمید نگران نباشید.در قسمت های بعدی به تفصیل راجع به آنها بحث می شود.

۱۰.عملیات های قسمت های ۸ و ۹ را برای صفحه دوم نیز انجام دهید با این تفاوت که در قسمت ورود متن ویرایشگر اسکریپت صفحه دوم کد زیر را بایستی تایپ کنید:

```
If (i$>'') Then
  LoadText("Text","i$")
End
```

۱۱.به صفحه ی اول باز گردید و با ۲ بار کلیک بر روی دکمه با عنوان Back> به قسمت خصوصیات آن بروید.در کادر ظاهر شده از قسمت Action گزینه More Action(اولین دکمه از سمت راست)را برگزینید تا ویرایشگر اسکریپت نمایان شود.از زیر پنجره ویرایشگر گزینه Mouse Up را انتخاب کنید تا ویرایشگر مربوط به رویداد Mouse Up ظاهر شود، نهایتا کد زیر را در قسمت ورود داده های رویداد Mouse Up وارد نمایید:

```
i=i-1
If (i<1) Then
  i=3
End
i$=CHAR(i)
name$='<SrcDir>\images\\' + i$ + '.jpg'
ReplaceImage("Page 2::HotSpot","name$")
DisplayValue("Text","i")
NextPage()
```

در ادامه با فشردن کلید OK از پنجره ویرایشگر Script خارج شوید و بار دیگر با زدن OK از پنجره خصوصیت دکمه خارج شوید.

۱۲.در صفحه اول به قسمت Actions دکمه با عنوان < Next بروید و در پنجره رویداد Mouse Upکد زیر را تایپ کنید:

```
i=i+1
If (i>3) Then
  i=1
End
i$=CHAR(i)
name$='<SrcDir>\images\\' + i$ + '.jpg'
ReplaceImage("Page 2::HotSpot","name$")
DisplayValue("Text","i")
NextPage()
```

و در آخر با ۲ بار فشردن کلید OK به ترتیب از پنجره های ویرایشگر و خصوصیت دکمه خارج شوید.

۱۳.عملیات های قسمت های ۱۱ و ۱۲ را نیز برای دکمه های <Back و> Next صفحه دوم نیز انجام دهید با این تفاوت که کدهای :

```
ReplaceImage("Page 2::HotSpot","name$")
```

به

```
ReplaceImage("Page 1::HotSpot","name$")
```

و

```
NextPage()
```

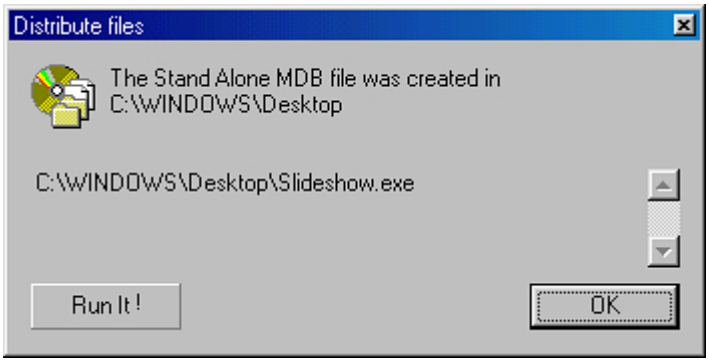
به

```
PrevPage()
```

تغییر می کنند.

۱٤.حال بر روی دیسک سخت خود یک پوشه به نام images ایجاد کنید و چند عکس را با پسوند JPG که نام آنها از عدد یک الی آخر می باشد را در آن پوشه کپی نمایید.مثلا چنانچه ۳ عکس داشته باشیم نام آنها به ترتیب 1.jpg،2.jpg و 3.jpg می شود.

۱۵.از منوی File گزینه Compile را جهت تولید نهایی و کامپایل برگزینید.از کادر بعدی که ظاهر می شود در قسمت Distribute to Destination توسط کلید Locate مسیر را برای ذخیره فایل نهایی مشخص نمایید و در ادامه دکمه OKرا فشار دهید .هنگامیکه فایل توسط MMB کامپایل شد کادری همانند شکل ظاهر می شود که بایستی دکمه OKرا از آن فشار داد.



۱۶.اکنون کافیست تا پوشه images را که قبلا ایجاد کردید به کنار فایل اجرایی که در مرحله قبل تولید شد ، منتقل شود تا برنامه به درستی عمل کند.به محل پوشه images بروید و آن را به کنار فایل اجرایی که در مرحله ۱۵ تولید شده است منتقل کنید.

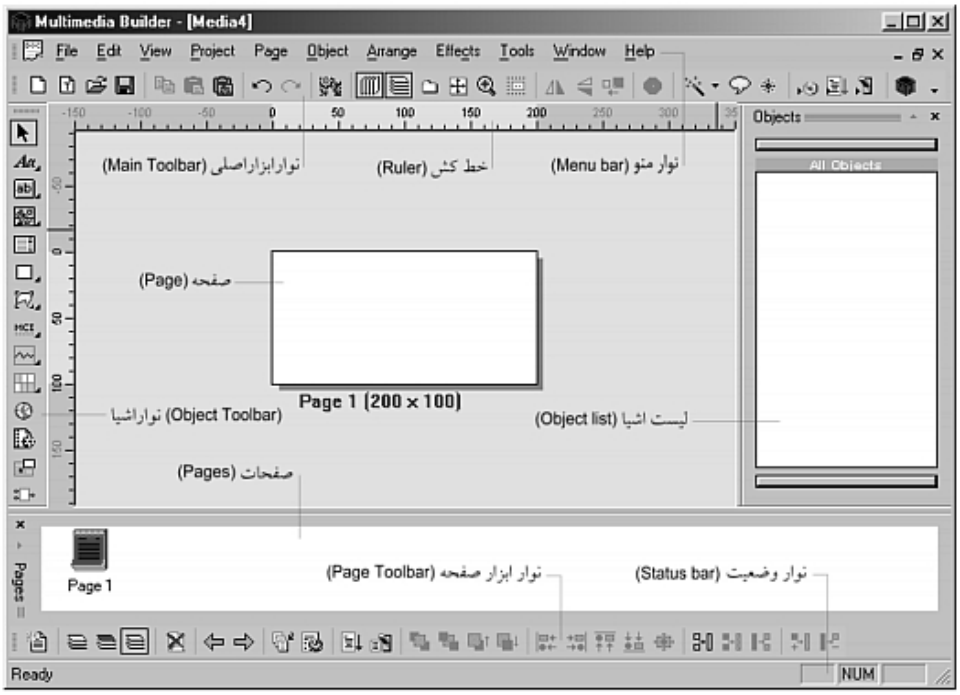
۱۷.اکنون برنامه شما آماده است تا مورد استفاده قرار گیرد .به همین منظور فایل تولید شده را اجرا نمایید و از حاصل کار لذت ببرید.

ایـن بخـش اختـصاص بـه بررسـی رابـط کـاربر(User Interface)برنامـهMultimedia Builder دارد.یـاد گیـری ایـن قسمت به مثابه ی زیر بنای روند یادگیری تلقی می شود.

هنگامیکه تصمیم به تولید یک برنامـه ی چنـد رسـانه ای مـی گیریـد اولـین قـدم در تولیـد آن نـرم افـزار ایجـاد ظـاهر آن مـی باشد،در اینجاست که رابط کاربر MMB بـه کمـک مـا مـی آیـد تـا بتـوانیم بـا بهـره گیـری از عناصـر ارائـه شـده بـه تکمیـل پروژه بپردازیم.البته باید یاد آور شـد کـه ایـن رابـط تنهـا بـرای سـاخت ظـاهر برنامـه سـود منـد نمـی باشـد بلکـه رونـد کـد نویسی برنامه نیز توسط همین رابـط صـورت مـی گیـرد.بـه طـور کلـی مهمتـرین ویژگـی ای کـه رابـط کـاربر بـه ارمغـان مـی آورد سهولت دسترسی به ابـزار هـا واسـتفاده از آنهـا مـی باشـد.در ایـن کتـاب رونـد بررسـی رابـط کـاربر را ابتـدا بـا بررسـی منو های برنامه شروع می کنیم،سـپس بـه سـراغ ابزارهـای کنتـرل و ویـرایش اشـیا مـی رویـم.در ادامـه تنظیمـات پـروژه مـورد بررسی قرار می گیرد تا به بررسی صفحه ها در MMB برسیم.در نهایت اشیا ارائه شده در MMB معرفی می شوند.

شرح رابط کاربرMMB(User Interface)

اولـین گـام در یـادگیری هـر نـرم افزاری،یـادگیری چگـونگی عملکـرد رابـط کـاربر آن نـرم افـزار مـی باشـد.بـه همـین منظـوردر ایـن قـسمت مـی‌خـواهیم بـه شـرح رابـط کـاربر MMB بپـردازیم. رابـط MMB نـسخه 4.9.7 بـه انـدازه کـافی قدرتمنـد مـی‌باشـد و فرآینـد کـاری بـسیار سـاده و مـوثری را فـراهم مـی کنـد. بـه کمـک ایـن رابطـه پیـشرفته کـار کـردن با MMB را آسان می یابید.شکل زیر رابطMMB ونواحی عمده را نشان میدهد.



شکل ۱-۲

ابتدا با بررسی منوهای رابط کاربر MMB شروع می کنیم.

«منوی File»

New

برای تولید یک سند جدید استفاده میشود.

New From Template

بـا اسـتفاده از ایـن گزینـه مـی تـوانیم براسـاس قالـب‌هـای ارائـه شـده قـالبی را انتخـاب کـرده و آن را تاحـدودی ویـرایش نموده و از آن استفاده نماییم.مزیت این قسمت در داشتن wizard جهت تکمیل مرحله به مرحله قالب میباشد.

Open

برای بازگشایی یک سند با فرمت MBD میباشد.

Close
این گزینه سبب بسته شدن سند جاری میشود.

Save
جهت ذخیره سند جاری بر روی دیسک میباشد.

Save As
جهت ذخیره یک کپی دیگر از سند جاری بر روی دیسک می‌باشد.

Compress & Export
گاهی اوقات پیش می‌آید که شما مایلید تنها به نشر فایل منبع یعنی فایل mbd.* بپردازید. برای مثال آن را بر روی شبکه برای دسترسی دیگران قراردهید. با استفاده از این قابلیت میتوانید فشردگی فایل و حجم فایل خروجی را تغییر دهید.

Compile
جهت کامپایل[1] نهایی پروژه می‌باشد.

Reduce size
این گزینه حجم ها و اقلام اضافی پروژه را حذف می نماید.

Import object
ممکن است در حین طراحی یک پروژه تمایل داشته باشید از یک شیء که قبلاً درجایی دیگر ودر زمانی دیگر طراحی گردیده است استفاده نمایید، به همین منظور میتوانید از این گزینه استفاده نمایید. البته پسوند شیء ها obm میباشد.

Export object
برای صدور یک قسمت از پروژه‌مان که مایلیم آن را ذخیره نماییم و در دفعات بعد از آن استفاده کنیم مناسب می باشد؛در دفعات بعد توسط گزینه Import object می‌توان آنها را به درون پروژه واردکرد.

MEF import
ازاین گزینه برای بارگذاری فایل‌های MEF که توسط برنامه Real-DRAW تولید می شود استفاده میشود.MEF مخفف (Meta Export Format) می‌باشد

۱- فرآیندی است که طی آن برنامه نوشته شده به زبان ماشین ترجمه می شود .

Webcam Image
همانطور که از نام این گزینه پیداست می توان تصاویری را که توسط web cam گرفته شده را بارگذاری نمود.

Import page
برای وارد کردن یک صفحه که از قبل صادر شده است استفاده می‌شود.

Export page
چنانچه بخواهیم میتوانیم صفحه پروژه رابا پسوند MMP صادر نماییم تا در دفعات بعد بتوانیم مکرراً از آن استفاده کنیم.

Print
این قسمت جهت چاپ صفحه‌ی جاری پروژه میباشد (در زمان طراحی)

Print preview
برای بازبینی قبل از چاپ میباشد

Print setup
این گزینه جهت تنظیم پارامترهای چاپ می‌باشد.

Exit
سبب خروج از برنامه MMB می شود.

«منوی Edit»

Undo
سبب لغو آخرین تغییر اعمال شده می شود.

Redo
سبب تکرار آخرین تغییری که به واسطه فرمان Undo ایجاد شده می‌شود.

Copy
جهت تهیه یک کپی از شیء یا اشیاء و ارسال آنها به Clipboard[1] می‌باشد.

۱-ابزاری است که عملیات کپی و چسباندن فایل ها و پوشه ها را در ویندوز مدیریت می کند.

Copy Incremental

این گزینه جهت تهیه چندین کپی از یک شیء می‌باشد.بدین صورت که با انتخاب یک شیء و برگزیدن این ویژگی تنها با کلیک کردن بر روی شیء و کشیدن،کپی جدیدی ازشیء اصلی ایجاد می شود.

Paste

جهت چسباندن شیء یا اشیاء از clip board به پروژهMMB می‌باشد.

Paste bitmap

چنانچه عکسی در clip board باشد توسط این گزینه می‌توان آن را در صفحه پروژه چسباند.

Delete

جهت پاک کردن شیء یا اشیاء درحال انتخاب می‌باشد.

Hide / Show

جهت مخفی کردن/ نمایان ساختن شیء یا اشیاء درون صفحه درهنگام طراحی می‌باشد.

Clone graphics

این قسمت برای تهیه یک کپی از شیء یا اشیا گرافیکی (عکس‌ها) می‌باشد. تفاوت بین این قسمت با قسمت Copy در این است که چنانچه شیء مادر (اصلی) را تغییر دهیم بلافاصله همان تغییر در اشیاء کپی برداری شده (فرزند) اعمال می‌شود.

Select all

جهت انتخاب تمام اشیاء موجود در صفحه می‌باشد.

Snap to grid

جهت پرش اشیاء بر اساس شبکه‌بندی از پیش تنظیم شده می‌باشد.

Edit grid

با این گزینه می‌توان شبکه‌بندی را تغییر داد.

Snap to Guides

Guides (راهنما) خطی است که با کلیک موس بر روی خط‌کش MMB و کشیدن آن به پایین و یا به سمت راست تولید می‌شود.از این خط جهت بهبود عمل ترازبندی استفاده می‌شود.حال با برگزیدن گزینه مذکور اشیاء به این خط راهنما پرش می‌کنند.

Edit Guides
جهت ویرایش و تنظیم خط‌های راهنما می‌باشد

«منوی View»

Object list
این گزینه لیست اشیاء موجود را در محیط طراحی نشان می‌دهد و یا حذف می‌کند.

Page list
این گزینه لیست صفحات راکه درپایین محیط طراحی قرار دارد مخفی و یا نمایان می‌سازد.

Toolbars
از این گزینه برای تنظیم نمایش مولفه‌های نوار ابزار استفاده می‌شود.

Status bar
از این گزینه جهت نمایان ساختن یا مخفی کردن نوار وضعیت استفاده می‌شود.

Master page
مشخص می کند که آیا صفحه جاری صفحه اصلی باشد یا خیر.

Quick object
این گزینه اشیاء موجود در درایوها را که قابلیت افزوده شدن به پروژه را دارند نمایان می‌سازد،که در نهایت با انتخاب شی وکشیدن آن به محیط طراحی توسط موس، آن شیء به محیط طراحی افزوده می‌شود.

Dimensions
از این ابزار جهت تغییر اندازه و موقعیت اشیاء استفاده میشود.

Zoom
برای بزرگ نمایی محیط کار و طراحی جهت تسلط بیشتر و در عین حال دیدن جزئیات صفحه از این ابزار استفاده می‌شود.

«منوی Project»

Project Settings
از ایـن قـسمت جهـت تنظـیم پـاره‌ای از ویژگـی‌هـای پـروژه نظیـر انـدازه صـفحات ،نـوار عنوان،شـکل صـفحات ،عکـس پس زمینه و... استفاده می‌شود.

Path replace
یک ابزار کارآمد دیگر جهت اعمال درست مسیرها درهنگام تولید و کامپایل نهایی پروژه می‌باشد.

Text replace
از این گزینه برای ویرایش و تنظیم و یا تصحیح کلی متون به کار رفته در پروژه استفاده می‌شود.

Check & distribute
از این گزینه جهت کامپایل نهایی پروژه استفاده می‌شود.

Embedded[1] files
از این قسمت برای به انضمام در آوردن سایر فایل‌ها استفاده می‌شود.

Comments
چنانچـه بخـواهیم فایـل منبعـی را منتـشر کنـیم مـی‌تـوانیم توضـیحاتی را بـه آن اضـافه کنـیم تـا کـاربران دیگـر در هنگـام بارگذاری فایل منبع از آن توضیحات مطلع شود.

Debug
از این گزینه جهت رفع عیب و بررسی جزئیات پروژه درهنگام اجرا استفاده می‌شود.

Embedded sound
برای به انضمام در آوردن فایل‌های صوتی استفاده می‌شود.

Run
از این گزینه جهت اجرای پروژه در زمان طراحی استفاده می‌شود.

«منوی Page»

Add page
صفحه‌ی جدیدی را به پروژه اضاف می‌کند.

۱- عملی است که طی آن فایل یا فایل هایی به بدنه اصلی پروژه الحاق می یابد .

Insert
این گزینه صفحه‌ای جدید رابه صفحات پروژه اضافه می‌کند با این تفاوت که می‌توان صفحه‌ی جدید را به قبل و یا بعد از آخرین صفحه موجود انتقال داد.

Delete page
صفحه منتخب را از پروژه حذف می‌کند.

Page manager
این قسمت کادری را جهت مدیریت قرارگیری صفحات ارائه می‌دهد.

Next page
در زمان طراحی به صفحه بعد می‌رود.

Previous page
در زمان طراحی به صفحه قبل می‌رود.

Master page
این گزینه سبب رفتن به صفحه‌ی اصلی می‌شود.

Master top layer
این گزینه سبب رفتن به صفحه اصلی از بالاترین لایه می‌شود.

Backgrourd
از این گزینه جهت تنظیم عکس پس زمینه صفحات استفاده می‌شود.

Debug
از این قسمت جهت رفع عیب و کنترل جزئیات صفحه منتخب جاری در زمان اجرا استفاده می‌شود.

Test current page
این قسمت برای اجرا و امتحان کردن صفحه جاری می‌باشد.

Check for lost object
این گزینه بر اساس تنظیمات جاری بررسی می‌کند که آیا شی‌ای در صفحه جاری گم شده است یا خیر.

Properties
این گزینه کادر خصوصیت هر صفحه را نشان می‌دهد.

«منوی Object»

Create
با کلیک برروی این گزینه لیست تمامی اشیاء موجود ظاهرمی‌شود که می توان بر اساس نیاز یکی را برگزید.

Convert to bitmap
این گزینه متنهای منتخب را به عکس تبدیل می کند.استفاده از این گزینه هنگامی مفید است که شک داشته باشیم نوع قلم(Font) مورد نظر ما در کامپیوتر کاربر نهایی موجود است.

Clone bitmap object
از این گزینه جهت تهیه یک نسخه از عکس منتخب جاری استفاده می‌شود با این تفاوت که اعمال هر تغییر بر روی نسخه‌ی اصلی سبب تغییر نسخه‌های کپی شده هم میشود.

Actions
این قسمت جهت دسترسی به بخشهایی می‌باشد که آن قسمت‌ها حاوی فعالیتی هستند. قسمت‌هایی ازقبیل Script, Sound, Interactions, Command & page

Properties
جهت دسترسی به کادر خصوصیت شیء منتخب می‌توان از این گزینه استفاده کرد.

«منوی Arrange»

Nudge
با استفاده از زیرشاخه‌های این گزینه می‌توان سبب تغییر مکان شیء شد.

Order
چنانچه بخواهیم یک شیء را بین لایه‌ها حرکت دهیم این گزینه به کمک ما می‌آید.

Align
برای تراز کردن موقعیت قرارگیری اشیاء از این قسمت استفاده می‌شود.

Combine
این گزینه اشیاء را با پس زمینه ادغام می‌کند و یک مجموعه واحد به عنوان پس زمینه تشکیل می‌دهد.

Group
برای گروه‌بندی اشیاء است.

Ungroup
جهت خنثی کردن گروه‌بندی است.

Add to group
می‌توان با استفاده از این گزینه اشیاء جدیدی را به گروه اضاف کرد.

Remove from group
سبب می‌شود تا شیء یا اشیاء منتخب از گروه خارج شود.

Group manger
این گزینه کادر مدیریت گروه‌ها را نمایان می‌سازد.

«منوی Effects»

Flip
جهت چرخاندن اشیاء در راستاهای افقی، عمودی به مقدار ۱۸۰° می‌باشد.

Blur
برای ایجاد جلوه بلور در عکسها می باشد.ابزار blur لبه‌ها و حاشیه‌های سخت و تیز تصویر را نرم و صاف می‌کند.

Sharpen
این گزینه جلوه Sharpen را به عکس ها اعمال میکند. از جلوه Sharpen برای سخت و تمیز کردن (واضح کردن)لبه‌ها و حاشیه‌های صاف و نرم موجود در تصویر استفاده میشود.

Convert to gray scale
به عکس جلوه سیاه و سفید با ته مایه خاکستری می‌بخشد.

Tile
سبب ایجاد جلوه‌ای همانند کاشی‌ها می‌شود.

Crop
با استفاده از این ابزار میتوان قسمتی از عکس منتخب را برش داد.

MMB Effects
این قسمت برای اعمال جلوه‌هایی نظیر سایه‌گذاری برجسته‌سازی براق‌سازی و ... می‌باشد.

Special Effects

ایـن گزینـه قـسمتی از جلـوه‌هـایی را کـه مـی تـوان بـر عکـسها اعمـال کـرد ارائـه مـی دهـد. افـرادی کـه نحـوه کـار بـا photoshop را می دانند با این گزینه آشنا هستند.

Inetractive Effects

این گزینه هم بـرای اعمـال چنـد جلـوه ویـژه دیگـر بـه عکـس هـا مـی‌باشـد. ایـن جلـوه هـا قابلیـت پویـایی و حرکـت را دارند.

Color Tweak Effects

جهت تغییر رنگ اشیایی که با کادر خصوصیت آنهـا نمـی تـوان رنـگ آنهـا را تنظـیم کـرد مـی تـوان از ایـن گزینـه اسـتفاده کرد.یادآور می شویم که این گزینه فراتر از تنها تغییر رنگ عمل می کند.

Restore original

چنانچه عکسی را تغییر داده باشیم با اعمال این گزینه می‌توانیم تغییرات اعمال شده را حذف کنیم.

Make new origiral

چنانچه به یـک عکـس تغییراتـی را اعمـال کنـیم و مایـل باشـیم یـک نـسخه ازهمـان عکـس تغییـر یافتـه بـه عنـوان نـسخه اصلی تهیه کنیم از این گزینه استفاده می کنیم.

«منوی Tools»

Designers Settings

این گزینـه انتخابهـایی را جهـت تنظـیم برخـی از خـصوصیات صـفحه طراحـی ازقبیـل اجـرا کننـده، تنظـیم صـداو... ارائـه می‌دهد.

Grid setting

برای تنظیم شبکه‌بندی می‌باشد.

Skin

MMB به همراه خود ٤ پوسته(Skin) جالـب بـرای رابـط کـاربر خـود ارائـه میدهـد کـه توسـط ایـن گزینـه قابـل تعـویض می‌باشند.

Smart Expanding Menus
کاربران ویندوز XP ویا۲۰۰۰ با این گزینه آشنا هستند.این گزینه سبب می‌شود تا منوها به صورت هوشمند عمل نمایند.بدین ترتیب که مولفه‌هایی را که مکرراً استفاده می‌شود نشان می‌دهد و بقیه مولفه ها را به صورت مجتمع در می‌آورد.

«منوی Window»

New window
سبب تولید یک پنجره جدید در محیط طراحی می‌شود.

Cascade
این گزینه پنجره‌ها را به صورت آبشاری مرتب می‌کند.

Tile
این قسمت سبب قرارگیری پنجره‌ها به صورت کاشی گونه می‌باشد.

«منوی Help»

Help topic
این گزینه راهنمای MMB را نمایان می‌سازد.

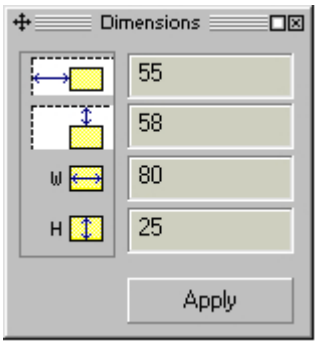
About MMBuilder
این گزینه کادر درباره برنامه را نشان می‌دهد که حاوی اطلاعاتی از قبیل اطلاعات ثبت برنامه می‌باشد.چنانچه اطلاعات ثبت برنامه MMB از قبیل نام و کد رمز را در اختیار داشته باشید، می توانید از این قسمت نسبت به ثبت برنامه اقدام نمایید.

Tip of the day
انتخاب این گزینه سبب میشود تا هر بار که MMB را اجرا می‌کنیم کادری حاوی نکته‌ای نمایان شود.

ابزارهای کنترل ویرایش

بدیهی است که در زمان طراحی نیاز مند ابزارهایی جهت تغییییر موقعیت و اندازه ویا مدیریت قرار گیری اشیا می باشیم.بنابراین قبل از فراگیری سایر قسمت ها لازم است تا این ابزارها را بشناسیم.

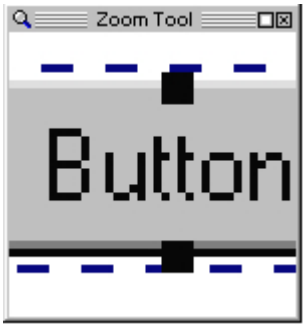
ابزار Dimensions



شکل۲-۲

این ابزار برای تغییر موقعیت و تغییر اندازه شی می‌باشد.

ابزار Zoom tool



شکل۲-۳

ازاین ابزار جهت بزرگ نمایی محیط کار و طراحی به خاطر داشتن تسلط بیشتر و در عین حال برای دیدن جزئیات استفاده می‌شود. چنانچه بر روی پنجره این ابزار راست کلیک کنیم با منویی مواجه خواهیم شد که قسمت‌های منو به شرح ذیل است.

- **Zoom factor**

این قسمت درصد بزرگ نمایی را مشخص می‌نماید.

- **Refresh period**

این گزینه دوره تازه‌سازی تصویر رامشخص می‌کند که این دوره درواحد میلی‌ثانیه می‌باشد.

- **Floating window**

چنانچه این گزینه را انتخاب کنیم پنجره ابزار Zoom به دنبال نشانه‌گر موس حرکت میکند.

- **Fixed window**

این قسمت جهت ثابت نگه‌داشتن پنجره ابزار Zoom میباشد.

- **Lock / Unlock view**

با استفاده از این ویژگی می توان ابزار Zoom را بر روی محدوده‌ای خاص ثابت نگه داشت و با تغییر مکان نما ابزار Zoom دیگر قسمتها و محدوده‌های دیگر را بزرگ‌نمایی نمی‌کند.

- **Show / Hide gird**

این گزینه شبکه‌بندی را بر روی ابزار Zoom نشان می‌دهد و یا محو می‌کند.

- **Grid color**

با این ویژگی می‌توان رنگ شبکه‌بندی را تعیین کرد.

ابزار Align

در تولید پروژه در MMBبه دفعات پیش می‌آید که بخواهیم اشیاء حاضر در صحنه ترازبندی شوند یا در صف خاصی در راستایی مرتب شوند.برای این منظور ابزار Align به کمک ما می‌آید. ویژگی Align متشکل از قسمتهای زیر است:

- **Left** 

اشیاء رانسبت به سمت چپ آنها ترازبندی می‌کند.مبنای این تراز اولین شیء انتخاب شده است.

- **Right** 

اشیاء را نسبت به سمت راست آنها ترازبندی می کند.مبنای این تراز اولین شیء انتخاب شده است.

- **Top**

اشیاء را نسبت به سمت بالای آنها ترازبندی می کند. مبنای این تراز اولین شیء انتخاب شده است.

- **Bottom**

اشیاء را نسبت به سمت پایین آنها ترازبندی می کند.مبنای این تراز اولین شیء انتخاب شده است.

- **Center**

این قسمت یک مرکز مجازی برای اشیاء انتخاب شده در نظر می‌گیرد سپس همگی آنها را به آن مرکز منتقل می‌کند.

ابزارگروه بندی (Grouping)

گروه‌بندی کردن یک ترفند کارآمد جهت ایجاد سهولت در امر طراحی و پیش‌برد پروژه می‌باشد به همین‌منظور این ویژگی نیز در MMB گنجانده شده است.

- **Group**

اشیاء منتخب رابه شکل یک گروه در می‌آید.

- **Ungroup**

گروه بندی را خنثی می کند.

- **Remove from group**

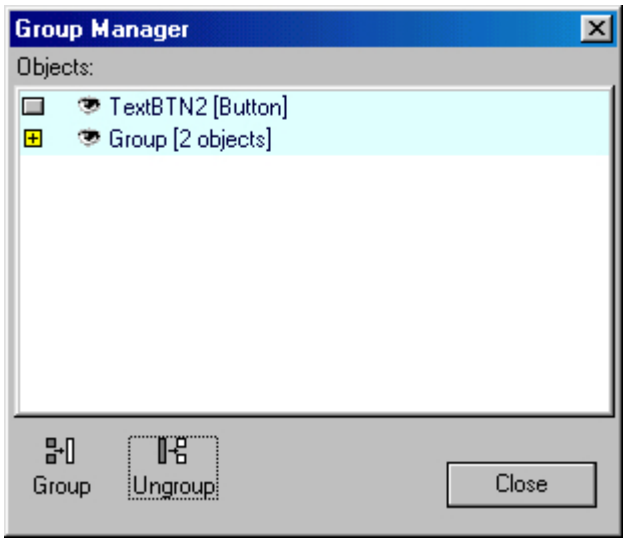
این قسمت عضو انتخاب شده گروه را از گروه خارج می‌کند.

- **Add to group**

شیءیا اشیاء درحال انتخاب را به گروه منتخب جاری اضافه می‌کند.

- **Group manger**

این ویژگی لیستی از گروههای موجود را ارائه می‌دهد.که می‌توان آنها را خنثی و یا ادغام کرد.



شکل۳-۲

ابزار Order

بعضی اوقات در هنگام طراحی ممکن است چند شیء بر روی شیء‌ای دیگر قرار گیرند یا برعکس زیر یک شیء. به همین سبب برای تغییردادن قرارگیری اشیاء بر روی هم وتغییر لایه های قرارگیری اشیا این ویژگی پیش‌بینی شده است.

- **Bring to front** – 

این ویژگی شیء منتخب را به بالاترین سطح انتقال می‌دهد یعنی بر روی تمام اشیاء.

- **Send to Back** – 

این ویژگی شیء منتخب را به پایین‌ترین سطح انتقال می‌دهد یعنی به زیر تمام اشیاء موجود.

- **Bring Forward** – 

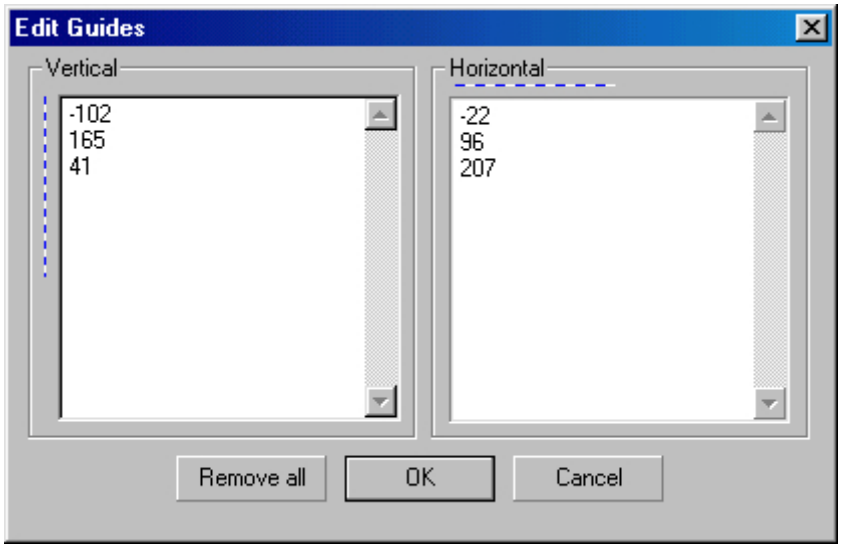
این قسمت شیء منتخب را به یک لایه بالاتر انتقال می‌دهد.

- **Bring Backward** – 

این قسمت شیء منتخب را به یک لایه پایین‌تر انتقال میدهد.

ابزار Guide

Guide (راهنما) خطی است که با کلیک موس بر روی خط‌کش MMB وکشیدن آن به پایین ویا به سمت راست تولید می شود. از این خط جهت بهبود عمل ترازبندی استفاده می‌شود.پس از آن با نزدیک کردن اشیاء به این خطوط اشیاء به این خطها پرش می‌کند. در MMB هیچ محدودیتی برای تعداد این خطوط وجود ندارد.



شکل ٤-٢

Edit Guide –

جهت ویرایش و تنظیم خطهای راهنما می‌باشد.

Snap to Guide –

چنانچه این گزینه فعال باشد خطوط راهنما عملاً قابل استفاده هستند. در غیر این صورت پرش به خطوط صورت نمی‌گیرد.

ابزار Grid

این ابزارکارآمد یک شبکه‌بندی خاص را بر روی صفحه پروژه درهنگام طراحی ایجاد می‌کند و هدف از آن باز ایجاد سهولت در امر جابه‌جایی اشیاء و ترازبندی آنها براساس این شبکه‌ها می‌باشد.

Snap to Grid –

جهت پرش اشیاء بر اساس شبکه‌بندی به منظور ترازبندی اشیاء می‌باشد.

Edit Gird –

با این گزینه می‌توان شبکه‌بندی را تغییر داد.

ابزار Flip

این گزینه جهت چرخاندن عکس‌ها در راستای افقی، عمودی، به مقدار ۱۸۰° میباشد.

Vertically – 

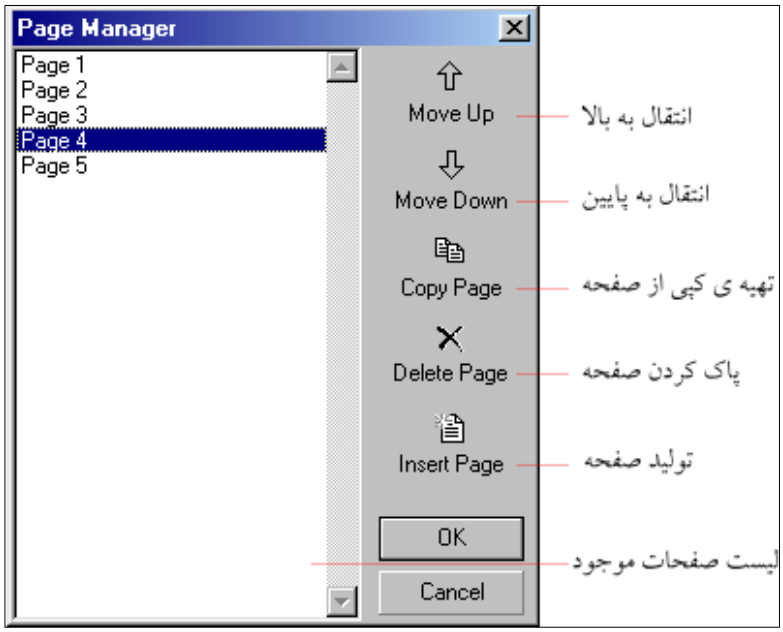
شکل‌ها را در راستای عمودی به میزان ۱۸۰ درجه می‌چرخاند.

Horizentally – 

شکل‌ها را در راستای افقی به میزان ۱۸۰ درجه می‌چرخاند.

ابزار Page managers

این ابزار برای مدیریت ترازبندی و تولید صفحات می‌باشد. این ابزار هنگامیکه پروژه ما از صفحات متعددی تشکیل شده است بسیار کارآمد و مفید می باشد.



شکل ۵-۲

تنظیمات پروژه «Project Setting»

تولید هر پروژه‌ای نیاز به یک سری تنظیمات کلی و اولیه دارد. به همین منظور قسمت Project Setting برای انجام این تنظیمات کلی در اختیار ماست.لازم به تذکر است که تنظیمات موجود بر کل صفحات پروژه اعمال می شوند.به عنوان مثال ۲ صفحه در یک پروژه نمی توانند دو اندازه ی مختلف داشته باشند. برای ایجاد تمایز در برخی از خصوصیات صفحه می توانیم از اسکریپت نویسی و فرامین موجود بهره بریم.در ادامه به شرح جزئیات این قسمت می‌پردازیم.

Windows size

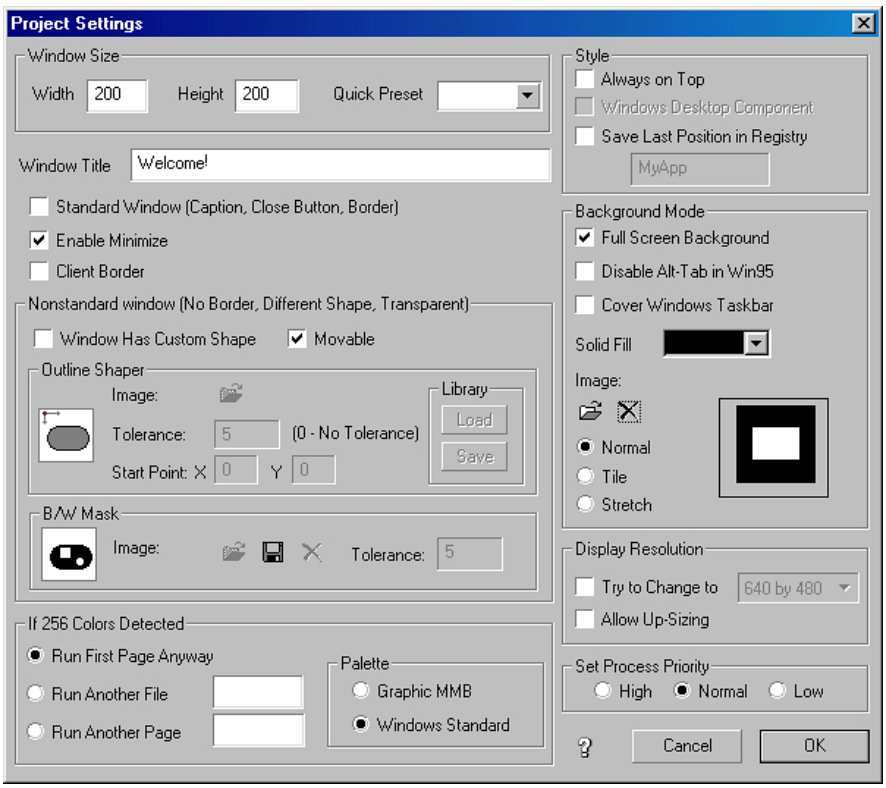
این قسمت اندازه پنجره برنامه را تعیین می‌کند. Width برای عرض و Height برای ارتفاع است.همچنین در قسمت Quick Preset چند اندازه که معمولاً استفاده می‌شود موجود است.

Windows Title

برای تعیین عنوان پنجره برنامه می‌باشد.

Standard Windows (Caption, Close Button, Border)

شکل۶-۲



در صورت فعال بودن این گزینه پنجره برنامه به صورت یک پنجره‌ی معمولی ویندوز ظاهر می شود. این پنجره همانند سایر پنجره‌ها می‌تواند دارای عنوان (Caption) دکمه بستن (Close Button) و حاشیه (Border) باشد.

Enable Minimize
این گزینه قابلیت Minimize شدن را فعال می‌کند.

Client Border
این گزینه حاشیه‌ای مشکی و باریک را به پنجره برنامه اضاف می‌کند.

Nonstandard Window
همان طور که می‌دانید در حالت عادی پنجره برنامه حالت استاندارد یک پنجره‌ی ویندوز را دارد. اما توسط این قسمت می‌توانیم شکل پنجره را به شکل دلخواه خود تغییر دهیم،برای مثال به دایره. این گزینه در صورت غیر فعال بودن گزینه Standard Windows فعال می‌شود. اما برای تغییر ظاهر برنامه ابتدا باید عکسی را که مایلیم پنجره به همان صورت در آید بارگذاری کنیم سپس می‌توانیم مختصات قرارگیری این عکس رانسبت به منتهی علیه سمت چپ وبالای پنجره برنامه جهت نتیجه بهتر تغییر دهیم. لازم به ذکر است که بهترین نتیجه هنگامی حاصل می‌شود که عکس ما حاشیه‌هایی واضح و پس زمینه‌ای یکدست داشته باشد. قسمت Library هم برای ذخیره پنجره‌ی جدیدی که تولید شده است می‌باشد تا در دفعات بعد هم بتوانیم از این پنجره استفاده کنیم.در صورت تمایل می توانید با استفاده ازTweak PlugIn که در قسمت Plug-In شرح داده شده است شکل پنجره را در هنگام اجرا مکررا تغییر دهید.

B/W Mask
توسط این قسمت می‌توانیم تصویری را بارگذاری کنیم و از آن به عنوان ماسکی بر روی صفحه استفاده کنیم. به عنوان مثال فرض کنید که تصویری داریم که در آن دایره‌ای قرار دارد و ضمناً این عکس زمینه‌ای یکدست دارد. چنانچه این تصویر را بر روی پنجره برنامه قرار دهیم تنها قسمتی از پروژه که زیر دایره و در محدوده‌ی دایره است قابل دید است. باید گفت که با استفاده ماهرانه از این قابلیت می‌توان پروژه‌هایی زیبا تولید نمود.

If 256 Color Detected
چنانچه پروژه ما با مد ۲۵٦ رنگ ناسازگار باشد می‌توانیم به هنگام مواجه با مد ۲۵٦ رنگ فایلی دیگر (Run another file)، صفحه‌ای دیگر از پروژه (Run another page) و یا اجرای خود برنامه تحت هر شرایطی (Run first page anyway) را برگزینیم تا پروژه به اجرای خود ادامه دهد.

Palette
در این بخش می‌توانیم مشخص کنیم که برنامه ما از پلت رنگ ویندوز (Windows standard) استفاده کند و یا از پلت رنگ MMB (Graphic MMB).

Style

در این قسمت می‌توان نوع قرارگیری پنچره برنامه را اعم از:

Always on Top –

همیشه بر بالا .

Windows Desktop Component -

این گزینه پنجره برنامه را همیشه بر روی دسکتاپ ویندوز اجرا می‌کند .

Save Last Position in Registry -

آخرین موقعیت قرارگیری پنجره در صفحه را در رجستری[1] تحت نامی که برمی‌گزینیم ذخیره می‌کند.

Background Mode

در صورت تمایل می‌توانیم برای پروژه پس زمینه‌ای را در نظر بگیریم. البته این پس زمینه با پس زمینه هر صفحه متمایز است. در این حالت عکس پس زمینه در زیر پنجره برنامه ودر کل پروژه قابل رؤیت است.در حالی که هر صفحه می‌تواند پس زمینه منحصر به خودش را داشته باشد.

Fullscreen Background

امکان داشتن پس زمینه تمام صفحه را ارائه می‌دهد. برای این عمل کافیست از قسمت Image دکمه Load Image (تصویر) را فشار دهیم.

Disable Alt – Tab in win 95

همان طور که می‌دانید با فشردن همزمان کلیدهای Alt و Tab در ویندوز پنجره‌ای نمایان می‌شود که حاوی لیستی از برنامه‌های در حال اجرا می‌باشد که توسط این لیست می‌توانیم بین برنامه‌ها نقل مکان نماییم. حال انتخاب این گزینه این قابلیت را غیر فعال می‌سازد.

Cover Windows Taskbar

انتخاب این گزینه در هنگام اجرای برنامه نوار وظیفه را مخفی نگه می‌دارد.

Solid Fill

می‌توانیم به جای عکس برای پس زمینه کل پروژه،از رنگ‌ها استفاده کنیم.

Display Resolution

توسط این قسمت می‌توانیم بر اساس نیاز معین کنیم که وضوح تصویر (Resolution) در هنگام اجرای برنامه چقدر باشد.

۱- محلی است که اطلاعات مورد نیاز سیستم در انجا ذخیره و نگهداری می شود.هرگونه تغییر در این بخش منجر به تغییر قسمت مربوط در ویندوز می شود.

Allow Up-Sizing
این انتخاب برای سازگاری برنامه در هنگام تغییر وضوح تصویر با سخت افزارهای قدیمی است.

Set Process Priority
این بخش مقدار استفاده برنامه از CPU را تعیین می‌کند، گزینه Low(پایین) سبب می‌شود که برنامه به مقدار کم از CPU استفاده نماید یا Normal(معمولی) باعث استفاده به صورت معمولی می‌شود و High (بالا) استفاده از CPU را تا حد امکان افزایش می‌دهد.

۱- توصیه می شود تنها در صورت نیاز از این بخش استفاده نمایید.

صفحه (Page)

محلی است کـه حـاوی گروهـی از اشـیا بـه همـراه توابـع آنهـا مـی باشـد تـا در نهایـت عملیـات مـورد نظـر صـورت گیـرد. بنابراین صفحات برای سازماندهی وتقسیم نیازهای پروژه می باشد.

ممکـن اسـت پـروژه هـا تنهـا متـشکل از یـک صـفحه باشـند،اما اسـتفاده از چنـدین صـفحه بـه سـهولت طراحـی و کـارایی بیشتر پروژه کمک می کند.یکـی از قابلیـت هـای کارآمـدی کـه صـفحات بـه ارمغـان مـی آورنـد اینـست کـه بـه راحتـی مـی توان در زمانهـای بعـد از تولیـد پروژه،اقـدام بـه بـه روز رسـانی صـفحات کـرد.امـا بایـد توجـه داشـت کـه اسـتفاده زیـاد از صفحات سبب افزایش حجـم برنامـه مـی شـود کـه ایـن افـزایش حجـم در همـه جـا خوشـایند نیـست،مثلا ممکـن اسـت در نظر داشته باشید پروژه را در اینترنت انتشار دهید و چنانچه حجم پروژه بالا باشد سبب بروز مشکلاتی می شود.

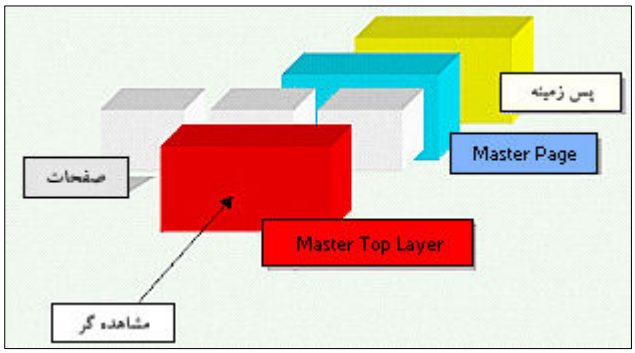
برنامه MMB همچنین ۲ صفحه ی ویژه را برای افزایش قابلیت ها و سهولت در امر طراحی ارائه می دهد.

Master Page –[1]

تمامی اشیا در این صفحه در کل صـفحات پـروژه قابـل دیـد هـستند.ایـن صـفحه در زیرتـرین لایـه بعـد از پـس زمینـه قـرار دارد.

Master Top Layer –[2]

تمامی اشیا در این صفحه نیز در کل صفحات پروژه قابل دید هستند.این صفحه در بالا ترین لایه قرار دارد.



شکل۷-۲

۱و۲-این دو صفحه از نوار ابزار صفحه (Page Toolbar) قابل دسترسی می باشند.

رویدادهای صفحه

رویداد یعنی اینکه حالتی به وقوع بپیوندد.به عنوان مثال عملکرد پدال گاز در وسیله نقلیه را در نظر بگیرید.هنگامیکه پدال گاز توسط راننده فشار داده می شود برسرعت وسیله نقلیه افزوده می شود. فشردن پدال گاز که منجر به انجام عملی می شو درا رویدا داین مثال گویند.صفحات در MMB دارای ۲ رویداد هستند:

Page Start– :

زمانی است که صفحه اجرا می شود.

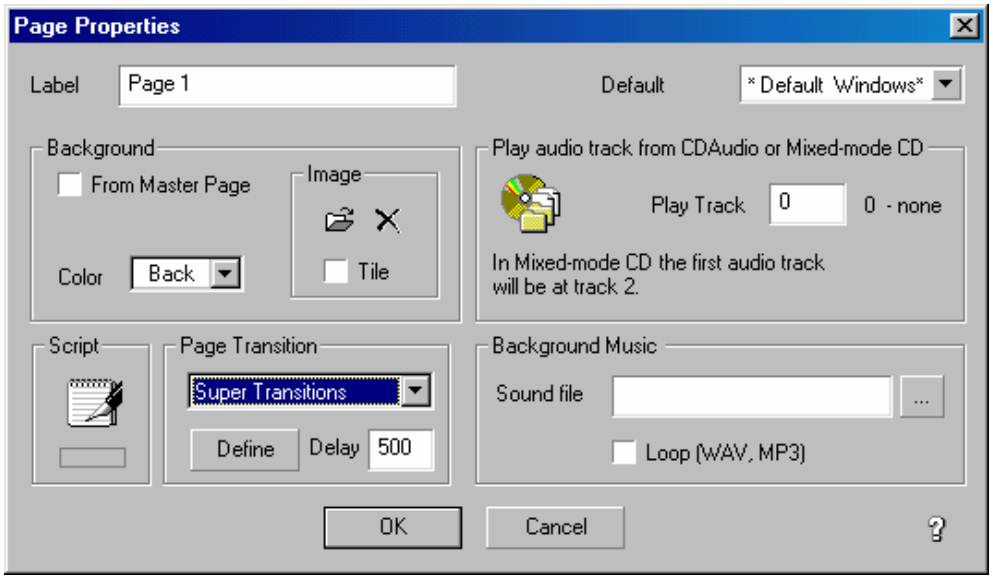
Page Exit– :

هنگامی است که از صفحه خارج می شویم.

بنابراین می توان با استفاده از این رویداد ها کد نویسی را انجام داد،بدین صورت که می توانیم به هنگام به وقوع پیوستن رویدادها به برنامه انجام عملی را نسبت دهیم.

خصوصیات صفحه

خصوصیت های هر صفحه از قسمت Page Properties ویا با ۲ بار کلیک بر روی آیکن هر صفحه در پنل صفحه قابل دسترسی است.



شکل۸-۲

Label
نام صفحه را مشخص می کند.

Background
این قسمت برای مشخص کردن پس زمینه صفحه می باشد.جنانچه گزینه From Master Page فعال باشد عکس پس زمینه همانند پس زمینه Master Page می شود.قسمت Imageبرای برگزیدن عکس می باشد و Color نیز برای تعیین رنگ می باشد.

Script
جهت نوشتن اسکریپت می باشد.

Page Transition
برای معین کردن Transition(واسط) می باشد.Transition جلوه ای بصری است که به هنگام تغییر صفحات نمایان می شود.

Default
نمایشگرموس(Cursor)صفحه را مشخص می کند.

Play Audio track from CD Audio or mixed mode CD
این قسمت یکی از آهنگ های CD صوتی را به عنوان آهنگ زمینه انتخاب می کند.

Background music
برای معین کردن آهنگ زمینه می باشد.گزینه LOOPجهت ایجاد تکرار می باشد.

اشیا (Objects)

اشـیا در MMB عناصـر تـشکیل دهنـده ی پـروژه هـستند،به همـین منظـور MMB طیـف وسـیعی از اشـیا و خـصوصیات آنهـا را ارائـه مـی دهـد.در ایـن قـسمت قـصد داریـم تـا بـه بررسـی اشـیا موجـود در MMB بـه همـراه خـصوصیات آنهـا بپردازیم.لازم بـه تـذکر اسـت کـه در پایـان ایـن قـسمت خـصوصیاتی را کـه در همـه اشـیا مـشترک مـی باشـند توضـیح داده ایـم تـا از تکـرار پرهیـز کنـیم.در MMB بـرای ایجـاد اشـیا کافیـست کـه آن شـی را از لیـست اشـیا ویـا نـوار ابزار(Toolbar)برگزینیم و در نهایـت بـا عمـل کـشیدن و رهـا کـردن مـوس شـی مربوطـه را ایجـاد کنـیم.بـرای دسترسـی بـه خـصوصیات اشـیا نیـز دو راه موجـود اسـت:۱-بـا دو بـار کلیـک بـر روی شـی ۲-گزینـه Properties از منـوی Object هنگامیکه شی در حالت انتخاب قرار دارد.در ادامه به بررسی اشیا ارائه شده به همراه ویژگی های آنها می پردازیم.

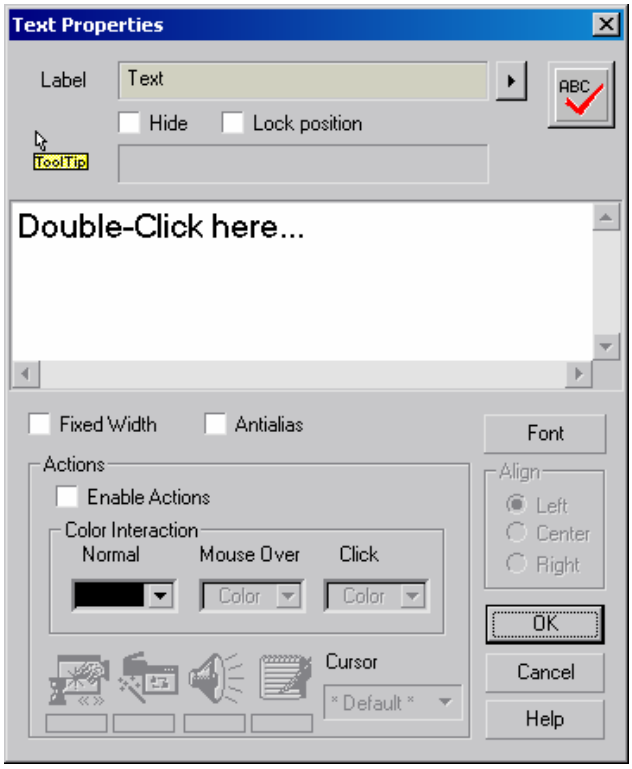
- **Text(متن)**

شکل۹-۲

این شـی بـرای نمـایش دادن متـون، کلمـات، عبـارات و ... مـورد نظـر شـما مـی‌باشـد. در ضـمن ایـن شـی مـی‌توانـد بـه صورت فعال باشد.

نکته: منظور از فعال اینست که شی می‌تواند حاوی اسکریپتی جهت انجام عملی باشد.

-خصوصیت‌های شی Text



شکل ۱۰-۲

Color Interaction (تعامل رنگ)

توسط این ویژگی می‌توانیم مشخص کنیم که بر اساس وضعیت موس نسبت به شی متن، متن چه رنگی شود.

در این قسمت ۳ حالت:

۱- **Normal** (معمولی) به هنگامی که موس بر روی متن قرار ندارد، یا کلیک نکند.

۲-**MouseOver** (موس بر روی متن) هنگامی است که نشانه گر موس بر روی شی متن قرار می‌گیرد.

۳-**Click** (کلیک) هنگامی که موس بر روی متن کلیک کند.

پیش‌بینی شده است.

Align (ترازبندی)

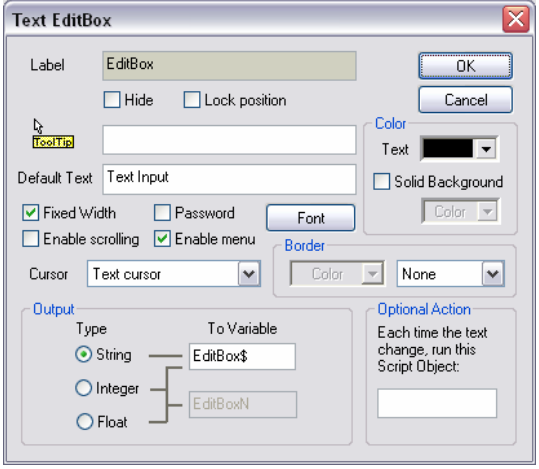
این ویژگی برای تراز کردن متن است.

- **EditBox (کادر ورود داده)**

این شی این امکان را فراهم می‌کند که اطلاعاتی را از کار گرفته و از آن در برنامه‌ی خود استفاده کنید.البته شی ارائه شده در MMB نسخه 4.9.7 برخی از ضعف های نسخه های قبلی را از میان برده است.

- خصوصیت‌های شی (EditBox)

-Default Text (متن پیش فرض)



توسط این کادر می‌توانید متن پیش فرض را برای شی EditBox در هنگامی که پروژه اجرا می شود تنظیم نمایند.

Color (رنگ)

با استفاده از این قسمت رنگ زمینه (Solid Background) و رنگ متن (Text) را تعیین می‌کنیم.

Out put (خروجی)

برای تعیین متغیری است که بایستی اطلاعات وارده شده به آن متغیر نسبت داده شود.بر اساس نیاز می‌توان نوع متغیر را به یکی از حالات String (رشته‌ای) Integer (صحیح) و یا نوع Float که شامل هر دو نوع متغیر است تغییر داد.متغیرها در قسمت بعد مفصلا شرح داده خواهند شد.

تذکر: برای نمایش اعداد با تعداد ارقام بیشتر از ٦ سعی کنید از نوع Float استفاده نمایید.

Fixed Width

این گزینه سبب می شود که طول شی Editbox در همه حال ثابت باشد.

Password

توسط این گزینه حروف به صورت رمزی ویکسان نمایان می شوند.

Enable Menu

جهت فعال سازی / غیر فعال کردن منوی Right Click به هنگام اجرا می باشد.

Optional Action (فعالیت دلخواه)

چنانچه بخواهیم به هنگام تغییر متن کادر ورود داده‌ها، فعالیت‌های خاصی صورت گیرد و یا اسکریپتی اجرا شود می‌توانیم از این قسمت استفاده کنیم.تنها کافیست که در کادر زیر این قسمت نام شی اسکریپت را معین کنیم.چنانچه در هنگام مطالعه اشیا توضیحات مربوط به برخی از قسمت های آن ها را نیافتید می توانید در ادامه و در بخش خصوصیت مشترک بین اشیا درباره آنها مطالعه نمایید.

- **Paragraph Text (پاراگراف)**

برای نمایش دادن متن‌های بلند خود می‌توانید از این شی استفاده کنید. با انتخاب شی و کشیدن کادر مورد نظر در صفحه و رفتن به قسمت خصوصیات شی می‌توانیم متن خود را وارد کنیم.

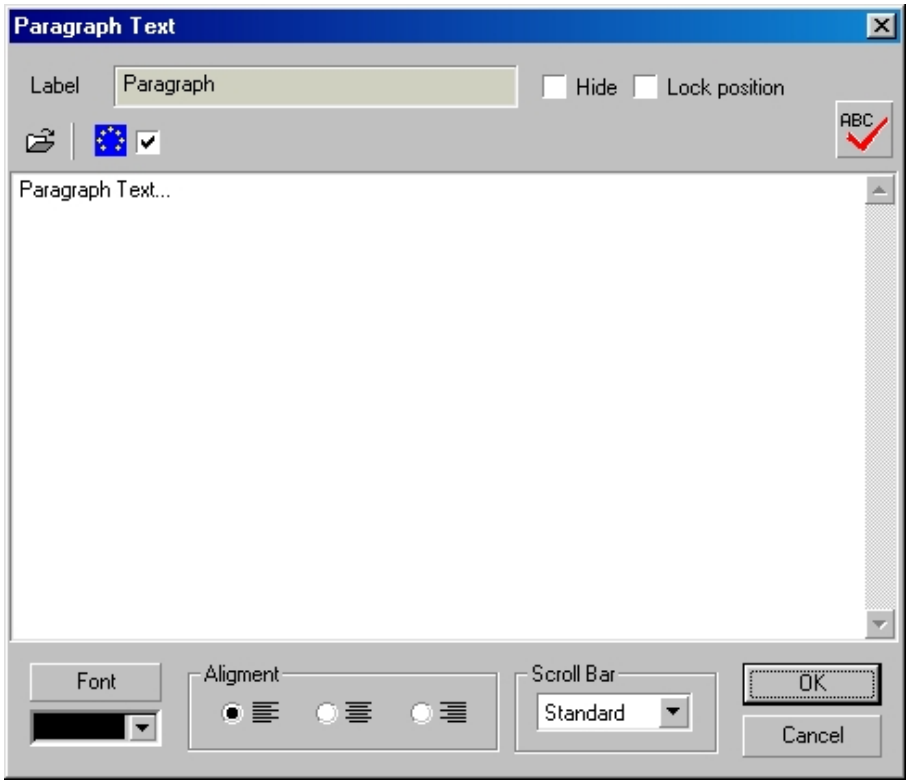
-خصوصیت‌های شی Paragraph

International Character Editor (ویرایشگر کاراکتر بین المللی)

این گزینه نوع ویرایشگر متن را تغییر می‌دهد.

Spell Checking (بررسی واژه‌ها)

چنانچه بخواهیم متن خودمان را از نظر غلط‌های املایی (تنها زبان انگلیسی) بررسی کنیم می‌توانیم از این ویژگی استفاده کنیم.



شکل۱۲-۲

Alignment (ترازبندی)

جهت تراز کردن متن می‌باشد.

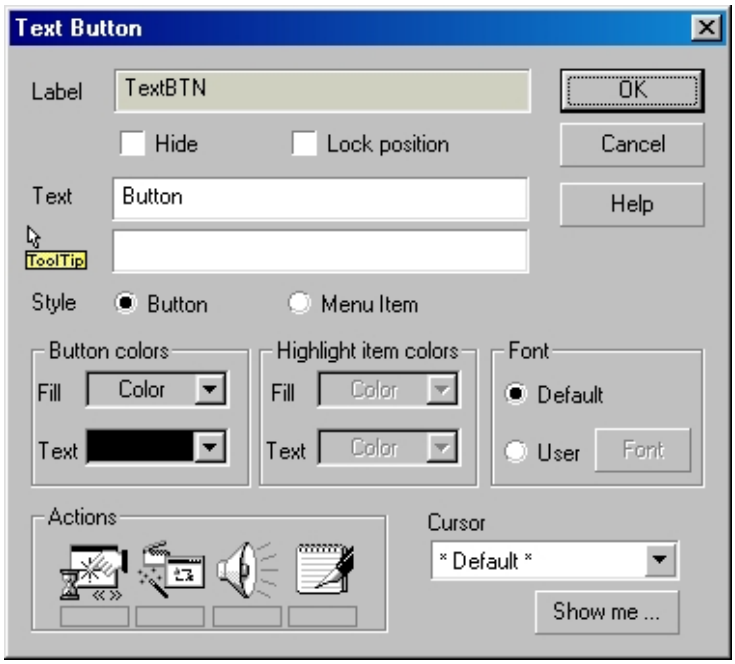
Scroll Bar (نوار نمایش)

اگرمتن مورد نظر ما تماماً در کادر مورد نظر نگنجد به طور خود کار نوارهای پیمایش در کنار کادر ظاهر می‌شود. برای شی Paragraph ،۲ نوع نوار پیش نمایش معمولی و تخت (Flat) موجود می‌باشد.

- **Text Button (دکمه‌ی متنی)**

این شی برای ساختن دکمه‌های استاندارد ویندوزی می‌باشد. ضمناً می‌توان رنگ و اندازه‌های این دکمه‌ها را تغییر داد، یا آنها را به گونه‌ای تنظیم کرد که همانند منوها عمل کنند.

- خصوصیت‌های شی Text Button



شکل۱۳-۲

Style(سبک)

توسط این قسمت می‌توان نوع دکمه را تعیین کرد. ۲ نوع موجود است:

۱- Button(دکمه استاندارد) ۲-Menu Item (همانند منو)

Button Colors (رنگ دکمه)

این قسمت برای تعیین رنگ دکمه (Fill) و رنگ متن (Text) می‌باشد. این قسمت در صورتی فعال می‌باشد که نوع دکمه همان حالت استاندارد باشد.

Highlight Item Colors

چنانچه نوع دکمهMenu Item باشد. می‌توان رنگ دکمه (Fill) و متن (Text) را تنظیم نمود.

- **Bitmap Button (دکمه تصویری)**

این گزینه به شما این امکان را می‌دهد که یک دکمه متشکل از ۳ عکس داشته باشید. (عکسی که در حالت عادی می‌بینید،در زمانی که موس بر روی آن قرار می‌گیرد و زمانی که بر روی آن Click می‌کنید) و یا اینکه شما می‌توانید با انتخاب گزینه Auto Button یک دکمه بسازید که فقط از یک عکس تشکیل شده باشد. مهمترین مزیت این شی همان داشتن فرم ۳ حالتی است.

- خصوصیت‌های شی Bitmap Button

Image (تصویر)

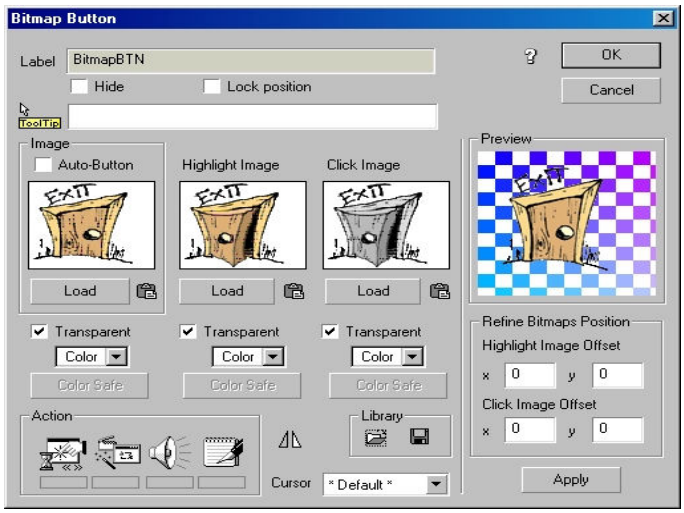
این قسمت برای تعیین ۳ عکس دکمه می‌باشد. قاب سمت چپ برای عکس پیش‌فرض، قاب وسط برای حالتی که موس بر روی دکمه واقع می‌شود(Highlight) و قاب سوم برای حالتی که با موس بر روی شی کلیک می‌کنیم (Click).

Preview (بازبینی)

این قسمت جهت بازبینی دکمه می‌باشد.

Transparent

با استفاده از این گزینه می‌توان رنگ پس زمینه‌ی عکسها را حذف کرد. مبنای شناسایی رنگ زمینه،پیکسل پایینی و منتهی علیه سمت چپ می‌باشد.



شکل ۱۴-۲

Auto Button

چنانچه بخواهیم دکمه ما تنها متشکل از یک عکس باشد،می‌توان از این گزینه استفاده کرد.

Color Safe

جهت رعایت موارد رنگ از نظر رنگ‌هایی که قابلیت نمایان شدن در اینترنت و... را دارند می‌توان از این گزینه استفاده کرد.

Refine Bitmap Position

اگر جهت تشکیل دکمه تصویری، عکس یا عکس‌هایی مناسب برای ۳ قسمت دکمه در اختیار نداشته باشیم و یا نخواهیم از ۳ تصویر استفاده کنیم می‌توانیم در حالتهای (Highlight) و (Click) به گونه‌ای عمل کنیم که تصویر آن قسمت ها تغییر مکان جزئی و محسوس بدهد. قسمت Highlight lmage offset برای حالت Highlight و قسمت Click lmage offset برای هنگام کلیک کردن است. در استفاده از این ویژگی تنها کافیست تا مقدار تغییر مکان را در کادرهای مربوطه وارد نمائیم.

Library

گاهی اوقات پیش می‌آید که می‌خواهیم دکمه‌ای را که ایجاد کرده‌ایم برای استفاده‌های آتی ذخیره کنیم.برای این امر می‌توانیم از Library استفاده کنیم. حال چنانچه بخواهیم دکمه‌های از پیش ذخیره شده را بارگذاری کنیم بایستی بر روی دکمه‌ی Load Button (شکل) کلیک کنیم.

Flip Horizontal

با این گزینه می‌توان تصاویر را در راستای افقی ۱۸۰° درجه چرخاند.

- **Alpha Button** (دکمه آلفا)

یک نوع مخصوص از دکمه است که به وسیله برنامه‌ای به نام Real-DRAW محصول شرکت ®Mediachance تولید می‌شود. مزیت این دکمه‌ها در کیفیت و سادگی آنهاست. در این دکمه‌ها هم می‌توان ۳ حالتی را که شی Bitmap Button دارا بود را در اختیار داشت.داشتن ویژگی Transparency از دیگر مزیت های این نوع دکمه است.

-خصوصیت‌های شی Alpha Button

Load *.3sb button

برای بارگذاری دکمه‌های Alpha می توان از این دکمه استفاده کرد.

- **Bitmap (بیت مپ)**

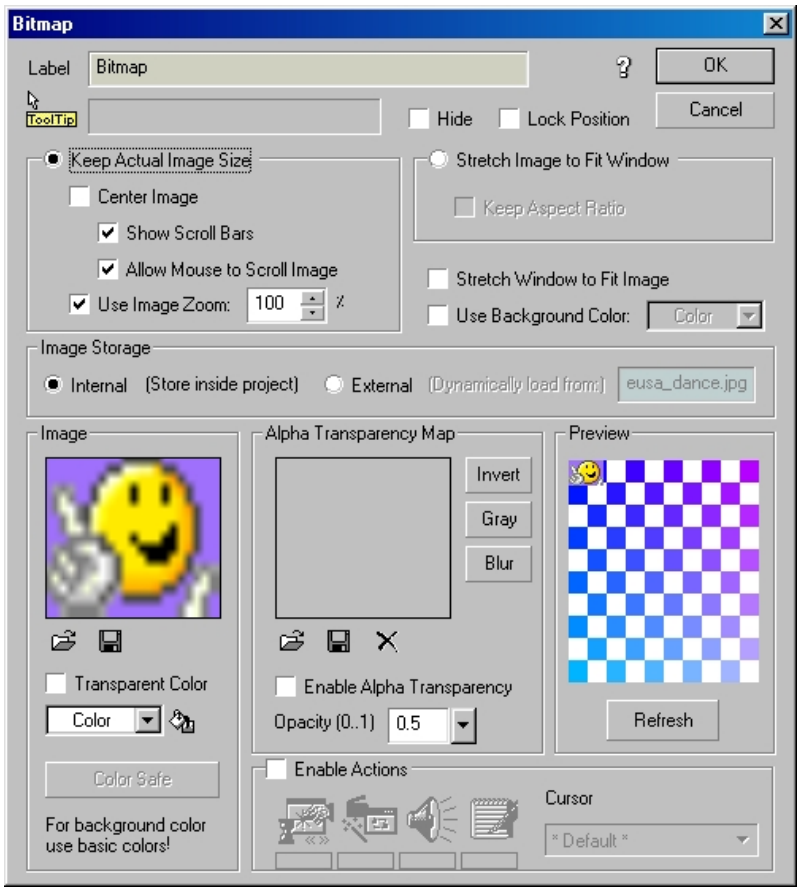
این شی برای نشان دادن تصاویر گرافیکی در پروژه می‌باشد.این شی می‌تواند به ۲ صورت فعال و غیر فعال باشد.
Bitmap از ۲ قسمت تشکیل شده است: ۱-Bitmap ۲-Alpha Transparency map

این شی می‌تواند از فرمت‌های TIFF, PNG,PCX,GIF,JPG,BMP حمایت کند.

خصوصیت های شی Bitmap

Keep Actual Image size

این قسمت اندازه واقعی تصویر را حفظ می‌کند و دارای مولفه‌های زیر است:



Center Image:

تصویر را در وسط کادر مورد نظر Bitmap نمایش می‌دهد.

Show Scroll bars

چنانچه تصویر از محدوده‌ای که برایش در نظر گرفته شده بزرگتر باشد نوارهای پیمایش ظاهر خواهند شد.

Allow Mouse to scroll Image

در صورت فعال بودن این گزینه عمل پیمایش را می‌توان توسط موس انجام داد.

Use Image Zoom

می‌توان تصویر را بزرگ‌نمایی یا کوچک‌نمایی کرد.

Stretch Image to fit windows

این قسمت برای ایجاد کشیدگی در تصویر جهت تطبیق در محدوده مشخص شده برای عکس می باشد.

Image storage

می‌توانیم مشخص کنیم که آیا تصویر به همراه فایل اجرایی الحاق شود یا در کنار فایل اجرایی در پوشه‌ای مخصوص قرار گیرد. گزینه‌ی Internal برای الحاق فایلهاست و External جهت ذخیره در خارج از فایل اجرایی می باشد.

Stretch windows to fit Image

اگر تصویر ما از محدوده‌ای مشخص شده برای تصویر بزرگتر باشد با استفاده از این گزینه می‌توان به MMB فهماند که پنجره را با عکس تطبیق دهد.

Use Background Color

جهت برگزیدن رنگ زمینه برای محدوده‌ی تصویر می‌باشد.

Image

این بخش تصویر بارگذاری شده را نشان می‌دهد.در این قسمت نیز ویژگی Transparencyوجود دارد.

Alpha Transparency map

در صورت فعال بودن این قسمت تصویر مربوطه بر اساس مقدار غلظت (opacity)حالت شفافیتی به خود می‌گیرد. حتی می‌توان از تصویری دیگر برای اعمال حالت شفافیت استفاده کرد. دکمه‌های Blur / Gray / Invert به ترتیب برای معکوس کردن حالت / اعمال یک پوشش خاکستری / جلوه بلور به کار می‌روند.

- **Animated GIF (GIF متحرک)**

با انتخاب این گزینه می‌توانید فایل GIF متحرک را وارد پروژه خود کنید. فایل GIF معمولاً در قسمت برنامه‌نویسی کنترل می‌شود. در قسمت تنظیمات Animated GIF می‌توانید توسط گزینه‌های Transparency (

شـفافیت)، Loop (تکـرار)، Speed (سـرعت) و... کنتـرل اجـرای GIF را بهبـود بخـشید. ایـن شـی بـه پویـایی پـروژه شما کمک شایانی می‌کند.

فایل های GIF متحـرک در اینترنـت بـسیار رایـج و مـشهور هـستند. بـه راحتـی مـی توانیـد باجـستجو در اینترنـت اقـدام بـه تهیـه آنهـا نماییـدو یـا اینکـه یـک Clipart CD را تهیـه نماییـد. نحـوه کـارکردن GIF هـم هماننـد سـایر اشـیاء مـی باشـد. برای مثال می توان آنها را گروه بندی نمود، حرکت داد و...

چنانچـه توسـط اسـکرپیت GIF پنهـان شـده را نمایـان سـازید نمـایش فایـل GIF از فـریم اول آن آغـاز مـی شـود.توسـط اسکریپت نویسی کنترل بیشتری بر روی GIF خواهید داشت

نکته: فایـل هـای GIF متحـرک احتیـاج بـه اسـتفاده ی بیـشتری از CPU نـسبت بـه GIF هـای ثابـت دارنـد. چنانچـه یـک GIF بـر روی یـک Bit map دارای Alpha Transparency قـرار گیـرد اجـرای GIF کنـدتر صـورت مـی گیـرد. بنـابراین از قـرار دادن مقـادیر زیـادی فایـل GIF متحـرک در پـروژه تـا حـد امکـان خـودداری کنیـد. فایـل هـای GIF متحرک برای بارز ساختن قسمتی ویژه مناسب می باشد !

-خصوصیات شی Animated GIF

Info (اطلاعات)

این قسمت حاوی اطلاعاتی درباره فایل GIF می‌باشد.

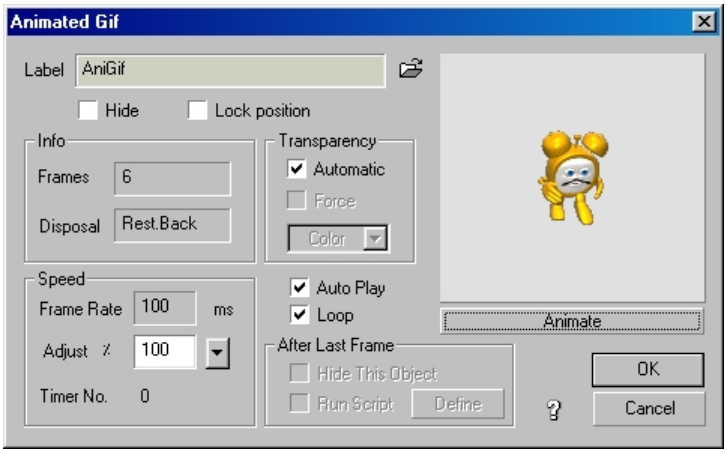
Transparency

برای حذف رنگ زمینـه تـصویر متحـرک اسـتفاده مـی‌شـود ایـن عمـل را مـی‌تـوان هـم بـه صـورت خودکـار(Automatic) و هم با اعمال نیرو (Force) به صورت دستی انجام داد.

Speed

تنظیم سرعت نمایش فایل GIF در این مکان صورت گیرد.

Auto play



با اعمال این گزینه در هنگام اجرای پروژه فایل GIF خودکار اجرا می‌شود.

Loop

سبب تکرار اجرای تصویر GIF می‌شود.

After Last Frame

در این مکان می‌توان مشخص کرد که پس از پایان اجرای GIF آیا شی پنهان شود (Hide the Object) و یا اسکریپتی (Run Script) اجرا شود.

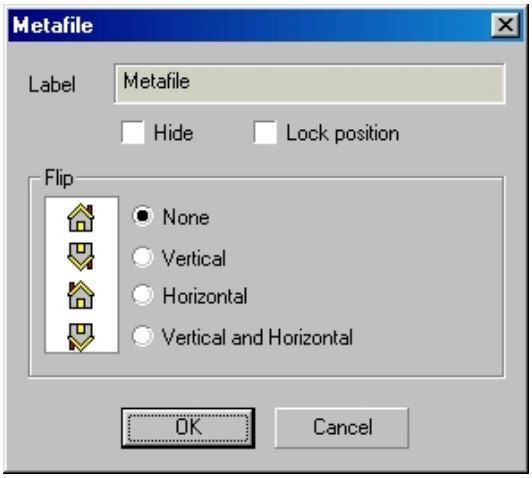
Animate

جهت بازبینی فایل GIF می‌باشد.

- **Meta File**

Windows Meta file (WMF) یک فرمت مشهور برداری می باشد. شما می توانید با استفاده از ابزارهای طراحی (از قبیل Corel draw) فایل هایی برداری تولید نمایید. فایل های با فرمت wmf به راحتی قابلیت تغییرمقیاس بدون تغییر کیفیت را در اختیار کاربر قرار می دهند.MMB از این فرمت پشتیبانی می نماید با این توضیح که فعلاً در نسخه ی جاری این شی غیر فعال است و حتی خصوصیات منحصر به فردی هم همانند سایر اشیاء ندارد.جهت قرار دادن wmf در صحنه ابتدا، بایست از منوی object گزینه Meta file را برگزینند. سپس با کشیدن موس محدوده wmf را مشخص کرده در ادامه کادری جهت بارگذاری فایل wmf آشکار می شود که می توان فایل دلخواه را بارگذاری نمود.

-خصوصیات شیMetaFile



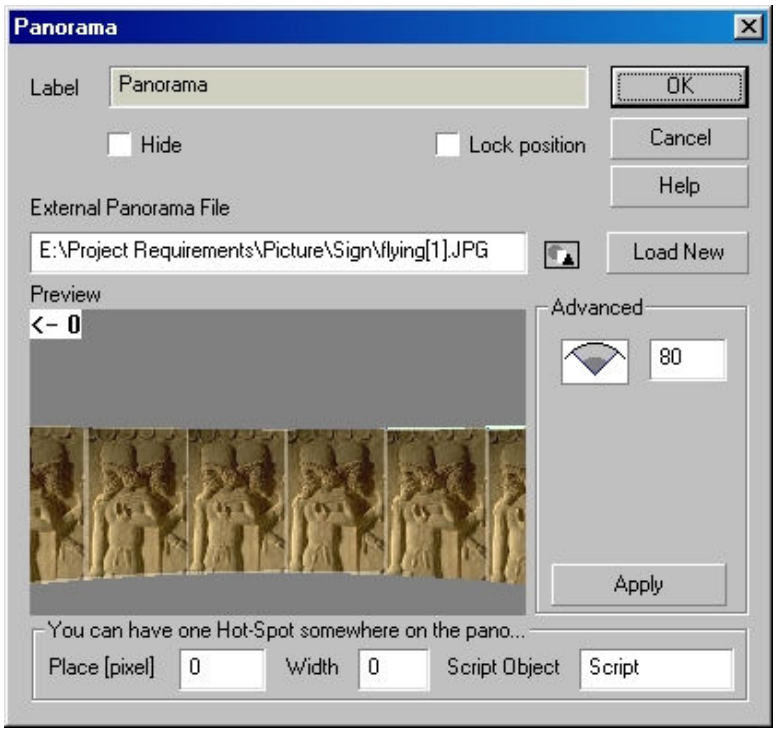
برای چرخاندن تصویر در راستاهای عمودی (Vertical) افقی (Horizontal) افقی و عمودی

(Vertical / Horizontal) و یا عدم چرخاندن (None) می‌باشد.

- **Panorama**

این شی سبب ایجاد جلوه‌ای بر روی عکس می‌شود که بیننده تصور می‌کند آن عکس در فضایی ۳ بعدی و قابل چرخش قرار دارد. این ویژگی برای تزئین پروژه بر اساس عکسها مفید است. در مواقعی هم که محور فعالیت شما عکس می‌باشد Panorama می‌تواند انتخاب خوبی باشد.

-خصوصیت‌های شی Panorama



Load New

با این گزینه تصویر جدیدی را انتخاب می‌کنیم.

Embedded File

می‌توان توسط این قسمت تصویر شیPanorama را به برنامه الحاق نمود.

Preview

برای باز بینی تصویر شی Panorama می‌باشد.

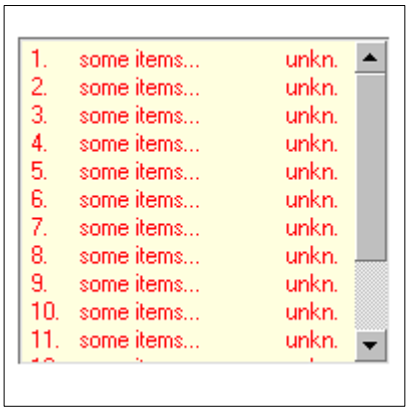
Advenced

برای تعیین زاویه دید می‌باشد.

You can have one hotspot somewhere on the pano…

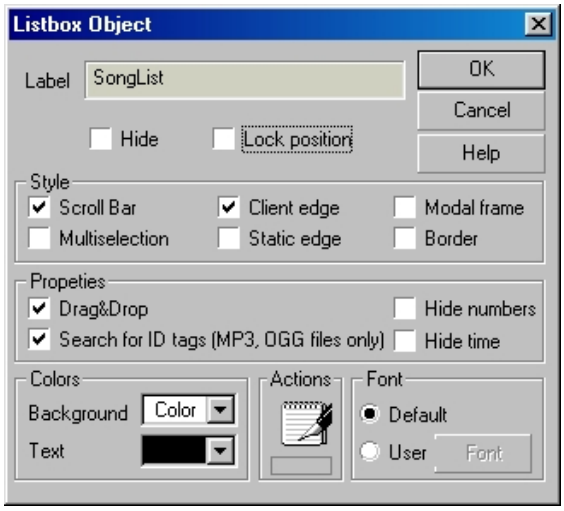
همـان طـور کـه واضـح اسـت مـی‌تـوان بـر روی شـی Panorama محـدوده‌ای مـشخص کـرد کـه هماننـد شـی Hotspot عمـل نمایـد و بـه هنگـام کلیـک بـر روی آن قـسمت،اسکریپتی کـه در قـسمت شـی Script بایـد معرفـی شـود اجـرا می‌شود.

- **List Box(کادر لیست)**



شـی ListBox قابلیـت نـشان دادن و لیـست کـردن تعـدادی از قـسمتها و انتخـاب‌هـا را فـراهم مـی‌نمایـد کـه کـاربر بـر حسب نیـاز مـی‌توانـد یکـی یـا بیـشتر از یـک قـسمت از انتخـاب‌هـا را برگزینـد. هـدف اولیـه MMB از نهـادن شـی List Box برای نشان دادن لیـست اجرایـی (Play List) فایـل‌هـای صـوتی اسـت. بـا ایـن حـال ایـن شـی مـی‌توانـد بـه عنـوان یک ListBox معمولی بـا قابلیـت چنـدین انتخـابی عمـل نمایـد. مـی‌توانیـد ListBox را توسـط متغیرهـای رشـته‌ای، متـون، آرایه‌ها و لیست اجرایی صوتی پـر نماییـد. بـه دنبـال اضـاف شـدن ایـن شـی بـه برنامـه انجـام کارهـای مختلـف وکارآمـدی امکان پذیر شـده اسـت.نکتـه حـائز اهمیـت در ایـن قـسمت اینـست کـه بـرای تعامـل بـا ایـن شـی توسـط فـرامین مربوطـه حتما لازم است تا شی قابل رؤیت باشـد،بدین معنـا کـه در هنگـام طراحـی نبایـد آنـرا مخفـی سـاخت.امـا چنانچـه شـرایطی پیش بیاید که مجبـور باشـیم ایـن شـی را مخفـی کنـیم مـی تـوانیم آن را بـه خـارج از محـدوده ی صـفحه در زمـان طراحـی انتقال دهیم تـا مجـازا ایـن شـی را پنهـان کـرده باشـیم.یـاد آوری مـی شـو دکـه بـا اسـتفاده ماهرانـه از ایـن شـی بـه همـراه EditBox می توان شی ComboBox را ایجاد نمود.

-خصوصیت های شی ListBox



Style

این قسمت چگونگی شمایل ListBox را معین می‌کند.

-Scroll bar(نوار پیمایش)

-Multi selection(چند انتخابی)

- Client Edge (خط حاشیه)

- Static Edge (لبه‌های ایستا)

-Modal Frame(لبه‌های برجسته)

-Border(حاشیه)

Properties

این قسمت یکسری دیگر از پارامترها را تنظیم می‌کند.

۱- شی ای است که علاوه بر خصوصیات شی ListBox قابلیت انتخاب مولفه هایش بر اساس عمل تایپ کردن مهیا می باشد.

Drag & Drop-

این قسمت قابلیت کشیدن و رها کردن را برای List Box فعال می‌کند.

Hide Numbers-

عددها را در کنار مؤلفه‌ها نمایان نمی‌سازد.

Search for ID tags-

این گزینه به جستجوی tag فایل‌های OGG و MP3 می‌پردازد.

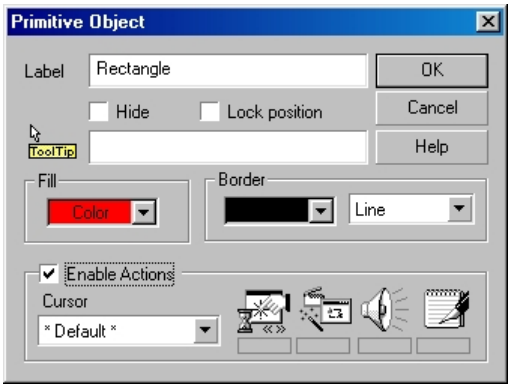
Hide time-

این گزینه زمان فایل‌های صوتی را به هنگام قرارگیری در لیست نشان نمی‌دهد.

Color-

در اینجا می‌توان رنگ زمینه (Background) و متن (Text) شی ListBox را تعیین کرد.

- **Rectangle(مستطیل)**



این شی برای کشیدن مربع و یا مستطیل می‌باشد. این شی می‌تواند فعال یا غیر فعال باشد.

- خصوصیت‌های شی Rectangle

Fill

رنگ شی را تعیین می‌کند.

Border

۲ قسمت موجود است، قسمت سمت چپ برای رنگ حاشیه شی و قسمت سمت راست برای چگونگی حاشیه شی اعم از: Line(خط)، None(هیچ)، Sunken(فرورفتگی)، Window(برجستگی) است.

- **Circle (دایره)**

توسط این قسمت به آسانی می‌توانید یک دایره و یا بیضی را ترسیم کنید. این شی هم می‌تواند فعال و یا غیر فعال باشد.

-خصوصیت های شی Circle

خصوصیت های این شی نیز کاملاً همانند شی مستطیل است.

- **Polygon (چند ضلعی)**

برای ترسیم چند ضلعی از این ابزار استفاده می‌شود.

- خصوصیت‌های شی Polygon

خصوصیات این شی هم مشابه مستطیل است.

- **Line(خط)**

این شی برای ترسیم خطوط مایل، شکسته و مستقیم می‌باشد.

-خصوصیت های شی Line

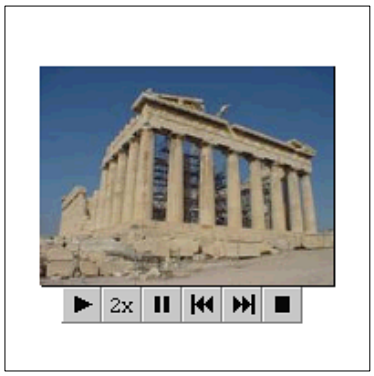
Position

برای تعیین موقعیت خط استفاده می‌شود. x1 و x2 به ترتیب طول نقطه‌های ابتدایی و انتهایی خط را مشخص می‌کند. y1 و y2 هم عرض نقاط ابتدایی وانتهایی را تعیین می‌کنند.

Line Color

رنگ خط را مشخص می‌کند.

- **Video**



برای وارد کردن فایلهای ویدئویی مورد نظرتان به پروژه می‌توانید از این شی استفاده کنید. Avi,Wmv,Dat,Mov,Mpeg از فرمتهای حمایت شده برای شی Video می‌باشند.شی Video یکی از اشیای غیر قابل اجتناب در اکثر پروژ ه های چند رسانه ای می باش دکه توسط اسکریپت نویسی و استفاده خلاقانه می تواند پروژه ای حرفه ای و کارا را به ارمغان بیاورند.در صورتی که نوع دیگری از فرمت های ویدیویی موجود باشد که در اینجا ذکر نشده باشد ، می توان با نصب Codec [1] مربوطه آن اجرای آن فرمت خاص را در MMB میسر ساخت.

-خصوصیت‌های شی Video

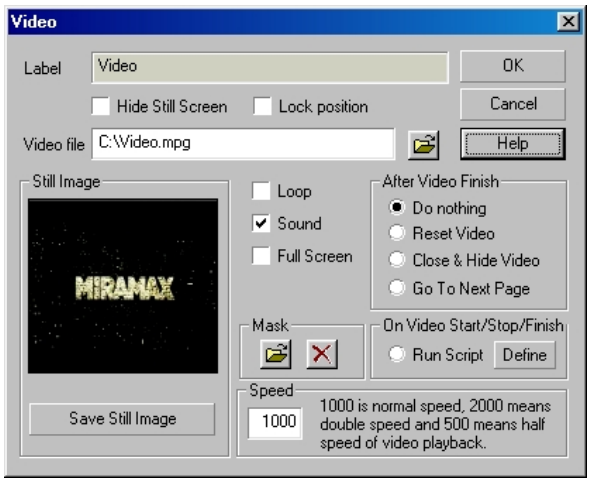
این کادر جهت وارد کردن فایل های ویدئویی می باشد. این کادر به شما اجازه می دهد تا هنگامی که فایل ویدئویی در حال اجرا نمی باشد عکسی را بر روی شی ویدئویی به عنوان عکس جاری نمایش دهید. هنگامی که توسط این کادر فایل ویدئویی را بارگذاری کردید، کارد دیگری جهت مشاهده فایل نمایان می شود که ازآنجا می توانید عکس جاری را به وسیله عقب و جلو بردن فایل ویدئویی تنظیم کنید.چنانچه فایل ویدئویی شما Mpg باشد و بخواهید تصویری را به عنوان عکس جاری برگزینید ممکن است موفق به انجام آن نشوید و عکس جاری تنها یک محدوده به رنگ مشکی باشد. این رفتار به خاطر MCI driver می باشد که مانع برگزیدن عکس جاری می شود. راه حل این مشکل این است که عکس مناسب با اندازه فایل ویدئویی بر روی آن قرار دهید. هنگامی که فایل اجرا است بالاتر از همه اشیاء حاضر در صفحه قرار می گیرد.

۱-ابزاری است که از آن برای فشرده سازی فایل های ویدویویی استفاده می شود.

البته در نسخه های جدید MMB اصلاحاتی در این زمینه صورت گرفته است .در نسخه 4.9.7 می توانید اشیا را بر روی شی ویدئو قرار دهید. در این حالت شی ویدئو در بالاترین سطح قرار می گیرد و هنگام اجرا قسمتی که در زیر شی دیگر قرار گرفته است قابل رویت نمی باشد.

Video File

مسیر فایل ویدئویی را مشخص می‌کند.



Still Image

تصویری را که در اولین فریم موجود است نشان می‌دهد.

Save Still Image

تصویر اولین فریم را ذخیره می‌کند.

Loop

سبب ایجاد تکرار در اجرای فایل ویدئویی می‌شود.

Sound

صدای فایل ویدئویی را پخش می‌کند.

Full Screen

فایل ویدئویی را به صورت تمام صفحه نشان می‌دهد.

Mask

این قسمت فایل تصویری را به عنوان ماسک بر روی فایل ویدئویی قرار می‌دهد. از این قسمت می‌توان برای ایجاد دریچه‌های دید جالبی استفاده کرد.

Speed

این قسمت سرعت اجرای فایل ویدئویی را تعیین می‌کند. سرعت پیش فرض 1000 است. 2000 به معنی ۲ برابر سرعت معمولی و 500 به معنای نصف سرعت معمولی است.

After Viedo Finish

این قسمت معین می‌کند که بلافاصله پس از اتمام اجرای فایل ویدئویی برنامه چه کاری انجام دهد.

Do nothing–

هیچ عملی صورت نمی‌گیرد.

Reset Video–

فایل ویدئویی به ابتدا باز گردانده می‌شود.

Close & Hide Video–

این گزینه پنجره ویدئو را بسته و محو می‌کند.

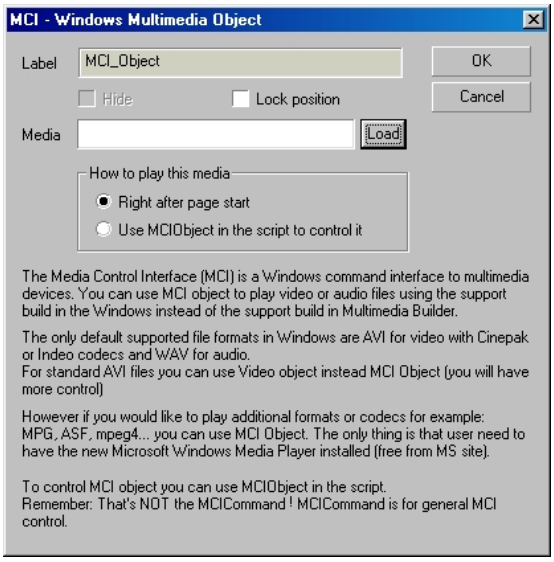
Go to next Page–

پس از اتمام اجرای فایل ویدئویی به صفحه‌ی بعد می‌رود.

On Video Start / Stop / Finish–

دراین قسمت برنامه می تواند بر اساس رویدادهای شروع / توقف / اتمام فایل ویدئویی اسکریپتی را اجرا کند.

- **MCI Object**



راه دیگری جهت اجرای فایل‌های ویدئویی (هم چنین صوتی) استفاده از شی MCI می‌باشد و به تبع آن می‌توان شی MCI را توسط فرامین MCI کنترل نمود.لازم به ذکر است که توسط طیف گسترده فرامین ارائه شده توسط MMB در نسخه های جدید نیاز به استفاده از این شی رو به کاهش گذاشته است.

-خصوصیت‌های شی MCI

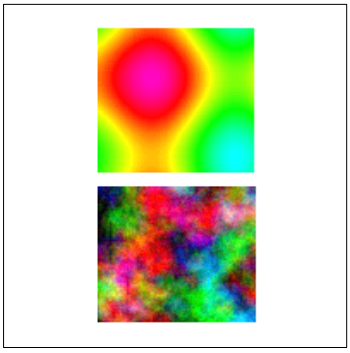
Media

برای بارگذاری فایلهای صوتی و تصویری می‌باشد.

How/To play this Media

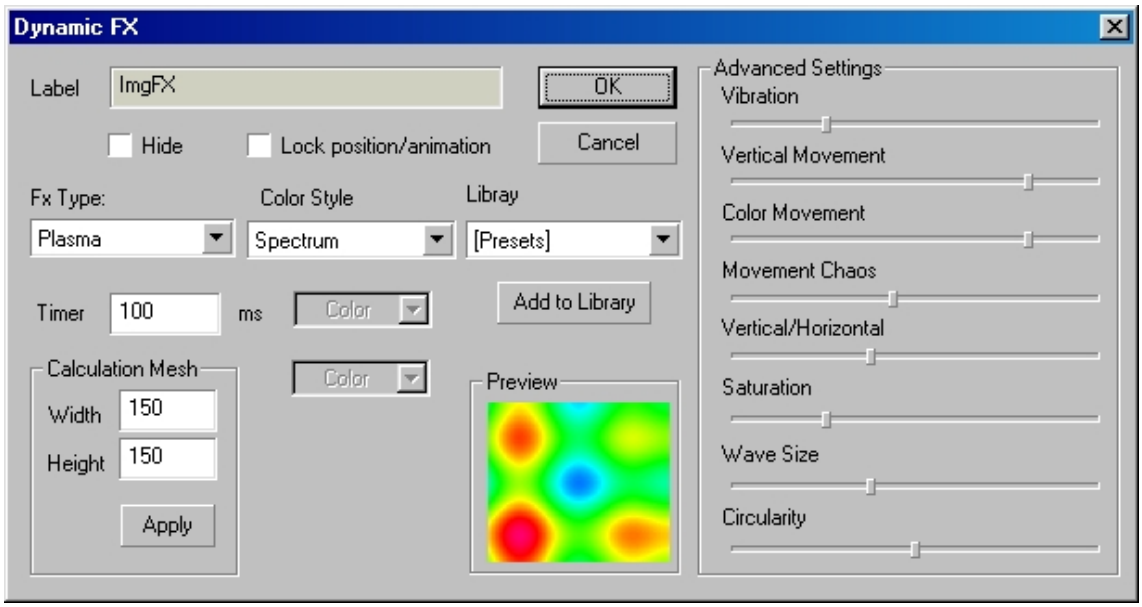
این قسمت معین می‌کند که فایل صوتی و یا تصویری چگونه اجرا شود. یعنی بلافاصله پس از اجرای برنامه page (Right after start) اجرا شود یا توسط اسکریپت(Use MCI Object in Script to Conrtol it) کنترل شود.

- **(Dynamic fx)**



این شی می‌تواند جلوه‌های گوناگون و جالبی را در صفحه و یا بر روی عکس‌ها ایجاد کند.جلوه‌هایی از قبیل آتش، دود و... Dy namic fx ها جای زیادی اشغال نمی‌کنند ولی فضای زیادی از CPU را به خود اختصاص می‌دهد.

-خصوصیت های شی Dynamic fx



fx Type

ایـن قـسمت نـوع fx را تعیـین مـی‌کنـد. در ایـن قـسمت نـوع‌هـای Plasma (پلاسـما) ، Smoke (دود) Fire (آتـش)، Fire / 3D Dots(نقطـه‌هـای ۳ بعـدی / آتـش)، Bumbing Light (نـور برجـسته) و Roto Zoom (بـزرگ‌نمـایی چرخنده) موجود است.

Color Style

ایـن قـسمت سـبک رنـگ را مـشخص مـی‌کنـد. ایـن قـسمت شـامل سـبک‌هـای Spectrum (طیـف)، Red Spectrum (طیـف قرمـز)، Blue Lumi(آبـی روشـن)، Green Lumi(سـبز روشـن)، Blend Color(آمیـزه رنـگ)، Acid(اسید) می باشد.

Add to Library

چنانچه در هنگام تنظیم سـبک رنـگ fx بـه ترکیـب جالـب و قـشنگی دسـت یـابیم و مایـل باشـیم کـه آن را ذخیـره کنـیم می‌توانیم از دکمه Add to Library استفاده کنیم.

Library

این قسمت شامل لیستی از ترکیباتی است که قبلاً به کتابخانه اضافه شده‌اند.

Timer

این قسمت سرعت حرکت D-fx را معین می‌کند.

Calculation Mesh

این قسمت میزان درهم رفتگی مولفه‌های D-fx را تعیین می‌کند.

Preview

پیش نمایشی را از D-fxنشان می‌دهد.

Advenced Setting

در این قسمت یک سری تنظیمات پیشرفته را می‌توانیم بر روی D-fx اعمال کنیم.

Horizontal Movment

این قسمت سرعت حرکت افقی مؤلفه‌ها را تنظیم می‌کند.

Vertical Movment

برای تعیین سرعت حرکت عمودی مؤلفه‌ها است.

Interference

این گزینه مقدار تداخل مؤلفه‌ها را تنظیم می‌کند.

Chuncks

غلظت مولفه‌ها را تعیین می‌کند.

Vibration

میزان لرزش مؤلفه ها ومقدار بی نظمی را تعیین می‌کند.

Color Movment

این گزینه سرعت حرکت وتغییر رنگ را تعیین می‌کند.

Saturation

این گزینه میزان اشباع رنگ را تعیین می‌کند.

Wave size

این قسمت اندازه موج را معین می‌کند.

Circularity

جهت تعیین میزان دایره‌وار بودن مولفه‌ها است.

Cut off

این گزینه محدوده‌ی بین ۲ مولفه‌ی fx را تعیین می‌کند.

Horizontal wave

این گزینه حالت موج افقی را مشخص می‌کند.

Vertical wave

این گزینه حالت موج عمودی را مشخص می کند.

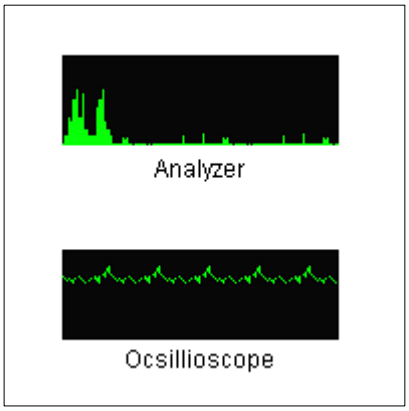
X,Y,Z Rotation

این گزینه‌ها به ترتیب چرخش در راستای X,Y,Z را مشخص می‌کند.

Burning Speed

سرعت سوختن را مشخص می‌کند.

- **Audio VISUALIZTION(رقص نور)**



توسط این قابلیت منحصر به فردMMB می‌توانید احساسی بصری به آهنگ‌های در حال اجرا ببخشید. بدین معنا که یک رقص نور جهت القای این حس در پروژه خود قرار دهید. از این رقص نور می‌توان برای اجرای فایلهای صوتی S3M,XM,OGG,Wav,MP3 استفاده نمود.در حقیقت این شی آنچنان سودمند نیست ،جز اینکه کاربران به هنگام گوش کردن به یک آهنگ تمایل به دیدن رقص نور را دارند. برای کنترل بهتر رقص نور می‌توانید نوع آن را نیز تغییر دهید. ۲ نوع رقص نور موجود است:

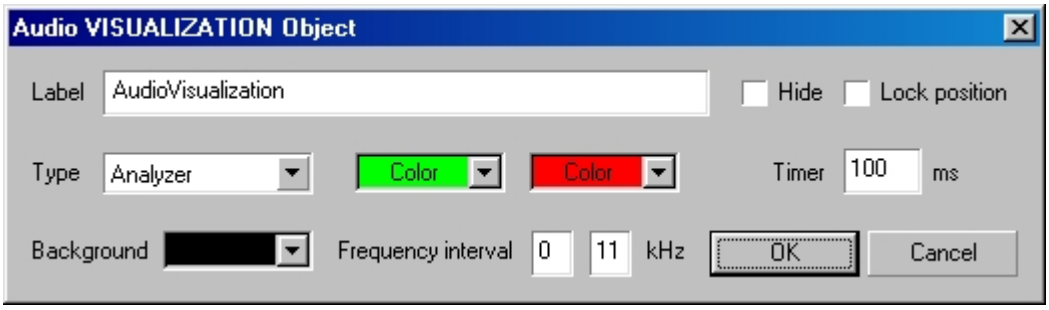
۱-Oscilloscope(نوسانی) ۲-Analyzer(تحلیلی)

-خصوصیت‌های شی AV

Timer

سرعت به‌روز رسانی را بر حسب زمان تعیین می‌کند.

Background

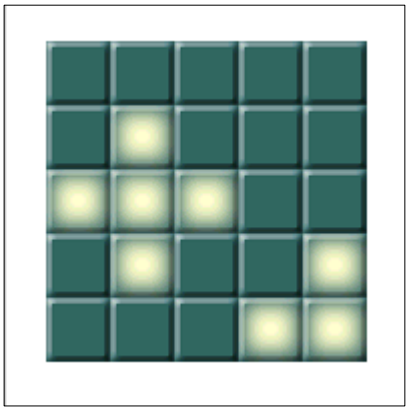


این قسمت رنگ زمینه رقص نور را تعیین می‌کند.

Frequency Interval

محدوده‌ی فرکانس‌ها را جهت پردازش معین می‌کند.

- **Image Matrix(ماتریس)**

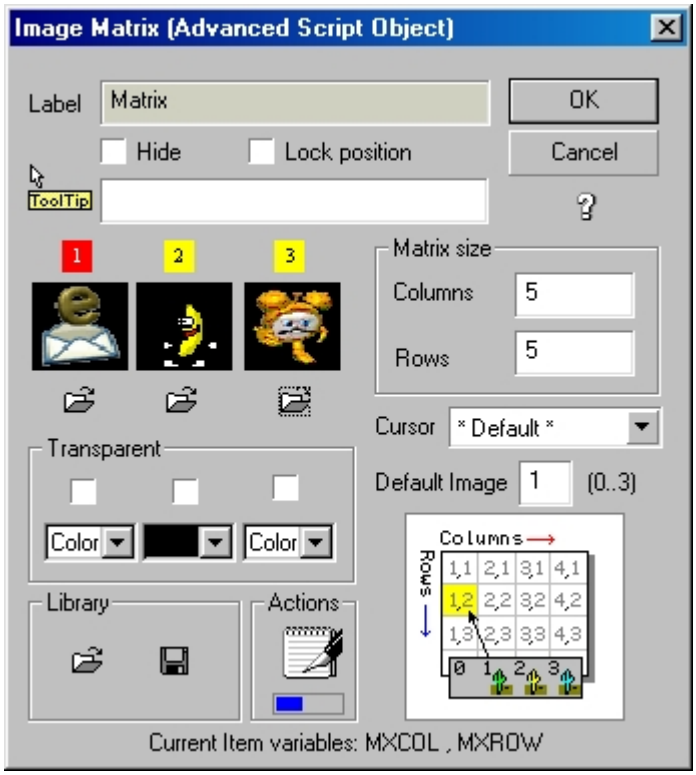


تصور کنید که قصد ساختن صفحه یک بازی را دارید که دارای ۵×۵ قسمت یکسان است و در هر قسمت صفحه می‌توانید ۳ شکل بر روی هم داشته باشید. چنانچه قوانین بازی حکم نماید که این عکس‌ها ترتیب‌شان تغییر نماید شما باید از فرامین Hide / show برای انجام این عمل استفاه کنید. اما MMB با استفاده از شی Matrix راه حل مناسبی را پیشنهاد می‌کند. به طور کلی، با این شی می‌توان اشیایی را تولید کرد که وضعیت آنها بایستی با کلیک موس تغییر کند. به عنوان مثال توسط این شی می‌توان اقدام به تولید Radio Button ،Check Box (دکمه رادیویی)و... نمود. تعداد سطرها و ستونهای شی ماتریس اندازه شی ماتریس را مشخص می‌کنند. ضمناً جهت استحصال نتیجه مطلوب باید هر ۳ شکل قرار داده شده هم اندازه باشند.برای مثال از این شی می توان به عنوان نشان دهنده وضعیت استفاده کرد.بدین ترتیب که برای وضعیت های مد نظر خود تصویری خاص را برگزینیم تا در هنگام برقراری شرایط نمایان شود.برای تعداد سطر و ستون ها محدودیتی وجود ندارد اما بهتر است که حتی الامکان از تعداد کمتری استفاده کنیم.

سطر

1.1	2.1	3.1
1.2	2.2	3.2
1.3	2.3	3.3

- خصوصیت های شی Image Matrix



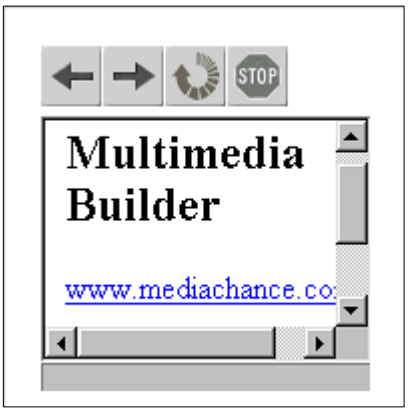
1,2,3-

این قسمت ها محل بارگذاری تصاویر می‌باشد که تنها مجاز می باشید برای هر خانه از ماتریس ۳ تصویر مجزا وارد نمایید.هرچند با استفاده ماهرانه می توان تعداد آنها را بیشتر نمود.

Matrix Size

این گزینه اندازه ماتریس را معین می‌کند. Columns (ستونها) و Rows(سطرها).

- **HTML Browser**



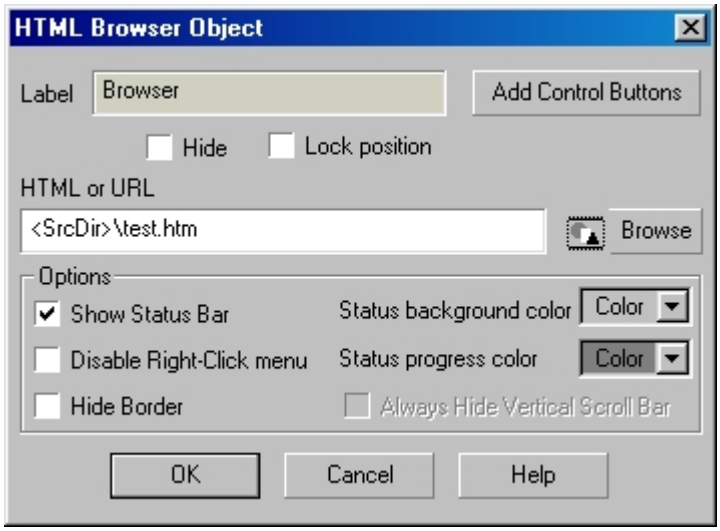
شی HTML یک جستجوگر کامل در MMB می‌باشد. این شی از کنترل‌های IE استفاده می‌نماید و این بدان معناست که این شی به Internet Explorer وابستگی دارد. ولی در عین حال این وابستگی مزیت‌هایی را به ارمغان می‌آورد:

- قادر به چاپ محتوی شی HTML هستید.
- می‌توانید از اسکریپتهای Jave ، اپلت‌ها، فرم و تمام چیزهایی که با IE سازگار است سود جوئید.
- امکان اتصال به اینترنت
- امکان استفاده ازFTP
- می‌توانید از شیHTML برای اجرای ActiveX و فایلهای Pdfاستفاده نمایید
- سازگاری با IE نسخه4.x به بالا
- قابل کنترل توسط اسکریپت نویسی
- سفارشی سازی ظاهر HTML

همان طور که بیان شد با استفاده از این شی می توانید محتویاتی را که برنامه Internet Explorer قادر به نمایش آنهاست را در MMB نمایش دهید.به عنوان مثال می توانید با پایگاه داده ها ارتباط برقرار کنید و یا تصاویر با فرمت های گوناگون را بار گذاری نمایید.

- خصوصیت‌های شی HTML

HTML or URL



این قسمت تعیین می‌کند که در هنگام اجرای برنامه شی HTML حاوی چه چیزی باشد. این محتوی هم می‌تواند یک فایل HTML الحاق شده باشد و هم آدرس سایتی خاص.

Options

این قسمت یکسری انتخاب‌ها را در اختیار ما قرار می‌دهد.

- **Show Status Bar**

در صورت فعال بودن نوار وضعیت را نمایان می‌سازد.

- **Disable Right Click Menu**

در صورت فعال بودن منویی را که به هنگام راست کلیک کردن نمایان می‌شود را غیر فعال می‌کند.

- **Hide Boarder**

حاشیه شی HTML را مخفی می‌کند.

- **Status Backgrund Color**

رنگ پس زمینه نوار وضعیت را تعیین می‌کند.

- **Status Progress Color**

این گزینه رنگ نوار پیشرفت نوار وضعیت را تعیین می‌کند.

- **Always Hide Vertical Scroll Bar**

نوار پیمایش عمودی را مخفی می‌کند.

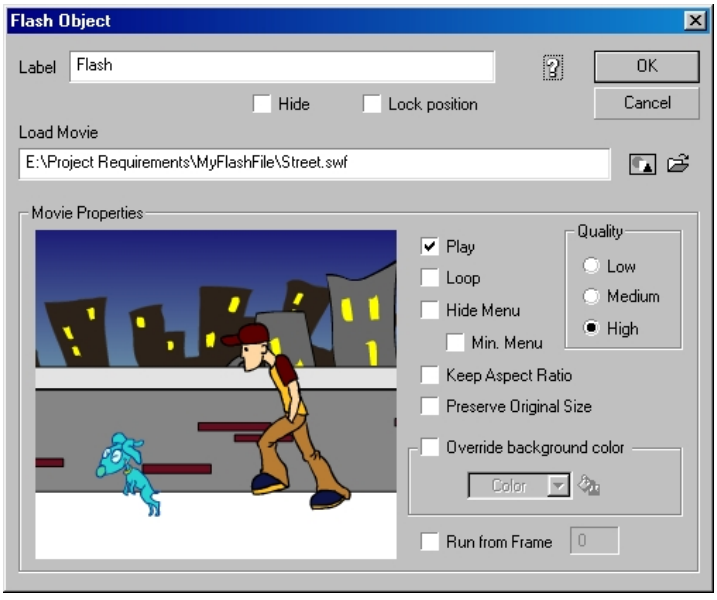
- **Add Control Button**

این قسمت چهار دکمه ی Back,Forward,Refresh,Stop را جهت تعامل با شی HTML ارائه می‌دهد.

- **Flash**

شی فلش (Flash) یک شی کارآمد و پرکاربرد است.از این شی جهت نمایش دادن کلیپ های ساخته شده توسط نرم افزار فلش استفاده می شود.جالب است بدانید که در MMBاز شی فلش تنها برای اجرای کلیپ(Clip)های فلش استفاده نمی‌شود بلکه توسط Action Script برنامه Macromedia Flash می‌توانید فرامین MMB فراخوانی کنید و پروژه را کنترل کنید. بنابراین با استفاده از شی فلش می‌توانید بر قدرت و توانایی‌های تولید خود بیفزایید. به عنوان مثال می‌توانید از توابع ریاضی و سایر فرامین فلش استفاده نمایید و نتیجه را به MMB ارسال کنید. البته این گونه از استفاده مستلزم دانستن Action Script فلش می‌باشد. شما می‌توانید فایل فلش مورد نظر خود را به برنامه الحاق نمایید و یا به صورت خارجی از آن استفاده کنید. حتی این قابلیت نیز فراهم است که فایل فلش مورد نظر را از روی اینترنت بارگذاری نمایید.

- خصوصیت‌های شی Flash



Load Movie

برای بارگذاری فایل‌های SWF می‌باشد.

Movie Properties

این قسمت جهت تنظیم پارامترهای فایل SWF بارگذاری شده است.

- Play

سبب اجرا شدن فایل SWF

بلافاصله پس از اجرا شدن برنامه می‌شود.

- Loop

سبب اجرای مکرر فایل SWF می‌شود.

- Hide Menu

این گزینه منویی را که به هنگام راست کلیک بر روی فایل SWF ظاهر می‌شود را مخفی می‌کند.

- Min.Menu

این گزینه به هنگام راست کلیک بر روی شی فلش منوی کوچک شده را نشان می‌دهد.

- Keep Aspect Ratio

این گزینه مانع از به هم ریختن نسبت وجوه می‌شود. این گزینه به هنگام تغییر اندازه شی کارآمد است.

- Preserve Original Size

انتخاب این گزینه سبب نگه داشتن اندازه اصلی تحت هر شرایطی می‌شود.

- Override Backgrund Color

این گزینه رنگ پس زمینه فایل SWF را مشخص می‌کند.البته این کار همیشه امکان پذیر نیست.

- Run from frame

این قسمت شروع اجرای فایل SWF را بر اساس فریم داده شده آغاز می‌کند.

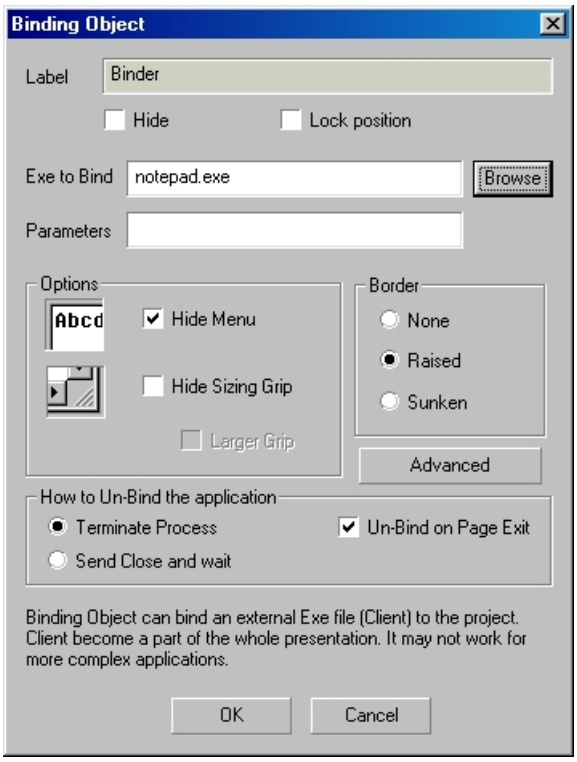
- Quality

این قسمت کیفیت فایل SWF را تعیین می کند. Low (کیفیت پایین) Medium (کیفیت معمولی) وHight (کیفیت بالا).

- **Binding object**

شی Binder راهی ساده است برای نهادن و یا برقراری ارتباط با یک فایل اجرایی(Exe) خارجی. در صورتی که فایل اجرایی مورد نظر به برنامه الحاق شود جزئی از اجرا کننده MMB محسوب می‌شود. به عنوان مثال یک انیمیشن فلش به همراه یک اجرا کننده (Flash Player) ،Notepad با یک متن و ...

نکته: توجه کنید که تمامی فایل‌های اجرایی (exe) نمی‌توانند در پروژه جا سازی شوند چرا که بعضی از آنها به خاطر عدم سازگاری سبب توقف فایل اجرایی و در نهایت تولید نقص در اجرای پروژه می‌شود. توسط MMB می‌توانید منوهای فایل اجرایی اتصال یافته را کنترل نمایید. حتی در صورت تمایل می توانید بیشتر از یک شی Binder را در پروژه خود قرار دهید.مثالی از این نوع برنامه اجرایی EditBox است که جهت رفع محدودیت های EditBox برنامه MMB ساخته شده است.



- خصوصیت‌های شی Binder

Exe to Bind

این قسمت فایل اجرایی(Exe) را جهت اتصال مشخص می‌کند.

Parameters

در این قسمت بر اساس فایل اجرایی می‌توان پارامترهایی را وارد کرد تا برنامه اتصال یافته عملی را انجام دهد.استفاده از این قسمت به هنگام به کار گیری برنامه های CommandLine سودمند می باشد.

Options

- Hide Menu

این گزینه منوهای فایل اتصال یافته را مخفی می‌کند.

- Hide Sizing Grip

این گزینه قسمتی از برنامه را که برای تغییر اندازه پنجره است را مخفی می‌کند.

- Larger Grip

محـدوده‌ای را کـه در سـمت راسـت و پـایین برخـی از برنامـه جهـت تغییـر انـدازه پنجـره برنامـه مـی‌باشـد را بزرگتـر می‌کند.

- Border

نوع حاشیه را تعیین می‌کند. None (هیچ)، Raised(برجسته)، Sunken(فرو رفته)

- How to Un-bind the application

این قسمت مشخص می‌کند که برنامه اتصال یافته چگونه از حالت اتصال خارج شود.

-Terminate Proccess(پایان بخشیدن به پردازش)

به صورت قاطع و بدون توجه به وضعیت اجرایی برنامه از آن خارج می‌شود.

-Send close and wait

روشی ملایمتر جهـت بـستن فایـل اجرایـی اتـصال یافتـه اسـت. بـدین معنـا کـه بـه برنامـه اجرایـی فرصـت بـسته شـدن و احیاناً پرسشی مبنی بر ذخیره فایـل وابـسته بـه برنامـه جاسـازی شـده را ارائـه مـی‌دهـد. از ایـن گزینـه در صـورت ناسـازگار بودن روش قبل می‌توان استفاده کرد.

-Un Bind On Page Exit

به محض انتقال از صفحه حـاوی شـی Binder اجـرای فایـل اجرایـی اتـصال یافتـه متوقـف مـی‌شـود.همچنـین در صـورت بازگشت به صفحه حاوی شی Binder مجدداً فایل جاسازی شده اجرا می‌شود.

Advanced Properties

برخـی از برنامـه‌هـای پیچیـده ممکـن اسـت دارای چنـدین پنجـره وابـسته باشـند و یـا برخـی از قـسمتهای آنهـا ایجـاد مزاحمت نماید. به همـین سـبب MMB ممکـن اسـت کـه پنجـره‌ای نادرسـت را جهـت بـاز کـردن و اتـصال انتخـاب کنـد. در ایـن مواقـع اسـت کـه خـصوصیات پیـشرفته بـه کمـک MMB مـی‌آیـد. بـرای حـل ایـن مـشکل مـی‌توانیـد بـه MMB بگویید که عنوان پنجره مورد نظر حاوی چه رشته‌ای است و یا بر عکس حاوی چه رشته‌ای نیست.

نکتـه: در برنامـه‌هـای سـاده از قبیـل FlashPlayer,Notepad احتیـاجی نیـست کـه بـه MMB بگوییـد پنجـره مـورد نظـر حاوی چه نوشته‌ای است. این ویژگی زمانی به کار می‌آید که MMB در تشخیص پنجره‌ها اشتباه کند.

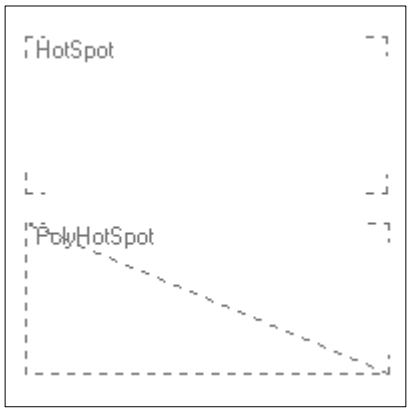
-Wait Before Binding

برخـی از برنامـه‌هـا ممکـن اسـت زمـان بیـشتری بـرای بارگـذاری نیـاز داشـته باشـند چـون MMB بلافاصـله سـعی در اجرای برنامه دارد ممکن اسـت برنامـه در زمـان شـروع سـریعاً اجـرا نـشود، بـه همـین منظـور بایـستی از ایـن ویژگـی سـود جست، چرا که این ویژگی سبب توقف MMB تا زمان بارگذاری کامل برنامه اتصال یافته می‌شود.

Additional Cut from Top or bottun-

از دیگر مزیت‌های Binder این اسـت کـه شـما بـه آسـانی قادریـد محـدوده‌ای مـشخص از بـالا و یـا پـایین برنامـه اتـصال یافته را پنهان نمایید. از این ویژگـی بیـشتر زمـانی اسـتفاده مـی‌شـود کـه بخواهیـد نـوار ابزارویـا نـوار وضـعیت یـک برنامـه را پنهان کنید.

- **Hot-Spot**



یک شی فعال مـی‌باشـد. Hot-Spot یـک شـی و یـا بـه عبـارتی محـدوده‌ای غیـر قابـل دیـد اسـت کـه در پـروژه تعریـف شده و کاربر نهایی می‌توانـد بـا کلیـک کـردن روی آن ویـا انتقـال مـوس بـر روی آن سـبب اجـرای اسـکریپتی شـود. Hot-Stop در جلوی دیدگان کـاربر نهـایی غیـر قابـل دیـد اسـت و تنهـا شـما در زمـان طراحـی مـی‌توانیـد محـدوده‌ی آن را کـه توسط خط چین مشخص شده است را مشاهده نمایید.اما باید گفت که ۲ نوع Hot-Spot موجود است:

۱-Hot-Spot (چهار ضلعی)

۲-Polygonal Hot-Spot (چند ضلعی)

که هر کدام بسته به نیاز بـه کـار مـی‌آیـد. از شـی Hotspot اسـتفاده هـای زیـادی مـی تـوان کـرد هماننـد قـرار دادن ان بـه عنوان چهارچوب بارگذاری عکس ها و یا برای غیر فعال سازی برخی اشیا در زمان اجرای برنامه.

- **Script**

شی Script یک شی غیر فعال است که در هنگام اجرای برنامه قابل دید نیست. شی Script می‌تواند توسط میانبرها فراخوانی ویا از جایی دیگر اجرا شود. این شی به شما این امکان را می‌دهد که کد برنامه‌ی خود را در صورت نیاز بدون استفاده از موس اجرا کنید. این شی در هنگام اجرای برنامه نامرئی می‌باشد و تنها اثر آن محسوس است.

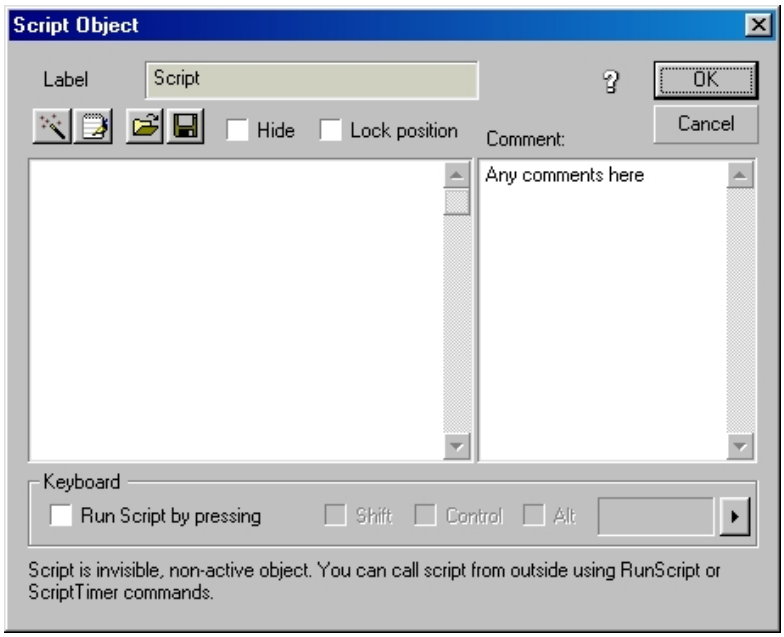
-خصوصیت‌های شی Script

Comment

جهت اضافه کردن توضیحاتی به کد برنامه می‌باشد.توضیحات می توانند سبب خوانایی بیشتر کد ها شوند.

Shortcuts

برای تعیین کلیدهای میانبر می‌باشد.بدین ترتیب چنانچه میانبری برای شی اسکریپت تعریف کرده باشیم ، اگر در زمان اجرای پروژه کلید های مشخص شده را بفشاریم اسکریپت مربوط هاجرا خواهد شد.



خصوصیت های مشترک بین اشیاء

همانطور که می‌دانید تمام اشیاء در MMB دارای خصوصیاتی می‌باشند که برخی ازاین خصوصیات تقریباً در تمامی اشیاء مشترک می‌باشد و ماهم به جهت جلوگیری از تکرار آنها قصد داریم تا آنها را در این قسمت توضیح دهیم. شایان ذکر است که به ۲ صورت می‌توانیم به کادر خصوصیت یک شیء در MMB دسترسی پیدا کنیم اول اینکه با ۲ بار کلیک بر روی شیء مورد نظر و راه دوم توسط گزینه properties از منوی Object است.بقیه خصوصیات منحصر به فرد هر شیء در قسمتهای قبلی بررسی شد.

- Label (نام شی)

در این کادر میتوانیم بر چسبی رابه شیء اختصاص دهیم.توصیه می‌کنیم که حتی‌الامکان از اسامی معنادار استفاده کنید چرا که بعداً در اسکریپت نویسی راحت‌تر می‌توانید با اشیاء تعامل برقرار کنید.

- Hide (پنهان سازی)

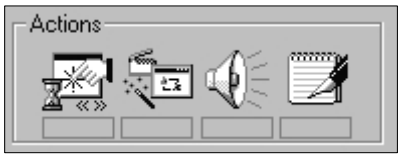
این گزینه شیء را هم در زمان طراحی و هم در زمان اجرا مخفی نگه می‌داردو تا زمانی که توسط دستوری شیء از نو نمایان شود،حالت مخفی بودن ادامه خواهد داشت. ضمناً هنگامی که شیء‌ای در زمان طراحی مخفی باشد کنار نام آن شیء در پنجره object ضربدر قرمزی نمایان می‌شود.

- Lock position (قفل کردن موقعیت)

با استفاده از این گزینه میتوان در هنگام طراحی،از حرکت شی، ممانعت کرد. این ویژگی هنگامی بیشتر کارآمد است که صفحه حاوی اشیاء ریز و درشت بسیاری باشد و ما بخواهیم از تغییر مکان ناخواسته اشیاء جلوگیری کنیم.

- Font (نوع قلم)

این ویژگی جهت تغییر نوع قلم متن‌های شی است.



- Actions (فعالیت‌ها)

توسط این قسمتها فعالیت‌های پروژه مشخص می‌شود. قسمت Action شامل چهار بخش:

- External command & Page actions (فرمان خارجی و فعالیت‌های صفحه)

با اعمال این قابلیت میتوان فعالیت‌هایی نظیر نصب یک برنامه، اجرای یک پروژه خارجی، جستجو و گردش در اینترنت ،ارسال نامه الکترونیکی ،باز کردن یک سند، نمایش فایل راهنما، اجرای فایل MMB دیگر، جستجو در دیسک سخت،بارگذاری واجرای فایل های صوتی،حرکت بین صفحات و خروج از صفحات را انتظار داشت.

- teractions with other object video (تعامل با اشیاء دیگر و ویدئو)

بسته به نوع رفتار موس (قرار گرفتن روی شیء یا کلیک بر روی آن) شی می‌تواند رفتاری از خود بروز دهد.برای مثال پنهان شدن و نمایش داده شدن،اجرای فایل ویدئویی و ... با استفاده از این قسمت میتوانید یک پروژه‌ی حرفه‌ای داشته باشید.

Sound Action (فعالیت صدا)

باز هم بسته به نوع قرارگیری موس (روی شیء یا کلیک بر روی شیء) MMB میتواند صدای مربوطه را اجرا نماید.لازم به تذکر است که MMB از چندین کانال حمایت میکند این بدین معناست که MMB میتواند همزمان چندین صدا را اجرا نماید و نگرانی از بابت بخش آهنگ زمینه به خاطر ایجاد وقفه وجود ندارد.البته در این حالت ویژگی DirectSound باید فعال باشد.

More actions (فعالیت‌های بیشتر)

با توجه به اینکه شما میتوانید یک سری فعالیت‌های اصلی وساده را به اشیاء اعمال کنید،برای کنترل بهتر پروژه‌ها و سودمند کردن آنها میتوان از این قسمت جهت اسکریپت نویسی استفاده کرد و در نهایت پروژه تعاملی‌تری را تولید نمود.

Cursors (مکان نما)

با استفاده از این قسمت می‌توان مکان نمایی را که به هنگام قرارگیری بر روی شیء بایستی ظاهر شود مشخص کرد.

Tooltip (توضیح مختصر)

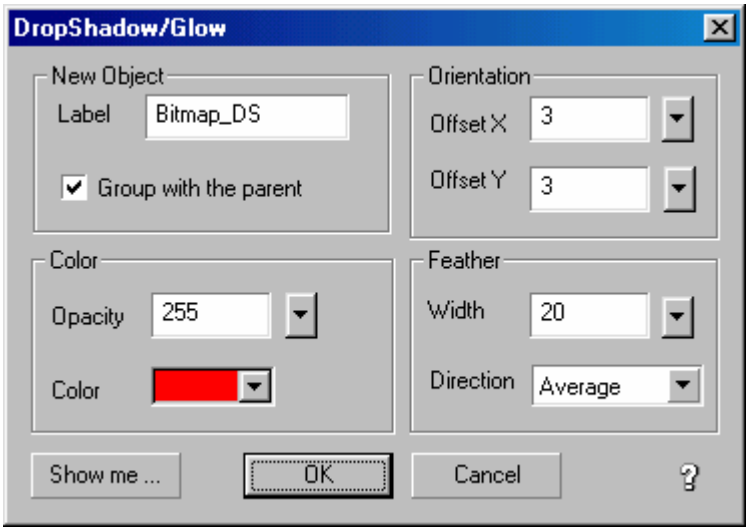
حتماً تا به حال پس از قرارگیری موس بر روی دکمه‌ای یا قسمتی در برنامه‌های دیگر مشاهده کرده‌اید که متن یا عبارت مختصری موقتاً جهت راهنمایی ظاهر میشود و پس از حرکت مجدد موس دوباره محو میشود این قابلیت همان tooltip می‌باشد که ما میتوانیم برحسب نیاز به اشیاء در MMB آن را اضاف کنیم.

جلوه های MMB (Effects)

همـان طـور کـه قـبلا در بررسـی منوهـای MMB مـشاهده کردیـد،می توانیـدجلوه هـایی را بـه اشـیا اعمـال نماییـد،جلوه هـایی نظیـر (برجـسته سـازی)Bevel ،(سـایه)Shadow،(آتـش)Fire،(درخـشش)Glow،CutOut.در ادامـه قـصد داریـم تـا پارامترهای این جلوه ها را بررسی نماییم.

Drop Shadow

این جلوه برای ایجاد سایه می باشد.اندازه سایه و رنگ سایه نیز قابل تنظیم است.



New Object

سایه ای را که MMBایجاد مـی کنـد در نهایـت بـه صـورت یـک شـی تـصویری در مـی آیـد کـه در ایـن قـسمت مـی تـوان برچـسبی را بـرای ایـن شـی در نظـر گرفـت.گزینـه Group with the parent سـایه و شـی اصـلی را در یـک گـروه قـرار می دهد.

Color

همان طور که واضح است این قسمت برای تعیین رنگ سایه می باشد.گزینه Opacityبرای تعیین غلظت می باشد.

Orientation

برای تعیین موقعیت سایه می باشد.

Feather

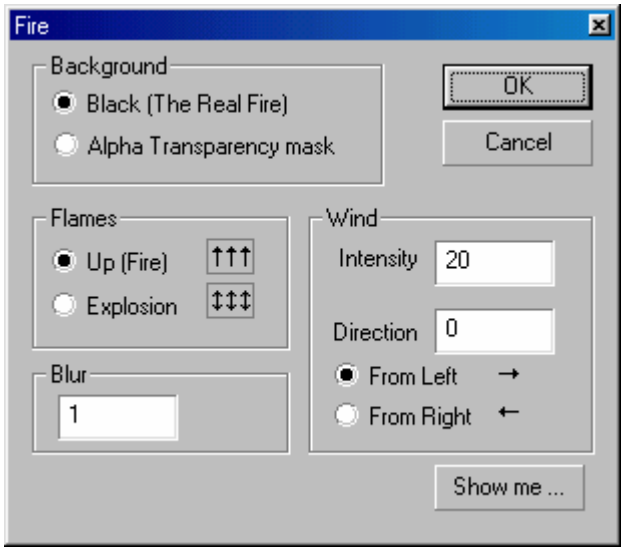
این قسمت برای ایجاد هاله ای اطراف سایه است.Width عرض این هاله را معین می کند و Direction نیز جهت هاله را مشخص می کند.

Glow

این قسمت جلوه درخشندگی را اعمال می کند.پارامترهای این جلوه نیز کاملا شبیه جلوه سایه است.

Fire

این قسمت جلوه آتش را بر حاشیه اشیا اعمال می کند.



Background

برای رنگ زمینه جلوه آتش می باشد.Black زمینه مشکی را به آتش اضاف می کند و Alpha Transparency mask زمینه آتش را به صورت شفاف در می آورد.

Flame

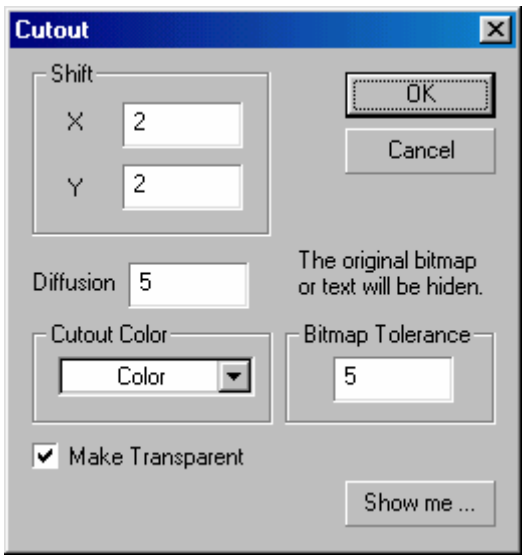
این قسمت برای مشخص کردن جهت شعله های آتش است.

Wind

توسط ایـن ویژگـی مـی تـوان اثـر بـاد را بـر آتـش اعمـال کـرد.Intensity شـدت بـاد را تعیـین مـی کنـد،Direction بـرای جهت می باشد.

CutOut

این قسمت حالت فرروفتگی را به تصاویر اعمال می کند.البته این جلوه از کیفیت بالایی برخوردار نمی باشد.



Shift

موقعیت را تعیین می کند.

Diffusion

برای مشخص نمودن حالت انتشاری و گستردگی نور است.

Cutout Color

رنگ این جلوه را معین می کند

Bitmap Tolerance

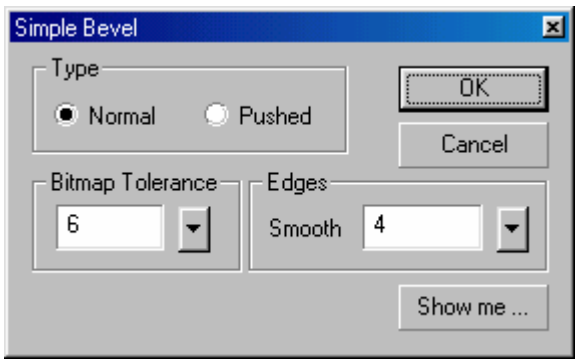
شدت تحمل Bitmap را مشخص می کند.

Make Transparent

خصوصیت Transparencyرا به این جلوه اضاف می کند.

Bevel

این جلوه برای برجسته سازی حاشیه تصاویر است.



Type

گونه ی برجسته شدن را مشخص می کند.

Bitmap Tolerance

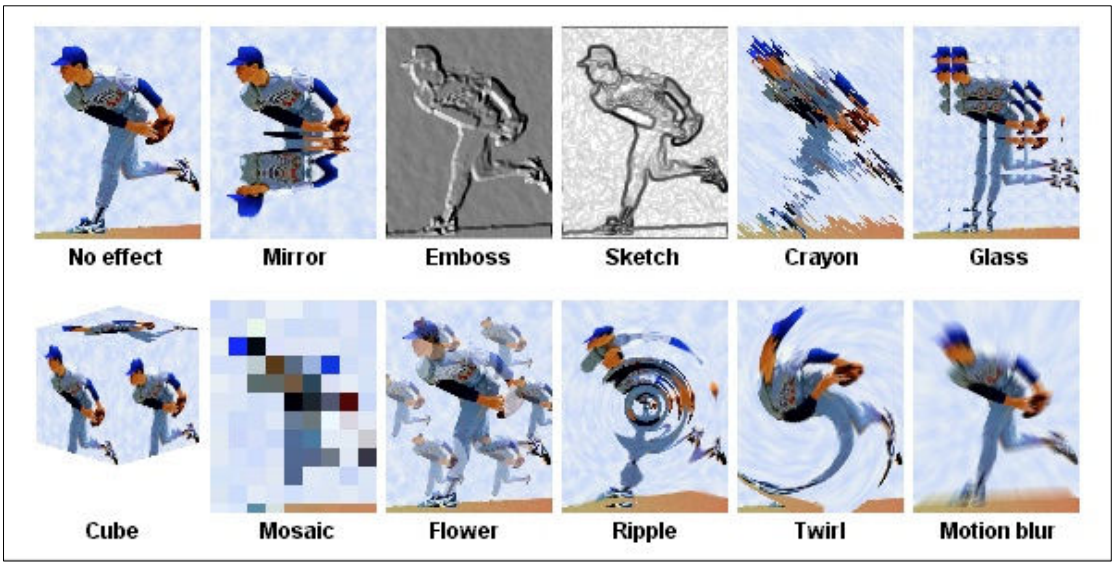
شدت تحمل Bitmap را مشخص می کند.

Edge

مقدار هموار بودن لبه ها را مشخص می کند.

Special Effects

MMB در ایـن قـسمت یکـسری از فیلترهـایی را کـه بـر تـصاویر اعمـال مـی شـود را ارائـه مـی دهـد.کـسانی کـه بـا برنامه Photoshop کار کرده باشـند بـا ایـن فیلتـر هـا آشـنایی دارنـد.تـصویر زیـر اثـر هـر یـک از ایـن فیلترهـا را نـشان مـی دهد:

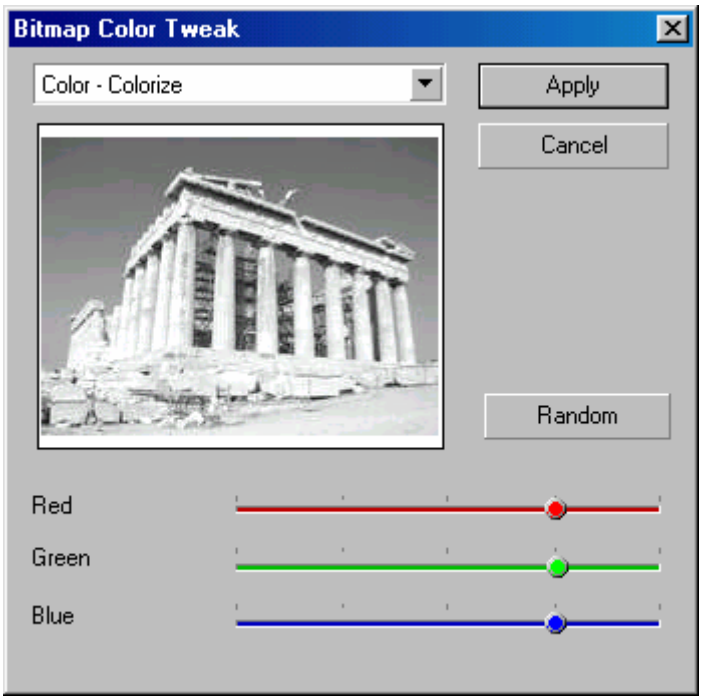


Interactive Effect

در این قسمت نیز سه جلـوه تـصویری دیگـر بـرای عکـسها ارائـه شـده اسـت.جلـوه Wrap حالـت خمیـدگی و مچـالگی در عکس ها ایجـاد مـی کنـد،Impressionist بـرای مخـدوش کـردن عکـس هـا اسـت و Metal Light بـرای اعمـال رنگـی بـا ته مایه های رنگ فلزات اسـت.لازم بـه ذکـر اسـت کـه ایـن جلـوه هـا در کـادر خـصوصیت آنهـا و توسـط نـشانه گـر مـوس اعمال می شود.

Color Tweak

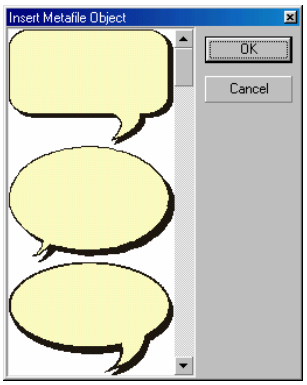
از این ویژگی کارامد برای تغییر رنگ،کنتراست(تمایز رنگ)،کیفیت،مقدار اشباع رنگ و... عکس هااستفاده می شود.



این قسمت گزینه های متعددی را در رابطه با تغییر رنگ و کیفیت عکس ها ارائه می دهد.هنگامی که یک گزینه را انتخاب نماییم بسته به نوع آن پارامترهای آن در زیر کادر Color Tweak ظاهر می شود.

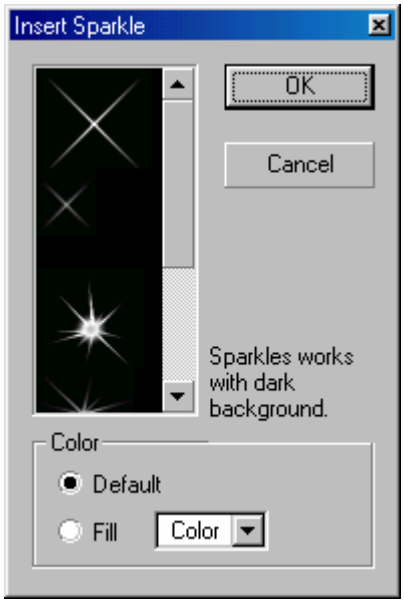
Bubble

برای ایجاد شکل حباب است.این ابزار از منوی ابزار قابل دسترسی است.کاربرد این اشکال به هنگامی است که بخواهیم مطلبی را به طور خاص بازگو کنیم،البته اسفاده از این ابزار بستگی به سلیقه طراح دارد.



Insert Sparkle

این قسمت نیز اشیایی براق را به پروژه اضاف می کند.رنگ این اشیا نیز قابل تغییر است.این ابزار از منوی ابزار قابل دسترسی است.



Crop

این ابزار جهت برش قسمتی از تصویر به صورت برشهای مستطیلی می باشد. این ابزار هنگام برش هیچ تاثیری بر درجه تفکیک (Res) و یا ابعاد تصویر نمی گذارد.

جهت برش تصویر به صورت زیر عمل نمایید:

۱-شی تصویر را انتخاب کنید.

۲-از منوی Effect قسمت Crop را برگزینید.

۳-یک مستطیل مجازی که محدوده برش را مشخص می کند بر روی تصویر بکشید.

چنانچه از نتیجه کار راضی نبودید و مایل به بازسازی تصویر اصلی می باشید،می توانید از منوی Effect گزینه Restore original را بگزینید و چنانچه از قسمت برش یافته راضی می باشید می توانید از منوی Effect گزینه

Make new original را برگزینید، این کار قسمت برش یافته را همانند یک تصویر اصلی می کند که بعداً باز می توانید به ویرایش آن بپردازید.

Make mew original

اگر قصد تغییر اندازه و یا ویرایش قسمت برش یافته از تصویر را داشته باشید بدون اعمال ویژگی Make mew original به همراه تغییر اندازه قسمت برش یافته، کل تصویر اصلی نیز تغییر می کند. حال برای جلوگیری از این حالت می توان از منوی Effect گزینه Make mew original را برگزید.

فصل سوم

اسکریپت نویسی

Multimedia Builder

- اسکریپت نویسی
- ویرایشگر اسکریپت
- ثابت ها
- متغیر
- آرایه
- عبارت شرطی
- حلقه ها
- جادوگر اسکریپت

به قسمت برنامه نویسی MMB خوش آمدید.همیشه با شنیدن برنامه نویسی تا حدودی تردید و ترسی در رابطه با یاد گیری آن ایجاد می شود.اما شیوه ی برنامه نویسی که MMB ارائه می دهد این شک و تردید را بی مورد می کند.به طور کلی قواعد و فرامینی که در این برنامه تعبیه شده است در مقایسه با سایر زبانهای برنامه نویسی بسیار ساده تر وقابل فهم تر است.

در این بخش ابتدا بر اصول برنامه نویسی تمرکز می شود،اصولی از قبیل مفاهیم ثابت ها،متغیرها و...سپس عبارت های شرطی و حلقه ها معرفی می شوند و در پایان نیز فرامین ارائه شده توسط MMB مورد بحث وبررسی قرار می گیرد.توصیه می شود که مطالب این قسمت را به دقت بررسی ومطالعه نمایید چرا که هر چه مهارتتان در برنامه نویسی بیشترو بهتر شود به همین نسبت می توانید نرم افزارهای قویتری تولید نمایید.

اسکریپت نویسی (Scripting)

یکی از سوالاتی که همیشه به ذهن کاربران MMB خطور می کند اینست که آیا آنها برنامه‌نویس محسوب می‌شوند یا خیر؟ درجواب باید گفت بله برنامه‌نویس محسوب می‌شوند حتی اگر یک خط برنامه رانوشته باشند.این مطلب اصلاً غیرطبیعی نیست چرا که هنگامیکه شما دستگاه پخش و ضبط ویدئویی خانگی خود را برای ضبط خودکار یک برنامه از تلویزیون تنظیم می نمایید عمل برنامه‌ریزی صورت پذیرفته است. نوشتن نمایشنامه خوب یک نوع برنامه‌ریزی و برنامه نویسی است،مهم نیست که چقدر بی‌ارتباط با مقوله‌ی کامپیوتر می‌باشد. به هر حال شما با انجام فعالیتهای مذکور برنامه‌نویس محسوب می‌شوید.

درحقیقت، پس از قرنها برنامه‌ریزی به صورت‌های گفته شده اخیراً پا درعرصه برنامه‌نویسی ماشینی نهاده‌ایم، ماشینها به واسطه برنامه‌ها کارها را ساده‌تر نموده اند. بله هر کاری که انجام کارهای ما را ساده‌تر نماید برای ما خوشایندست و قریب به یقین همه آرزو داریم تا بتوانیم به نحوی از این قدرت ساده‌سازی بهره‌مند شویم.اما آرزوی شما برآورده شده است و آن چیزی نیست جز MMB که شما را در محقق شدن آرزوهایتان یاری می‌کند.

اصول برنامه‌نویسی درMMB

در کار با MMB همانند سایر برنامه‌ها قواعد و قوانین خاصی وجود دارد که باید از آنها پیروی کنیم.در مقایسه با زبان‌های بشری وسایر زبانهای برنامه نویسی MMB یکی از ساده‌ترین و سریعترین زبانها را جهت فراگیری برنامه‌نویسی ارائه می‌دهد.

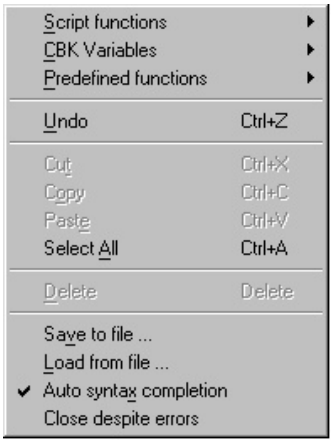
دانستنیهای اصلی

۱- برنامه‌نویسی توسط خطوط صورت می‌گیرد.همانند اینکه شما اقدام به تایپ یک داستان می‌نماییدو خط به خط آن را توسط ماشین تایپ می کنید.عمل نوشتن در MMB توسط صفحه کلید صورت می‌گیرد تاکدهای برنامه نوشته شود (یا همان داستان نوشته شود).و همان‌طور که ممکن است حدس زده باشید یک جمله در هر خط.

۲- جمله‌ها در کد برنامه هایی که توسط MMB تولید می‌شوند خط کد (codeline) نامیده می‌شوند.هر خط به طورمعمول نمایانگر یک دستور می‌باشد البته در این باره استثناهایی هم وجود دارد.

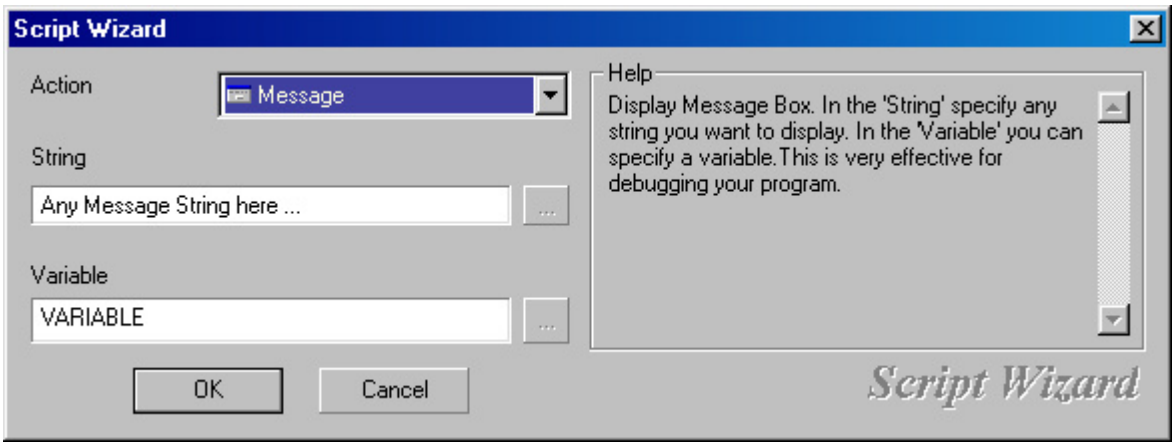
۳- احتیاجی به حفظ کردن نام فرامین MMB نیست. MMB به شما به چند طریق زیر درنوشتن کد برنامه کمک می‌نماید.

٤- کامل کردن قواعد (Syntax completion) : MMB درهنگام تایپ کد برنامه توسط شما به طور دائم نظارت می‌کند که چه چیزی را می نویسید و چنانچه براساس متون نوشته شده فرمانی را از لیست فرامین خود مناسب دید باقی کد برنامه را به صورت خودکار کامل می‌نماید، حال تنها چیزی که باید توسط شما تایپ شود یکسری پارامترهای مخصوص ومتغیرهای مدنظر شماست.

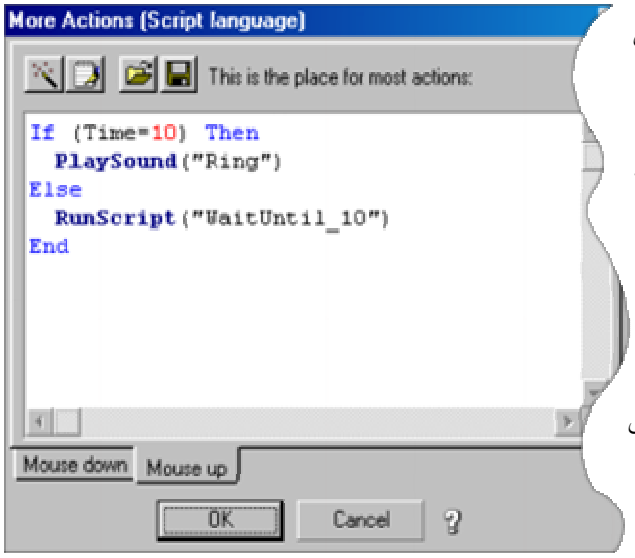


٥- منوی فرامین با استفاده از راست کلیک(Rightclick menu): یک لیست سریع فرامین توسط عمل راست کلیک دراختیار شماست تا بتوانید با انتخاب سریع فرمان مورد نظر عمل کدنویسی را تسریع بخشید.

٦-جادوگراسکریپت(Script wizard): توسط این ویژگی همواره یک لیست درونی از فرامین دراختیار شماست که با انتخاب هر فرمان شرح مختصری از آن به همراه محل ورود پارامترهای موردنظر جلوی روی شما ظاهر خواهد شد و همان طور که درابتدای این بحث بیان شد نیازی به حفظ کردن اسامی فرمان ها نمی‌باشد.



۷- **راهنمـای MMB**.هـر وقـت کـه احتیـاج بـه اطلاعات و جزئیـات بیـشتری داشـتید بـه راهنمـای MMB رجوع نمایید تا مشکلتان را رفع نمایید.



۸- **متمایزسـازی و پیکربنـدی کـدهای نوشـته شـده**.MMB بـه طـور خودکـار متنهـای مختلـف را رنگی می‌کند تـا درنهایـت سـبب آسـانی فهـم و افـزایش خوانـایی کـد شـود.بـه عـلاوه MMB بـه طـور خودکـار نحـوه قرارگیـری فـرامین را تنظـیم مـی‌کنـد.در شکل،شـما پیکربنـدی کـدهای نوشـته شـده راملاحظـه مـی‌نماییـد هـر خـط کـد بیان‌کننده شـرایط و فـرامین اسـت کـه برنامـه مـا را شـکل مـی‌دهنـد در مثـال بـالا برنامـه بـه قـرار زیر است:

- چنانچه ساعت ۱۰ باشد ساعت زنگ می زند.
- اگرساعت ۱۰ نباشد برنامه تا ساعت ۱۰ صبرمی‌کند.

اما چه جزئیاتی را می‌توان از مثال بالا دریافت نمود؟

- اسامی (ساعت) با رنگ مشکی نشان داده می‌شود.
- اعداد به صورت قرمز نمایان می شوند.
- شرایط وفرامین به صورت آبی نمایان می‌شوند (اگر ساعت ۱۰ باشد / اگر ساعت ۱۰ نباشد).
- یکسری فعالیتهای تعیین شده به صورت مشکی درمی‌آید.
- نحوه قرارگیری خطوط سبب خوانایی بیشتر کد می‌شود.

قواعد برنامه‌نویسی (Scripting Syntax)

فرامین از مهمترین قسمت‌های یک کـد مـی‌باشـند و توسـط آنهابـه MMB دسـتور مـی دهیـد کـه چـه کـاری را انجـام دهـد. به عنوان مثال تـصور کنیـد کـه شـما فرمانـده هـستید (برنامـه‌نـویس) ویـک ارتـش الکترونیکـی در اختیـار داریـد(MMB) هر چه که شما فرمان دهید صـورت مـی پـذیرد امـا نـه هـر فرمـانی! فرامینـی کـه MMB قـادر بـه شناسـایی و درک آنهاسـت را باید صادر کنیم.

در MMB چندین گونه فرمان موجود است، اما نترسید آنها تقریبا به یک صورت هستند با این تفاوت که نامهای متفاوتی دارند.

```
Command("Parameter1","Parameter(s)2")
```

تمامی فرامین در MMB حداقل از ظاهری شبیه فرمان بالا برخوردار هستند اما به مثال زیر توجه فرمایید.

همانند اینکه شما می‌توانید گلدان را بدون گل داشته باشید در MMB هم می‌توانید فرمان با پارامتر و یا بدون

گلدان دارای گل
فرمان با پارامتر

گلدان بدون گل همانند
فرمان بدون پارامتر

پارامتر داشته باشید در جدول زیر گونه‌های مختلف فرامین توصیف شده است:

گونه فرمان	مثال
بدون پارامتر	`Commandname( )`
با یک پارامتر	`Commandname("parameter")`
با چند پارامتر	`Commandname("parameter1","parameter2")`
با چند متغیر	`Commandname("variable1","variable2")`

توصیفات ذکر شده تنها صورت کلی از انواع دستورها هستند.جهت اطلاعات بیشتر با ما همراه باشید.

پارامترها (Parameters)

یکسری از داده‌ها را ارائه می‌دهند که توسط فرامین استفاده می‌شوند چندین گونه از این پارامترها موجود می‌باشد.

مثال فرمان	گونه پارامتر
`Commandname("hello, MMBers")`	متن (text)
`Commandname("4329")`	عدد (text)
`Commandname("my text object")`	نام اشیا (object lable)
`Commandname("60")`	زمان / درصد (time/ percentage)
`Commandname("c:\myvoice. Mp3")`	نام فایل / مسیر (file name/ path) پارامترهای ثابت fixed paramerters

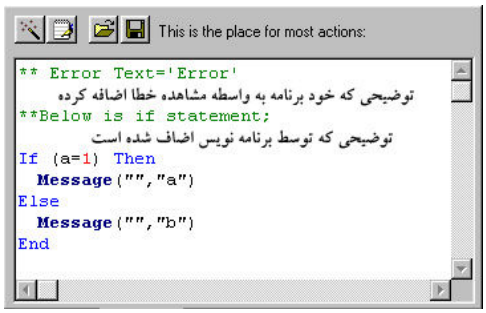
نگران نباشید در ادامه به طور کامل درباره‌ی پارامترها مطالب تکمیلی را خواهید آموخت.اما توجه نمایید که چگونه پارامترها از قسمت فرمان متمایز شده‌اند،بله آنها درون یک جفت("") کوتیشن(علامت نقل قول) محصور شده‌اند و چنانچه بیشتر از یک پارامتر موجود باشد بین آنها کاما قرار می‌گیرد.

```
Commandname("param 1")
```

```
Commandname("param 1","param 2")
```

گذشته از همه‌ی مطالب گفته شده باید به این حقیقت توجه کنیم که اسکریپتهای MMB به صورت سطری و یکی پس از دیگری پشت سر هم نوشته می‌شود.در MMB هیچ محدودیتی جهت تعداد خطهای یک برنامه وجود ندارد.و به خاطر بسپاریم که در MMB فرامین نوشته شده حساس به حروف (case sensitive) نیستند،بدین معنی که COMMAND NAME با command name هیچ فرقی ندارد اما باید توجه داشت که متغیرها و پارامترها حساس به حروف هستند یعنی کوچکی و بزرگی حروف در آنها حائز اهمیت است.

توضیح اسکریپت (Remark)



توضیحات یکی از قسمتهای مهم در کدنویسی هستند چرا که سبب خوانایی کد نوشته شده می‌شوند.شما می‌توانید آنها را میان خطوط کد قراردهید و یا در ابتدای کد پروژه بنویسید. قاعده ی نوشتن توضیح تنها به این صورت است که ابتدا باید دو ستاره (* *) تایپ کنیم و هر آنچه را که خواستیم در ادامه بنویسیم.

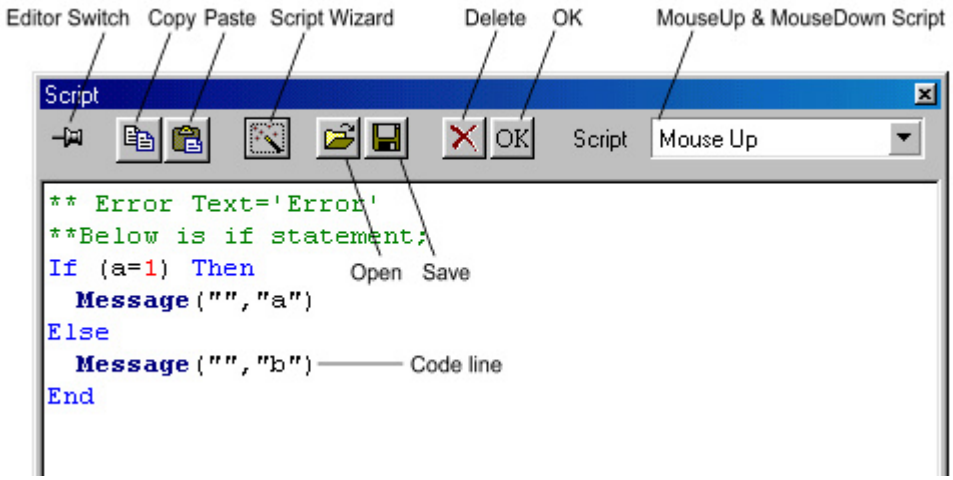
اما شرایطی هست که MMB خود توضیحی را به کد برنامه اضافه می کندو آن درصورتی است که MMB حین بستن پنجره اسکریپت نویسی که کد برنامه را از جهت صحت بررسی می‌کند چنانچه خطایی را مشاهده کند آن را با اضاف کردن ۲ ستاره و کلمه Error به توضیح تبدیل می‌کند. می‌توانیم جهت رفع خطا با برداشتن ۲ستاره و کلمه Error واعمال درست کد از ایجاد این نوع توضیح پرهیز کنیم.

ویرایشگر اسکریپت (Script Editor)

اما اجازه دهید قبل از فراگیری نحوه عملکرد و کاربرد ویرایشگر به معرفی اشیایی از MMB که دارای پنجره اسکریپت نویسی هستند بپردازیم:

- Text (متن)
- Button (دکمه)
- List Box (کادرلیست)
- Script object (شی اسکریپت)
- Hot-spot
- Image matrix (شبکه تصویر)
- اشیاء گرافیکی

اما اکنون نوبت به خود ویرایشگر می‌رسد:



جدول زیر در رابطه با قسمت‌های مختلف پنجره ویرایشگر توضیح می‌دهد.

قسمت	توضیح
Editor switch	این دکمه سبب تعویض پنجره اسکریپت نویسی از حالت کوچک به پیشرفته می‌شود.
Copy/Paste	قسمت کپی‌برداری و چسباندن متون از کلیپ برد(Clipboard).
Script wizard	این دکمه wizard فرامین را بازگشایی می کند که در انتخاب هر فرمان توضیح مختصر آن و کادر ورود پارامترها هم ظاهر می‌شود.
Open/Save	MMB را قادر می‌سازد که اسکریپت‌ها را ازفایل‌های متنی rtf, text بازگشایی و بارگذاری نموده و یا در صورت لزوم با همان فرمتهای مذکور ذخیره نماید.
MouseUp/Mouse Down	با توجه به این انتخاب پنجره‌ای جدید جهت نوشتن کد ظاهر می شود . Mouse up بدین معناست که هنگامی که نشانه‌گر موس بر روی شیء حاوی اسکریپت ازحالت کلیک خلاص می‌شود کد اجرا شود. Mouse down زمانیست که با موس کلیک کرده و هنوز دکمه موس را رها نکردیم.

مزیتی دیگر که پنجره ویرایشگر دارد این‌است که با ۲ بار کلیک بر روی یک فرمان، Wizard برنامه MMB ظاهر می شود تا سریعا بتوانیم درصورت لزوم تغییرات را اعمال کنیم.

تا کنون متوجه شدید که اسکریپتهای MMB کجا باید نوشته شود، پس به جاست که از آن درست استفاده کنیم و سعی در مراقبت از کدهایمان داشته باشیم چرا که ساعتهای زیادی را صرف نوشتن آنها خواهیم کرد.

بررسی عملی ویژگی های ذکر شده

پس از بررسی تئوریک قسمت های پیشین اکنون نوبت ان است تا دست به کار شوید و ویژگی های برشمرده شده را عملا بیازمایید، به همین منظور در این بخش راهنمای ساخت برنامه ای که در آن عمل کشیدن و رها کردن شی توسط موس انجام می شود را ارائه کرده ایم.توصیه می شود در حین ساخت برنامه ویژگی های ذکر شده را بررسی نمایید.در این خود آموز فرا می گیرید که چگونه می توان یک شی را توسط موس به مکانی دیگر منقل نمود یا به بیانی دیگر عمل Drag & Drop .

۱- یک شی (مثلا یک عکس یا یک مسطتیل)را در صفحه قرار دهید.ما در این تمرین ازیک مستطیل استفاده کرده ایم.

۲- کد زیر را به درون رویداد MouseDown شی منتقل نمایید:

```
i=1
X=ObjectX(Rectangle)
Y=ObjectY(Rectangle)
Xm=MouseX()
Ym=MouseY()
StartX=Xm - X
StartY=Ym - Y
RunScript("Script")
```

۳- سپس یک شی اسکریپت ایجاد نمایید و کد زیر را به درون آن منتقل نمایید:

```
If (i=1) Then
Xm=MouseX()
Ym=MouseY()
NewX=Xm - StartX
NewY=Ym - StartY
MoveObject("Rectangle1","NewX,NewY")
Refresh()
ScriptTimer("Script","0")
End
```

٤- نهایتا عبارت i=0 را به قسمت رویداد MouseUp شی مستطیل اضاف کنیدتا برنامه کامل شود.

۵- کلید F5 را برای اجرای پروژه فشار دهید.

متغیر Variable

تغییرات، تغییرات، باز هم تغییرات. این روزها تغییرات بسیار شده و در هر جنبه زندگی بسیار است. مثلاً هنگامی که شما برای خرید به فروشگاه مراجعه می کنید و تمام پول خود را در آن جا صرف می کنید متغیر در این جا کیف پول شماست چرا که از حالت پر به خالی تبدیل شده است.

برنامه شما بستگی به متغیرها دارند و مرتبا داده هایی که به متغیرها اختصاص داده شده اند در حال تغییر هستند از متغیرها به عنوان ابزاری جهت نگه داری داده ها و تغییر آنها به صورت موقت استفاده می شود. در MMB دو نوع متغیر موجود است:

- **متغیر عددی (Integer)**: که برای ذخیره اعداد استفاده می شود.این نوع متغیر برای انجام اعمال ریاضی حائز اهمیت است.

- **متغیررشته ای (String)**: متغیر رشته ای جهت ذخیره انواع کاراکترها، کلمات، جملات و پاراگرافهاست.

اما متغیرها شامل ۲ قسمت هستند:

- **برچسب متغیر**: نامی است که به متغیر اختصاص داده می شودومتغیر بعداً توسط همین نام فراخوانی می شود این نام می تواند تنها متشکل از حروف و یا ترکیبی از حروف و عدد باشد مثل Test 12,My Var و...

فراموش نکنیم که شروع نام متغیرها نباید عددباشد.به عنوان مثال نام 3th_Year به عنوان نامی غیر معتبر در نظر گرفته می شود.

- **محتوی متغیر**: که برای متغیرهای عددی همان عدد است و برای متغیر رشته ای همان حروف.

توجـه نماییـد کـه در MMB لازم نیـست کـه نـوع متغیـر از پـیش تعیـین شـود و ایـن امـر بـه صـورت خودکـار بـه وسـیله برنامه صورت می گیرد.

متغیرهای عددی (Numerical Variable)

برخی از مواقع ما احتیاج به انجـام عملیـات هـای ریاضـی داریـم و اینجاسـت کـه پـای متغیرهـای عـددی بـه میـان مـی آیـد. در MMB ۲ نوع عدد موجود است که ما از آنها جهت ذخیره در متغیرها استفاده می کنیم:

- اعداد حقیقی (اعم از اعشاری) مثل 5. 4.2 , 900.21 و $\frac{2}{3}$

- اعداد صحیح (گرد شده) مثل ٤، ١٢، ١٠٠٠

توصیه می شود که حتـی الامکـان از نامهـای دارای مفهـوم کـه نـه خیلـی کوتـاه و نـه خیلـی بلنـد هـستند اسـتفاده شـود. امـا مهمترین قسمت در زمینه متغیرها انتساب آنهاست که برای این کار بایستی ابتدا:

الف)نام متغیر را می نویسیم:

Glass

ب)سپس به دنبال آن علامت = را قرار می دهیم.

Glass =

پ) و در نهایت مقدار متغیر را پس از علامت مساوی قرار می دهیم:

Glass =1555

امـا ایـن نکتـه مهـم اسـت کـه مـا بتـوانیم مقـدار یـک متغیـر را بـه متغیـر دیگـری اختـصاص دهیم،بـرای انجـام ایـن عمـل کافیست که به ترتیب زیر عمل نماییم:

الف) نوشتن متغیر مقصد: (منظور همان است که در نهایت مقدار به آن داده می شود)

face2

ب) علامت = را در ادامه قرار می دهیم:

face2 =

پ)سپس نوشتن متغیر اصلی و منبع را بعد از گذاشتن علامت مساوی انجام می دهیم:

face2 = face1

و چنانچه مقدار متغیر face1 , 255 باشد همین مقدار متعاقباً به face2 انتقال داده می شود.از تکنیک متغیر به متغیر (Variable to Variable) اکثراً در مواقعی که احتیاج به انجام اعمال ریاضی است استفاده می شود.

حال که صحبت از اعمال ریاضی است باید گفت که می توان از چهار عمل اصلی ریاضی در اختصاص دهی و تعریف متغیرها سود جست.برای توضیح بیشتر به جدول زیر توجه کنید:

عملگر	شرح	مثال
+	این عملگر،عمل جمع را بین ۲ یا چندعدد و همچنین ۲ یا چند متغیر عددی انجام می دهد و نتیجه را به متغیر مربوطه اختصاص می دهد.	a =12+43 a = b+c Weekend=Sat+Sun Days of week = 1 + 2+3+4 +5+6+7
–	این عملگر، عمل تفریق را بین ۲ یا چند عدد و همچنین ۲ یا چند متغیر عددی صورت می دهد و نتیجه را به متغیر مربوطه اختصاص می دهد.	a =43-20 b = c – a Workday=week– weekwnd Tax = 100 -11- 65
×	این عملگر، عمل ضرب را بین ۲ یا چند عدد و همچنین بین ۲ یا چند متغیر عددی صورت می دهد و نتیجه را به متغیر مربوطه اختصاص می دهد.	a=43*12 a=c*b Month = week*4 Sales = 24.99* buyers
/	این عملگر، عمل تقسیم را بین ۲ یا چند عدد و همچنین بین ۲ یا چند متغیر عددی صورت می دهد و نتیجه را به متغیر مربوطه اختصاص می دهد.	a=43/12 a=c/b Month=Year/12 SpecWeight=Spec Quantily/30

اما با استفاده از همین چهار عمل اصلی ترکیباتی حاصل می شود که به قرار زیر است:

-عملیات ریاضی با ۲ شماره و یا بیشتر

a=432/21

```
MyNum=4+1+5
```

-عملیات ریاضی با دو متغیر یا بیشتر:

```
a = b*c
Price = Quantity + Item price – Taxes
```

-عملیات ریاضی با ترکیبی از اعداد و متغیرهای عددی

```
a= b + 42* Number of clicks
Price = 2032 * Quantity + 83/ Number of Members
```

هنگامیکه در سمت راست این انتساب ها مقدار محاسبه می شود بلافاصله به متغیر نسبت داده می شود و پس از آن در هر کجا می توانیم از آن استفاده نماییم. به عنوان مثال:

```
Price = 2032 * Quantity + 83/Number of Members
Message("Price of your order is: ", "Price")
```

در این اسکریپت ابتدا مقدار محاسبه می شود و سپس به متغیر Price نسبت داده می شود سپس توسط فرمان Message به نمایش در می آید.

متغیر رشته ای (String Variable)

متغیرهای رشته ای در همه جا هستند، حروف، کلمات، جملات، پاراگرافها و...از جمله متغیرهای رشته ای هستند. مهمترین قسمت یک متغیر رشته ای بر چسب(Label) آن می باشد. برچسب متغیر رشته ای از یک اسم و کارکتر " $ " تشکیل شده است. MyName$,Title$ از جمله برچسب های معتبر می باشند. باید همیشه به خاطر داشته باشیم که پس از برچسب متغیر رشته ای حتماً علامت $ را باید درج کنیم چرا که توسط این علامت MMB می فهمد که این متغیر رشته ای می باشد. ضمناً توصیه می شود که حتی الامکان از اسامی دارای مفهوم که نه خیلی زیاد کوچک و نه خیلی زیاد بزرگ باشد برای نامگذاری متغیرها استفاده نمایید چرا که این امر سبب ایجاد سهولت و خوانایی برنامه شما می شود.

اما عمل انتساب در MMB برای متغیرهای رشته ای چگونه صورت می گیرد؟

الف) ابتدا برچسب متغیر را می نویسیم:

Sentence

ب)پسوند $ را اضافه می کنیم:

Sentence$

پ) در ادامه علامت مساوی را قرار می دهیم:

Sentence$=

ت) و در نهایت محتوی متغیر را که بایستی بین یک جفت (') محصور شود را می نویسیم:

Sentence$ = ' This is an example '

همچنین برای افزودن Backslash(\) به صورت زیر ۲ بار بایستی Backslash را تایپ نماییم:

Path$ = ' C:\Windows\\ ' ⟶ (نتیجه می دهد) C:\Windows\

اما برای افزودن کاراکتر(‘)به محتوی رشته می توانیم از کاراکتر (\ )قبل از علامت (‘)استفاده کنیم،همانند مثال زیر:

String$='Ali\'s Book' ⟶ Ali's Book

همانند متغیرهای عددی در اینجا هم می توانیم محتوی یک متغیر را به یک متغیر دیگر مستقیماً نسبت دهیم. برای این منظور:

الف) ابتدا بر چسب متغیرمقصد را می نویسیم:

```
MySecondString$
```

ب) علامت = را در ادامه می نویسیم:

```
MySecondString$ =
```

پ) سپس نام متغیر منبع را می نویسیم:

```
MySecondString$ = MyFirstString$
```

چنانچه محتوی متغیر منبع (This is an example) باشد همین عبارت عیناً به متغیر مقصد نسبت داده می شود. جهت ادغام رشته ها و انتساب آنها به یک متغیر دیگر از عمل جمع استفاده می شود.استفاده از عمل جمع هیچ محدودیتی ندارد. به عنوان مثال اگر متغیر اول به صورت زیر باشد:

```
FirstPart$= ' to be - or not to be … '
```

و متغیر دوم به صورت زیر باشد:

```
SecondPart$ = ' The question is now '
```

برای ادغام دو متغیر و انتساب آن همانند زیر عمل می کنیم.

```
CompleteSentence$= FirstPart$ + SecondPart$
```

که نتیجه آن به صورت زیر می شود:

to be – or to be not… The question is now

یادآوری می کنیم که همین نتیجه را می توانستیم از ترکیب ۲ محتوا و سپس انتساب آنها بگیریم.

اما جدول زیر اغلب ترکیبات را در انتساب متغیرهای رشته ای بیان می کند.

مثال	شرح
Name$= ' Alex Green'	انتساب متن به متغیر:این متن انتساب یافته ثابت است .
User password$=Edit Box$	انتساب متغیر به متغیر:در این حالت مقداریکی از متغیرهای رشته ای به متغیر دیگری داده می شود.
A$=User Name$+ ' 19 Years old'	انتساب متغیر + متن (یا بر عکس) به متغیر: در این حالت ابتدا توسط عمل ادغام محتوی متغیر تشکیل می شود و در نهایت به متغیر دیگر تخصیص داده می شود.

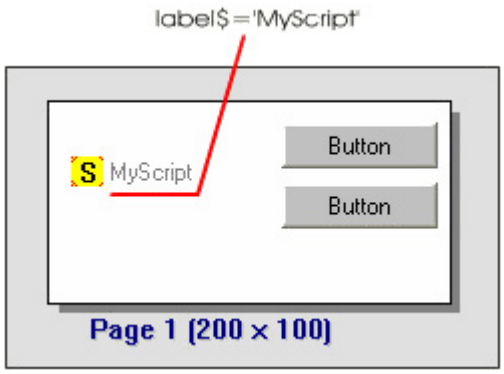
در زیر مثالی از گونه ای که در آن از ادغام استفاده شده آمده است:

```
Player1$= 'Hamed'
Player2$= 'Hadi'
MatchPlayers$ = Player1$ + 'Vs' + Player2$
Message("And now, match: ","MatchPlayers$")
```

که در نهایت عبارت And now, match: Hamed Vs Hadi در کادر پیغام نمایان می شود.

متغیرهای رشته ای به عنوان برچسب اشیاء

انعطاف پذیری MMB این امکان را فراهم می کند که بتوانیداز یک متغیر رشته ای به عنوان منبعی برای برچسب اشیاء استفاده کنید.



با این عمل در هر فرمانی که باید در آن برچسب شی را قرار داد می توان از متغیر رشته ای به جای نام ثابت شی استفاده کرد.به عنوان مثال در فرمان RunScript که یک پارامتری می باشد می توان به جای اسم ثابت شی Script از متغیر رشته ای استفاده کرد همانند مثال زیر:

```
Label$ ='MyScriptScript'
RunScript("Label$")
```

استفاده از این ویژگی در زمانی مفید است که اشیاء تقریباً نامی متحدالشکل داشته باشند مثل Text3, Text2, Text1 و... و ما بخواهیم در حلقه ها از نام اشیاء استفاده کنیم.

ثابت ها (Constants)

چیزهای زیادی وجود دارندکه ما می‌توانیم آنها را تغییر دهیم.اما با این حال تعداد زیادی از چیزها هم هستند که قادر به تغییر دادن آنها نیستیم. اجازه بدهید تا در مورد حالت دوم بحث نماییم. زمین یکبار در۲٤ ساعت به دور خودش می‌چرخدو ما قادر نیستم که آن را تغییر دهیم، سنگی را به بالا پرتاب می‌کنیم اندکی بعد به زمین می‌خوردو هیچ راه دیگری نیست. اما رایانه‌ها، همیشه از ثابت‌ها استفاده می‌کنند و بدون آنها ادامه کار میسر نیست.به عنوان مثال رایانه‌ها احتیاج به ولتاژ ثابت دارند و نبود این ثابت سبب ایجاد اختلال می‌شود. ثابت‌ها در دیدگاه نرم‌افزاری با یک نام یا برچسب شناسانده می‌شوند.

موارد استفاده از ثابت‌ها در MMB :

– برای اطلاعات سیستمی و چند رسانه‌ای گوناگون مثل اندازه صفحات، زمان فایل‌های صوتی و تصویری،مسیرهای پروژه در ویندوز،عناوین فایل‌ها و ...

– به عنوان پارامترهای یک فرمان که به MMB بگوید دقیقا چه کاری را انجام دهد، مثلا این پنجره را باز کن، آن را تغییر بده و ...

اما همان‌طور که ممکن است حدس زده باشید،بعضی از ثابت‌ها می‌توانند تغییر کنند البته غیرمستقیم. به عنوان مثال اگر شما زمان و عنوان یک فایل صوتی را تغییر دهید محتوی ثابت تغییر می‌کند و MMB خود این تغییرها را اعمال می‌نماید.

یکی ازویژگی های مهم MMB اینست که می‌تواند مقدار ثابت‌ها را دریافت و در متغیرهای رشته‌ای (String) یا عددی (Integer) ذخیره نماید و به همین خاطر دامنه‌ی فعالیت ما بر اساس اطلاعات دریافتی بیشتر می‌شود.

جهت انتقال اطلاعات از یک ثابت به یک متغیر همانند زیر عمل می‌کنیم. (ابتدا متغیر عددی)

الف) برچسب متغیر عددی را می‌نویسیم:

Current year

ب) یک علامت مساوی در ادامه قرار می‌دهیم:

Current year =

ج) و در نهایت ثابتی را که می‌خواهیم از آن اطلاعات بگیریم می‌نویسیم:

Current year = CBK- year

اما جهت انتقال اطلاعات از ثابت به متغیر رشته‌ای به صورت زیر عمل می‌کنیم.

الف) برچسب متغیر رشته‌ای را می‌نویسیم:

windows_ver$

ب) در ادامه یک علامت مساوی قرار می‌دهیم:

windows_ver$ =

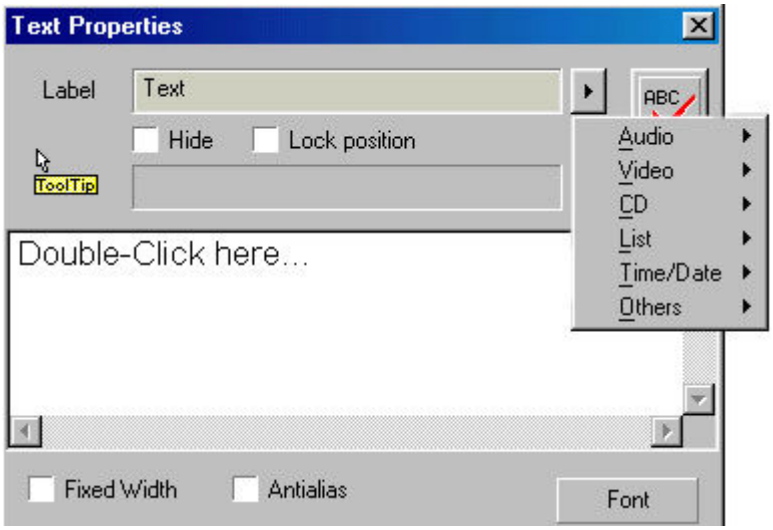
ج) و پس از آن ثابتی را که می‌خواهیم اطلاعات از آن دریافت کنیم می‌نویسیم

windows_ver$ = WinVer( )

همان طور که در ۲ حالت قبل مشاهده کردید بعضی از ثابت‌ها اطلاعات را به صورت عددی باز می‌گردانند و برخی بصورت رشته‌ای (بستگی به محتوای ثابت‌ها دارد).

ثابت‌های CBK

کمی عجیب به نظر می‌رسند، اما آنها چیزی جز ثابت‌های سیستمی، چند رسانه‌ای و...نیستند. ثابت‌های CBK جزئی از خصوصیات و امکانات متن محسوب می‌شود. بنابراین به سراغ آنها می‌رویم تا ببینیم که چه چیزهایی می‌توانیم پیدا کنیم، تغییر دهیم و یادبگیریم.



با توجه به شکل پس از کادر label شی متن پیکان کوچکی است که با فشردن آن منوی حاوی ثابت‌ها که به ٦ دسته تقسیم شده‌اند ظاهر می‌شود.پس از اینکه ثابت مورد نظر را برگزیدید،اگر برنامه را اجرا نمایید مشاهده می‌کنید که شی متن مقدارثابت را بازمی‌گرداند.

در جدول زیر شرحی از ثابت‌ها و مثالهایی برای آنها آمده است.

نام CBK (Audio CBK)	شرح CBK	مثال
CBK- Total	این ثابت طول مدت فایل صوتی بارگذاری شده را نمایش می‌هد.فرمت زمان نشان داده شده به صورت MM: SS (Seconds ثانیه) (Minutes دقیقه) است.	21: 12
CBK- TotalSec	این ثابت طول مدت زمان فایل بارگذاری شده را در واحد ثانیه نمایش می‌دهد.	360
CBK- Time	این ثابت زمان جاری فایل صوتی بارگذاری شده را نمایش می‌دهد.	11: 08
CBK-TimeSec	این ثابت زمان جاری فایل را در واحد ثانیه نمایش می‌دهد.	110
CBK-MP3Type	نوع (Type) فایل mp3 بارگذاری شده را نمایش می‌دهد.	MPEG1 layer 3
CBK- MP3Bit	این ثابت Bitrate فایل صوتی بارگذاری شده را بازمی‌گرداند. فرمت این گونه است Number rbit/ s.	128 rbps
CBK-MP3Freq	این ثابت بسامد نمونه سازی شده فایل صوتی را بازمی‌گرداند. بسامد در واحد Hz است.	44. 1 KHz
CBK-MP3Name	این ثابت نام فایل صوتی بارگذاری شده را بازمی‌گرداند، بدون پسوند و مسیر فایل.	INFINITY
CBK-1D3Song	این ثابت عنوان فایل صوتی mp3 را که از ID3 tag گرفته شده را نشان می‌دهد.	Beauty
CBK-Channels	این ثابت کانال فایل صوتی را که یکی از ۲ گونه Stereo و Mono است باز می گرداند.	Stereo
CBK-ID3Artist	این ثابت نام هنرمند و تولید کننده فایل MP3 بارگذاری شده را بر اساس ID3tag باز می گرداند.	James-Robins
CBK-ID3Album	این ثابت نام آلبوم حاوی فایل MP3 را باز می گرداند. این نام بر اساس ID3tag گرفته می شود.	The Ultimate collection
CBK-ID3Year	این ثابت بر اساس ID3tag سال تولید فایل صوتیMP3 بار گذاری شده را باز می گرداند.	2002
CBK-ID3Genre	این ثابت بر اساسID3tag طبقه (ژانر) فایل صوتیMP3 بارگذاری شده را نمایش می دهد	Soul
CBK-Numtracks	این ثابت تعداد Track های یک CD صوتی را باز می گرداند.	14

Volume	این ثابت حجم صدای اصلی (Master Volume) را در واحد درصد بیان می کند (100تا 0).	70
CBK-Curltemlist	این ثابت انتخاب جاری درSongListرا نمایش می دهد.	After All
CBK-NuminList	این ثابت تعداد قسمتهای موجود درSongList را نمایش می دهد.	31
CBK-TotalList	این ثابت مجموع کل زمان فایلهای صوتی موجود درSong List را نشان می دهد. فرمت به گونه MM:SS است.	135:28
CBK-TotalListSec	این ثابت مجموع کل زمان فایلهای صوتی موجود درSong List را در واحد ثانیه نشان می دهد.	1130
(VideoCBK) CBK-Vname	این ثابت نام فایل ویدویی در حال اجرا را بدون پسوند و مسیر فایل باز می گرداند.	The Matrix
CBK- Vtotal	این ثابت طول مدت فایل جاری ویدوئی را با فرمت MM:SSباز می گرداند.	180:19
CBK-VTotalSec	این ثابت طول مدت فایل جاری ویدوئی را در واحد ثانیه باز می گرداند.	10700
CBK-VTime	این ثابت زمان جاری فایل در حال اجرا را با فرمت MM:SS باز می گرداند.	94:32
CBK-VtimeSec	این ثابت زمان جاری فایل ویدوئی را در واحد ثانیه باز می گرداند.	48000
CBK-Vtotal Frames	این ثابت تعدادفریمهای فایل ویدوئی رابازمی گرداند.	394025
CBK-VFrame	این ثابت فریم جاری فایل ویدوئی را باز می گرداند.	249253
(Data &Time CBK) CBK – Year	سال جاری را نشان می دهد.	2004
CBK-Month	این ثابت ماه جاری از سال جاری را نشان می دهد.	July
CBK-MonthNum	این ثابت ماه جاری را بر اساس عدد نشان می دهد.(بین 1 تا 12).	7
CBK-Day	این ثابت روز جاری را نشان می دهد.	Tuesday
CBK- DayNum	این ثابت روز جاری را بر اساس عدد نشان می دهد. Wednesday=4 , Sunday=1 , Monday=2 , Tuesday=3 , Saturday=7 Thursday=5 , Friday=6 ,	3
CBK- DateNum	این ثابت روز و ماه جاری را بر اساس عدد نشان می دهد.	22
CBK- DateShort	این ثابت بر اساس فرمت DD/MM/YY تاریخ را نشان می دهد.	21/03/2004
CBK- DataLong	این ثابت بر اساس فرمت Month Day , Year تاریخ جاری را باز می گرداند.	July 23,2004
CBK-TimeHMS	این ثابت زمان جاری سیستم را بر اساس , AM/PM و H:M:S(ثانیه:دقیقه:ساعت) نشان می دهد.	01:06:12 PM

CBK-Time24	این ثابت زمان جاری سیستم را بر اساس HH:MM:SS و ٢٤ ساعت نشان می دهد.	13:07:12
CBK-Hour	ساعت جاری را که بین 0 تا 23 است را باز می گرداند.	13
CBK-Minute	دقیقه جاری که بین 0 تا 59 است را باز می گرداند.	54
CBK-Second	ثانیه جاری که بین 0 تا 59 است را باز می گرداند.	12
CBK-PageName	این ثابت نام صفحه باز جاری را باز می گرداند.	Intro Page
CBK-Error	این ثابت خطاهایی را که توسط موتور صوتی FMODیا شیء ویدوئی ارسال شده را باز می گرداند.(لیست خطاها در ضمیمه)	
CBK-URLPath	این ثابت URL جاری را از شی HTML (در صورت موجود بودن) باز می گرداند.	www.rasane.net
CBK-OpenFile	پس از استفاده از کادر محاوره ای بازگشایی فایل این ثابت نام فایل انتخاب شده را بدون مسیر آن باز می گرداند.	Test. Txt
CBK-OpenDir	پس از استفاده از کادر محاوره ای بازگشایی فایل این ثابت مسیر فایل انتخاب شده را باز می گرداند.	C:\Mydoc\
CBK-ReturnVal	فرمان Runیک مقدار بازگشتی را از برنامه اجرا شده ارسال می کند(درصورتی که کدبازگشت موجودباشد.)	1
CBK-Selcolor	این ثابت مقدار RGB را از کادر محاوره ای انتخاب رنگMMB باز می گرداند.کادر محاوره ای رنگ از سیستمRGB (R قرمز، G سبز، B آبی) استفاده می کند هرقسمت رنگ می تواند دارای سطحی بین 1 تا 255 باشد هر چه مقدار عددی بیشتر باشد برشدت رنگ افزوده می شود.مقدار بازگشتی یک آرایه رشته ای است که قسمتها توسط کاما از هم جدا می شوند (اول:قرمز،دوم:سبز،سوم: آبی).جهت دریافت مقدار رنگ کافیست به آسانی مقدار CBK-Selcolor را به یک متغیر رشته ای نسبت دهیم . CBK- Sel color=Color$	R,G,B 128,5,64
CBK-MP3EOF	چنانچه بر چسب یک شی اسکریپت را به CBK-MP3EOF تغییر دهیم و آن را یا در صفحه جاری و یا در Master Toplayer قرار دهیم بلافاصله پس از اتمام اجرا فایل صوتی MP3 آن اسکریپت اجرا خواهد شد.MMB ابتدا در صفحه جاری به دنبال اسکریپت می گردد و در صورت فقدان آن در Master TopLayer به دنبال آن می گردد.	

CBK-Menu	چنانچه بر چسب گروهی از اشیاء جاری CBK-Menu باشد این گروه هر وقت که در خارج از محدوده آن کلیک شود ناپدید خواهد شد. این ویژگی به ما در ساختن منوهای جهنده (Pop-Up) کمک خواهد نمود. هنگامی که کاربر در صفحه کلیک نماید تمام گروههایی که با این ثابت نامگذاری شده اند محو خواهند شد به جز گروهی که زیر نشانه گر موس است.	
CBK-Exit	اگر نام و بر حسب یک شی حاوی اسکریپت را به CBK-Exit تغییر دهید و آن را یا در صفحه جاری و یا در Master Toplayer قرار دهید با فشردن کلید Esc بر نامه طبق معمول بسته نمی شود. و به جای آن شی که با CBK-Exit نامگذاری شده است فراخوانی می شود.از این ویژگی می توان برای تولید کادرهایی که به کاربر پیغام آیا مطمئن هستید را می دهند،استفاده کرد.	

ثابت های سیستمی System Constans

اطلاعات سیستمی نقش مهمی را در تولید و تنظیم پروژه شما ایفا می کند.در MMB ثابتهای سیستمی اطلاعاتی راجع به صفحه (Screen)، پنجره (Window)، پردازشگر (Proccesser) و حافظه (Memory) را در اختیار ما قرار می دهد. باید توجه داشت که این ثوابت به عنوان پارامتر در سایر فرامین قابل استفاده هستند.

```
ScriptCommand("System Constant","param 2")
```

ثابت های سیستمی به شرح زیر هستند:

ScreenWidth( )

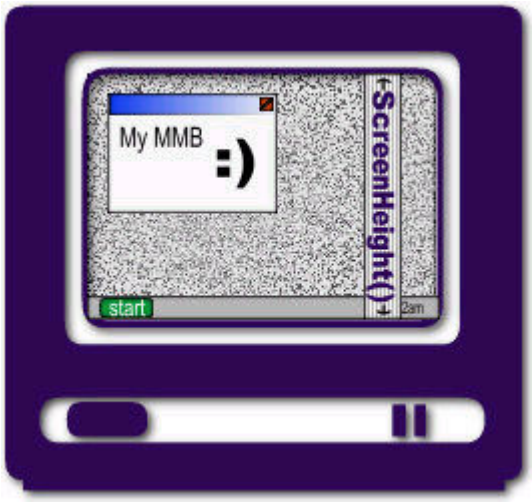
این ثابت عرض صفحه را در واحد پیکسل (Pixel) و به صورت عدد صحیح نشان می دهد.

مثال:

```
Message("Display width is:","ScreenWidth()")
```

ScreenHeight( )

این ثابت ارتفاع صفحه را در واحد پیکسل و به صورت عدد صحیح نشان می دهد.



مثال:

```
Message("Display height is:","ScreenHeight()")
```

WorkAreaWidth( )

این ثابت عرض ناحیه کار را در واحد پیکسل باز می گرداند.

مثال:

```
Message("Display work area width is:","WorkAreaWidth()")
```

WorkAreaHeight( )

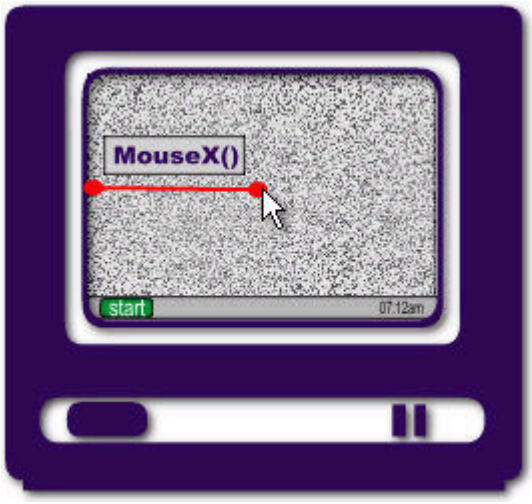
این ثابت ارتفاع ناحیه کار را به صورت عدد صحیح و در واحد پیکسل باز می گرداند.

مثال:

```
Message("Display work area height is","WorkAreaHeight()" )
```

MouseX( )

این ثابت مقدار طول محل نشانگر موس را نسبت به سمت چپ صفحه نمایش در واحد پیکسل باز می گرداند.

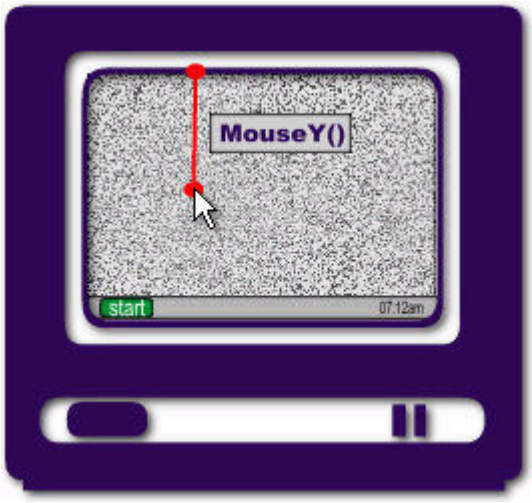


مثال:

```
Message("Current Mouse X Position :", "MouseX()")
```

MouseY( )

این ثابت مقدار عرض محل نشانگر موس را نسبت به سمت بالای صفحه نمایش در واحد پیکسل باز می گرداند.



مثال:

```
Message("Current Mouse Y Position :", "MouseY()")
```

ProcType()

این ثابت نام تولید کننده، نوع و سرعت CPU را باز می گرداند.

مثال:

```
Var$ = ProcType()
Message("Cpu in this Machine is: ","Var$")
```

ProcFreq( )

این ثابت سرعت CPU را در واحد MHZ(مگاهرتز) بیان می کند.

مثال:

```
Message("CPU Freq in MHz is:","ProcFreq()")
```

GetMemory( )

این ثابت مقدار کل حافظه آزاد را باز می گذارد. این مقدار بازگشتی همانند 512/231 است و جهت جداسازی آن می توان از فرامین MMB برای رشته ها استفاده کرد.

مثال:

```
Massage("Machine Memory status:","GetMemory()")
```

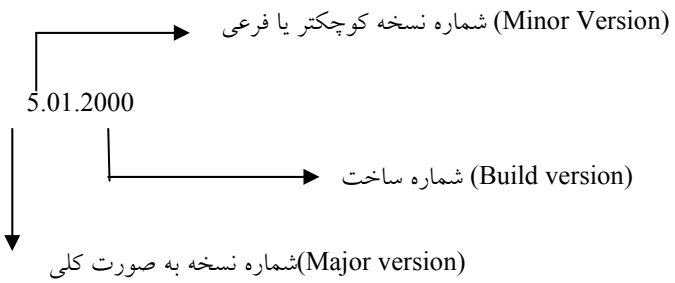
UsingWinNT( )

این ثابت ۲ مقدار 0 یا 1 را باز می گرداند. که اگر:

-1 باشد سیستم دارای ویندوز NT است.
– 0 باشد سیسم دارای ویندوز NT نیست.

WinVer( )

این ثابت نسخه سیستم عامل ویندوز را به صورت رشته باز می گرداند. این مقدار بازگشتی متشکل از چند رقم است که با حرف کاما از یکدیگر جدا شده اند چنانچه می خواهید تنها قسمتی از این مقدار را نمایان سازید می توانید از فرامین MMB برای رشته ها استفاده کنید.



مثال:

```
Var$ = WinVer()
Message("Windows Version is:","Var $")
```

جدول استاندارد نسخه های ویندوز به صورت زیر است:

Windows 95	4.00.950
Windows 95 SP1	4.00.(>950)/4.00.(<=1080)
Windows 95 OSR2	4.(<10).(>1080)
Windows 98	4.10.1998
Windows 98 SP 1	4.10.(>1998)/4.10.(<2183)
Windows 98 SE	4.10.(>=2183)
Windows Me	4.90.3000
Windows NT 3.51	3.51.1057
Windows NT 4.0	4.00.1381
Windows 2000	5.00.2195
Windows XP/SP1	5.01.2600

ScreenColors( )

این ثابت مقدار رنگهای جاری مورد استفاده در سیستم را باز می گرداند. این مقدار بازگشتی عددی صحیح است.

مثال:

```
Message("Current graphic color Mode:","ScreenColors()")
```

PubWidth( )

این ثابت عرض پنجره برنامه را بدون احتساب اندازه حاشیه و عنوان، با یک عدد صحیح بیان می کند.

مثال:

```
Message("Project window width is:","PubWidth()")
```

PubHeight()

این ثابت ارتفاع پنجره برنامه را بدون احتساب اندازه حاشیه و عنوان پنجره با یک عدد صحیح بیان می کند.

مثال:

```
Message("Project window height is:","PubHeight()")
```

ClientWidth( )

این ثابت عرض پنجره برنامه را به همراه حاشیه آن باز می گرداند می کند.

مثال:

```
Message("Project window width+broder is:","WinWidth()")
```

ClientHeight( )

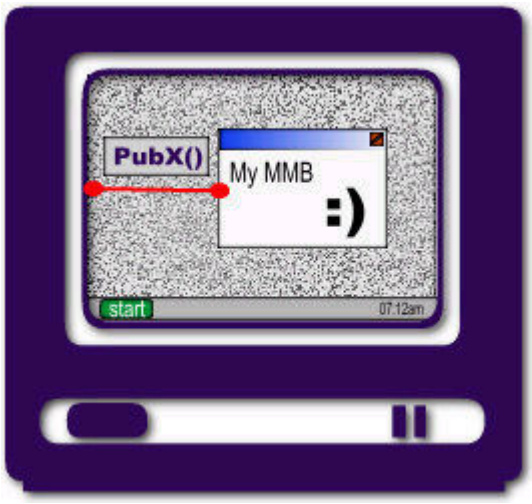
این ثابت ارتفاع پنجره برنامه را به همراه حاشیه آن و عنوان پنجره بیان می کند.

مثال:

```
Message("project window Height+boarder title is:","WinHeight()")
```

PubX( )

این ثابت مختصات طول پنجره برنامه را نسبت به سمت چپ صفحه در واحد پیکسل باز می گرداند

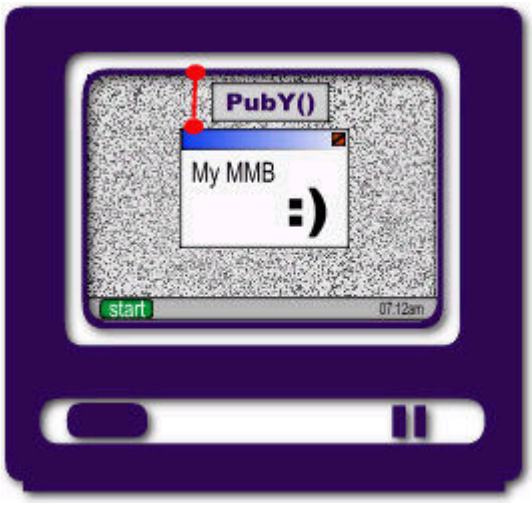


مثال:

```
Message("Project window X position :","PubX()")
```

PubY( )

این ثابت مختصات عرض (ارتفاع) پنجره را نسبت به بالای صفحه در واحد پیکسل باز می گرداند.

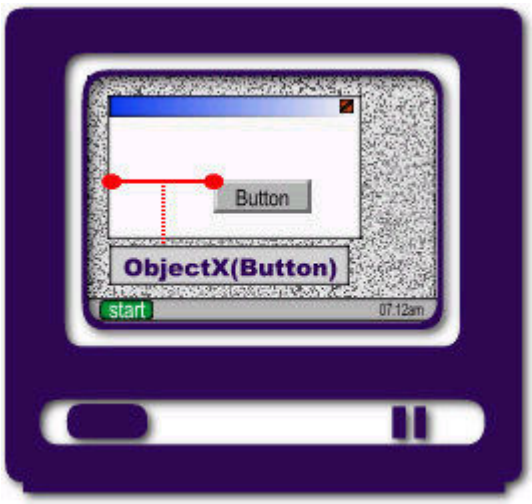


مثال:

```
Message("Project window Y position:","PubY()")
```

ObjectX(Object Lable)

این ثابت مختصات طول شی را نسبت به منتهی علیه سمت چپ پنجره برنامه باز می گرداند. جهت مشخص کردن شی کافیست نام شی را بین ۲ پرانتز قرار دهیم.



مثال:

```
X = ObjcetX(Circle)
```

```
Message(“X Value is:” , “X” )
```

ObjectY(Object Lable)

این ثابت مختصات عرض شی را نسبت به منتهی علیه سمت بالای پنجره برنامه باز می گرداند. جهت مشخص کردن شی کافیست نام شی را بین ۲ پرانتز قرار دهیم.

مثال:

```
Y= ObjectY(Circle)
Message(“Y value is:”,“Y”)
```

ObjectWidth(Object Lable)

این ثابت عرض شی مربوطه را باز می گرداند نام شی را بایستی بین ۲ پرانتز قرار دهیم.

مثال:

```
ObjectWidth = ObjectWidth(Rectangle)
Message("Object width is:","Object Width()")
```

ObjectHeight(Object Lable)

این ثابت ارتفاع (طول) شی مربوطه را باز می گرداند. نام شی را بایستی بین ۲ پرانتز قرار دهیم.

مثال:

```
ObjectHeight = ObjectHeight(Rectangle)
Message("Object Height is:","Object Height")
```

IsVisible (Object Lable)

این ثابت بر اساس حالات نمایان بودن شی اعم از قابل مشاهده بودن،غیرقابل دید و یا عدم وجود شی به ترتیب مقادیر 1,0,1– را باز می گرداند. تنها کافیست برای بررسی حالت شی نام شی را میان پرانتزها بنویسیم.

IsMinimized( )

این ثابت بررسی می کندکه آیاپنجره برنامه شماکوچک شده(Minimize) و به نوار وظیفه(Task Bar) منتقل شده است یا خیر و بر اساس آن یکی از ۲ مقدار زیر را باز می گرداند:

1- در این حالت پنجره برنامه Minimize شده است.

0- در این حالت پنجره برنامه Minimize شده نیست.

مثال:

```
Message("Project window status:","IsMinimized()")
```

MouseLButton()

این ثابت وضعیت دکمه سمت چپ موس را باز می گرداند.این ماکرو چنانچه دکمه سمت چپ در حال فشردن باشد،عدد ۱ ودر غیر اینصورت عدد صفر را باز می گرداند.

MouseRButton()

این ثابت وضعیت دکمه سمت راست موس را باز می گرداند.این ماکرو چنانچه دکمه سمت راست در حال فشردن باشد،عدد ۱ ودر غیر اینصورت عدد صفر را باز می گرداند.

MouseMButton()

این ثابت وضعیت دکمه وسط موس را باز می گرداند.این ماکرو چنانچه دکمه وسط در حال فشردن باشد،عدد ۱ ودر غیر اینصورت عدد صفر را باز می گرداند.

مثال:

```
Mstate = MouseRButton()
If ( Mstate = 1) Then
Run("<Embedded>\App.exe")
End
```

CurrentObject()

این ثابت نام شی را که از درون آن فراخوانی می شود باز می گرداند.

مثال:

```
obj$=CurrentObject()
```

ماکروهای مسیر (Path Macros)

در MMB دو گونه مسیر یکی ثابت(Fixed) و دیگری پویا(Dynamic) موجود است.

- **مسیر ثابت (Fixed Path)** : که به ترتیب از چپ به راست به حالت زیر است:

مسیر فایل (نام فایل) که ثابت است + پوشه، پوشه ها + نام درایو

Drive:\ Folder(s) \ *.*

به عنوان مثال:

C:\MyDoc\App.exe

به خاطر اینکه مسیرها در این نوع ثابت هستند،کاربر نمی تواند به دلخواه خود مسیر را تغییر دهد و این روزها این روش تقریبا قدیمی و غیر حرفه ای شده است.

- **مسیر پویا (Dynamic Path)**: در این حالت ماکروهای MMB به کمک ما می آیند تا بتوانیم مسیر دهی را به صورت پویا انجام دهیم.ماکروهای مسیر در MMB مقدارهای رشته ای هستند که ما از آنها به عنوان پارامترهای فرمان استفاده می کنیم.

مثال:

```
ScriptCommand("Param 1","<Project Folder>\MyApp.exe")
```

در فرمان بالا از ماکروی <Project Folder> جهت دریافت پوشه ای که برنامه در آن واقع است استفاده شده است.پس درماکرو مسیر نام کامل فایل را می نویسیم (در این حالت My App.exe)گذشته از ماکروهای MMB تعدادی ماکرو در برخی از Plug-in های MMB هم وجود دارند که جهت فراگیری آنهابایستی به راهنمای آن Plug-In مراجعه کرد.

اما حالا اجازه بدهید تا به بررسی ماکروهای مسیر خود MMB بپردازیم:

<SrcDir>

تقریبا پر استفاده ترین ماکرو مسیراست که پوشه برنامه (که در حال اجراست) را نشان می دهد در حالیکه معمولا سایر فایلهای مورد نیاز پروژه ما در این پوشه است ماکرو <SrcDir> بیشتر مورد استفاده قرار می گیرد.

اگر مسیر کامل برنامه ما به صورت زیر باشد:

C:\Program File\My Project Folder\MyApp.exe

سپس با استفاده از ماکروی مسیر به صورت زیر در می آید:

<SrcDir>\MyApp.exe

همان طور که واضح است این گونه مسیردهی ساده تر و کاراتر است و به جای مسیردهی ثابت می توانیم با استفاده از ماکروها مسیردهی پویا را جهت افزایش کارایی به خدمت بگیریم.اما باید توجه کنیم که هنگامی که MMB ماکروها را ترجمه کرد در نهایت مسیر به صورت ثابت در می آید. به عنوان مثال در مسیر زیر:

که به صورت مقابل ترجمه می شود

Run (“<SrcDir>\MyApp.exe”) ⟶ C:\My App\MyApp.exe

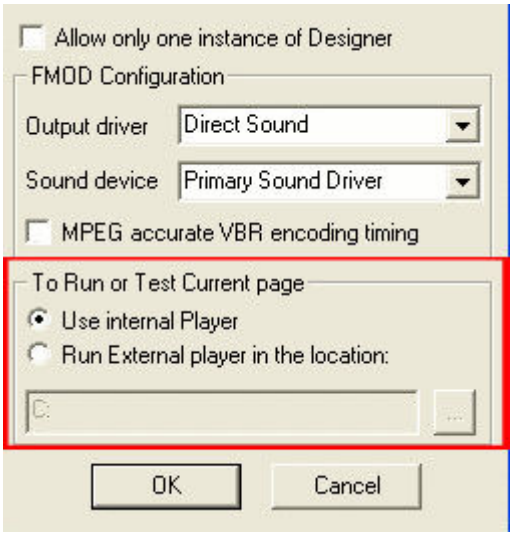
و بر اساس آن برنامه مورد نظر اجرا می شود.

توجه: MMB فایلهای پروژه را به دو گونه مختلف اجرا می کند:

Design Mode: این انتخاب در زمان طراحی پروژه صورت می گیرد و چنانچه با استفاده از گزینه Run برنامه را اجرا نمایید ماکرو (<SrcDir>) پوشه نصب برنامه MMB می شود چرا که اجرا کننده(Player) در این پوشه است. این گونه هنگامی که گزینهUse Internal Player از منوی Designer setting ⟵ Tools انتخاب شده باشد فعال است.

RunTime Mode: این گزینه برنامه کامپایل شده را به عنوان یک فایل اجرایی مستقل هنگامی که گزینه Run External player in the Location از منوی Designer Setting ⟵ Tools انتخاب شده باشد در نظر می گیرد و <SrcDir> همان پوشه فایل اجرایی می شود.

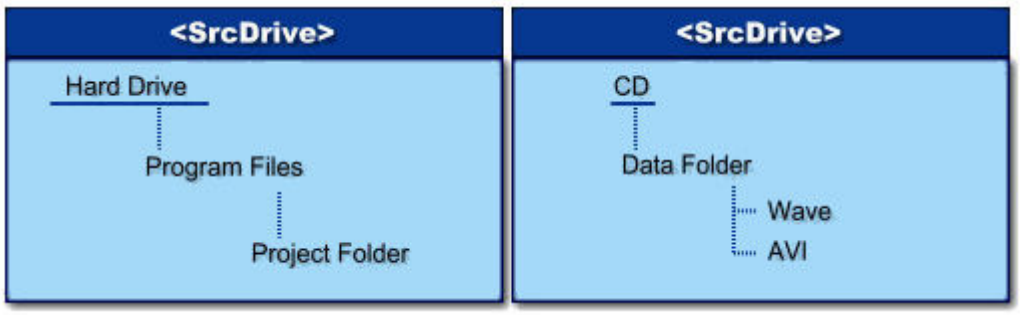
و اما اجرا کننده خارجی(External player) چیست؟



فایل اجرایی (exe) است که در پوشه ای قرار داده شده است و به عنوان یک اجرا کننده مستقل عمل می کند. هنگامیکه شما پس از انتخاب گزینه Run External player in the location در زیر این گزینه مسیر را معین نمایید ماکرو <SrcDir> در این صورت همان مسیری است که شما تعیین کرده اید.و در نهایت باید سایر فایل های پروژه در این پوشه قرار گیرند.

<SrcDrive>

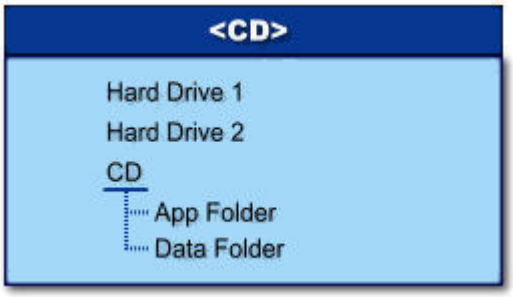
این ماکرو درایو منبع فایل اجرایی برنامه را نشان می دهد.



به عنوان مثال اگر برنامه در درایو C باشد این ماکرو C:\ را باز می گرداند.

<CD>

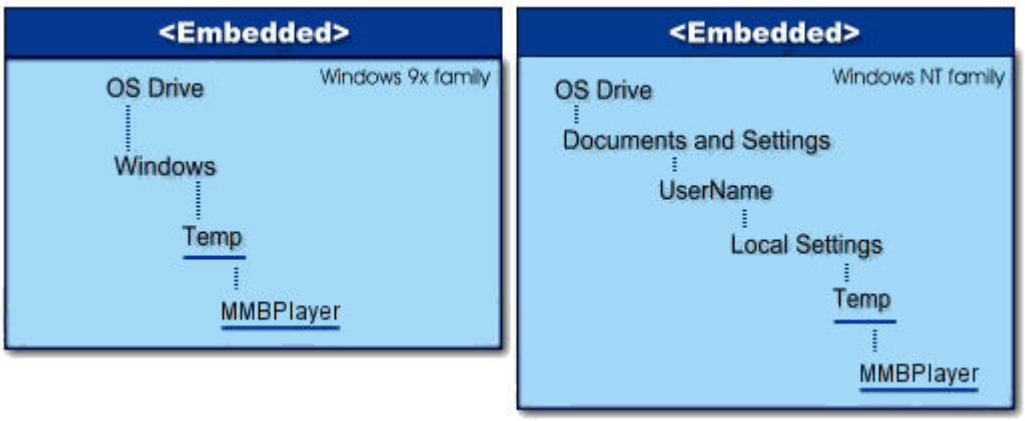
این ماکرو اولین درایو CD را باز می گرداند.به عنوان مثال اگر نام اولین درایو CD, E: باشد،ماکرو <CD>,



E:\ را باز می گرداند.استفاده از این ماکرو در مواقعی کارآمد است که بخواهیم برنامه هایمان را از روی درایو CD یا DVD اجرا کنیم اما باید توجه داشت که این ماکرو تعداد درایوهای CD را حساب نمی کند تنها نام اولین درایو CD موجود را باز می گرداند.برای بدست آوردن نام و برچسب تمام درایوها می توان از PlugIn ها ی ارائه شده استفاده نمود.

<Embedded>

این ماکرو پوشه ای را که برنامه شما جهت فایل های جاسازی شده (Embedded) استفاده می کند باز می گرداند.



به عنوان مثال اگر فایل صوتی زیر را به صورت جاسازی شده داشته باشیم.

Music.Wav

مسیر آن با استفاده از ماکرو به صورت زیر است:

< Embedded >\Music.wav

و هنگامی که شما برنامه را اجرا نمایید این ماکرو به صورت:

(در ویندوزهای XP، ۲۰۰۰ و NT)

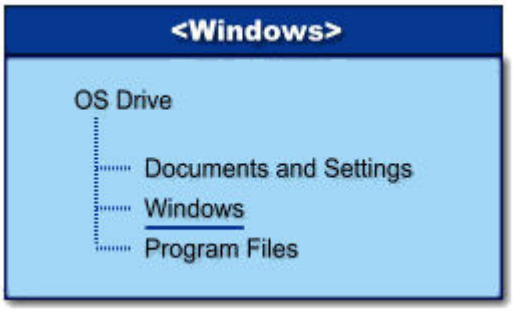
C:\Documant And setting\User Name\Local Setting\Temp\Music.wav

(در ویندوز های 9x)

C:\Windows\Temp\Music.wav

در می آید.

<Windows>



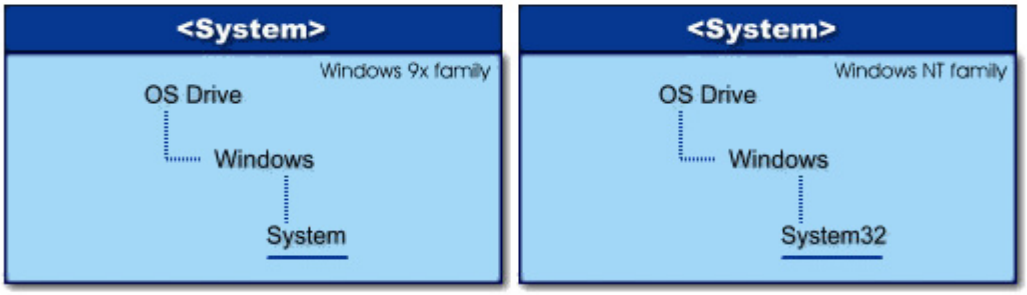
این ماکرو پوشه ویندوز را باز می گرداند.به عنوان مثال چنانچه Windows در درایو C نصب شده باشد و بخواهیم فایل Win.ini را پیدا کنیم به صورت زیر عمل می کنیم:

<Windows>\Win.ini

و در زمـان اجـرا MMB مـسیر بـالا را بـه صـورت C:\Windows\Win.ini ترجمـه خواهـد کـرد.معمـولاً از ایـن ماکروهـا جهت فراخوانی فایل های INI و سیستمی استفاده می شود.

<System>

این ماکرو پوشه سیستم ویندوز را باز می گرداند.

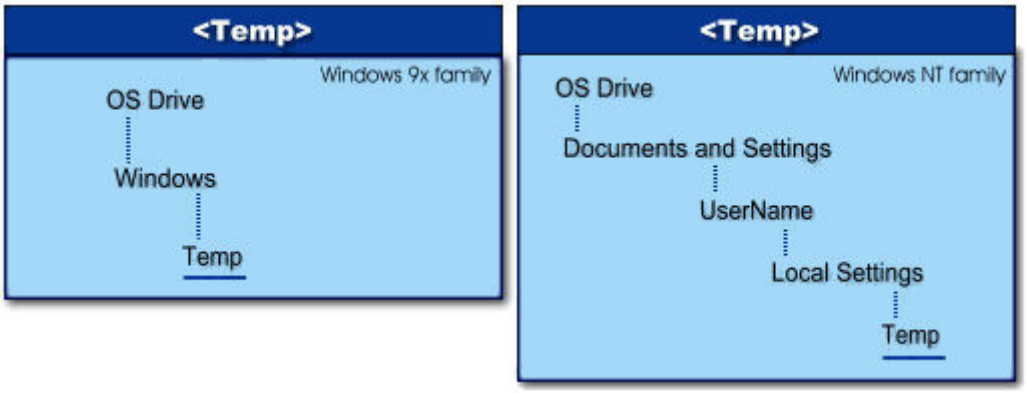


بـه عنـوان مثـال مـایلیم کـه برنامـه dxdiag.exe واقـع در پوشـه System در درایـو :C را اجـرا کنـیم. مـسیر دهـی را بـه صورت زیر انجام می دهیم:

<System>\dxdiag.exe

کـــــــــه در نهایـــــــــت MMB آن را بـــــــــه C:\windows\system\dxdiag.exe در وینـــــــــــدوزهای 9x و C:\windows\system32\dxdiag.exe در ویندوزهای NT ترجمه می کند.

<Temp>



این ماکرو پوشه ای از ویندوز را که برای قرار دادن فایل های موقتی است را باز می گرداند. به عنوان مثال مسیر دهی مقابل<Temp>\TempList.txt در نهایت به صورتهای زیر

در ویندوز 9x :

C:\Windows\Temp\TempList.txt

و در ویندوز NT :

C:\Documents and Settings\User\Local Settings\Temp\TempList.txt

در می آیند.

<File>

این ماکرو نام و مسیر فایل منتخب را که توسط کادر انتخاب فایل برگزیده شده را باز می گرداند.چنانچه فایل بازگشایی شده برای مثال C:\test.txt باشد توسط ماکرو <File> می توان همین مسیر را دریافت کرد. لازم به تذکر است در هر بازگشایی فایل محتوی این ماکرو متناسب با آن فایل تغییر می کند.پس برای ذخیره مسیرها بایستی از چندین متغیر رشته ای یا از آرایه های رشته ای سود جست.

<List>

این ماکرو جهت نگهداری محتوی Internal SongList و شی ListBox می باشد.بنابراین به آسانی می توان محتوی این دو شی را به یکدیگرویا به شی دیگری انتقال داد.

<This>

این ماکرو،یک ماکروی مخصوص فرامین[1] MCIاست که به Deviceاعلام می کند پنجره MMBپنجره پدر(اصلی) می باشد.

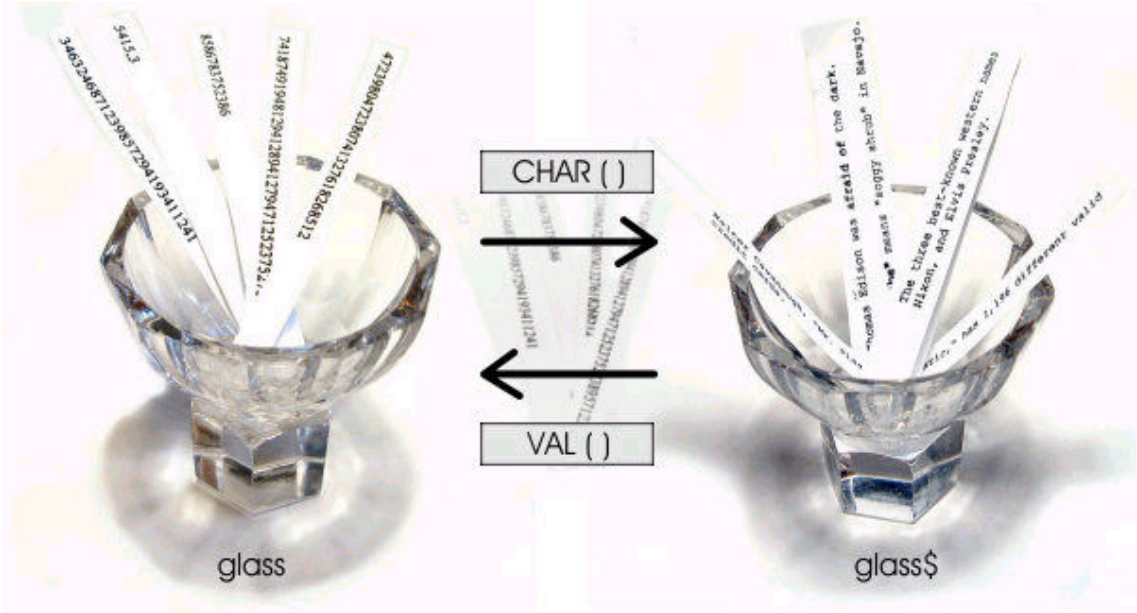
مثال:

```
MCICommand("open <SrcDir>\sample.mpg alias MPEG style child parent <This>")
MCICommand("put MPEG window at 100 50 200 200")
MCICommand("window MPEG state hide")
MCICommand("play MPEG")
```

۱- فرامینی هستند که عملیات های مربوط به صوت و تصویر توسط آنها صورت می گیرد.

« توابع وابسته به متغیرها »

همان طور که مـا در زنـدگی روزمـره اقـدام بـه ترجمـه یـک مـتن از زبـانی بـه زبـان دیگـر مـی کنـیم در MMB متغیرهـا از حالتی به حالت دیگر ترجمه می شوند. استفاده از توابع وابسته به متغیرها سبب تغییر گونه آنها می شود.



جهت رسیدن بـه ایـن هـدف (تبـدیل و ترجمـه متغیرهـا) بـه عمـل ترجمـه نیازمنـدیم. در MMB مـا عمـل ترجمـه را بـین متغیرهای عددی و رشته ای انجام می دهیم.

تابع CHAR

این تابع متغیر عددی را بـه متغیـر رشـته ای تبـدیل مـی نمایـد جهـت ایـن عمـل بایـستی نـام متغیـر عـددی را بـین ۲ پرانتـز قرار داد.

```
CHAR(Numeric Variable Label)
```

مثال:

```
a=12
b$=CHAR(a)
```

تابع VAL

برای تبدیل محتوی متغیر رشته ای به عددی از این تابع استفاده می کنیم.این تابع هنگامی مفید است که نیاز باشد بر روی محتوی متغیرهای رشته ای اعمال ریاضی صورت گیرد.

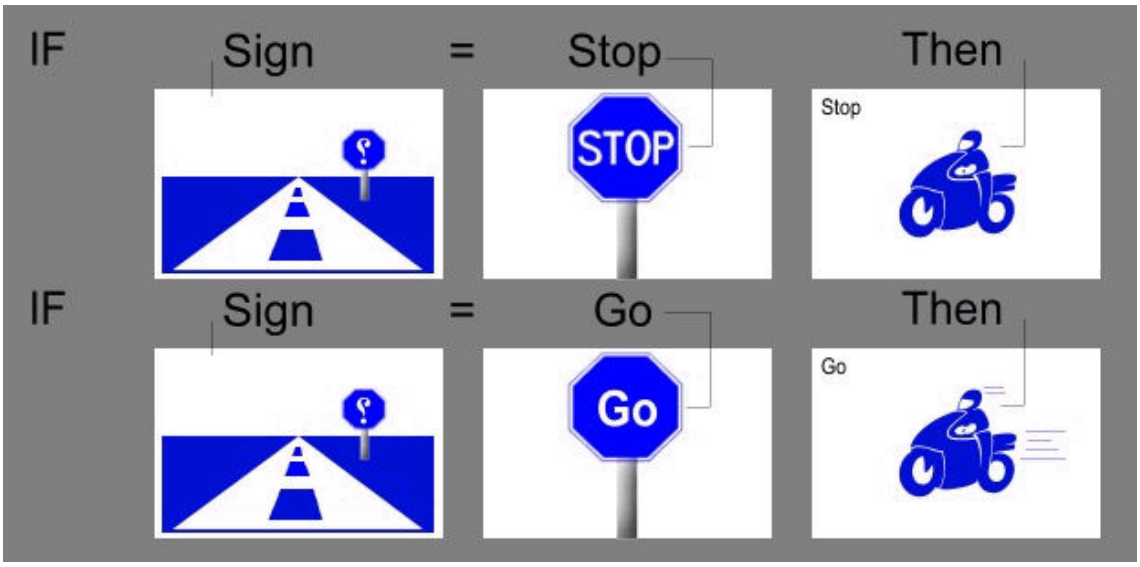
```
VAL(String Variable$)
```

مثال:

```
a$='152'
b=VAL(a$)
```

عبارت شرطی (If statment)

تصمیم گـرفتن مطلبـی اسـت کـه عبـارت شـرطی I f راجـع بـه آن صـحبت مـی کنـد. توسـط علائـم راهنمـایی و رانـدگی شما توقف می کنید، آهسته می رانید ویا دور می زنیـد. همیـشه بـه هنگـام اجـرای برنامـه شـرایطی پـیش مـی آیـد کـه بایـد بر اساس آن شرایط تصمیمی گرفته شود.عمل تصمیم گیری در MMB توسط عبارت شرطی Ifصورت می گیرید.



بر اساس تابلوها در شکل بالا موتور سوار یا حرکت می کند و یا می ایستد .

عبارت شرطی If شامل قسمت های زیر است:

الف) If

ب)پرانتزی که حاوی شرط است.

پ) علامت مساوی داخل پرانتز.

ت) کلمه Then

ث) فرمان بعدی که بر اساس شرط اجرا می شود.

ج) کلمه End جهت بستن عبارت شرطی

مثال:

```
If (EditBox$='2')Then
 Message("Number is: ","EditBox$")
End
```

چنانچه در مثال بالا شرط برقرار باشد MMB فرمان Message را اجرا نموده و چنانچه شرط برقرار نباشد MMB از اجرای دستور شرط صرف نظر کرده و به سراغ بقیه قسمتها می رود.

اما همان طور که مشاهده کردید در حالت بالا شرط برقرار بود اما حال اگر شرط برقرار نبود چه کنیم؟ پاسخ ساده است،می توانیم حالت مخالف شرط اصلی را به این قسمت توسط کلمه Else اضاف کنیم. به مثال زیر توجه کنید:

```
If(a$='2')Then
 Message("Number is: ","a$")
Else
 a$='3'
 Message("Number is: ","a$")
End
```

در این حالت چنانچه شرط اصلی که همان a$ = '2' است فراهم نباشد MMB از انجام دستور اولی صرف نظر کرده و قسمت پس از E lse را اجرا می نماید. البته براساس گفته های بالا اگر احتیاج به استفاده از Else ندارید هیچ ضرورتی ندارد که آن را بنویسید.

دستورات if چندتایی

با استفاده از چند دستور If می توان تعداد شروط بیشتری تولید و در نتیجه تصمیمات بیشتری را اتخاذ کرد.

مثال:

```
If(a=2)Then
  If(b=3)Then
     Message("You guesed right!","")
    End
Else
  a=4
  Message("That is wrong,"")
End
```

استفاده از محدوده ی متغیرها در عبارت شرطی I f

استفاده تنها از علامت مساوی درهمه جا کارآمد نمی باشد چرا که گاهی اوقات لازم است تا اعمال تصمیم بر اساس محدوده ی از مقدارها صورت گیرد. به عنوان مثال می خواهیم شرطی داشته باشیم که هر کس سنش زیر ۲۰ سال است بتواند وارد شود. خب در این حالت منطقی و به صرفه نیست که اقدام به نوشتن ۲۰ عبارت شرطی کنیم.

بلکه به جای آن می توانیم تنها بـا یـک دسـتور و بـا مـشخص کـردن محـدوده، منظورمـان را بـه MMB بفهمـانیم. بـه خـاطر این مثال و هزاران مثال مشابه است که حالتهای دیگری پدید می آید که همگی در جدول زیر آمده است:

نام	عملگر
مساوی	=
مخالف	<>
کوچکتر از	<
بزرگتر از	>
کوچکتر یا مساوی	<=
بزرگتر یا مساوی	>=

به عنوان مثال:

```
If(Age<=20)Then
  Message("You can enter","")
End
```

که چنانچه سن کوچکتر یا مساوی ۲۰ باشد پیغامی مبنی بر اجازه ورود ظاهر می شود.

```
If (Age > 20)Then
  Message("You are not allowed")
End
```

که در این حالت پیغامی مبنی برعدم اجازه ورود ظاهر می شود.

چند حالت در یک عبارت شرطی I f

یکی از قابلیت های کارآمد عبـارت شـرطی If امکـان اسـتفاده از چنـد وضـعیت در یـک خـط مـی باشـد بـر همـین اسـاس ۲ عملگر "& " (و) ، " | " (یـا) موجـود اسـت.شـیوه ی اسـتفاده بـدین صـورت اسـت کـه ابتـدا عبـارت If را نوشـته پرانتـز را باز می کنیم سپس شرط اول را قـرار مـی دهـیم بـا توجـه بـه نیازمـان یـا عملگـر "& " و یـا " | " را قـرار مـی دهـیم سـپس پرانتز را بسته و کلمه then را می آوریـم و در ادامـه بـا توجـه بـه شـرط مـی تـوانیم فرمـان مـورد نظـر را قـرار دهـیم. حـال به مثال زیر توجه نمایید.

```
If (Age=10| Age=20)Then
  Message("You can go")
End

If (Age>20 & Job='Student') Then
  Message("You can enter!")
End
```

بر اساس مثال بالا چنانچه سن کاربر یا ۱۰باشد و یا ۲۰ کاربر مجاز به رفتن می شود.و در مثال دوم چنانچه هر دو شرط بر قرار باشد پیغام خطا ظاهر می شود.لازم به ذکر است که در این حالت می توانیم بیش از دو متغیر را مورد آزمایش قرار دهیم.

```
If (Age>20 & Job="Student") Then
  Message("You can enter")
End
```

متغیر در مقابل متغیر

در MMB این امکان فراهم است تا شرطی را داشته باشیم که MMB در آن مستقیماً به امتحان و بررسی متغیرها بپردازد. همانند مثال زیر:

```
If (User1$ = User2$) then
  Message("User are identical !" ,"")
End
```

بر اساس شرط بالا چنانچه محتوی ۲ متغیر یکسان باشد پیغام ظاهر می شود. همین حالت را می توانیم برای متغیرهای عددی داشته باشیم.

عبارت های شرطی if با متغیرهای آمیخته شده

همان طور که ممکن است حدس زده باشید این امکان فراهم است که در قسمت شرط از۲ نوع متغیر استفاده کنیم برای درک بهتر به مثال زیر دقت نمایید.

```
If (User Name $= 'Hadi' & User Height > 170 ) then
  Message("Hadi" is higher than 170 cm","")
End
```

در کد بالا ابتدا نام شرط بررسی می شود، سپس شرط قد کاربر، چنانچه هر دو شرط موجود باشد پیغام ظاهر می شود.

استفاده ازعملگر < >

گاهی اوقات پیش می آید که نمی خواهیم محدوده ای را بررسی نماییم و فقط یک مقدار مورد نظر ماست که تنها بر اساس آن مقدار شرط حالت درست پیدا می کند به همین خاطر می توانیم از علامت "< >" استفاده کنیم به مثال زیر دقت کنید:

```
If (Number < > 1382 ) then
  Message("Entered number is wrong !","")
End
```

که چنانچه مقدار عددی 1382 نباشد برنامه پیغام خطا را نمایش می دهد.

استفاده از if تو در تو

گاهی اوقات امکان دارد که بخواهیم شرط، بر اساس شرطی دیگر بررسی شود. در این حالت است که از ساختار تودرتو بهره می جوییم. البته باید توجه داشت که بیشتر اوقات برای پرهیز از این حالت می توان از حالت ترکیبی که قبلاً اشاره شد استفاده کرد.

مثال:

```
If (UserName $= 'Hadi' )then
   If (UserHeight > 170 ) then
       Message ("Hadi is Higher than 170 cm ","")
   End
End
```

همچنین در این حالت می توان قبل از شرط های تودرتوی بعدی فرمانی را قرار داد. از جمله مواردی که احتیاج به نوشتن دارد کلمه E nd برای هر دستور If است. باید دقت داشت که همان تعداد که دستور If داریم باید به همان تعداد هم End داشته باشیم. البته محل نوشتن این End ها هم مهم است چرا که نوشتن اشتباه آن ممکن است ۲ شرط را با هم ادغام کند.

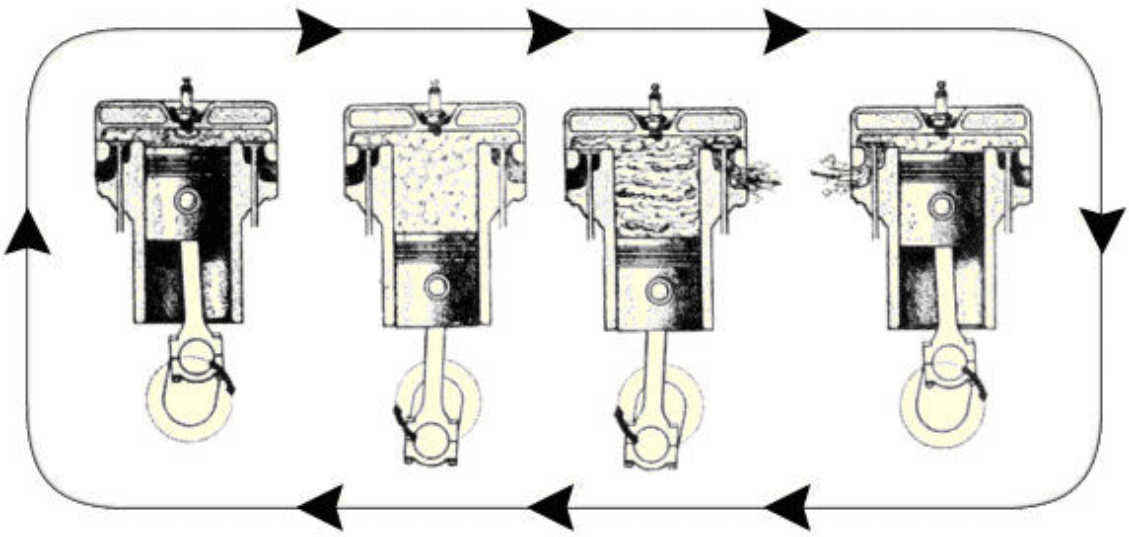
```
If (UserName$='Hadi') Then
   Message("Hello Hadi !","")
     If (UserHeight>170) Then
```

```
        Message("Hadi is higher than 170 cm","")
          If (UserWeight>70) Then
              Message("Hadi weighs more than 70 kilos","")
              Return()
          End
         Message("Hadi weighs less than/or 70 kilos","")
    End
End
```

همان طور که در بالا مشاهده می کنید این کد از ۳ دستور If تشکیل شده است که هر کدام با تورفتگی مشخص شده اند.

حلقه های For…next

راه رفتن، چرخش به دور میدان، نفس کشیدن و... این ها همه نشانی از وجود حلقه ها در زندگی روزمره می باشد. که همگی باید توسط ما انجام شود که برخی خودآگاه است و برخی ناخودآگاه به واسطه همین تکرارها بود که ماشین وارد زندگی انسان شد چرا که آنها احساس ندارند و از انجام کارهای تکراری خسته نمی شوند. کمی پیشرفته تر به کامپیوتر ها می رسیم که روزانه کارهای متعددی را که همگی آنها دارای تکرار هستند انجام می دهند. در MMB همانند سایر محسط های برنامه نویسی ابزار حلقه برای انجام کارهای تکراری ارائه شده است.



عملیات تکرار در MMB توسط عبارت For…Next انجام می شود که در ادامه به شرح ان می پردازیم.

اما قاعده استفاده از حلقه For…nextبه صورت زیر است:

الف) ابتدا کلمه For را می نویسیم:

For

ب) سپس متغیر عددی را پس از کلمه For می نویسیم .از این متغیر به عنوان شمارشگر(Counter) استفاده می شود. پس از آن علامت مساوی را پس از متغیر می آوریم:

For Counter =

پ)در ادامه عددی را که می خواهیم حلقه با آن شروع شود می نویسیم، مثلاً 1

ت) پس از آن کلمه to را نوشته و در نهایت محدوده حلقه را با نوشتن عدد مورد نظر مشخص می کنیم.

For Counter = 1 to 10

ث) سپس فرامینی را که می خواهیم مکرراً اجرا شوند می نویسیم.

ج) و در آخر هم کلمه Next را به همراه نام متغیر عددی را که قبلاً به عنوان شمارشگر مشخص کردیم می نویسیم.

Next Counter

حال به مثال زیر توجه کنید:

```
For Counter = 1 to 100
     Commands
Next Counter
```

در حلقه مثال بالا فرمان ها به تعداد 100 بار اجرا می شود که می توان این محدوده 100 بار را با تغییر حد بالا و پایین تغییر داد. هنگامی که برای مثال بار 50 ام از 100 بار است مقدار متغیر ما (شمارشگر) همان 50 می باشد که نهایتاً تا 100 ادامه می یابد. از این حالت می توان به صورت پویا در فرامین استفاده کرد. بدین معنی که از مقدار جاری متغیر شمارشگرمی توان در برنامه و در فرامین استفاده کرد. نباید فراموش کنیم که می توانیم محدوده را طوری تعیین کنیم که سیر نزولی داشته باشد، یعنی بر خلاف مثال بالا که از 1 تا 100 ادامه دارد از 100 تا 1 باشد. پس از اینکه حلقه به اتمام رسید MMB بلافاصله فرمانهای پس از حلقه را اجرا می کند. ما در مثال بالا حد

پایین و بالا را عدد 1 قرار دادیم که الزاماً این طور نیست یعنی می توانستیم حد پایین را از 1 یا از 100 شروع کنیم،تعیین این مقدار بستگی به نیازمان در هنگام کدنویسی دارد. در زیر مثالی برای حالتی که می توانیم از محتوی متغیر شمارشگر استفاده کنیم آمده است:

```
For Runer = 1 to 10
   Var$ = 'You are runner no. ' + CHAR(Runer)
   Message("","Var$")
Next Runer
```

در مثال بیان شده با استفاده از مقدار جاری متغیر شمارشگر شماره دونده را حساب کردیم.

ابزارهای حلقه

به طور قطع مواقعی پیش می آید که بخواهیم حلقه محدوده مشخصی نداشته باشد یعنی تا بینهایت ادامه یابد، اینجاست که صحبت از حلقه ی بینهایت (Infinity Loop)به میان می آید.

```
For Check = 1 to Infinity
  Check = Check + 1
  DisplayValue("Text ","Check")
Next Check
```

بلوک فرمان بالا تا هنگامی که برنامه ما در حال اجراست و یا کامپیوتر خاموش نشده باشد به روند خود ادامه می دهد. آیا باید از بابت اجرا نشدن بقیه کدها به هنگام اجرای حلقه ی بینهایت نگران بود؟ جواب خیر است. فرامینی پیش بینی شده اند که اجرای سایر قسمتها را امکان پذیر می سازند.

Refresh ( ) -

این فرمان زمانی را در اختیار سایر قسمتها می کذارد تا توسط برنامه به کار گرفته شوند.به محض اینکه ما در حلقه اسکرپیت خود از فرمان ( ) Refresh استفاده کنیم وارد یک پروسه نیمه موازی شده ایم، این بدین معنا است که هنگامی که حلقه در حال اجرا است و ادامه دارد، می توانیم با شی های دیکر تعامل داشته باشیم و یا اسکرپیت های کوچک دیکری را به اجرا درآوریم.

مثال(حلقه معمولی):

```
For n=0 to 10000
 a=n+1
Next n
```

حلقه معمولی باعث توقف MMB در حین انجام عملیات حلقه می شود. لذا شما قادر به کلیک کردن بر روی هیچ دکمه ای در حین عملیات حلقه نخواهید بود.

مثال(حلقه موازی):

```
for n=0 to 10000
 a=n+1
 Refresh ( )
Next n
```

فرمان ( ) Refresh داخل حلقه سبب ایجاد تعامل های موازی در حین عملیات حلقه می شود.این بدان معنا است که هنگامی که حلقه هنوز در حال اجرا است قادر خواهیم بود بر روی هر دکمه ای جهت اجرای اسکریپتی دیگر کلیک نماییم.باید به خاطر داشته باشید که هر حلقه دیگر اعم از موازی یا غیر موازی سبب توقف حلقه اول تا زمان اتمام خودش (حلقه ثانویه) خواهد شد.

به نمودار زیر توجه کنید:

تنها در صورتی که مطمئن هستید که چه کاری توسط حلقه موازی انجام می شود اقدام به استفاده از آن نمایید.

مثالی جهت استفاده از حلقه بینهایت:

با استفاده از حلقه موازی می توانیم حلقه بینهایت تولید کنیم (اما باید به خاطر داشت که راه خروجی را بایدتعبیه کنیم).مثال زیر سبب چسبیدن دایره ای به نشانه گر موس تا زمانی که هیچ شی دیگری سبب مقدار دهی متغیر stop به ۱ شود،می شود.

```
Stop = 0
For n=0 to Infinity
MoveObject(" Circle","MouseX( ),MouseY( )")
If (Stop = 1) then
   Return ( )
End
   Refresh ( )
Next n
```

Pause ("Time") –

این فرمان سبب ایجاد وقفه در بلوک کد برنامه بر اساس مقدار ورودی می شود واحد این زمان وقفه میلی ثانیه (۰/۰۰۱ ثانیه) است. در زیر مثالی برای ( )Refresh آمده است. این مثال جهت مقایسه می باشد:

```
For Check = 1 to Infinity
 Check = Check +1
 DisplayValue("Text ","Check" )
 Refresh( )
Next Check
```

هنگامی که برنامه در حلقه به ( )Refresh می رسد به قسمتهای دیگر قبل از شروع مجدد حلقه اجازه فعالیت می دهد. حال به مثالی برای (" ") Puase توجه کنید:

```
For Check = 1 to Infinity
 Check = Check +1
 DisplayValue("Text ","Check" )
 Puase("500")
Next Check
```

در این حالت برنامه به هنگام رسیدن به فرمان ۵۰۰ میلی ثانیه یا نیم ثانیه توقف می کند و سپس روند تکرار حلقه از سر گرفته می شود.

Return ( )–

یک ابزار کارآمد دیگر( ) Return است و هنگامی مورد استفاده قرار می گیرد که در حلقه بینهایت شرطی به وقوع بپیوندد،در اینجاست که حلقه باید متوقف شود. برای فهم بیشتر این مسئله به مثال زیر توجه کنید:

```
For Check = 1 to infinity
 Check = Check +1
 DisplayValue ("Text ","Check" )
 If (User Input $ = 'Go' ) Then
  Return( )
 End
Puase(" 500")
Next Check
```

مثال بالا آنقدر هوشمند می باشد که به محض برقراری شرایط حلقه را متوقف کند. چنانچه بخواهیم فرمان یا فرمانهایی پیش از محقق شدن شراط اجرا شود می توانیم در بلوک شرط و قبل از Return فرامین را قرار دهیم.توجه نمایید که MMB به محض رسیدن به ()Return اجرای اسکریژت را متوقف کرده و از اجرای باقی فرامین ژس از ()Return صرف نظر می کند و به ابتدای کد باز می گردد.

Break()-

ممکن است حالت پیش بیاید کـه تنهـا مایـل باشـیم اجـرای حلقـه متوقـف شـود و سـایر فـرامین ژس از حلقـه متوالیـا اجـرا شـوند. در ایـن حالـت از فرمـان ()Break اسـتفاده مـی کنـیم. MMB بـه محـض رسـیدن بـه ()Break اجـرای حلقـه را متوقف می کند و بلافاصله اولین خط پس از کلمه کلیدی Next را اجرا می کند.

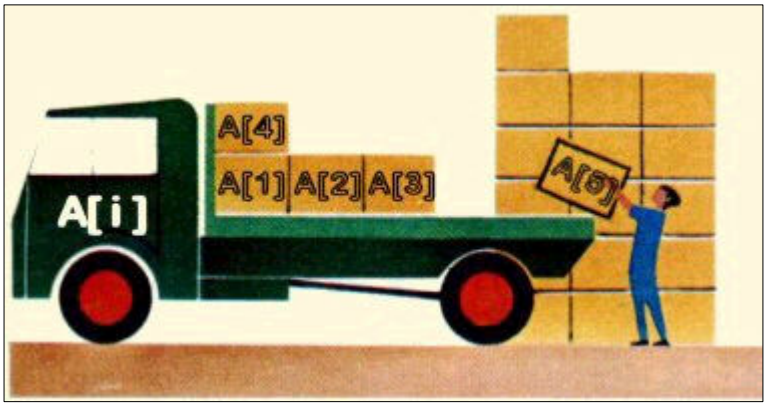
مثال:

```
maxloop=5
For i=1 To maxloop
  Message("loop 1","i")
  For n=1 To 10
    If (n=3) Then
      Break()
    End
    Message("loop 2","n")
  Next n
Next i
```

قطعه کد بالا عملا بـی اسـتفاده اسـت. امـا در هنگـام اجـرا ۵ پیغـام از حلقـه I اجـرا مـی شـود و ۲ پیغـام از حلقـه n.

آرایه Array

حتما تا به حال به قفسه بندی ها توجه کرده اید،هدف از ایجاد قفسه ها نظم دادن به قرار گیری اشیا همسان می باشد.برای مثا در یک فروشگاه تمام صابونها در یک قفسه و در کنار هم هستند،یا همانند شکل بسته های چیده شده در یک کامیون.



همان طور که واضح است انتقال اجناس همسان توسط یک کامیون منطقی تر از آن است که اجناس را تک تک به مقصد اتقال دهیم.در MMB نیز می توانیم از آرایه ها استفاده کنیم.البته آرایه های تک بعدی(1-Dimensional) دلیل این نوع نا مگذاری اینست که محتوی این آرایه در یک خط مرتب می شوند.آرایه ها در MMB گروهی از متغیرها تحت یک نام منفرد هستند.در MMB از آرایه ها همانند متغیر ها استفاده می شود.جدول زیر مقایسه ای بین آرایه ها و متغیرها را نشان می دهد.

Variable	Array
A$	A$[ ]
Name$	Name$[n]
Clicks$	Clicks[n]
User_address$	User_address$[n]
User_Counter	User_counter[n]

در جدول بالا ستون سمت چپ متغیرها هستند و ستون سمت راست آرایه ها هستند.همان طور که مشاهده می کنید آرایه ها نوع توسعه یافته متغیرها هستند.آرایه ی [ ]User_Counter را در نظر بگیرید می خواهیم ببینیم این آرایه از چه قسمتهایی تشکیل شده است.User_Counter برچسب آرایه است،[ ] براکت ها جهت مشخص کردن شمارنده و n به عنوان شمارنده(Index) محسوب می شود.

چنانچه قسمت متغیر ها را مطالعه کرده باشید متوجه شده اید که متغیرها تقریبا در همه جا می باشند.یکی از نقش های متغیرها اینست که مؤلفه های آرایه ها را مشخص نمایند.البته در این میان از متغیرهای عددی استفاده می شود.

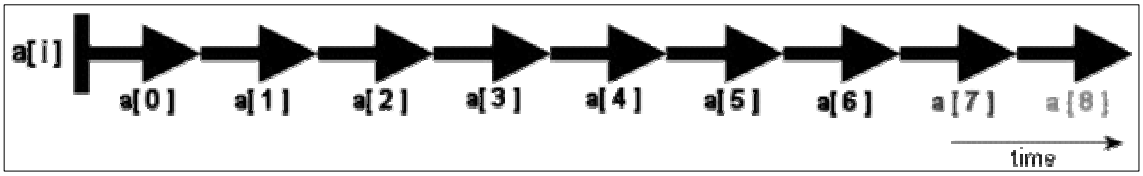
مثالی از آرایه هایی که شمارشگر آنها عددهای معمولی هستند:

```
a[1]
twain$[72]
my_finger[534]
tool$[102]
```

مثالی از آرایه هایی که شمارشگر آنها متغیرهای عددی هستند:

```
a[i]
twain$[chapter_no]
my_finger[102]
```

از آرایه ها معمولا بیشتر در حلقه های For…nextاستفاده می شود.



هنگامیکه مولفه های آرایه ها پر می شوند،جدید ترین مؤلفه بالاترین مقدار را به خود اختصاص می دهد.

همچنین برای مشخص کردن مولفه ها در آرایه ها می توانیم از اعمال ریاضی استفاده می کنیم:

```
a[I+1]
twain$[chapterNo_4]
```

اما چطور می توانیم آرایه ها را پر کنیم؟این امر بستگی به نوع آرایه ها دارد.

«آرایه های عددی»

همانند متغیرهای عددی ،آرایه هـای عـددی هـم بـرای نگهـداری اعـداد اسـتفاده مـی شـوند.از آرایـه هـای عـددی نیـز بـرای اعمال ریاضی استفاده می کنیم.اجازه دهید تا آرایه ای را مقدار دهی کنیم:

```
a[I]=c
myfinger[I=15
```

در مثال بالا در آرایـه ی a[ ]، مقـدار c بـه مولفـه ۱ آن نـسبت داده شـده اسـت و در قـسمت دوم مقـدار ۱۵ بـه آرایـه نـسبت داده شده است.

خواندن محتوی آرایه ها نیز به صورت زیر است:

```
c=a[1]
finger=myfinger[I]
```

در مثال بـالا نیـز مؤلفـه شـماره ۱ آرایـه a[ ] بـه متغیـر c اختـصاص داده مـی شـود.ودر قـسمت دوم مقـدار مؤلفـه آرایـه بـه متغیر منتقل می شود.

«آرایه های رشته »

همانند متغیرهـای رشـته ای ،آرایـه هـا ی رشـته ای نیـز بـرای ذخیـره کلمات،عبـارات متنـی و کـلا رشـته هـا مـورد اسـتفاده قرار می گیرند.آرایه های رشته ای همانند زیر هستند:

a$[I]

aبرچسب آرایه است ،$به MMB می گویـد کـه نـوع آرایـه رشـته ای اسـت،[ ] براکـت هـا و مؤلفـه هـا کـه هـم مـی تواننـد عدد باشند و هم متغیر عددی.مقدار دهی آرایه های رشته ای همانند زیر است:

```
a$[I]=c$
finger_name$[I]=name$
```

در مثـال بـالا مؤلفـه اول آرایـه a$[ ] مقـدار متغیـر c$ را مـی گیـرد و در قـسمت دوم آرایـه finger_name$[ ] محتـوی متغیر name$ را دریافت می کند.

مثالی از استفاده آرایه ها در حلقه ها:

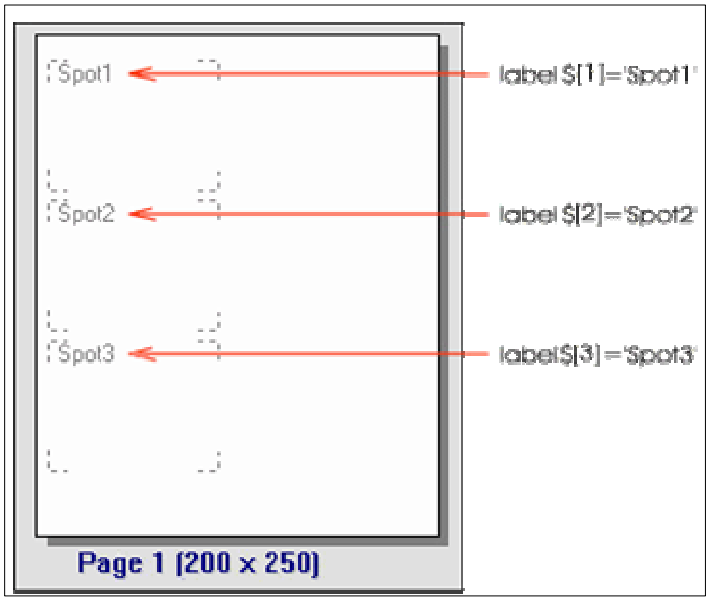
```
For I=1 to 50
 a[I]=I
Next I
```

مثال بالا ۵۰ مؤلفه برای آرایه ایجاد می کند که هر کدام مقدار متناظرش را دریافت می کند،اما به مثال زیر جهت خواندن محتوی آرایه ها توجه کنید:

```
For I=1  to 50
 a=a[I]
 Message("Array item content is :","a")
Next I
```

آرایه های رشته ای به عنوان برچسب اشیا

انعطاف پذیری MMB به شما اجازه می دهد تا از محتوی آرایه های رشته ای به عنوان منبعی برای برچسب اشیا استفاده کنید.



این ویژگی زمانی مفید است که اشیا از اسامی متحدالشکل برخوردار باشند.

مثال:

```
For I=1 to 20
  text$='Hello No. '+CHAR(I)
  Label$[I]=text$ + CHAR(I)
  LoadText("Label$[I]","text$")
Next I
```

توابع آرایه ها

- **GetArrayItem(Array$,SeparatorChar$,Position)**

ایـن تـابع بـر اسـاس موقعیـت داده شـده مؤلفـه ای از آرایـه را بـاز مـی گردانـد.$SepratorCharکـاراکتری را مـشخص مـی کند که آن کاراکتر مرز بین مؤلفه هارا مشخص می کنـد بـه ایـن کـاراکتر جـدا سـاز نیـز اطـلاق مـی شـود.کـاراکتر جـدا سـاز هر کاراکتری می توانـد باشـد ،امـا توصـیه مـی شـود کـه از کاراکترهـای (#)،(*)و(|) اسـتفاده نماییـد.پـارامتر Position بـرای تعیین موقعیت مولفه ی مورد نظر می باشد.

مثال:

```
Items$ = 'item1#item2#item3#item4#item5#'
For i = 1 To 5
 ArrayItem$ = GetArrayItem(Items$,#,i)
 Message("Obtained array item: ","ArrayItem$")
Next i
```

- **GetArrayNum(Array$,SepratorChar$)**

این تابع تعداد مؤلفه های درون آرایه را باز می گرداند. $SepratorChar کـــاراکتر جدا ساز را مشخص می کند.

مثال:

```
Items$ = 'item1#item2#item3#item4#item5#'
NumOfItems=GetArrayItem(Items$,#)
For i = 1 To NumOfItems
 ArrayItem$ = GetArrayItem(Items$,#,i)
 Message("Obtained array item: ","ArrayItem$")
Next i
```

جادوگر اسکریپت (MMB Script Wizard)

همان طور که پیش تر گفته شد این قسمت برای ساده کردن و در عین حال سریع کردن برنامه نویسی می باشد.در این جادوگر لیستی از فرامین MMB موجود می باشد که با انتخاب هر فرمان توضیحی مختصر به همراه محل ورود پارامترهای فرمان ظاهر می شود،البته برخی از فرامین که هیچ پارامتری را دریافت نمی کنند مسلما کادر ورود پارامتر برای آنها ظاهر نخواهد شد.بهتر دیدیم که شرح فرامین MMBرا براساس این جادوگر انجام دهیم.

« توابع اصلی »

قسمت بنیادی اسکریپت نویسی همان فرامین می‌باشند. برخی از این فرامین تابع نیز هستند چرا که مقدار خروجی تولید می‌کنند. صورت کلی فرامین و توابع به صورت زیر است:

Function("Parameter 1" , " Parameter 2")

در این صورت به جای کلمه Function نام فرمان مورد نظر قرار می‌گیرد. Parameter 1 معمولاً محل نوشتن نام شی می‌باشد اما می‌تواند چیزهایی دیگر از قبیل مسیر یک فایل، زمان و... نیز باشد. Parameter 2 نیز محل قرار دادن سایر پارامترهای مورد نیاز هر دستور است. ممکن است گاهی اوقات قسمت Parameter 2 چند قسمت شود که هر قسمت با علامت کاما از هم جدا می‌شود. به طور کلی پارامترهای هر دستور بر اساس خود دستور معین می‌شود. در ادامه می‌خواهیم به شرح و بررسی فرامین و توابع Wizard (جادوگر) MMB بپردازیم.

فرمان « Hide »

شی یا گروهی از اشیاء را پنهان می‌کند. در قسمت پارامتر این فرمان نام شی را می‌نویسیم:

```
Hide("Object Name")
```

فرمان « Show »

این فرمان شی یا گروهی از اشیاء را نمایان می‌سازد. اگر شی یکGIF متحرک باشد به هنگام نمایان شدن شروعش از فریم نخست می‌باشد. در قسمت پارامتر این فرمان بایستی نام شی مورد نظر را بنویسیم.

```
Show("Object Name")
```

فرمان « Invert »

برای نمایان ساختن و یا پنهان کردن شیای به کار میرود. چنانچه شی قابل دیدن باشد آنرا پنهان و چنانچه پنهان باشد آن را آشکار میسازد. نام شی مورد نظر در قسمت پارامتر این فرمان میآید.

```
Invert("Object Name")
```

مثال:

```
Hide("Rectangle 1")
a$ ='Text'
Show("a$ ")
Invert("Circle")
```

نکته: در MMB این قابلیت فراهم است که خارج از صفحه حاوی شی و یا از محلی دیگر همانند Master page حالت نمایش اشیاء را تغییر دهیم. برای این کار بایستی همانند مثال زیر عمل کنیم:

```
Hide("Page 10::Circle")
```

مثال بالا شی Circle را در صفحه ۱۰ پنهان میکند. توجه کنید که ابتدا محل شی(صفحه) را مینویسیم (همانند Page 10)

سپس : : را مینویسیم و نهایتاً نام شی مورد نظر.

مثال:

```
Hide("Master Page :: Rect1")
Hide("Page 10::Circle")
Show("Page 1:: Text")
```

فرمان « RunMBD »

این فرمان باعث اجرای فایل دیگری با پسوند M BD میشود. همچنین میتوانید تعیین کنید که این فایل در صفحه جاری اجرا شود و یا در صفحهای دیگر.استفاده از این دستور برای تولید محصولات با حجم بالا بسیار کارا وموثر میباشد، با این عمل شما بدنه محصول خود را به چند قسمت مجزا تقسیم میکنید که به جای حرکت بین صفحات بین این فایلها حرکت میکنید. با استفاده از این فرمان دیگر نگرانی از بابت مشکل با حافظه وجود نخواهد داشت. چرا که به محض بارگذاری فایلی جدید،فایل قبلی از حافظه پاک میشود. یکی از استفادههای این

فرمـان در جـایی اسـت کـه مایـل هـستیم هـر صـفحه خـصوصیت‌هـای منحـصر بـه فـردی داشـته باشـد. البتـه بایـستی در استفاده از این فرمان دقت داشت چرا که ممکن است مسیرها به درستی ذکر نشود.

```
RunMBD("Path to MBD" , "Page Label")
```

پـارامتر اول مـسیر فایـل MBD را مـشخص مـی‌کنـد و پـارامتر دوم نـام صـفحه را. مـی‌تـوانیم بـرای اجـرای ایـن فایـل MBD صفحه‌ای مجازی در نظر بگیریم.

فرمان « Run »

از این فرمان برای اجـرای فایـل‌هـای اجرایـی خـارجی اسـتفاده مـی‌شـود. بـرای تغییـر مکـان پنجـره فایـل اجرایـی، بـه ایـن معنا که پنجره فایل مـورد نظـر روی پنجـره‌هـای فایـل‌هـای دیگـر اجـرا شـود و یـا خیـر (بیننـده ایـن پنجـره را بـالای بقیـه ببیند)، ۲ پارامتر موجود است که این کار را انجام دهد:

1- Top 2- Top Most

حال ببینیم که تفاوت این ۲ پارامتر در چیست؟

Top-

این پـارامتر ایـن اطمینـان را مـی‌دهـد کـه پنجـره فایـل مـا همیـشه بـر روی سـایر پنجـره‌هـا کـه از قبـل بـاز شـده‌انـد اجـرا می‌شود. اما پس از کلیک کردن روی پنجره‌ای دیگر این پنجره به زیر سایر پنجره‌ها می‌رود.

Top Most-

این پارامتر باعث حفظ وضعیت قرارگیری پنجره از نظـر بـالاتر بـودن نـسبت بـه سـایر پنجـره‌هـا مـی‌شـود. (تـا حـد امکـان بالاتر از سایر پنجره‌ها)

مثال:

بـرای اجـرای برنامـه Notepad ویـنـدوز کـه بـالاتر از بقیـه باشـد و در ضـمن سـند مـورد نظـر مـا در بـالاترین سـطح قـرار گیرد،از کد زیر استفاده می‌کنیم:

```
Run("C:\Windows\Notepad.exe" , "Top Most  C:\My doc.txt")
Run(" Path to executable file" , " Parameters")
```

در این فرمان پارامتر اول مسیر فایل اجرایی را مشخص می‌کند و پارامتر دوم نحوه عملکرد پنجره و خود برنامه را مشخص می‌کند.

علاوه بر آن چنانچه از پارامترهای Top \ Top Most استفاده شود،MMB تا زمان اجرا و آغاز برنامه خارجی صبر خواهد کرد. این وقفه گاهی اوقات می‌تواند خطرناک و مشکل آفرین باشد. همانند مواقعی که برنامه کاربردی اتصال یافته اجرا نمی‌شود و همین عدم اجرای صحیح باعث گیرکردن و از کار افتادن اجرا کننده‌ی MMB می‌شود. برای رفع این مشکل راه حلی هست و آن اینکه می‌توان با اختصاص دادن زمانی معین در اجرای برنامه اصلی(که توسط شما تولید شده است)وقفه ای ایجاد نمودو پس از سپری شدن زمان مقرر شده فرمان اجرای آن فایل را داد. با استفاده از این ترفند نگرانی از بابت عدم اجرای فایل اجرایی و از کار افتادن اجرا کننده‌ی MMB از بین می‌رود. ضمناً فراموش نشود که واحد این زمان بر حسب میلی ثانیه است.

مثال:

```
Run("C:\Windows\Notepad.exe","Top Most (2000)  C:\Mydoc.txt")
```

اما باید توجه کنیم که در استفاده از فرمان Run محدودیت‌هایی وجود دارد، محدودیت‌هایی از قبیل:

-Top \ Top Most از پنجره‌های فرزند حمایت نمی‌کنند و تنها برای اولین پنجره باز شده از برنامه به کار می‌روند.

- برخی از برنامه‌های غیر استاندارد که فاقد PID (Proccess ID) هستند ممکن است که به درستی عمل ننمایند.

- چنانچه شما برنامه منبع خود را (که قبلاً در MMB توسط خودتان تولید شده و فایل خارجی را فراخوانی می‌کند) ببندید به هیچ وجه برنامه فراخوانی شده به طور خودکار بسته نخواهد شد.

- چنانچه می‌خواهید برنامه‌های سیستمی از قبیل Regedit.exe,Notepad.exe و ... را به همراه پارامترهای Top \ Top Most اجرا نماید باید مسیر کامل این فایل‌ها را وارد نمایید. بر خلاف حالتی که این فراخوانی فاقد ۲ پارامترTop \ Top Most می‌باشد که نیازی به نوشتن مسیر کامل نیست و تنها در صورت نوشتن نام فایل اجرایی، برنامه‌های سیستمی اجرا می‌شوند.

این فرامین به ترتیب از راست به چپ برای رفتن به اولین صفحه، رفتن به آخرین صفحه، رفتن به صفحه بعد و رفتن به صفحه قبل می‌باشند. این فرامین هیچ پارامتری را دریافت نمی‌کنند و در موقع استفاده از آنها بایستی ۲ پرانتز را به صورت ( ) در انتهای آنها بیاوریم.

مثال:

```
First Page()
Next Page()
```

فرمان « Page »

توسط این فرمان می‌توان به صفحه‌ی مورد نظر انتقال یافت. برای این کار تنها کافیست تا نام صفحه را در قسمت پارامتر این فرمان بنویسیم. شایان ذکر است که در استفاده از فرمان Page می‌توانید با نوشتن LAST PAGE در قسمت پارامتر این فرمان به آخرین پنجره‌ای که قبلاً باز شده باز گردید.

Page("Page Label")

مثال:

```
Page("Page 5")
```

فرمان « Exit »

سبب خروج از برنامه می‌شود.

```
Exit()
```

فرمان « Minimize »

این فرمان باعث انتقال برنامه به نوار وظیفه می‌شود.

```
Minimize()
```

فرمان « Exit Timer »

این فرمان باعث خروج از برنامه پس از گذشت چند میلی ثانیه می‌شود. زمان دلخواه را می‌توان در قسمت پارامتر این فرمان وارد کرد. توجه نمایید که استفاده از فرمان Page Timer می‌تواند باعث لغو این فرمان شود. همچنین کلیک نمودن بر روی هر شی فعالی باعث لغو این فرمان حین به کارگیری این فرمان خواهد شد. پس می‌توانید با

گذاشتن دکمه‌های مناسب در برنامه از کلیک های ناخواسته بر جاهای دیگر جلوگیری نمایید. این قابلیت برای مواقعی که می خواهیم وقایعی صورت گیرد و سپس از برنامه خارج شویم مناسب است.

ExitTimer("MiliSec")

مثال:

```
ExitTimer("1000")
```

فرمان «Page Timer »

باعث انتقال به صفحه‌ای دیگر پس از گذشت مقدار معینی زمان بر حسب میلی ثانیه می‌شود. چنانچه صفحه‌ی خاصی را بیان ننمایید برنامه به صفحه‌ی بعدی می‌رود.

PageTimer("MiliSec", "Page")

در ضمن می‌توانیم به جای پارامتر P age از پارامترهای زیر نیز استفاده کنیم:

- THIS_SCRIPT:باعث اجرای اسکریپت در همان صفحه می‌شود.

- THIS_PAGE:باعث اجرای مجدد همان صفحه جاری می‌شود.

- IF_IDLE: به صفحه دیگر منتقل می‌شود چنانچه کاربر عمل خاصی را جز صبر کردن انجام ندهد.

نکته: کلیک بر روی هر شی فعال باعث لغو این فرمان می‌شود. استفاده از فرمان PageTimer یا ExitTimer دیگر باعث لغو PageTimer جاری می‌شود.

مثال:

```
PageTimer("10000"," Page 5" )
```

فرمان « DisplayValue »

باعث نمایان شدن جزء صحیح متغیری در شی متنی می‌شود این فرمان برای شمارشگرها مناسب می‌باشد.

DisplayValue("Text Object", "Variable")

پارامتر اول شی متنی را که نمایش دهنده مقدار است تعیین می‌کند و پارامتر دوم متغیر مورد نظر را معین می‌کند.

مثال:

```
DisplayValue("Text 1", "Counter")
```

فرمان « Message »

این فرمان برای نمایان ساختن کادرپیغام(Message Box) می‌باشد. البته باید یادآور شد که این کادر پیغام ارائه شده توسط MMB همانند سایر کادرهای پیغام از انعطاف پذیری بالایی برخوردار نیست.

Message("Any String", "Variable")

پارامتر اول محتوی پیغام را مشخص می‌کند و پارامتر دوم هم برای نشان دادن محتوی متغیر چه عددی و چه رشته‌ای است.چنانچه بخواهیم پیغام در چند خط نمایان شود بایستی از عبارت (\n)در بین محتوی پیغام استفاده کنیم.

مثال:

```
age=20
Message ("Your age is : " , "age")
```

فرمان « Return »

در قسمت‌های قبل بررسی شد.

فرمان‌های «AGif Reset, AGif Stop, AGif Play»

این فرامین به ترتیب از راست به چپ باعث اجرای IF G شده، باعث توقف GIF و بازگشت به اولین فریم GIF می شود و اجرا را از سر می‌گیرد. بایستی توجه کرد که اجرای یک فایل به مشخصات آن هم بستگی دارد.

AGIFPlay("Ani GIF Object")

AGIFStop("Ani GIF Object")

AGIFReset("Ani GIF Object")

در این فرامین در قسمت پارامتر بایستی نام شی GIF را وارد نماییم.

مثال:

```
AGIFPlay("Start")
```

فرمان « RunScript »

از این دستور برای اجرای شی اسکریپتی استفاده می شود.همچنین ازاین فرمان در پاره‌ای از اوقات می‌توان برای تولید حلقه استفاده نمود.

RunScript("Script object")

در قسمت پارامتر باید نام شی اسکریپت را قرار دهیم.

مثال:

```
RunScript("Script 1")
```

فرمان « ScriptTimer »

این فرمان شی اسکریپتی را پس از گذشت چند میلی ثانیه اجرا می‌کند.

ScriptTimer(“Script Object” ,”MiliSec”)

پارامتر اول نام شی اسکریپتی را مشخص می‌کند و پارامتر دوم مقدار زمان بر حسب میلی ثانیه.

فرمان « OpenFile »

این فرمان یک کادر بازگشایی فایل را نمایان می‌سازد و سپس می‌توانیم توسط ثابت‌های CBK-OpenDir, CBK-OpenFile,<File> و متغیر $OpenFile مقدارهای بازگشتی را دریافت کنیم.

OpenFile("Filter ","Default Extension")

در قسمت پارامتر این فرمان بایستی فیلتر را معین کنیم که تنها کادر بازگشایی چه نوع فایل و یا فایل هایی را شناسایی کند. در زیر مثالی برای فیلتر آمده است:

MPEG Files (*.mpg)|*.mpg|All Files*|*.*||

ایـن فیلتـر در نهایـت سـبب ایجـاد ۲ انتخـاب بـرای کـادر بازگـشایی مـی‌شـود توجـه نماییـد کـه کـاراکتر " | " بـرای جـدا کـردن قـسمتها (پـسوندها) و کاراکترهـای " || " بـرای پایـان عبـارت بـه کـار مـی‌رونـد. پـارامتر دوم ایـن دسـتور پـسوند پیش فرض را تعیین می‌کند.

مثال:

```
OpenFile(" Text File (*.txt )| .txt| All File |*.*| | , *.txt")
Message("The file you selected is: " , "Open file$" )
LoadText("paragraph" , " < File > ")
```

فرمان «SaveFile»

فرمان مذکور کادر محاوره ای ذخیره فایل را نمایان می سازد.

- SaveFile(“Filter”,”Defualt Extension”)

اولین پارامتر ایـن فرمـان فیلتـر را مـشخص مـی کنیـد ودومـین پـارامتر بـرای تعیـین پـسوند پـیش فـرض مـی باشـد.در زیـر مثالی برای فیلتر آمده است:

Text File (*.txt)| .txt| All File |*.*| | , *.txt

نحـــوه مـــشخص کـــردن فیلتـــر هماننـــد دســـتور OpenFile مـــی باشـــد.در ایـــن دســـتور نیـــز ۳ ثابـــت <File>،CBK_OpenFileوCBK_OpenDir مقدار دهی می شوند.

مثال:

```
SaveFile("TXT Files (*.txt)|*.txt|All Files (*.*)|*.*|","*.txt")
Message("Selected file in OpenFile$ variable: ","OpenFile$")
file$=<File>
Message("Selected file in <File> constant: ","file$")
file$=CBK_OpenFile
Message("Selected file name: ","file$")
folder$=CBK_OpenDir
Message("Folder path of selected file: ","folder$")
```

مثال:

در ایـن مثـال ابتـدا کـد منبـع یـک فایـل HTML بـه درون یـک ListBox بـار گـذاری مـی شـود وسـپس توسـط فرمـان SaveFile می توان فایل HTML را در جایی دیگر ذخیره کرد.

```
OpenFile("HTML Files (*.htm;*.html)|*.htm;*.html|All Files
|*.*||","*.htm;*.html")
If (OpenFile$<>'')Then
ListBoxDeleteItem("ListBox","-1")
ListBoxAddItem("ListBox","OpenFile$")
OpenFile$=''
End
Filename$='CBK_OpenFile
SaveFile("HTML Files (*.htm;*.html)|*.htm;*.html|All Files|*.*||","filename$")
If (OpenFile$>'') Then
SongListSave("ListBox","OpenFile$")
OpenFile$=''
End
```

فرمان « FileString »

این فرمان به جستجوی زیـر رشـته‌هـای مـشخص شـده در مقـداری کـه توسـط < File > دریافـت مـی‌شـود مـی‌پـردازد. بـه عنوان مثال چنانچـه < File > دارای زیـر رشـته wav باشـد مقـدار متغیـر بازگـشتی ایـن فرمـان 1 و در غیـر ایـن صـورت صفر می‌شود.

FileString ("SubString ","Variable")

پـارامتر اول زیـر رشـته را مـشخص مـی‌کنـد و پـارامتر دوم متغیـری را بـرای دریافـت وضـعیت معـین مـی‌کنـد. ایـن متغیـر بایستی از نوع عددی باشد.

مثال:

```
FileString(".Swf "," Status")
```

فرمان «InstallFont»

از این فرمان هنگامی استفاده می‌شـود کـه نـسخه‌ای از فونـت بـه کـار رفتـه در پـروژه شـما در سیـستم کـاربر نهـایی موجـود نباشد. توسط این فرمان می‌توانیـد موقتـاً ایـن فونـت را نـصب نماییـد. توجـه نماییـد کـه بـه هنگـام خـروج از برنامـه فونـت هم پاک خواهد شد. معمولاً این فرمان در هنگام شروع صفحات مورد استفاده قرار می‌گیرد.

InstallFont ("Path to font")

در قسمت پارامتر مسیر فونتی که مایلیم نصب شود را باید مشخص کنیم.

مثال:

```
InstallFont("< Scr Dir>\funny.ttf")
```

ضمناً شما می‌توانید از فونت‌های الحاق شده (Embedded) استفاده نمایید.

فرمان « ClipBoard »

این فرمان جهت ارسال و یا دریافت محتوی متغیر از Clip Board ویندوز می‌باشد.

Clipboard ("SEND/GET ","String Variable")

چنانچه بخواهیم متغیر را ارسال کنیم از کلمه SEND (فرستادن) در پارامتر استفاده می‌کنیم و اگر بخواهیم متغیر را دریافت کنیم از کلمه GET (گرفتن) در پارامتر اول استفاده می‌کنیم. پارامتر دوم متغیری را مشخص می‌کند اطلاعات در آن قرار می‌گیرد و یا از آن گرفته و به Clip Board ارسال می‌شود. لازم به ذکر است که این فرمان تنها بر اساس متن و عدد کار می‌کند و قادر به ارسال شی‌هایی از قبیل عکس به کلیپ برد نمی‌باشد.

مثال:

```
a$ = 'Text to Clip board'
Clipboard ("SEND ","a$" )
```

فرمان « FileExist »

این فرمان به امتحان اینکه فایل مشخص شده موجود است یا خیر می‌پردازد. و در صورت وجود مقدار ۱ را به متغیر اختصاص می‌دهد و در صورت عدم وجود مقدار صفر را به متغیر اختصاص می‌دهد.

FileExist("Path to file","Variable")

پارامتر اول مسیر فایل مورد نظر را تعیین می‌کند. و پارامتر دوم متغیر را مشخص می‌کند.

مثال:

```
FileExist("C:\My file.exe","Isexist" )
```

فرامین «WaveStop, WavePlay»

ایـن فـرامین بـه ترتیـب از راسـت بـرای اجـرا و یـا توقـف اجـرای فایـل صـوتی خـارجی و یـا داخلـی بـا فرمـت Wave اسـتفاده مـی‌شـود. در قـسمت Command ایـن فرمـان شـما مـی‌توانیـد بـا قـرار دادن کلمـه Loop باعـث اجـرای مکـرر فایل صوتی شوید. لازم به تذکر است کـه بهتـرین نتیجـه بـه هنگـام اسـتفاده از Loop مـوقعی بدسـت مـی‌آیـد کـه Direct Sound موجود باشد.نکته بـسیار مهـم دیگـر اینـست کـه چنانچـه فایـل صـوتی الحـاق شـده باشـد لازم نیـست در قـسمت مسیر فایل Wave Play مسیر کامل را بنویسیم تنها کافیست که نام فایل صوتی بدون پسوند نوشته شود.

WavePlay ("Path to file ","Loop")

پارامتر اول مسیر فایل صوتی Wav را مشخص می‌کند و پارامتر دوم جهت ایجاد تکرار استفاده می‌شود.

WaveStop ( )

سبب توقف اجرا می شود.

مثال:

```
WavePlay("D:\Blue.Wav "," Loop" )
```

فرمان‌های «Audio Rewind, Audio Pause, Audio Stop, Audio Play, Audio Open»

فرامین گفته شده بـه ترتیـب از راسـت بـه چـپ بـرای بازگـشایی فایـل صـوتی، اجـرای فایـل صـوتی، توقـف اجـرای فایـل صـوتی، وقفـه موقـت در اجـرا و جلـو و عقـب بـردن مـی‌باشـند. ایـن فـرامین بـرای کنتـرل فایلهـای ASF, WMA,OGG,MP3 می‌باشند. در ضمن این فایل‌ها بایستی خارجی باشند نه الحاق یافته.

AudioOpen(" Path to file")

در قسمت پارامتر مسیر فایل صوتی مـشخص مـی‌شـود. اگـر در قـسمت پـارامتر از رشـته خـالی اسـتفاده کنـیم یعنـی چیـزی ننویسیم ایـن فرمـان کـادر بـاز کـردن فایـل را نمـایش خواهـد داد. هماننـد سـایر کادرهـای بازگـشایی ایـن کـادر بازگـشایی نیز قابلیت انتخـاب چنـدین فایـل همزمـان را بـرای مـا فـراهم مـی‌کنـد و اسـامی و مـسیر و تمـام فایـل‌هـای انتخـاب شـده درون یک لیست اجرایی درونی و در ماکروی<List> ذخیره می‌شود.

AudioPlay( )

AudioStop( )

AudioPause( )

AudioRewind("seconds" ,"Command")

پارامتر اول زمانی را بر حسب ثانیه معین می‌کند که بایستی بر اساس آن به عقب بازگشت. برای اینکه عمل جلو بردن را انجام دهیم می‌توانیم در پارامتر اول در کنار زمان علامت منفی را قرار دهیم. در پارامتر دوم نیز می‌توانیم کلمه RELATIVE را قرار دهیم تا عمل‌های عقب و جلو بردن نسبی صورت گیرد.

مثال:

```
AudioOpen("F:\Music\Bird.mp3")
AudioRewind("-3000","")
```

فرامین «CDPlay, CDStop, CDPause , CDTrack ,CDForward,

CDBackward,CDSkipForward,CDSkipBackward,CDPlayerPause,Which CD Track»

همان طور که واضح است فرامین گفته شده به ترتیب از چپ به راست، برای اجرای CD صوتی، توقفCD صوتی، ایجاد وقفه موقت در اجرای صوتی، مشخص کننده Track آغازین، جلو بردنCD صوتی، عقب بردن CD صوتی، باعث جلو رفتن به مدت ۱۰ ثانیه می‌شود، باعث عقب رفتن به مدت ۱۰ ثانیه می‌شود، باعث توقف CD می‌شود و فرمان آخر شماره فایل صوتی در حال اجرا را باز می‌گرداند. لازم به تذکر است که چنانچه قصد اجرایCD های Mixed-Mode را دارید شماره اولین Track صوتی به جای یک از ۲ شروع می‌شود زیرا اولین Track داده‌های دیگر می‌باشند.

CDPlay( )

CDStop( )

CDPause( )

CDTrack("Track # ")

در این فرمان در قسمت پارامتر شماره Track (شروع از یک قرار) می‌گیرد.

CDForward( )

CDBackward( )

CDSkipForward( )

CDSkipBackward( )

CDPlayerPause( )

WhichCDTrack(“Track Number”)

قسمت پارامتر متغیری را مشخص می‌کند که شماره Track در حال اجرا به آن اختصاص داده می‌شود.

مثال:

```
WhichCDTrack("Track No" )
```

فرامین « Midi Stop, Midi Play »

این فرامین به ترتیب برای اجرای فایل‌های با پسوند MIDI و توقف اجرای آنها می‌باشند.

MidiPlay("Path to MIDI " , "LOOP ")

پارامتر اول مسیر فایل MIDI را مشخص می‌کند و پارامتر دوم سبب ایجاد تکرار در اجرای فایل می‌شود.نوشتن پارامتر دوم اختیاری است.

MidiStop( )

مثال:

MidiPlay("C:\MyMusic\Fall.MIDI")

فرامین « MOD Stop, MOD Play »

به ترتیب از راست برای اجرا و توقف فایلهای MOD می‌باشند.

MODPlay("Path to MOD")

در قسمت پارامتر مسیر فایل صوتی قرار می‌گیرد.

MODStop( )

مثال:

```
MODPlay("C:\MyMusic\Fall.MOD")
```

فرامین «StopSound , PlaySound»

فرمان سمت راست باعث اجرای کلیه فایل‌های صوتی پیشتیبانی شده توسط MMB می‌شود.ضمناً از این فرمان می‌توانیم پس از فرمان File Open استفاده کنیم و در قسمت پارامتر PlaySound می توانیم به جای مسیر از ماکروی <File> استفاده کنیم. فرمان دوم هم کلیه فایل‌های صوتی در حال اجرا از جمله آهنگ زمینه را متوقف می‌کند.

PlaySound("Path")

در قسمت پارامتر بایستی فایل صوتی را مشخص نمود.

StopSound( )

مثال:

```
PlaySound("C:\MyMusic\Fall.Mp3")
```

فرامین « VolumeDown , VolumeUp »

به ترتیب از راست به چپ، باعث افزایش حجم صدا بر اساس مقدار ورودی می‌شود و فرمان بعدی حجم صدا را هر با به میزان ۵٪ کاهش می‌دهد.

VolumeUp("Volume")

در قسمت پارامتر می‌توان درصدی را که می‌خواهیم بر اساس آن صدا بلندتر شود را وارد کرد(مقادیر صحیح بین 100 تا 100-). با استفاده از این فرمان نیز می‌توان حجم صدا را کاهش داد. برای این منظور بایستی در کنار درصد علامت منفی را نیز قرار دهیم.

VolumeDown( )

مثال:

```
VolumeUp("50")
```

فرامین «PrintRect, PrintPage, PrintText»

فرمان اول از سمت راست سبب چاپ محتوی شی متنی می‌شود، فرمان دوم باعث چاپ کل صفحه می‌شود و فرمان سوم سبب چاپ محدوده‌ی محصور بین کادر شی فرضی مستطیل شکل خواهد شد.

PrintText("Text Object","FONT – FRM – OBJECT")

در این فرمان پارامتر اول شی متنی را مشخص می‌کند و پارامتر دوم سبب چاپ متن با همان فونت اصلی خودش می‌شود، اما چنانچه این پارامتر حذف شود MMB متن را بر اساس تنظیمات قسمت چاپ، چاپ خواهد نمود.

PrintPage("fit")

PrintRect("Rectangle", "fit")

پارامتر اول شی‌ای را که به منزله‌ی شی فرضی مستطیلی عمل می‌کند را مشخص می‌کند و پارامتر دوم هم درصد اشغال صفحه را مشخص می‌کند:

مثال:

```
PrintText("Paragraph","FONT – FRM – OBJECT")
```

فرامین TTS Install ، Init TTS، Say، Speak Text، Stop TTS، Pause TTS، Resume TTS، Pitch TTS، Speed TTS

- Text – to- Speech چیست؟

به یک فرآیند گویند که طی آن یک متن به صوت دیجیتالی مبدل می‌شود و سپس بازگو می‌شود. فرامین مذکور جهت انجام همین فرآیند می‌باشند. TTS مخفف عبارت Text – to- Speech است.

Install TTS، باعث نصب MS Text – to- Speech از یک مسیر ویژه می‌شود. ابتدا کامپیوتر کاربر از نظر داشتن این ویژگی بررسی می‌شود و چنانچه این ویژگی از قبل موجود نباشد MMB به سرعت اقدام به نصب آن می‌نماید. تنها کافیست فایل اجرایی نصب TTS را بر روی CD مربوطه قرار دهید. تا MMB عملیات نصب را به صورت خودکار انجام دهد. نام این فایل اجرایی Ms tts 221.exe است.

اگر در طراحی پروژه خود قصد استفاده از TTS را داشتید حتماً لازم است که دستور Install TTS را در ابتدای برنامه بگنجانید. چنانچه کابر نهایی از قبل این ویژگی را نصب کرده باشد برنامه بدون توقف به اجرای باقی فرامین

می‌پردازد. Init TTS باعث بنیان نهادن ویژگی TTS می‌شود. این فرمان ممکن است برای بنیان نهادن TTS چند ثانیه را صرف کند و اغلب بر روی سیستم‌های با سرعت پایین بیشتر طول می‌کشد. لذا بهترین مکان برای قرار دادن این فرمان بلافاصله پس از فرمان TTS Install می باشد.MMB در هنگام بنیان نهادن ویژگی TTS توسط این دستور کادر سیاه رنگی مبنی بر این عمل را نشان می‌دهد.البته این ویژگی در ویندوزهای 9xممکن است به درستی عمل نکند.زیرا راه انداز کارت صوتی ممکن است که با این ویژگی سازگارنباشد.بهترین کار در این صورت بازدید از سایت تولید کننده کارت صوتی به منظور دریافت اطلاعات بیشتر می باشد.Say، باعث گفتن متنی می شود البته بایستی TTS از قبل نصب شده باشد. SpeakText سبب گفتن متن هر شی که حاوی متن است می‌شود. در این حالت نیز باید TTS از قبل نصب شده باشد. Stop TTS گفتن را متوقف می‌کند. Pause TTSسبب ایجاد مکث در گفتن متن می‌شود و Resume TTS باعث از سرگیری مجدد گفتن می‌شود.Pitch TTS ،این فرمان برای تنظیم زیر و بم صدا استفاده می‌شود. Speed TTS سرعت گفتن را بر اساس واحد کلمه در ثانیه تنظیم می‌کند.

InstallTTS( )

InitTTS( )

Say("Text")

متنی را که می‌خواهیم قرائت شود در بین پرانتز در قسمت پارامتر قرار می‌گیرد.

SpeakText("Text object")

در قسمت پارامتر شی متنی مشخص می‌شود.

StopTTS( )

PauseTTS( )

ResumeTTS( )

PitchTTS("Value")

در قسمت پارامترمقدارهای بین ۵۰ تا ۲۰۰ هرتز را می‌توان وارد کرد. مقدار پیش فرض ۱۰۰هرتزاست.

SpeedTTS("Value")

مقدار سرعت در قسمت پارامتر فرمان می‌آید، مقادیر می‌تواند بین ۳۰ تا ٤٥٠ باشد. مقدار پیش فرض ۱۵۰ است.

فرامین(MP3 List)

در این قسمت یک لیست اجرایی (Play List) درونی موجود می‌باشد که شما می توانید فایل های پشتیبانی شده توسط MMB را به این قسمت اضافه نمایید. یا در صورت تمایل لیست اجرایی خارجی را (M3L\M3u) وارد نمایید.

SongListReset( )

این فرمان باعث پاک کردن یک آهنگ قسمت از لیست اجرا می شود.

SongListAdd("< Src Dir > \ Path ")

باعث افزودن یک لیست اجرایی دیگر به لیست اجرایی درونی می شود.

SongListDel("number")

باعث پاک کردن آهنگ مشخص شده از لیست می شود.

SongListPlay("number")

این فرمان باعث اجرای آهنگ های درون لیست، شروع از یک می شود.

SongListNext( )

باعث اجرای آهنگ بعد آهنگ جاری از لیست اجرایی می شود.

SongListPrev ( )

باعث اجرای آهنگ قبلی آهنگ جاری می شود.

SongListLoad("Path")

این فرمان باعث وارد کردن یک لیست خارجی به درون لیست درونی می شود و یا در حالت دیگر می توانید مستقیماً اقدام به وارد کردن فایلهای صوتی نمایید.

مثال:

```
OpenFile("PlayLists(*.m3l;*.m3u;*.pls)|*.m3l;*.m3u;*.pls|AllFiles|*.*||","*.m3l;*.m3u;*.pls")
SongListLoad("OpenFile$")
```

SongListRND( )

باعث ایجاد لیست درونی تصادفی می شود.

SongListEdit( )

این فرمان باعث بازگشایی یک کادر ویژه می شود که کاربر می تواند بر حسب نیاز به ویرایش لیست اجرایی بپردازد و آن را ذخیره و یا بارگذاری نماید. همچنین در این کادر ویژه می توان با درگ کردن فایلها از Explorer به درون این کادر فایلها را به لیست اضافه نمود. ضمناً این قابلیت فراهم است تا به توسط فرامین Song ListLoad ، SongListDel و SongListSave به بارگذاری، ذخیره و پاک کردن این لیست پرداخت اما این شرایط برای فایلهایی که از Explorer درگ شده اند، صادق نیست.

SearchForSong("Path")

این فرمان بر اساس مسیر داده شده به جستجوی فایل های پشتیبانی شده توسط MMB می گردد و پس از یافتن فایلها لیست اجرایی را پر می کند.

مثال:

```
SearchForSong("<SrcDir>\MyMP3\")
```

یا

```
OpenFile("Play Lists(*.mp3;*.ogg;*.asf;*.wma)|*.mp3;*.ogg;*.asf;*.wma|All
Files|*.*||","*.mp3;*.ogg;*.asf;*.wma")
SelDir$=CBK_OpenDir
SearchForSongs("SelDir$")
```

SongListSave("ListBoxObject","Path to save")

این فرمان باعث ذخیره کردن محتوی لیست اجرایی درونی در یک فایل می شود.

فرامین تصویر(External Image Commands)

فرمان «ViewJPG»

VeiwJPG(" Path to EXTERNAL JPG " , " CENTER ")

از ایـن فرمـان بـرای مـشاهده فایـل هـای تـصویری بـا پـسوند BMP و JPG اسـتفاده مـی شـود. پـارامتر اول ایـن دسـتور مسیر فایل عکس را مشخص مـی نمایـد و پـارامتر دوم سـبب نمـایش دادن عکـس در وسـط صـفحه مـی شـود. ایـن فرمـان بـرای CD هـای حـاوی عکـس مناسـب مـی باشـد.MMB در هنگـام نمـایش دادن عکـس مربوطـه آن را بـه درون پنجـره ای قرار می دهد که چنانچه کاربر بر روی آن کلیک نماید باعث بستن پنجره می شود.

ReplaceImage("Bitmap Object" , "PATH TO EXTERNAL JPG File")

این فرمان سبب جایگزینی شی عکـس مـشخص شـد. بـا عکـس جدیـدی مـی شـود.بـه عنـوان مثـال فـرض کنیـد کـه شـی عکسی در صفحه دارید، حـال بـر حـسب نیـاز احتیـاج بـه تغییـر آن داریـد، مـی توانیـد بـا ایـن فرمـان ایـن کـار را صـورت دهید.

جای مناسب دیگری که می توان از ایـن فرمـان سـود جـست در کاتـالوگ هـا و برنامـه هـایی اسـت کـه بـر اسـاس مقـداری عکـس معنـا دار مـی باشـند اسـت و بـدین وسـیله هـم سـرعت اجـرای برنامـه را بـالا بـرده و هـم مـی توانیـد محـصول بزرگتـری را تولیـد نماییـد. در ایـن فرمـان پـارامتر اول، شـی عکـسی را مـشخص نمـوده و پـارامتر دوم بـرای مـسیر دهـی عکس دوم که جایگزین خواهـد شـد اسـت. از دیگـر ویژگـی هـای ایـن فرمـان اینـست کـه مـی توانیـد بـه جـای مـشخص کردن شـی عکـسی از Hot-Spot اسـتفاده نماییـد و از آن بـه عنـوان محـل نمـایش عکـس جدیـد اسـتفاده نماییـد. بـدیهی است که فرامین اختصـــاص داه شـــده بـه شـی Hot-Spot بـه قـوت خـود بـاقی مـی مانـد. ضـمناً بـرای حـذف عکـس جاری از محل بارگذاریش در صفحه می توان پارامتر دوم فرمان را خالی گذاشت.

یکی از راه هـای بهینـه سـازی پـروژه اسـتفاده مناسـب از ایـن فرمـان مـی باشـد. فـرض کنیـد کـه در هنگـام شـروع برنامـه عکـسی بارگـذاری مـی شـود حـال مـی توانیـد بـا اسـتفاه از دسـتور Replace Image و پـارامتر دوم کـه خـالی اسـت در هنگـام بـستن صـفحات عکـس دار مقـداری از حافظـه را کـه توسـط آنهـا اشـغال شـده اسـت را بـه سیـستم بـاز گردانیـد. دستور Replace Image فرمـت دیگـری هـم دارد کـه بـرای بارگـذاری یـا حـذف عکـس در صـفحات اسـتفاده مـی شـود. برای مثال می توانید از صفحه جاری عکسی را در صفحه ای دیگر بارگذاری نمایید.

مثال:

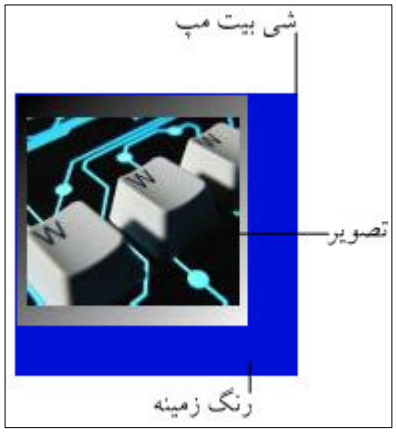
```
ReplaceImage("Page 1::Bitmap2")
```

فرمان « UseImageBkgColor »

این فرمان رنگی را به عنوان زمینه در محـدوده ی شـی عکـس نمایـان مـی سـازد.ایـن فرمـان ۲ پـارامتر دارد کـه یکـی بـرای مشخص کردن نام شی عکس است و دومی برای فعال یا غیر فعال کردن رنگ زمینه.

اما جهت پارامتر دوم می تون به ترتیب از اعداد 0 (برای غیر فعال سازی) و ۱ (برای فعال سازی) استفاده کرد.

UseImageBkgColor("Image","Switch")



فرمان « SetImageBkgColor »

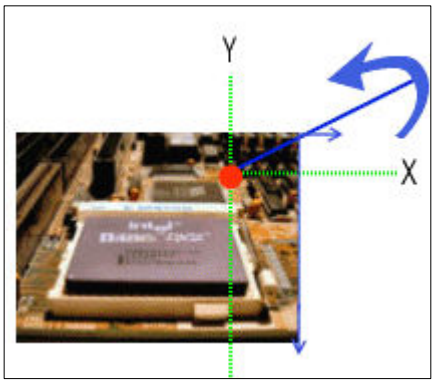
این فرمان رنگ زمینه را تعیین می کند.

SetImageBkgColor("Image ","RGB Value")

این فرمان ۲ پارامتر دارد که یکی برای مشخص کردن نام شی عکس است و دیگری برای مقدار رنگ که در کانال RGB معرفی می شود است.

فرمان « SetImageOrigin »

از این فرمان جهت تنظیم نقطه شروع شی عکس استفاده می شود (به عبارتی تکیه گاه شی عکس) تنظیم این نقطه بر اساس منتهی علیه بالا سمت چت شی عکس صورت می گیرد. کارکرد این فرمان برای ۲ حالت تغییر اندازه شی و چرخش شی می باشد و چرخش شی و تغییر اندازه عکس حول این نقطه صورت می گیرد.



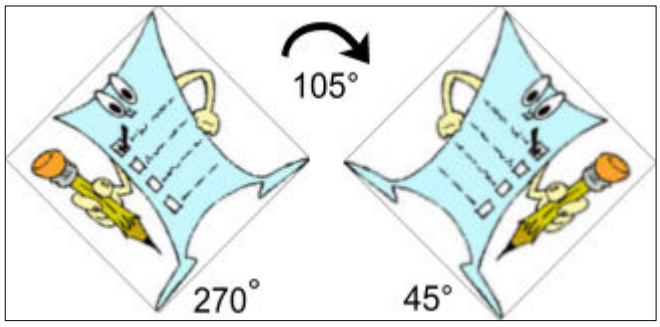
SetImageOrigin("Image","X,Y")

این فرمـان هـم متـشکل از ۲ پـارامتر اسـت کـه اولـین آن بـرای نـام شـی و دومـی بـرای مـشخص کـردن نقطـه مـی باشـد. قسمت دوم خود مستقل از دو قسمت Xجهت فاصله افقی و Y جهت فاصله عمودی است.

مثال:

SetImageOrigin("Image" , "80,30")

فرمان « RotateImageRel »



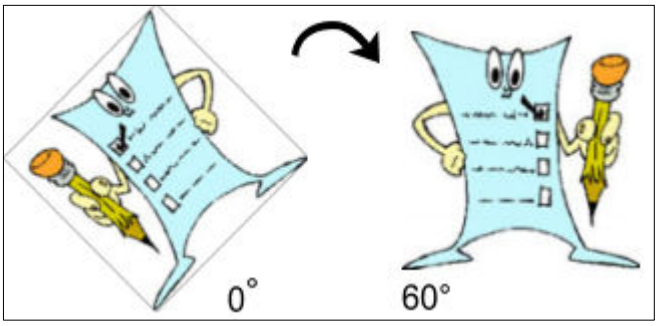
این فرمان سبب چرخش عکس به نسبت زاویه جاری عکس می شود.

RotateImageRel("Image","Degree")

جهت استفاده از این فرمان باید ۲ پارامتر را وارد کرد: اولی نام شی تصویر می باشد و دومی مقدار زاویه چرخش.

فرمان « RotateImageTo »

این فرمان سبب چرخش مطلق شی می شود. بدین معنی که اگر زاویه جاری بین تصویر و افق ۴۵° درجه باشد و ما بخواهیم تصویر چند درجه ای بچرخد، زاویه جاری صفر در نظر گرفته می شود و تصویر به مقدار زاویه داده شده می چرخد.



RotateImageTo("Image","Degree")

همانند قبل این فرمان هم ۲پارامتر دارد که یکی جهت معین کردن نام شی تصویر و دیگری جهت مشخص کردن مقدار زاویه.

فرمان « ScrollImageView »

گاهی اوقات است که تصویر بارگذاری شده در محدوده ی شی تصویر تماماً قابل دید نیست و چاره ی کار استفاده از دستور پیمایش است. این دستور ۲ پارمتر دارد اولی برای مشخص کردن نام شی و دومی برای مشخص کردن مقدار پیمایش. در پارامتر دوم مقدار X برای طول Y برای عرض است. همچنین می توان از مقادیر منفی استفاده کرد.

ScrollImageView("Image","X,Y")

فرمان « ZoomImageView »

از این فرمان برای بزرگ نمایی و یا کوچک نمایی تصویر استفاده می شود.برای این دستور ۲ پارامتر موجود است یکی جهت مشخص کردن نام شی و دیگری برای مقدار تغییر از نظر کوچکی وبزرگی، مقدار پیش فرض ۱۰۰ است.

ZoomImageView("Image","Value")

فرمان « RestoreImage »

چنانچه بخواهیم تغییرات اعمال شده به شی تصویر از قبیل چرخش و تغییر اندازه را خنثی کنیم از این فرمان استفاده می کنیم. این فرمان تنها یک پارامتر دارد و آن هم تنها نام شی می باشد.

RestoreImage("Image")

فرمان "ImageOpacity"

ImageOpacity("OpacityLevel")

توسط این فرمان می توان شفافیت تصویر را بین محدوده ۱ تا ۱۰۰ تغییر داد.

مثال :

```
ImageOpacity("50")
```

توابع ماتریس«Matrix function»

فرمان «MatrixSet»

Matrix Set ("Matrix Object [Column , Row] , "Image index (0…3) ")

این فرمان باعث تنظیم یک تصویر برای خانه ای از ماتریس می شود.در قسمت اول پارامتر اول نام شی ماتریس را قرار می دهیم،در قسمت دوم همین پارامتر به ترتیب از چپ شماره ستون و سطر را می نویسیم.در ادامه در پارامتر دوم شماره تصویر مورد نظر را می آوریم.

به عنوان مثال تمایل داریم که در موقعیت ۳ و ۲ (سطر و ستون) ماتریس تصویر شماره ۲ را نمایش دهد:

MatrixSet ("My Matrix [2,3] " . " 2")

چنانچه بخواهیم سطرها و یا ستون ها را به یک تصویراز ماتریس تغییر دهیم می توانیم به جای شماره سطر یا ستون از عدد صفر استفاده نماییم.

MatrixSet ("My Matrix [5,0] " . " 2")

این گونه از فرمان باعث تنظیم تصویر شماره ۲ برای تمام سطور می شود.

MatrixSet ("My Matrix [5,0] " . " 2")

این گونه از فرمان باعث تنظیم تمام ستونها برای نشان دادن تصویر شماره ۲ می شود.

MatrixSet ("My Matrix [0,0] " . " 2")

این گونه از فرمان باعث تنظیم تمام سطرها و ستون ها و سطور برای نشان دادن تصویر ۲ می شود.

فرمان« MatrixGet»

MatrixGet ("Matrix Object [Column, Row] " . "Variable")

این فرمان شماره عکس جاری در موقعیت مشخص شده را به یک متغیر اختصاص می دهد.

مثال:

```
MatrixGet("MyMatrix[2.3]","Cur")
If  ( Cur = 1 ) then
  MatrixSet("MyMatrix[2.3]"," 2")
```

```
End
```

برخی اوقات لازم است که بدانیم کاربر بر روی کدام درایه و موقعیت ماتریس کلیک کرده است برای این کار می توانیم از ثابت های MxRow, MxCol و نسبت دهی آنها استفاده نماییم.دو فرمان زیر را در نظر بگیرید.

MatrixSet("MyMatrix [0,0] " . " 1")

MatrixSet("MyMatrix [Mx C ol, Mx Row] " . " 2")

اولین فرمان باعث تنظیم کلیه قسمتها برای نشان دادن شکل ۱ می شود و دومین فرمان قسمت کلیک شده را با شکل شماره ۲ تعویض می کند.

فرمان «SysCommand»

SysCommand("Command”,”Parameters")

این فرمان برای کنترل و تغییر خصوصیات و ویژگی های پیشرفته پنجره ها استفاده می شود. البته یک استثناء و آن هم برای Copy File وجود دارد. پارامتر اول نوع دستور است و پارامتر دوم بر حسب نوع دستور معین می شود.

فرامین:

SysCommand("ResizeWindows", "Width,Height")

از این فرمان برای تغییر اندازه پنجره استفاده می شود. در قسمت width (عرض) و height (ارتفاع) می توان مقدار مورد نیاز را قرار داد. این مقدارها می توانند ثابت و یا متغیر باشند.

مثال:

```
SysCommand("ResizeWindows","0,400")
```

این فرمان در مثال بالا باعث تغییر اندازه پنجره به ترتیب به ارتفاع ٤٠٠ پیکسل و عرض بدون تغییر می شود.

```
a=200
b=100
SysCommand("ResizeWindows","a,b")
```

این فرمان اندازه را به مقدار 200×100 تغییر می دهد.

SysCommand(“CenterWindow”,””)

این فرمان باعث تغییر مکان پنجره به وسط صفحه نمایش می شود.

SysCommand("MoveWindows", "X,Y")

این فرمان باعث تغییر مکان پنجره پروژه بر اساس پارامترهایX و Y می شود.

نکته: MMB از قبل ۲ متغیر از پیش تعیین شده ScreenHeight و ScreenWidth را در اختیار شما جهت دریافت اندازه پنجره جاری قرار می دهد.

SysCommand("AlwaysOnTop","")

این فرمان پنجره پروژه را به بالای تمامی پنجره ها و بر روی همه ی پنجره های دیگر موجود منتقل می نماید.

SysCommand("NotAlwaysOnTop", "")

این فرمان باعث می شود که موقعیت قرارگیری پنجره پروژه از نظر قرار گیری بر روی سایر پنجره ها به موقعیت اولیه و معمولی اش باز می گردد.

SysCommand("CopyFile", "Source$,Destination$")

این فرمان باعث کپی شدن یک فایل از منبع به مقصد می شود.چنانچه پوشه مقصد موجود نباشد این دستور آن را ایجاد می نماید در این دستور پارامتر $Source منبع را مشخص و $Destination مسیر را مشخص می نماید.

مثال:

```
destdir$ = 'C:\Program Files\Copy Test'
Dest$ = destdir$ +' \TestFile.txt'
Source$ ='< Embedded >\Text.txt'
SysCommand("CopyFile" , "Source$ , Dest$" )
```

فرامین متحرک سازی«Animation Commands»

MoveObject("Object", "X,Y ,W, H")

از ایـن دسـتور بـرای تغییـر مکـان یـک شـی (گـروه) بـر اسـاس پارامترهـای X,Y اسـتفاده مـی شـود. همچنـین بـا مقـدار دادن پارامترهای W (عرض)وH (ارتفاع) مـی تـوان همزمـان انـدازه شـی را تغییـر داد. البتـه شـی هـای متنـی از ایـن قاعـده مستثنی است و این شی ها قابلیت تغییر اندازه توسط این دستور را ندارند.

مثال:

```
For i=0 to 100
MoveObject("Bitmap","i,20")
Refresh()
Pause("30")
Next i
```

ترفنـد: ترکیـب ثوابـت( )MouseXو()MouseY بـا MoveObject وحلقـه next … for سـبب حرکـت بلادرنـگ شـی بر اساس حرکت موس می شود.

MoveTo("Object", "X,Y,Steps,Type")

این فرمان ساده باعث حرکـت دادن شـی از مکـان جـاری بـه مکـان مـشخص شـده بـر اسـاس پارامترهـای X,Y مـی شـود. پـارامتر Steps مقـدار جهـش حرکـت را بیـان مـی کنـد و پـارامتر Type نـوع حرکـت طـولی را تعیـین مـی کنـد کـه آیـا شتاب دارباشد یا خیر؟ بـرای پارامـتر Type می تـــوان از عبـارت هـــایEASY TO و

EASYFORM یا هیچ استفاده کرد.

EASY TO: باعث شروع با سرعت معمولی شده و هر چه به انتهای مسیر نزدیک می شود سرعت کم تر می شود.

EASYFORM:حرکت از حالت آهسته شروع شده و به مراتب سرعت بیشتر می شود.

«فرامین ListBox»

-ListBoxAddItem("Object", "Parameter")

این فرمان سبب اضافه شدن یک گزینه به ListBox می شود.پارامتر اول شی ListBox را معین می کند.

در قسمت پارامتر دوم این فرمان می توانید از عبارت های ذیل استفاده نمایید:

- متن (this is test)

- متغیر رشته ای ($text)

- آرایه ی رشته ای ($[ ]text)

- فایل متنی (C:\My Documents\ testfile.txt)

- لیست اجرایی صوتی

(*. Pls,m3u *.m3l) ⇦ (<SrcDir>\ mix.m3u)

- Reset جهت پاک کردن محتویات شی ListBox

- RANDOMIZE جهت تصادفی پر کردن ListBox

مثال:

```
OpenFile("Play Lists(*.m3l;*.m3u;*.pls)|*.m3l;*.m3u;*.pls|All
Files|*.*||","*.m3l;*.m3u;*.pls")
ListBoxAddItem("ListBox","RESET")
ListBoxAddItem("ListBox","OpenFile$")
ListBoxDeleteItem("Object","Number")
```

این فرمان بر اساس شماره داده شده در قسمت Number گزینه ای را پاک می نماید چنانچه بخواهیم تمامی محتویات پاک شود از عدد ۱- استفاده می کنیم.

نکته: شماره گذاری گزینه های ListBox همیشه از ۱ یک شروع می شود.

این مثال پیشرفته گزینه های انتخاب شده را توسط دکمه Delete از صفحه کلید پاک می کند.

تمرین:

-یک شی Script ایجاد نمایید و میانبر آن را به کلید Delete نسبت دهید.

-اسکریپت های ذیل را در شی اسکریپتی کپی نمایید.

- یک ListBox در صفحه ایجاد نمایید.

```
ListBoxGetSelectedItems("ListBox","SelItemsArray$,NumSelItemsArray$,#,NumOfSel
Items")
for i=NumItems to 1
```

```
numitem$ = GETARRAYITEM(NumSelItemsArray$,#,i)
numitem = VAL(numitem$)
ListBoxDeleteItem("SongList","numitem")
next i
```

-ListBoxSelectItem("Object" , "Number")

ایـن فرمـان سـبب انتخـاب گزینـه ای بـر اسـاس پـارامتر Number مـی شـود. چنانچـه بخواهیـد تمـامی آیـتم هـا انتخـاب شود باید در قسمت پارامتر این فرمان از عدد (۱-) استفاده نمایید.

مثال:

```
ListBoxSelectItem("ListBox","3")
```

-ListBoxDeselectItem("Object" , "Number")

این فرمان بر خلاف فرمان قبل عمـل مـی نمایـد بـدین معنـی کـه بـر اسـاس پـارامتر Number گزینـه یـا گزینـه هـایی را از حالـت انتخـاب خـارج مـی نمایـد جهـت درآوردن تمـامی گزینـه هـا از حالـت انتخـاب بایـد از عـدد ۱- در قـسمت Number استفاده کرد.

مثال:

```
ListBoxDeselectItem("List Box"," -1")
```

-ListBoxGetItem("Object" , "Param1.Param2,Param3,Param4")

از این فرمان برای دریافت تمام گزینه هـای یـک List Box بهـره بـرده مـی شـود. اولـین پـارامتر بایـد یـک متغیـر رشـته ای جهت نگهداری رشته گزینه ها باشد کـه ایـن رشـته هـا توسـط یـک کـاراکتر محـدودگر از هـم متمـایز مـی شـود. همچنـین پارامتر دوم هم باید یک متغیر رشته ای باشـد کـه بتوانـد تقـدم گرینـه هـا را ذخیـره نمایـد. پـارامتر سـوم کـاراکتر محـدودگر را مشخص می نماید. و پارامتر چهارم که باید متغیر عددی باشد برای دریافت تعداد گزینه ها است.

مثال:

```
ListBoxGetItems("ListBox","ItemsArray$,NumItemsArray$,#,NumOfAllItems")
for i=1 to NumOfAllItems
GetItem$ = GETARRAYITEM(ItemsArray$,#,i)
Message("Selected Item...","GetItem$")
next i
```

-ListBoxGetSelectedItems("Object", "Param1,Param2,Param3,Param4")

این فرمان محتوی گزینه ی انتخاب شده را دریافت می دارد. اولین پارامتر این دستور باید یک متغیر رشته ای باشد که جهت نگهداری محتوی گزینه ها است.همچنین پارامتر دوم باید یک متغیر رشته ای باشد که تقدم گزینه ها را دریافت دارد. پارامتر سوم کاراکتر محدودگر جهت سوا کردن رشته ها از هم می باشد و پارامتر چهارم بایستی یک متغیر عددی باشد تا تعداد گزینه ها را دریافت کند.

مثال:

```
ListBoxGetSelectedItems("ListBox","SelItemsArray$,NumSelItemsArray$,#,NumOfSelItems")
for i=1 to NumOfSelItems
SelItem$ = GETARRAYITEM(SelItemsArray$,#,i)
Message("Selected Item...","SelItem$")
next i
```

-ListBoxSortItems("Object","Type")

این فرمان مولفه های شی ListBox را بر اساس نام (NAME) ، زمان (TIME) ویا برعکس (REVERSE) یا تصادفی (RANDOM) مرتب می کند. اولین پارامتر این دستور برای مشخص کردن نام شیListBox است و در قسمت پارامتر دوم می توان از کلمات گفته شده بالا برحسب نیاز استفاده کرد.

مثال:

```
ListBoxSortItems(“ListBox”,”NAME”)
```

-ListBoxMoveItem("Object","Number")

این فرمان حالت انتخاب را به جایی که توسط پارامتر دوم معرفی می شود انتقال می دهد. پارامتر اول این فرمان برای تعیین نام شیListBox است و پارامتر دوم موقعیت حرکت را مشخص می کند. پارامتر دوم هم می تواند عدد باشد و هم متغیر عددی.

توجه: شماره مؤلفه های ListBox همیشه از یک شروع می شود.

-ListBoxParam("Object","Switch")

هنگامی که بخواهیم یکسری از خصوصیات شی List Box را که در ادامه بحث می شود تنظیم کنیم از این فرمان استفاه می کنیم. به طور معمول پارامتر اول برای نام شی است و در قسمت پارامتر دوم کلمات زیر جای می گیرد که به ترتیب با انتساب آنها با ON و OFF می توان آن خصوصیت را فعال یا غیر فعال کرد.

DRAG & DROP: این پارامتر جهت ایجاد و حالت کشیدن و انداختن می باشد که به وسیله آن می توان آیتمی را با موس جا به جا کرد.

NUMBERS: از این پارامتر جهت نشان دادن / حذف اعداد کنار آیتم های لیست استفاده می شود.

TIMES: از این پارامتر جهت نشان دادن / حذف زمان کنار آیتم های لیست استفاده می شود.

BACKGROUND: از این پارامتر برای تعیین رنگ زمینه بر اساس کانال RGB استفاده می شود .با انتساب عدد مورد نظر رنگ تعیین می شود.

TEXT: این پارامتر برای تعیین رنگ متن ها بر اساس کانال RGB می باشد.

IDTAGS: این پارامتر هم در صورت فعال بودن تگ های ID فایل های صوتی را نشان می دهد. جهت فعال سازی و غیر فعال سازی این پارامتر می توان از OFF/ON استفاده کرد.

مثال:

```
ListBoxParam("ListBox","NUMBERS = OFF”)
ListBoxParam("ListBox","TIMES = ON”)
```

فرامین« Audio Visualization»

-AudioVisualizationType("AV Object" , "AV Type")

این فرمان نوع رقص نور را مشخص می نماید. پارامتر AV TYPE می تواند دو عبارت زیر باشد:

- ANALYZER

- OSCILLOSCOPE

-AudioVisualizationColor("AV Object" , "Color")

این فرمان رنگ جدیدی را برای شی رقص نور تعیین می نماید. پارامتر Color می تواند به حالت های زیر باشد:

جهت تعویض رنگ زمینه	BACKGROUND = RGB
جهت تعویض رنگ منحنی OSCILLOSCOPE	R,G,B

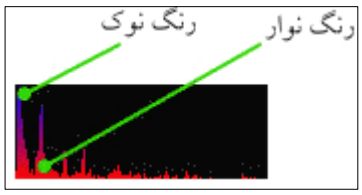
مثال:

```
AudioVisualizationColor("AudioVis","BACKGROUND=150,79,205")
```

این مثال رنگ زمینه شی AV را تغییر می دهد..

```
AudioVisualizationColor("AudioVis","RGB(255,0,0),RGB(0,0,255)")
```

این مثال رنگ نوارها و نوک آنها را در نوع ANALYZER تغییر می دهد



فرمان« LoadText»

LoadText("Text object" , " Variable $ ")
LoadText("Text object" , " Path ")
LoadText("Text object" , " < List > ")
LoadText("Text object" , " < List > Number")
LoadText("Variable $ " , " Path ")
LoadText("Variable $ " , " < List > Number ")

یک فرمان مهم برای متغیرهای رشته ای مـی باشـد.ایـن فرمـان سـبب بارگـذاری یـک رشـته، یـک فایـل از یـک مـسیر، یـک لیست یا یک مولفـه منفـرد بـه درون یـک شـی متنـی مـی شـود. ایـن فرمـان قـادر اسـت یـک فایـل متنـی را از مـسیری بـه درون شی متنی بارگذاری نماید، یا به یک متغیر رشته ای اختصاص دهد.

همچنین این فرمان می تواند محتوی یک متغیر رشته ای را به درون یک شی متنی بارگذاری نماید.

```
C$='<SrcDir>\My File.txt'
LoadText("Object" ,"C$")
```

این مثال یک فایل متنی را به درون یک شی متنی بارگذاری می نماید.

```
C$='what ever text'
LoadText("Object","C$")
```

این مثال متن مشخص شده را درون شی متنی نمایش می دهد.

چنانچه مسیر مشخص شده یک فایـل متنـی اشـتباه و یـا نـاقص باشـد در نهایـت در شـی متنـی همـان مـسیر اشـتباه و نـاقص نمایان خواهد شد.

« فرمت مستقیم Direct Format»

فرمت قبلی به MMB می گفت که چه کنـد. امـا چنانچـه در اسـتفاده از ایـن فرمـت (قبلـی) شـک داشـته باشـیم بهتـر اسـت از فرمت مستقیم بهره جوییم.

FILE : Variable

در این حالت ،محتوی فایل متنی در شی متنی نمایان می شود..

STRING : Variable

در این حالت تنها مسیر فایل در شی متن نمایش داده می شود.

مثال:

```
Path$='<SrcDir>\MyFile.txt'
LoadText("text","FILE:Path$")
```

این مثال سبب بارگذاری محتوی فایل مشخص شده به درون شی متنی می شود.

LoadText("text" , "STRING : Path$")

این گونه از فرمان عیناً رشته اختصاص یافته به متغیر$Path را در شی متنی نمایان می سازد.

نکته: چنانچه بخواهید یک رشته طولانی را به خطهای کوچکتری تقسیم نماید باید از پارامتر \n در رشته استفاده نماید.

مثال:

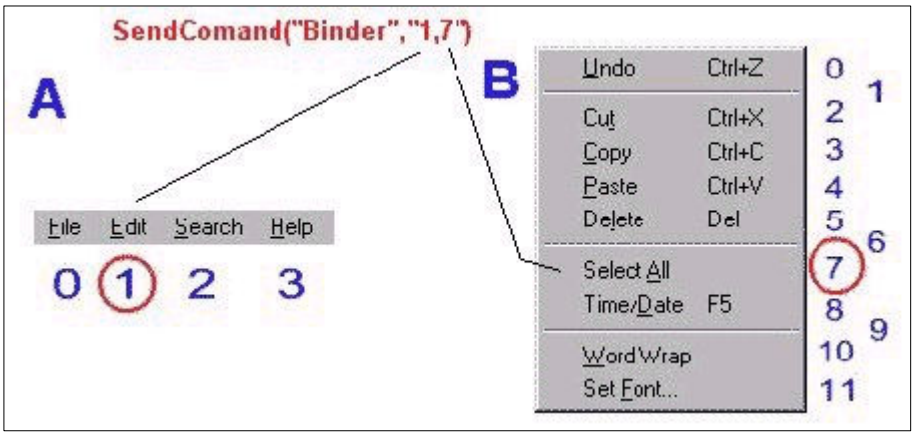
```
LongStr$='This is a very long string \n splited into two rows'
LoadText("text","LongStr$")
```

فرامین کنترل«Binder»

MMB قادر به فراخوانی مولفه های منو های برنامه های اتصال یافته می باشد.قاعده کار به این صورت است که بایستی شماره منو ومولفه را توسط فرمان SendCommand در اختیار MMB گذاشت.

«فرمان SendCommand»

SendCommand(“Object”,”N1,N2”)



پارامتر اول شی Binderرا مشخص می کند و پارامتر دوم برای شماره منو و مولفه مورد نظر است.همانند شکل می توانیم برای بدست آوردن شماره منو و مولفه عمل نماییم.

« ارسال رشته به شی جاسازی شده »

جهت ارسال رشته ای به شی جاسازی شده (اینجا Note pad) می توانیم از فرمان های MMB سود جوییم. ابتدا رشته را به Clipboard می فرستیم و سپس با انتخاب تمام رشته ها و عمل چسباندن، رشته را منتقل می کنیم.

مثال:

```
C$ = 'Hello form MMB'
Clipboard("SEND",”C$")
SendCommand("Binder", "1,7")
SendCommand("Binder", "1,4")
```

با ایده ای مشابه می توان رشته ای را از شی اتصال یافته دریافت نمود، ابتدا تمام رشته را انتخاب می کنیم. سپس آنها را توسط کپی کردن به Clipboardارسال نموده و توسط فرمان

Clipboard("Get","Str$") آن را دریافت می کنیم.

مثال:(در Notepad)

```
SendCommand("Binder", "1,7")
SendCommand("Binder", "1,3")
Clipboard("Get" , "Str$ ")
LoadText("TextDisp" , "C$")
```

فرامین کنترل شی «HTML »

Browser("HTML Object","Command")

پارامتر اول این فرمان نام شی HTMLرا مشخص می کند و پارامتر دوم می تواند یکی از کلمه های زیر باشد:(چنانچه با Mirosoft IE کار کرده باشید،با عملکرد فرامین زیر به خوبی آشنا می باشید)

- Back :فرمان به عقب را به شی HTMLصادر می کند.

- Forward :فرمان به جلو را به شی HTMLمی دهد.

- Stop :فرمان توقف را صادر می کند.

- Refresh :فرمان تازه سازی را صادر می کند.

- OpenFile :کادر بازگشایی فایل را نمایان می سازد(تنها فایلهای سازگار با IE قابل انتخاب می باشند).

- Print :فرمان چاپ سند جاری را صادر می کند.

- URL :مسیر سند مورد نظر را مشخص می کند.این مسیر می تواند بر روی اینترنت هم باشد.

می توانید از اتصالی خاص در شی HTML استفاده کنید که به MMB بگوید به صفحه ای دیگر برود یا اسکرپتی را اجرا نماید.

قاعده به صورت زیر است:

< a href = "page pag 2 "> Go to page 2 < \ a >

این اسکرپیت باعث رفتن به صفحه ی ۲ پروژه می شود.

< a href = " script : Script " > Run Script < \ a >

این دستور برای اجرای اسکرپتی خاص استفاده می شود

« فرامین کنترل شی Flash»

Flash("Object", "Event")

با این فرمان می توانید مشخص کنید که فایل فلش چگونه اجرا شود.پارامتر اول این دستور، شی فلش را معین می کند و پارامتر دوم هم برای مشخص کردن رویداد می باشد.اما رویدادهای ممکن به قرار زیر است:

- PLAY :فایل فلش را اجرا می کند.
- STOP :فایل فلش را که در حال اجرا است متوقف می نماید.
- LOOP :سبب ایجاد تکرار در اجرای فایل فلش می شود.

- SHOW MENU :منوی Right - click را نشان می دهد.
- HIDE MENU :منوی Right - click را پنهان می کند.
- BACK :به فریم قبلی نسبت به فریم جاری می رود.
- FORWARD :به فریم بعدی نسبت به فریم جاری می رود.
- REWIND :سبب به عقب برگشتن فایل فلش می شود.

هر چیز غیر از رویدادهای بالا از قبیل رشته ها به منزله ی مسیر فایل تلقی می شود.

مثال:

```
OpenFile("Flash Files (*.swf)|*.swf|All Files|*.*||","*.swf")
if (OpenFile$ <> '') then
Flash("Flash","OpenFile$")
Flash("Flash","PLAY")
end
```

FlashGetVar("Object", "FLASHVARname, MMBVARname")

فـــرمان بالا متغیر را از کلـــیپ فلش دریافت داشـــــته و در مـــتغیر برنامه MMB ذخیره می نماید.

توجه: متغیر مـورد فراخـوانی بایـد از قبـل در فلـش موجـود باشـد.چـون کـه ترتیـب اجـرای اسـکریپت هـای فلـش ترتیـب خاصی ندارد ممکن است فراخوانی یـک متغیـر بـی نتیجـه باشـد. بنـابراین چنانچـه مطمـئن نیـستید کـه متغیـر مـورد نظـر از قبل در شی فلش موجـود اسـت یـا اینکـه بـه درسـتی مقـداردهی شـده اسـت، توسـط فرمـان ( ) Pause در برنامـه MMB اجرای پـروژه MMB را تـا زمـان مقـداردهی متغیـر در فلـش متوقـف سـازید. راه حـل دیگـر کنتـرل MMB از طریـق شـی فلش و اسکریپت نویسی شی در فلش مـی باشـد کـه لازمـه آن داشـتن فایـل منبـع (*.fla) کلیـپ فلـش مـی باشـد.پـارامتر اول این فرمـان شـی فلـش را معـین مـی کنـد و پـارامتر دوم بـه ترتیـب از چـپ نـام متغیـر درون کلیـپ فلـش و نـام متغیـر MMBرا معین می کنند.

مثال:

```
FlashStr$ ='MmbText.mmbstr'+'GetDateTime'
FlashSetVar("Flash","FlashStr$")
```

سبب تنظیم GetDateTime به متغیرMmbText.mmbstr واقع در کلیپ فلش می شود.

Pause (" 100 ")

به شی فلش مهلت انجام فعالیت های مورد نیاز را میدهد.

متغیرهای $ Date و $ Time توسط فرامین زیر مستقیماً از شی فلش مقداردهی شده اند.

**FsCommand ("mmb "," Date $ = "+ MMB Cur Date Full)

**FsCommand ("mmb "," Time $ = "+ MMB Cur Time Full)

Load text (" Date Str" , "Date $")

Load text ("Time Str" , " Time $")

اما متغیر $Day مستقیماً توسط فرمان FlashGetVar از MMB خوانده می شود.

FlashGetVar("Flash" , "MmbText.MMB cur day , Day $")

LoadText(" DayStr" , "Day$")

FlashSetVar ("Object", “Flash VAR name , MMB VAR name ")

این فرمان سبب تنظیم مقدار جدید به متغیر کلیپ فلش می شود.

FlashSetFrame("Object" , "FRAMENUM")

این فرمان بر اساس پارامتر FRAMENUM شی فلش رابه فریمی که مد نظر است منتقل می کند.

FlashGetFrame("Object" , "VARIABLE")

شماره را فریم جاری را گرفته و در متغیر مشخص شده توسط VARIABLE ذخیره می کند.

FlashGetPorp (" OBJCET" , " , PROPERTY , VARIABLE ")

با این دستور می توان چگونگی خصوصیات تعیین شده برای شی فلش را دریافت کرد.

پارامتر PROPERTY (خصوصیت)می تواند یکی از عبارت های زیر باشد:

- SCALE مقیاس کلیپ را باز می گرداند.

- BGCOLOR رنگ زمینه کلیپ فلش را به صورت hexadecimal در کانال RGB باز می گرداند.

مثال: (رنگ قرمز= F F0000)

-QUALITY کیفیت کلیپ فلش را باز می گرداند.
- PLAYING در حال اجرا بودن یا نبودن کلیپ فلش را می سنجد.
- MOVIE نام کلیپ فلش را باز می گرداند.
- TOTALFRAMES تعداد فریم های کلیپ فلش را باز می گرداند.

قسمت دوم ازپارامتر دوم این دستور برای معین کردن متغیری می باشد که مقدارهای بازگشتی خصوصیات در آن قرار می گیرد.نوع متغیر با توجه به خصوصیت تعیین می شود.

مثال:

فرمانی جهت گرفتن رنگ زمینه:

FlashGetPorp("Flash" ,"BGCOLOR, BGColor$ ")

فرمانی جهت تشخیص در حال اجرا بودن یا نبودن کلیپ فلش:

FlashGetPorp("Flash" , "PLAYING,Play$")

فراخوانی فرامین MMB مستقیماً از Action Script فلش:

جهت برقراری ازتباط بین کلیپ فلش و MMB از فرمان fs Command فلش استفاده می شود.

این فرمان ۲ پارامتر دارد: Command و Arguments .چنانچه بخواهید توسط این فرمان، فرمانی را به MMB ارسال کنید پارامتر Command باید همیشه رشته "MMB" باشد و پارامتر دوم Command باید حاوی یک اسکریپت MMB می باشد.

مثال:

fscommand("mmb", "Time $, "+ MMBCurTimeFull)

این فرمان سبب مقدار دهی متغیر $ Time توسط فلش می شود.

fscommand ("mmb","LoadText(\"DirStr\",\""+ DirString +"\")")

این فرمان سبب فراخوانی فرمان Load Text از MMB می شود و سپس متن DirStr را در MMB پر می کند.

جهت قرار دادن علامت نقل قول در رشته بایستی آن را با کاراکتر Back flash همراه کرد. به این عملیات escaping یک کاراکتر می گویند کاراکترهای متعددی هستند که نمی توان درAction script به تنهایی از آنها استفاده کرد و نشان داد.جدول زیر آنها را نشان می دهد:

Character	Escape sequence
(ASCII 8) کاراکتر Backspace	\ b
(ASCII 12) کاراکتر Form – feed	\ f
(ASCII 10) کاراکتر Line- feed	\ n
(ASCII 13) کاراکتر Carrige Return	\ r
(ASCII 19) کاراکتر Tab	\ t
کاراکتر "	\ "
کاراکتر '	\ '
Back slash	\\
یک بایت مشخص شده در مبنای هشت	\ 000-\377
یک بایت مشخص شده در مبنای 16	\ × 00 - ff×
یک کاراکتر بین المللی (Uni Code) مشخص شده در مبنای 16	\ u 0000 – \ uff-ff

جهت اطلاعات بیشتر پیرامون fs Command از راهنمای برنامه فلش استفاده نمایید

فرمان «ColorPicker»

ColorPicker()

این فرمان کادر محاوره ای انتخاب رنگ را نشان می دهد . این دستور هیچ پارامتر ورودی را قبول نمی کند. به محض اینکه کاربر رنگ را برگزیند و دکمه O k را فشاردهد ثابت CBK-SelColor مقداردهی می شود. لازم به

ذکر است رنگ بر مبنای حالت RGB می باشد.مقدار بازگشتی یک آرایه است که توسط کاما از هم جدا شده اند مثل 100,210,250 که جهت خرد کردن آنها می توان از فرامین رشته استفاده کرد.

فرمان "FontPicker"

این فرمان کادر انتخاب فونت را نمایان می سازد. هنگامی که کاربر فونت مورد نظر خود را برگزیند ثابت CBK_SelFont مقدار دهی می شود. این مقدار یک رشته ساده است که همانند زیر می باشد :

FONTNAME|FONTSTYLE|FONTSIZE|FONTSCRIPT|FONTEFFECT|

FONTNAME : نام فونت
FONTSTYLE : گونه فونت (پر رنگ، ایتالیک و...)
FONTSIZE : اندازه فونت
FONTSCRIPT : شیوه کدگذاری فونت برای زبان
FONTEFFECT : جلوه اعمال شده از قبیل زیر خط دار و...

هر کدام از این مقادیر را می توانید توسط تابع **GetArrayItem** دریافت کنید.

مثال :

```
FontPicker()
FontParams$=CBK_Font
If (FontParams$<>'') Then
  item$[1]=GetArrayItem(FontParams$,|,1)
  item$[2]=GetArrayItem(FontParams$,|,2)
  item$[3]=GetArrayItem(FontParams$,|,3)
  item$[4]=GetArrayItem(FontParams$,|,4)
  item$[5]=GetArrayItem(FontParams$,|,5)
  msg$= 'FONTNAME = ' + item$[1] + CHR(13) + CHR(10) + 'FONTSTYLE = ' +
item$[2] + CHR(13) + CHR(10) + 'FONTSIZE = ' + item$[3] + CHR(13) + CHR(10) +
'FONTSCRIPT = ' + item$[4] + CHR(13) + CHR(10) + 'FONTEFFECT = ' + item$[5] +
CHR(13) + CHR(10)
  Message("Font parameters:","msg$")
End
```

در این مثال کادر انتخاب فونت نمایان شده و پس از انتخاب فونت ، مقادیر به صورت پیغام نمایش داده می شود.

فرمان «SaveVariable»

SaveVariable("Name","Variable")

با استفاده از این فرمان می توان محتوی یک متغیر را در رجیستری کامپیوتر ذخیره کرد.رجیستری یک کامپیوتر،یک پایگاه داده ای است که حاوی اطلاعاتی راجع به پیکربندی سیستم می باشد. با استفاده از MMB می توانید یک متغیر را در رجیستری ذخیره نمایید و یا آن را متعاقباً بارگذاری نمایید. مسیر ذخیره یک متغیر با استفاده از Save Variable به صورت زیر است:

HKEY – CURRENT – USER\Software\MMBplayer\Project EXE\RegName

که قسمت Project EXE نام فایل EXE شماست و قسمت RegName نام همان کلید است که می خواهید متغیر را در آن ذخیره نمایید ،می توان این نام را در قسمت ProjectSetting -> Style تنظیم نمود. حال که دریافتید که متغیرها در کجا ذخیره می شوند بهتر است راجع به چگونگی ذخیر سازی هم صحبت کنیم. این فرمان ۲ پارامتر دارد که اولی جهت تعیین نام متغیر به هنگام ذخیره در رجیستری می باشد و پارامتر دوم نیز که محتوی متغیر را مشخص می کند و هم می تواند رشته ای باشد و هم عددی. به عنوان مثال می خواهیم متغیر $Pass را ذخیره کنیم:

```
Pass$='My Password'
SaveVariable("Password","Pass$")
```

به همین ترتیب می توانیم برای متغیرهای عددی عمل کنیم:

```
HiScore = 5500
SaveVariable("User Hi Score" , "HiScore")
```

فرمان «LoadVariable »

LoadVariable(“Name”,”Variable”)

این فرمان متغیری را که توسط SaveVariable از قبل ذخیره شده است را بارگذاری می نماید این فرمان هم دارای ۲ پارامتر است که اولی نام متغیر ذخیره شده در رجیستری است و دومی محتوی متغیر را مشخص می کند و هم می تواند رشته ای باشد و هم عددی.

در زیر مثالی برای بارگذاری یک متغیر رشته ای از رجیستری آمده است:

```
LoadVariable("Password","Pass$")
```

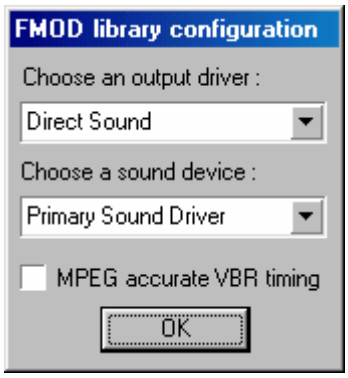
و یک مثال برای متغیر عددی:

```
LoadVariable("User Hi Score","HiScore")
```

فرمان « FMODConfig»

از این فرمان برای نشان دادن کادر پیکربندی کتابخانه FMOD استفاده می شود.این فرمان این قابلیت را فراهم می کند تا کاربر بتواند راه انداز صدای خروجی و ... را تنظیم نماید. همچنین این قابلیت فراهم است که بتوان این تنظیمات را از طریق اسکریپت کنترل کرد. فرمت دستور به شکل مقابل است:

FMODConfig(“Param1","Param2")



چنانچه ما از این دستور بدون هیچ پارامتری استفاده کنیم، کادر تبادلی زیر ظاهر می شود.

اما این فرمان ۲ پارامتر دارد که به وسیله کاما از هم متمایز می شوند، عبارتهای زیر برای قسمت اول پارامتر استفاده می شوند و نوشتن عدد کنار این قسمتها به منزله همان اسامی می باشد.:

1-DircetX Sound(DirectX صدای)

2-Windows Media WaveOut

3-No Sound(بدون صدا)

و برای قسمت دوم پارامتر عبارت های زیر موجودند:

0- MPEG Accurate VBR (Variable Bit Rate) timing off

1- MPEG Accurate VBR (Variable Bit Rate) timing on

برای مثال با استفاده از قسمتهای بالا فرمان به صورت زیر در می آید:

```
FMODConfig("1,0")
```

فرمان « BackgroundPlay »

BackgroundPlay("Path","LOOP")

این فرمان یک فایل صوتی را بازگشایی کرده و به عنوان آهنگ زمینه اجرا می کند. این فرمان ۲ پارامتر دارد که اولی برای مسیر فایل صوتی است. برای مسیر دهی هم می توان از ماکروهاا استفاده کرد و هم از مسیر های ثابت. پارامتر دوم مربوط به تنظیم تکرار آهنگ زمینه در حین اجرای برنامه می باشد.

```
BackgroundPlay("<ScrDir >\Moon.MP3","LOOP")
```

فرمان بالا فایل صوتی Moon.MP3 را به عنوان آهنگ زمینه مکرراً اجرا می کند.

فرمان « BackgroundPause »

BackgroundPause( )

این فرمان سبب ایجاد وقفه در اجرای آهنگ زمینه می شود و این فرمان هیچ پارامتری ندارد.

فرمان « BackgroundStop »

BackgroundStop( )

این فرمان اجرای آهنگ زمینه را به کل لغو می کند ،این فرمان هیچ پارامتری ندارد.

فرمان « VideoLoad »

VideoLoad("Object","Path")

از این فرمان برای بارگـذاری یـک فایـل ویـدئویی اسـتفاده مـی شـود. ایـن فرمـان ۲ پـارامتر دارد اولـی جهـت معـین کـردن نام شی ویدئویی و دومی جهـت مـسیر فایـل ویـدئویی کـه مـسیر مـی توانـد هـم ثابـت و هـم پویـا باشـد.چنانچـه در ایـن فرمـان پـارامتر دوم یعنـی مـسیر فایـل مـشخص نباشـد ،بـه هنگـام اسـتفاده از ایـن دسـتور کـادر بازگـشایی فایـل ویـدئویی نمایان می شود.

فرمان «VideoPlay»

VideoPlay("Object")

این فرمان فایل ویـدئویی بـار گـذاری شـده را اجـرا مـی کنـد.در قـسمت پـارامتر ایـن فرمـان بایـستی نـام شـی ویـدئویی را مشخص نماییم.

فرمان « VideoPause »

VideoPause("Object")

از ایـن فرمـان جهـت ایجـاد وقفـه در اجـرای فایـل ویـدئویی اسـتفاده مـی شـود .چنانچـه فایـل ویـدئویی در حـال توقـف باشد این فرمان سبب اجـرای مجـدد فایـل ویـدئویی مـی شـود. ایـن فرمـان تنهـا یـک پـارامتر دارد و آن هـم تنهـا نـام شـی ویدئویی می باشد.

فرمان « VideoRewind »

VideoRewind("Object","Time/Frame,RELATIVE")

این فرمان سبب عقب برگرداندن فایل ویدئویی می شود. می توانیم مقدار بازگشت را تعیین نماییم، اما این مقدار بستگی به تنظیمات فایل ویدئویی دارد که این تنظیمات توسط فرمان Video Param تعیین شده و بر نوع میلی ثانیه و فریم است.این فرمان ۲ پارامتر دارد یکی جهت مشخص کردن نام شی و دیگری جهت تنظیم مقدار بازگشت.چنانچه مایل باشیم که عقب گرد نسبی صورت گیرد می توانیم کلمه RELATIVE را پس از مقدار بازگشت بیاوریم،همانند مثال زیر:

VideoRewind("Video","-30,RELATIVE")

فرمان « VideoSpeed »

VideoSpeed("Object","Time/Frame,RELATIVE")

این فرمان سرعت اجرای فایلی را که توسط فرمان VideoLoad بارگذاری شده است را تنظیم می کند. با استفاده از اولین پارامتر این فرمان می توان نام شی ویدئویی مورد نظر را مشخص کرد. دومین پارامتر سرعت اجرا را معین می کند که این پارامتر هم می تواند عدد ثابت باشد و هم متغیر عددی. البته باید توجه داشت که سرعت می تواند بین محدودی 2000 تا 0 باشد.

مقدار پیش فرض سرعت1000 است و این بدان معناست که سرعت بالای 1000 سبب افزایش سرعت نسبت به حالت معمول و سرعت پایین 1000سبب کاهش سرعت اجرا نسبت به حالت معمول می شود. جهت استفاده نسبی از تنظیمات جاری می توانیم از عبارت RELATIVE بهره ببریم. شایان ذکر است که می توان با استفاده از شی Matrix سرعت جاری را به صورت گرافیکی در مقیاس درصدی یا هر واحد دیگر بیان کرد.

مثال:

```
VideoSpeed("Video","2000,RELATIVE")
```

فرمان « VideoParam »

VideoParam("Object","Switch")

هنگامی که بخواهیم یکسری پارامتر اضافه تر از قبیل تکرار،قطع صدا و ... را تنظیم نماییم از این فرمان استفاده می کنیم. همانند سایرین اولین پارامتر این فرمان برای معین کردن نام شی ویدئویی است. و در قسمت پارامتر دوم از کلمات زیر استفاده می شود:

FULLSCREEN : این کلمه حالت تمام صفحه را (وبر عکس آن را) فراهم می کند.

LOOP : این فرمان جهت ایجاد تکرار (یا بر عکس) می باشد که خود ۲ حالت دارد :

۱- LOOP=ON که سبب ایجاد تکرار می شود.
۲- LOOP=OFF که سبب لغو تکرار فایل ویدئویی می شود.

MUTE : این کلمه سبب قطع صدای فایل ویدئویی (یا برعکس) می شود ک خود دارای ۲ قسمت است:

۱-MUTE=ON که سبب قطع صدا می شود.

۲-MUTE=OFF که سبب لغو قطعی صدا می شود.

MODE : این حالت تنظیم می کند که آیا ویدئو بر اساس زمان اجرا شود یا بر اساس فریم، با این تفاسیر ۲ حالت وجود دارد:

۱- MODE=TIME بر اساس زمان
۲- MODE=FRAME بر اساس فریم

نکته: ماکرو CBK-VTime تنها در حالتی کاراست که MODE به MODE=TIME تنظیم شده باشد. CBK-Vframe هم تنها در صورت وجود عبارت MODE=FRAME فعال می شود.

باید گفت که MODE پیش فرض مد زمان(TIME) است. چنانچه بلافاصله پس از بارگذاری فایل ویدئویی CBK-Vframe عدد 0 را نشان داد این بدان معناست که فایل ویدئویی ازمد فریم پشتیبانی نمی کند و در نتیجه باید گفت که نمی توان از فرامین جلو بردن / عقب بردن فایل ویدئویی استفاده کرد.

فرمان « VideoScale »

VideoScale("Object","X,Y")

این فرمان اندازه پنجره ویدئو را تعیین می کند. اولین پارامتر این دستور برای تعیین نام شی ویدئویی است. دومین پارامتر جهت تعیین عرض و ارتفاع می باشد.

مثال:

```
VideoScale("Video","50,50")
```

«فرامین MCI»

فرامین MCI برای کنترل اجرای فایلهای صوتی و تصویری می باشند.یادگیری این قسمت تا حدودی مشکل است،چرا که مقوله MCI مبحثی با جزئیات زیاد است و استفاده از آن تنها برای کاربران حرفه ای توصیه می شود.البته باید گفت که امکانات موجود در MMB برای کنترل فایلهای صوتی وتصویری در حد مطلوبی است وتقریبا نیازی به استفاده از این فرامین ندارید.اما به هر ترتیب در این قسمت قصد داریم تنها به ذکر چند فرمان مهم و پر کاربرد MCI بپردازیم.

MCICommand("status cdaudio number of tracks")

فرمان بالا تعداد شیار(Track)های یک CD صوتی را باز می گرداند،بله درست حدس زده اید،این دستور یک تابع نیز هست بدین معنی که یک مقدار خروجی تولید می کند به طور کلی فرامین MCI خروجی خود را به متغیر MCIResult نسبت می دهند.

مثال:

```
MCICommand("status cdaudio number of tracks")
```

```
DisplayValue("Text","MCIResult")
```

MCICommand("play cdaudio from # to #")

این فرمان بر اساس شماره های وارد شده Track یا Track های مربوطه از CD صوتی را اجرا می کند.

مثال:

```
MCICommand("play cdaudio from 3 to 4")
```

MCICommand("status cdaudio length track #")

تابع مذکور طول Track مشخص شده را از CD صوتی باز می گرداند.

مثال:

```
MCICommand("status cdaudio length track 2")
DisplayValue("Text","MCIResult")
```

MCICommand("play cdaudio")

CD صوتی را اجرا می کند.

MCICommand("stop cdaudio")

اجرای CD صوتی را متوقف می کند.

MCICommand("status cdaudio mode")

تابع بالا وضعیت اجرایی CD صوتی را بررسی می کند.چنانچه CD صوتی در حال اجرا باشد تابع مقدار 2– را باز می گرداند.در غیر اینصورت مقدار 1– باز گردانده می شود.

مثال:

```
MCICommand("status cdaudio mode")
If (MCIResult== -2) Then
 Message("CD is now playing","")
Else
 MCIResult== -1
 Message("CD is not playing")
End
```

MCICommand("set cdaudio door open")

فرمان بالا در درایو CD را باز می کند.

MCICommand("set cddaudio door closed")

این فرمان در درایو CD را می بندد.

برای بررسی سایر فرامین MCI به راهنمای MCI و مثالهای درون CD همراه کتاب مراجعه نمایید.

توابع رشته ها[1]

به هنگام کار با رشته ها نیاز است تا اعمال خاصی از قبیل تبدیل انها به حروف بزرگ و بالعکس ، اندازه گیری آنها و ... صورت گیرد که خوشبختانه MMB توابعی سودمند جهت کار با رشته ها در اختیار برنامه نویس قرار می دهد.

CHR(Value)

این تابع براساس عدد وارد شده معادل اسکی آن را باز می گرداند.

مثال:

```
string$=CHR(169) + 'Rasane 2003'
LoadText("Text","string$")
```

در نهایت محتوی متغیر $string صورت زیر خواهد شد:

© Rasane 2003

ORD(character$)

این تابع عدد معادل کاراکتر مورد نظر را بر اساس جدول اسکی باز می گرداند.

مثال:

```
character$='#'
RetVal=ORD(character$)
```

نتیجه کد بالا عدد ۳۵ خواهد بود.

LEN(string$)

این تابع طول رشته مورد نظر را باز می گرداند.

۱- چنانچه به توابع بیشتری در زمینه کار با رشته ها احتیاج دارید ، می توانید از Plugy Plug-In استفاده نمایید.

مثال:

```
LoadText("string$","c:\test.txt")
RetVal=LEN(string$)
```

نتیجه کد بالا عدد ۱۱ خواهد بود.

LOW(string$)

این تابع حروف رشته را به حروف کوچک تبدیل می کند.

مثال:

```
string$='ThIs TeXt WiLL bE CoNvErtED to LoWeRcAsE'
RetString$=LOW(string$)
```

نهایتا عبارت به صورت this text will be converted to lowercase در می آید.

UPP(string$)

این تابع حروف رشته را به حروف بزرگ تبدیل می کند.

مثال:

```
string$='ThIs TeXt WiLL bE CoNvErtED to UppErCasE'
RetString$=UPP(string$)
```

نهایتا عبارت به صورت THIS TEXT WILL BE CONVERTED TO UPPERCASE در می آید.

POS(Substring$,String$)

این تابع به جستجوی زیر رشته(Substring)در رشته(String) داده شده می پردازد.چنانچه زیر رشته در رشته موجود باشد این تابع موقعیت اولین حرف زیررشته را باز می گرداند.اگر زیر رشته در رشته داده شده موجود نباشد تابع مقدار صفر را باز می گرداند.

مثال:

```
string$='Have a nice day!'
substring$='nice'
RetVal=POS(substring$,string$)
```

کد بالا عدد ۸ را باز می گرداند.

توجه نمایید که تابع بالا حساس به حروف می باشد و قبل از استفاده از این تابع بایستی حروف رشته مورد نظر را به حروف کوچک تبدیل کنیم.

NOL(FileName$)

تابع بالا تعداد خطوط فایل متنی بارگذاری شده را باز می گرداند.چنانچه هیچ فایلی برگزیده نشده باشد مقدار بازگشتی این تابع صفر خواهد بود.

مثال:

```
OpenFile("txt Files (*.txt)|*.txt|All Files|*.*||","*.txt")
If (OpenFile$<>'') Then
  RetVal=NOL(OpenFile$)
End
```

کد بالا تعداد خطوط فایل متنی بارگذاری شده را باز می گرداند.

StrCopy(string$,index,count)

این تابع براساس مقادیر ورودی، رشته ای را از رشته اصلی باز می گرداند. index شروع را مشخص می کند و count طول رشته را معین می کند.اگر عددی که شروع را مشخص می کند از طول رشته بیشتر باشد این تابع هیچ رشته ای را باز نمی گرداند.همچنین اگر عددی که طول رشته را معین می کند از طول رشته بیشتر باشد،تابع تنها رشته شروع از index تا آخر رشته اصلی را باز می گرداند.

مثال:

```
string$='Have a nice day!'
ReturnExt$=StrCopy(string$,1,4)
```

کد بالا کلمه Have را باز می گرداند.

StrDel(string$,index,count)

تابع بالا براساس مقادیر داده شده مقداری از رشته اصلی را کسر می کند و باقی مانده رشته اصلی را باز می گرداند.index شروع را مشخص می کند وcount طول را مشخص می کند.چنانچه index از طول رشته بزرگترباشد این تابع هیچ کاراکتری را از رشته اصلی حذف نمی کند.

مثال:

```
string$='Have a nice day!'
ReturnExt$=StrDel(string$,1,5)
```

کد بالا عبارت !a nice day را باز می گرداند.

StrIns(SourceStr$, DestStr$, Index)

تابع گفته شده رشته مقصد را با رشته اصلی در موقعیت بیان شده ادغام می کند.$SourceStr رشته اصلی را بیان می کند و$DestStr رشته مقصد را معین می کند. index نیز برای تعیین موقعیت می باشد.

مثال:

```
sourcestr$='Have a nice day!'
deststr$=' too'
count=LEN(sourcestr$)
ReturnStr$=StrIns(sourcestr$,deststr$,count)
```

کد مثال بالا عبارت !Have a nice day too را باز می گرداند.

StrGet(string$,Int)

این تابع بر اساس مقدار (Int)داده شده کاراکتر متناظرش را باز می گرداند.

مثال:

```
string$='Have a nice day!'
count=LEN(string$)
ReturnStr$=StrGet(string$,count)
```

کد مثال بالا کاراکتر ! را باز می گرداند.

StrSet(String$, Int,C$)

این تابع بر اساس مقـادیر داده شـده رشـته ای جدیـد را در رشـته اصـلی جـایگزین مـی کنـد.String$ رشـته اصـلی را معـین می کند،Int موقعیت را بیان می کند و C$ رشته ای را که بایستی جایگزین شود مشخص می کند.

مثال:

```
string$='Have a nice day!'
count=LEN(string$)
c$='!!!'
ReturnStr$=StrSet(string$,count,c$)
```

کد بالا در نهایت عبارت Have a nice day!!! را باز می گرداند.

StrOfChar(C$, Int)

این تابع بر اساس مقدار(Int) معین شده همان تعداد از کاراکتر مشخص شده را باز می گرداند.

مثال:

```
c$='#'
ReturnStr$=StrOfChar(c$,5)
```

مثال بالا ##### را باز می گرداند.

StrChange(String$, FromStr$, ToStr$)

این تـابع براسـاس مقـادیر داده شـده عبـارتی را جـایگزین عبـارتی مـشخص شـده از رشـته اصـلی مـی کنـد.String$ رشـته اصلی را مشخص مـی کنـد،FromStr$ عبـارتی را کـه بایـد جـایگزین شـود را معـین مـی کنـد و ToStr$ عبـارت جدیـد را مشخص می کند.

مثال:

```
string$='My name is ali'
fromstr$='ali'
tostr$='reza'
ReturnStr$=StrChange(string$,fromstr$,tostr$)
```

مثال بالا در نهایت جمله My name is reza را باز می گرداند.

StrToFile(FileName$, String$, Append, LineFeed)

تابع بالا رشته مشخص شده را در فایلی و یا در انتهای فایلی ذخیره می کند.$FileName فایل مقصد را معین می کند،$String رشته ای که قرار است ذخیره شود را معین می کند،Append شرایط ذخیره کردن رشته در فایل را مشخص میکند که چنانچه به جای آنTRUE را قرار دهیم رشته مورد نظر درانتهای فایل مشخص شده و پس از آخرین کاراکتر ذخیره می شود.اما اگر فایل مشخص شده از قبل موجود نباشد این فرمان فایل را ایجاد می کند.اگرLineFeed و Append هر دو TRUE باشند تابع، رشته مورد نظر را در خط بعد از آخرین خط فایل مورد نظر ذخیره می کند.اما چنانچه هر دو عبارت Append وLineFeed ، Falseباشد تابع رشته مورد نظر را در فایلی جدید و یا بر روی فایل موجود ذخیره می کند.

مثال:

```
file$='c:\temp\test.txt'
string$= 'this string will be append to text file'
ReturnVal=StrToFile(file$,string$,TRUE,FALSE)
```

کد بالا رشته مورد نظر را در فایل test.txt ذخیره می کند.

StrFromFile(FileName$, FromLine, NumOfLines)

این تابع رشته یا رشته ها از فایل مورد نظر بارگذاری می کند.$FileName برای مشخص کردن فایل متنی می باشد،FromLine شماره خطی را که بایستی عمل بارگذاری از آنجا صورت گیرد مشخص می کند وNumOfLine تعداد خطوطی را که باید بارگذاری شوند معین می کند.اگر به جای NumOfLine از عدد 1– استفاده کنیم تمام خطوط فایل متنی بارگذاری می شود.چنانچه NumOfLine= -1 و FromLine بزرگتر از صفر باشد تابع تمام خطوط پس از خط معین شده را باز می گرداند.

مثال:

```
file$='c:\temp\test.txt'
fromline= 1
numoflines=10
ReturnStr$=StrFromFile(file$,fromline,numoflines)
```

کد مثال بالا ۱۰ خط ابتدایی فایل test.txt را باز می گرداند.

ExtractExt(FileName$)

تابع بالا پسوند فایل مشخص شده را باز می گرداند.

ExtractDir(FileName$)

این تابع پوشه فایل مورد نظر را باز می گرداند.

ExtractName(FileName$)

این تابع نام فایل مشخص شده را باز می گرداند.

ExtractDrive(FileName$)

تابع بالا نام درایو فایل مورد نظر را باز می گرداند.

مثال:

```
Path$='c:\MyFiles\text.txt'
ReturnExt$=ExtractExt(Path$)
ReturnDir$=ExtractDir(Path$)
ReturnName$=ExtractName(Path$)
ReturnDrive$=ExtractDrive(Path$
```

مثال:

کد زیر عبارتی را دریافت می کند و ترتیب قرار گیری حروف آنرا تصادفا تغییر می دهد.

```
StrNew$=''
String$='House'
StrLen=Len(String$)
For i=StrLen To 1
 RndNum=RND(StrLen-1)+1
 GetLetter$=StrCopy(String$,RndNum,1)
 StrNew$=StrDel(String$,RndNum,1)
 StrLen=Len(String$)
Next i
LoadText("Text","StrNew$")
```

البته برای مشاهده نهایی بایستی یک شی Text در صفحه قرار دهید.

فصل چهارم

PlugIn

Multimedia Builder

- PlugIn چیست؟
- کنترل رویداد Event Handling
- Tweak PlugIn
- Freeplay PlugIn
- Slideshow PlugIn
- Eval PlugIn
- MMBDLL PlugIn

تا به حال عملا شما قسمت های عمده MMB را فرا گرفته ایدو تقریبا آماده ی تولید پروژه های معمولی می باشید،اما ممکن است در ابتدای کار هیچ محدودیتی را به جهت تولید فرامین و قابلیتهای ارائه شده احساس نکنید.اما هر چه که پیش بروید و تصمیم بـه تولیـد پروژه های بزرگتر بگیرید احتمالا به محدودیتهایی بر خواهید خورد .اما نگران نباشید چاره این کار از قبل اندیـشیده شـده اسـت وآن چیزی نیست جز PlugIn،قابلیتی که ویژگی هایی اضاف بر سازمان را به بدنه اصـلی MMB اضـاف مـی نمایـد.هـم اکنـون بـه مـدد PlugIn راه پیش روی خود را برای تولید پروژه مورد نظر هموارتر احساس می کنید.در ایـن بخـش قـصد داریـم بـه بررسـی نحـوه عملکرد PlugIn های MMB بپردازیم.در ادامه چند PlugIn معروف این برنامه را مورد کنکاش قرار خواهیم داد.

Plug-In چیست؟

اکثر برنامه‌های امروزی این قابلیت را فراهم می‌کنند تا کاربر بتواند ازPlug-In ها استفاده نماید. برای مدت‌های مدیدی این امکان فراهم نبود تا یک کاربر بتواند تعدادی توابع اضافه بر سازمان برنامه در اختیار داشته باشد و می‌بایست تا نسخه بعدی برنامه صبر می‌کرد. اما پس از چندی اشخاصی این قاعده را شکستند و این آرزوی کاربران را برآورده ساختند. پس ازآن با تدوین استانداردها برنامه‌نویسان و توسعه دهندگان اجازه یافتند تا براساس ضوابطی توابع دلخواه خود را در صورت امکان به بدنه برنامه اصلی اضاف نمایند. به همین دلیل این امکانات و توابع اضافه نام plug-in را به خود گرفت. امروزه می‌توانیم ردپای plug-In را در اکثر برنامه‌ها از قبیل برنامه‌های طراحی و ویرایشی و ... ببینیم. اما حال ببینیم plug-In ها دقیقاً چه هستند و چگونه کار می‌کند؟

مطمئناً همه‌ی شما ایستگاههای فضایی را دیده‌اید که حول زمین در حال چرخش هستید. حالا تصور کنید که MMB ساختار مرکزی این ایستگاه فضایی است.

البته شاید شکل بالا شبیه یک ایستگاه فضایی کامل نباشد چرا که سلولهای خورشیدی ندارد. اما به هر حال با تصوری که از شکل بالا پیدا کردید فرض کنید که می‌توانید در این ایستگاه فضایی به ادامه حیات و فعالیت بپردازید.

اما پس از مدتی احساس خواهید کرد که به امکاناتی اضافه احتیاج دارید امکاناتی از قبیل اسباب راحتی و... اما نگران نباشید این احساس نیاز از قبل پیش‌بینی شده و تنها کافیست که وسایل را تهیه و با اتصال آنها به ایستگاه فضایمان را گسترش دهیم. حالا با استفاده از قابلیت توسعه توانستید فضایتان وقدرتان را بهبود بخشید اما اگر بخواهیم به واقعیت نزدیکتر شویم و قدرتمان را در MMB افزایش دهیم دقیقاً باید چه کنیم؟

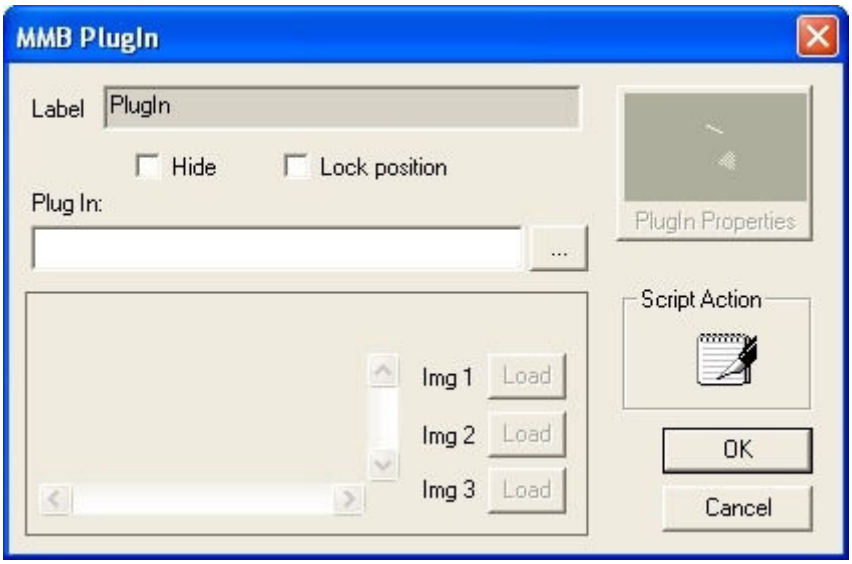
لازم نیست نگران باشید تنها کافیست برنامه MMB و plug-In های مورد نظر را در اختیار داشته باشید.اما حتماً از خود می‌پرسید plug-In های MMB چگونه هستند؟ بله plug-In های MMB تنها فایلی ساده با پسوند DLL هستند که امکان دارد در حالت عادی پنهان شده (Hidden) باشند. (می‌توانید از قسمت Folder option ویندوز آنها را درصورت مخفی بودن نمایان سازید. همچنین می‌توانید در حین اجرای یک برنامه حاوی plug-In، plug-In آن را از شاخه temp ویندوزبیابید). برای مثال فایل‌های misc.dll, tweak.dll, posxy.dll یکسری از plug-In های موجود هستند. اما این plug-In ها را از کجا باید تهیه کنیم؟ پاسخ ساده است شما می‌توانید با استفاده از سایت‌های زیر اقدام به تهیه آنها نمایید:

۱- http: //www.mmbstation.com (سایت شخصی مؤلف)
۲- http://www.bokzy.com
۳- http://www.Advanced-Microtechnologies.com
٤- http://www.Yomoweb.com
٥- http: //www.B8dae9.Hestultra.com
٦- http://www.squashproduction.com

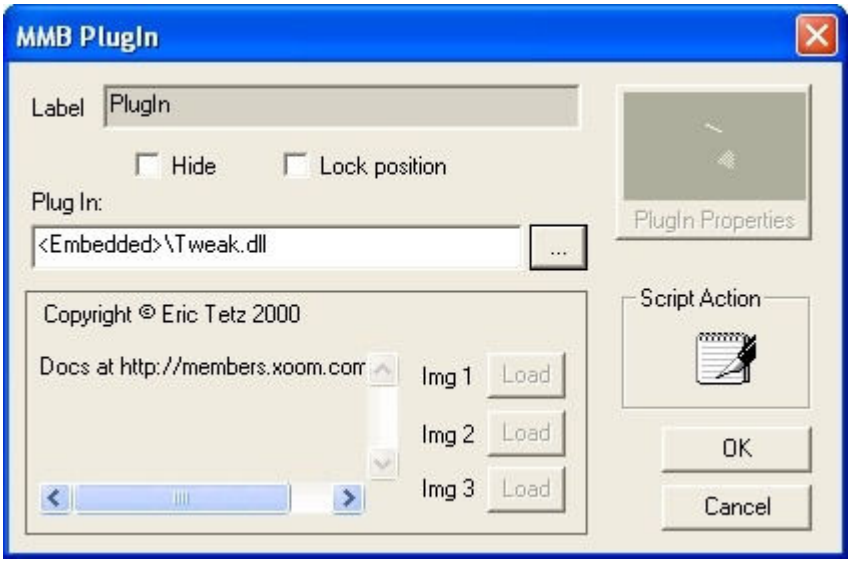
توجه: ممکن است به مرور زمان و به خاطر مشکلاتی لینک‌های بالا کار نکند بنابراین میتوانید از موتورهای جستجو استفاده نمایید.

بسیار خب پس از اینکه plug-In های مورد نیاز را جمع‌آوری کردید وقت استفاده از آنهاست. ابتدا قبل از هرکاری بایستی یک پنجره در صفحه پروژه برای plug-inبکشیم تا در ادامه این پنجره به عنوان نگهدارنده فایل plug-In ما باشد. از منوی Object با انتخاب قسمت plug-In پنجره مورد نظر را می‌کشیم .

برخی از plug-In ها هستند که به صورت بصری عمل می‌کنند.به عنوان مثال plug-In هایی که جلوه‌هایی نظیر بارش برف و باران و حرکت ستارگان را ایجاد می‌کنند. بنابراین این plug-Inها نیاز به پنجره‌ای مناسب و قابل دید دارند، اما درکنار همین plug-In های بصری تعداد دیگری plug-In هم هستند که بصری نیستند و نیاز به پنجره‌ای بزرگ و قابل دید ندارند بلکه فقط وجود پنجره‌ای کوچک در هر جایی برای آنها کافیست. پس از اینکه پنجره را ایجاد کردید بر روی آن ۲ بار کلیک نمایید تا پنجره‌ای همانند شکل ظاهر شود.

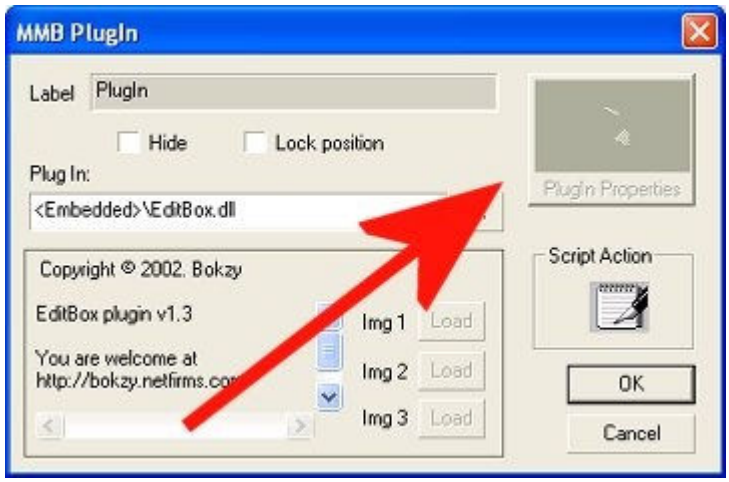


همانند سایر پنجره‌های خصوصیت برای اشیاء دیگر گزینه‌های Lock position و Hide عمل مشابهی را انجام می‌دهند و label (برچسب) هم نام شیء Plug-In را مشخص می‌نماید که باید این نام را به خاطر بسپارید. دروسط این پنجره کادر ورودی هست که کنار آن دکمه‌ای قرار گرفته که توسط  این دکمه کادری تبادلی جهت مسیردهی فایل plug-In ظاهر می‌شود. خب پس چرا منتظرید؟! دکمه را فشار دهید... اگر هیچ plug-In ای را جهت مسیر دهی ندارید به زیرشاخه plug-In از پوشه نصب برنامه MMB بروید و در آنجا یک Plug-In بصری به نام Tenblobs.dll راکه همراه برنامه MMB ارائه شده است را خواهید یافت. بلافاصله پس ازاینکه Plug-In را برگزیدید ، پیغامی مبنی براینکه آیا میخواهید Plug-In را الحاق نمایید؟ ظاهر میشود. به آن پاسخ مثبت بدهید.دیگر احتیاجی نیست که نگران این فایل پس از اتمام پروژه‌اتان باشید زیرا MMB تمام فایل‌های الحاق شده را به همراه فایل نهایی بسته‌بندی می‌کند. پس از الحاق یک Plug-In پنجره خصوصیت به صورت شکل درمی‌آید.



باتوجه به شکل مسیر منتهی به فایل Plug-In دارای ماکروی <Embedded> است که پیش‌تر درباره این ماکرو بحث کردیم. درزیر این کادرمسیر، قسمتی است که حاوی اطلاعاتی از قبیل نام مؤلف Plug-In ،پست الکتریکی ، سایت و ... مؤلف میباشد.

اکنون دکمه‌ی ok را فشار دهید تا عمل نصب Plug-In تمام شود. در اینجاست که پس ازیکبار اجرای برنامه احساس می‌کنید که چیزی کم است،بله درست فهمیدید بر اساس سایر Plug-In های برنامه‌های دیگر شما به این



تصور رسیده‌اید. اما اگر Plug-In شما بصری باشد در کادر خصوصیت Plug-In قسمتی در بالا سمت راست که درشکل نشان داده شده فعال می‌شود که میتوانید به آنجا بروید و تنظیمات موجود را تغییردهید.

اما خصوصیات و کنترل Plug-In های غیر بصری را چگونه می‌توان در دست گرفت؟ در اینجاست که فرامین به کمک ما می‌آیند. فرامینی که در دل اسکریپت‌های MMB جا خوش کرده اند.اما اجازه دهید که ببینیم در پشت پرده چه خبرست؟ همانطور که گفتیم فایلهای Plug-In برنامه MMB، DLL هستند که DLL مخفف عبارت Dynamic link library (کتابخانه اتصال پویا) است به این معنا که DLLها مجموعه‌ای از توابع و فرامین هستند.پس با این تفاسیر ارتباطی پویابین MMB و Plug-In ها برقرار است.این ارتباط می‌تواند به ۳ صورت زیر باشد:

۱- تنظیم متغیر در MMB و دریافت متغیر عددی توسط Plug-In

۲- تنظیم متغیر در MMB ودریافت متغیر رشته‌ای توسط Plug-In

۳- اجرای فرمانی خاص که ممکن است این فرمان خروجی به همراه داشته باشد.

برای استفاده از Plug-In ها بایستی با مفاهیم متغیرها و نحوه استفاده از آنها آشنا باشید.چنانچه این گونه نیست از ادامه خواندن این قسمت تا زمانیکه مفاهیم متغیرها را فراگیرید پرهیز نمایید. خب گذشته از اینها صورت پیشرفته‌تر ۳ حالت قبل به صورت زیردر می‌آید:

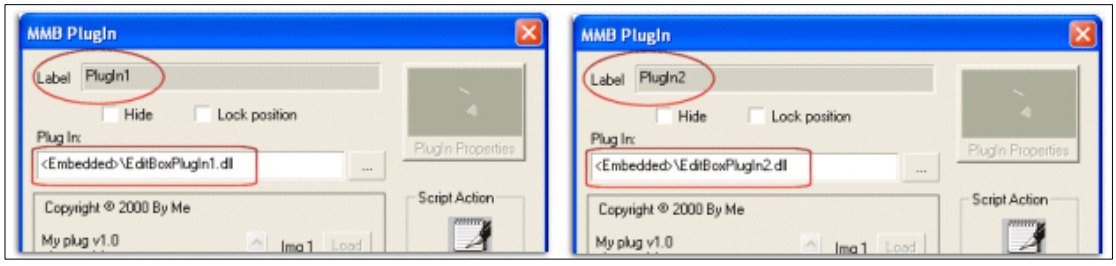
۱- تنظیم پارامترها براساس متغیرها (همیشه لازم نیست)

۲- اجرای فرامین Plug-In براساس پارامترها

۳- دریافت متغیرهای خروجی پس از اجرای فرامین (همیشه اینطور نیست)

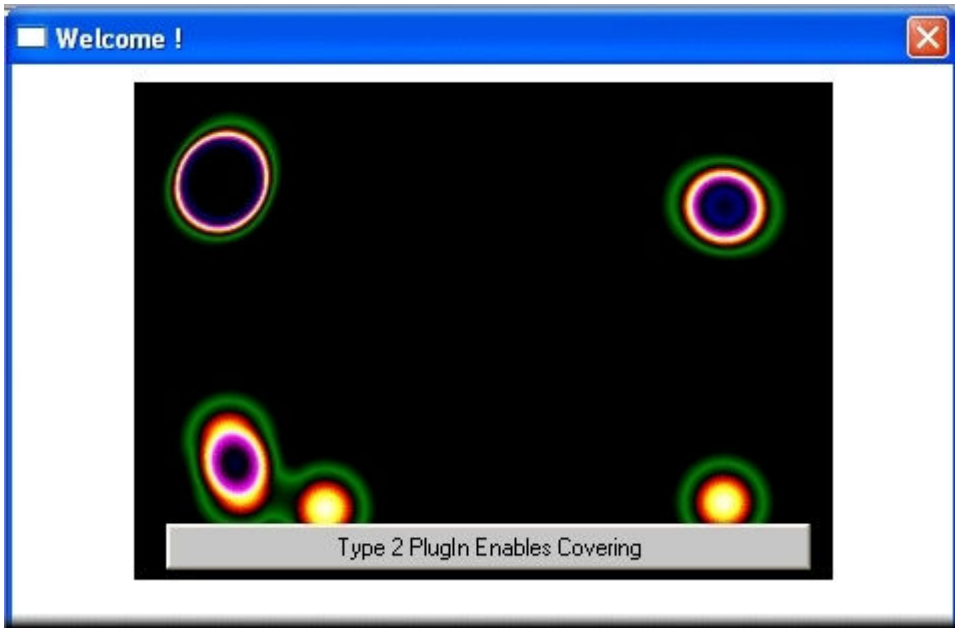
۳ حالت گفته شده زیربنای درک عملکرد Plug-In ها می‌باشند. همانطور که قبلاً گفته شد Plug-In های MMB بر ۲ نوع هستند: ۱- نوع غیربصری و قابل کنترل توسط فرامین ۲- نوع بصری

غالب Plug-In های MMB (نزدیک به ۹۸٪) ازنوع اول هستند که محدوده فعالیت آنها در قسمت پنجره تعبیه شده است. باقی فعالیتها توسط فرامین داخل Plug-In صورت می‌گیرد که یا منجر به تغییر حالت Plug-In یا ارسال خروجی به متغیرها می شود. این نکته حائز اهمیت است که Plug-In های ازنوع اول تقریباً مستقل از MMB عمل می‌کند و بر سایر اشیاء حاضر در پروژه هیچ تاثیر مستقیمی اعمال نمی‌کنند. عدم تعامل با سایر اشیاء ضعفی برای MMB شناخته می‌شود که شاید در نسخه‌های بعدی برطرف گردد.نکته‌ی دیگری که در خصوص نوع اول باید گفت اینست که نمی‌توان از یک Plug-In در ۲ پنجره استفاده کرد در مفهومی عام‌تر نمی توان



آنها را به اشـــتراک گذاشت. راه‌حل این نقیصه استفاده از Plug-In ۲ مشابه اما با نامهای مختلف است.

همانطورکه شـکل بـالا نـشان مـی‌دهـد دو Plug-In مـشابه بـا نامهـای مختلـف داریـم کـه ایـن نـام‌هـا در جـای خـود مهـم است، چـرا کـه در ادامـه توسـط همـین نامهـا و بـه مـدد از فـرامین از Plug-In هـا اسـتفاده مـی‌کنـیم. امـا نـوع دوم بـصری هستند که قبلاً راجع به آنها صحبت کردیم.شکل زیر یک نمونه از PlugInهای بصری است.



بگذارید پس از شرح قسمتهای مهم قبل به جمع‌بندی مطالبی که تا کنون آموختیم بپردازیم:

- Plug-In ها همه جا هستند.

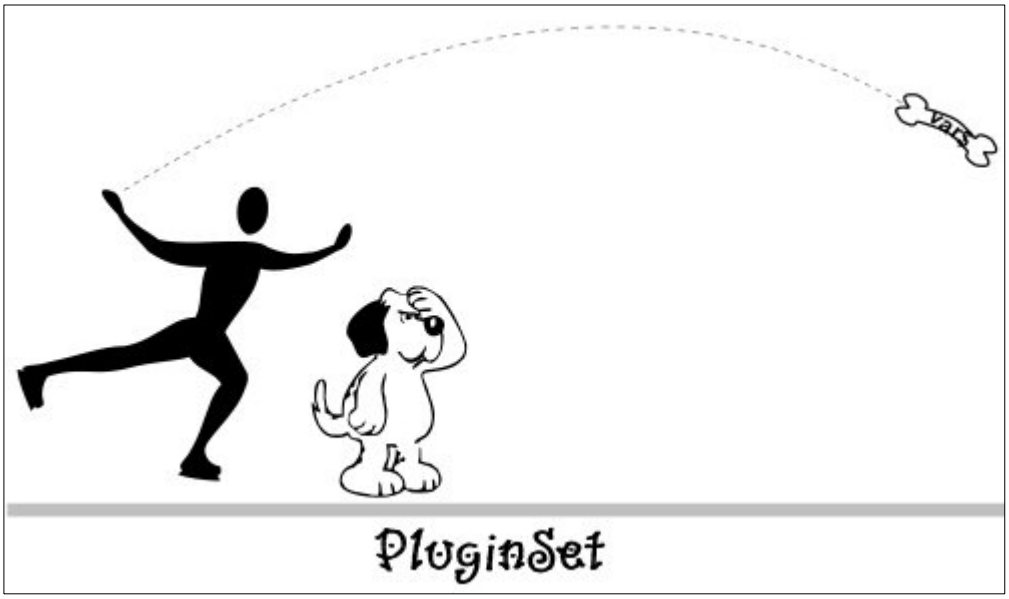
- قسمت‌هایی اضاف بر سازمان هستند.

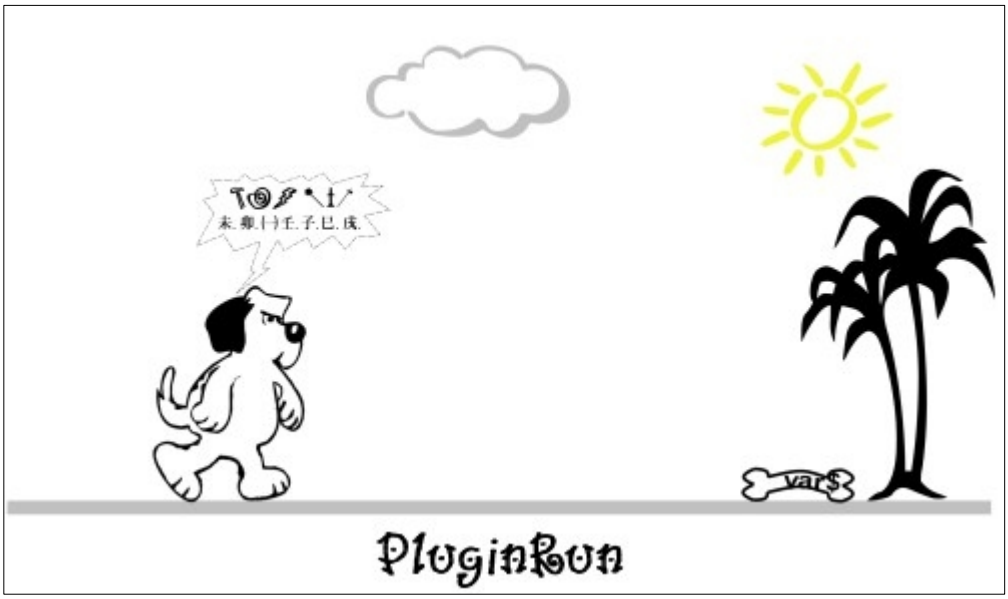
- هدف Plug-In ها افزایش قابلیت برنامه‌هاست.

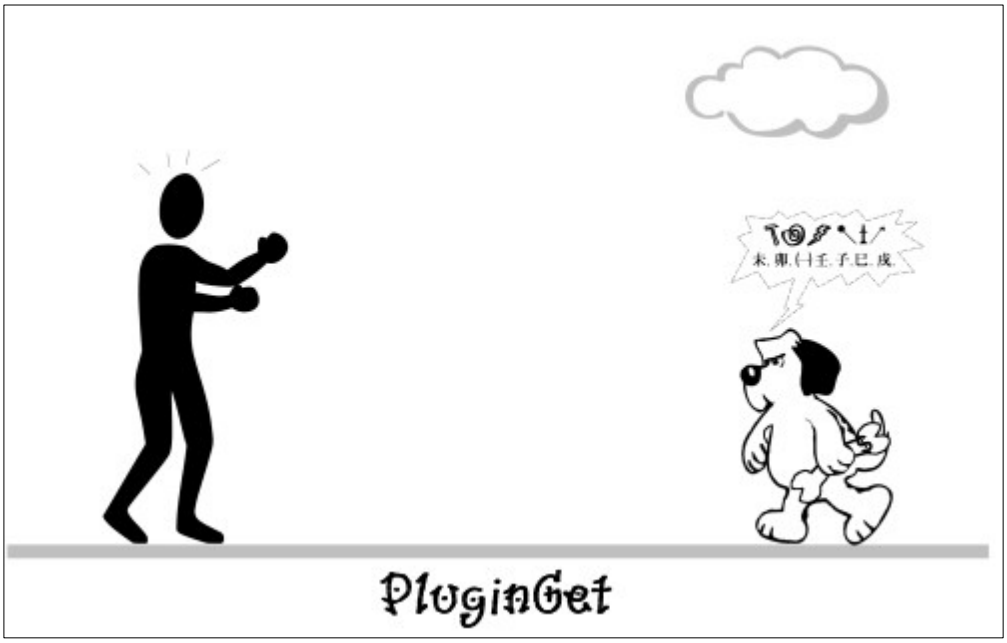
خب دیگر تئوری بس اسـت.بـه جاسـت کـه تمـرین را شـروع کنـیم. گفتـیم کـه تنظـیم خـصوصیات Plug-In هـای نـوع ۱ توسط اسکریپت نویـسی انجـام مـی‌شـود. در زبـان اسـکریپت‌نویـسی MMB، ۳ فرمـان مـرتبط بـا Plug-In هـا وجـود دارد که شـما از آنهـا جهـت ایجـاد ارتبـاط بـین MMB و Plug-In اسـتفاده مـی‌کنیـد. ایـن فـرامین بـه ترتیـب در جـدول زیـر بحث شده‌اند.

PlugInSet	ایـن فرمـان جهـت تنظـیم پـارامتر بـرای فـرامین Plug-In مـی‌باشـد. ایـن فرمـان پارامترهـا را کـه همـان متغیرهـای عـددی و رشـته‌ای هـستند را بـه فایـل Plug-In ارسـال مـی‌کنـد و Plug-In آنهـا را ذخیـره و بررسی میکند .هـر بـار اسـتفاده از ایـن دسـتور بـا یـک متغیـر کـه محتـوایش تغییـر کـرده سـبب جـایگزینی محتوی جدید در فایل Plug-In می‌شود.
PlugInRun	این فرمان توابـع و فـرامین تعبیـه شـده در فایـل Plug-In را براسـاس پارامترهـای تنظـیم شـده اجـرا میکنـد. چنانچـه بخواهیـد نـام فـرامین موجـود در Plug-In را بدانیـد بـه راهنمـای همـراه Plug-In مراجعـه کنیـد. البته همیـشه Plug-In هـا احتیـاج بـه پـارامتر ورودی ندارنـد و یـا در حـالاتی هـیچ مقـدار خروجـی تولیـد نمی‌کنند و تنها سبب تغییر حالت Plug-In و انجام سایر فعالیت‌ها می‌شود.
PlugInGet	ایـن فرمـان وظیفـه دریافـت مقـدار خروجـی کـه بـا اسـتفاده از فرمـان PlugInRun تولیـد شـده اسـت را برعهده دارد.پس طبیعتاً این فرمان پس از دستور PlugInRun استفاده میشود.

حال به تصاویر زیر دقت کنید که تفهیم مطلب را ساده‌تر می‌نماید.




var$
PluginRun


PluginGet

همانند سایر فرامین MMB این ۳ فرمان هم دارای چند قسمت هستند که در ادامه بحث می‌شوند.

PlugInSet	
در عمل	در تئوری
PlugInSet ("Plug-In” , “Variable")	PlugInSet
شرح	
در رابطه با خود PlugInSet که قبلاً صحبت شد.پارامتر اول همان نام شیء Plug-In را مشخص می‌کند که از کادر خصوصیت Plug-In قسمت label قابل تغییر است و به واسطه همین نام MMB می‌فهمد که با چه Plug-In ای رابطه برقرار کند. درقسمت Variable میتوانید از متغیری استفاده کنید و آن را به Plug-In ارسال کنید این متغیر هم میتواند عددی باشد هم رشته‌ای، نوع متغیر بسته به نوع فرمان است و می‌توانید از روی راهنمای Plug-In آن را تشخیص دهید.	
PlugInRun	
در عمل	در تئوری
PlugInRun(“Plug-In" , "Command")	PlugInRun
در رابطه با فرمان PlugInRun که بحث شد.پارامتر اول نام شیء Plug-In را مشخص می‌کند و پارامتر دوم فرمانی از فرامین Plug-In را اجرا می‌کند. جهت اطلاع از فرامین یک Plug-In بایستی به راهنمای ارائه شده همراه Plug-In رجوع کرد.ممکن است فرمان اجرا شده Plug-In خروجی به همراه داشته باشد که در ادامه میتوان توسط فرمان PlugInGet آن را دریافت کرد.	
PlugInGet	
در عمل	درتئوری
PlugInGet("Plug-In” , “Variable")	PlugInGet
شرح	
همان‌طور که می‌دانید از این فرمان جهت دریافت خروجی یک فرمان پیشتر اجرا شده استفاده می‌شود. قسمت اول این فرمان نام شیء Plug-In را مشخص می‌کند.قسمت دوم نام متغیری را مشخص می‌کند که حاوی مقدار خروجی است این متغیر هم می‌تواند عددی باشد وهم رشته‌ای.نوع متغیر را می‌توان با توجه به راهنمای همراه Plug-In مشخص ساخت.	

مثال

اکنون ۳ فرمان گفته شده را در کنار هم می‌آوریم تا بتوانیم از Plug-In استفاده کنیم.

```
var$ = “text”
PlugInSet(“Plug-In” , “ var$”)
PlugInRun(“Plug-In” ,  “uppercase”)
PlugInGet(“Plug-In” , “var$”)
Message(“out put is” , “var$”)
```

در بلـوک کـد بـالا ابتـدا محتـوی متغیـر $var توسـط فرمـان PlugInSet بـه Plug-In ارسـال مـی‌شـود.سـپس دسـتور PlugInRun بـــــــــــر اســـــــــــاس متغیـــــــــــر دریـــــــــــافتی دســـــــــــتور مربوطـــــــــــه (uppercase) را اجرا مـی‌کنـد ویـک خروجـی تولیـد مـی‌نمایـد. در ادامـه دسـتور PlugInGet مقـدار خروجـی را دریافـت کرده و به MMB انتقال مـی‌دهـد.کـه درآخـر میتـوان از ایـن مقـدار بـا اسـتفاده از سـایر فـرامین سـود جسـت. حالـت گفتـه شده گونه‌ای بـود کـه Plug-In خروجـی تولیـد کـرد امـا حـالتی پدیـد مـی‌آیـد کـه هـیچ خروجـی تولیـد نمـی‌شـود،یعنی اینکـه احتیـاجی بـه اسـتفاده از فرمـان PlugInGet نیـست.ایـن حالـت بـرای مثـال ممکـن اسـت بـرای مـواقعی کـه می‌خـواهیم انـدازه یـک فونـت را تغییـر دهـیم پدیـد آیـد. چیزهـایی کـه نیـاز اسـت ،یکـی انـدازه نهـایی فونـت و فرمـانی جهت اجرای آن است و هیچ نیازی به خروجی نمی‌باشد.این گونه از ارتباط همانند مثال زیر می‌باشد.

مثال:

```
fontsize = 16
PlugInSet("Plug-In" , " fontsize")
PlugInRun("Plug-In" ,"Set fonts size")
```

** توجه کنید که فعلاً تمام این Plug-In ها فرضی و مجازی هستند.

ممکـن اسـت گونـه‌هـای مختلفـی در اسـتفاده از فـرامین MMB بـرای ارتبـاط بـا Plug-In هـا پدیـد آیـد و ایـن بـستگی به نوع فرمان Plug-In دارد که می‌توان از طریق راهنمای Plug-In آن را مشخص ساخت.

بگذاریـد چـون صـحبت دربـاره راهنمـای Plug-In شـد،قدری راجـع بـه آن صـحبت کنـیم. هماننـد سـایر راهنماهـا، راهنمای Plug-In هـا هـم بـسته بـه نیـاز حـاوی تعـدادی مثـال هـستند کـه ایـن مثالهـا هـدف اصـلی هـر دسـتور را نـشان می‌دهد و همیشه نمی‌تـوان بـا کپـی و چـسابدن آنهـا از راهنمـا بـه ویرایـشگر اسـکریپت انتظـار عملکـرد صـحیح را از آنهـا داشت.گاهی اوقات لازم است که با توجه به خطـوط قبلـی کـد برنامـه و براسـاس نیازمـان کـد ایـن مثـال‌هـا را تغییـر دهـیم و نهایتاً از آنها استفاده کنیم. نکتـه‌ی دیگـری را کـه بایـد یـادآور شـد ایـن اسـت کـه در غالـب راهنماهـا بـر چـسب -Plug In ، PlugIn در نظر گرفته مـی‌شـود کـه چنانچـه بخـواهیم نـام دیگـری داشـته باشـیم بایـد در فـرامین مـرتبط بـا Plug-In این برچسب‌ها را تغییر دهیم.

اما در رابطه با حالتهای به کارگیری ۳ دستور قبل صـحبت مـی‌کـردیم.حالـت‌هـای دیگـری هـم کـه ممکـن اسـت پدیـد آیـد به صورت زیر است:

الف) استفاده از یک دستور بدون هیچ ورودی و خروجی

ب) استفاده از یک دستور بدون مقدار ورودی

حالت الف ممکن است مواقعی پیش بیاید که فرمان ما تنها عملی را بایستی انجام دهد برای مثال ویندوز را Shutdown کند.

حالت ب ممکن است مواقعی پیش بیاید که فرمان Plug-In مستقلاً عمل کرده و مقدار خروجی تولید کند. مثلا هنگامی که بخواهیم ساعت و تاریخ را دریافت کنیم.اگر پا را فراتر ازاین حالات بگذاریم حالت‌های دیگری به وجود می‌آید که خود از ترکیب چند حالت قبل پدید می‌آید،همانند مثال زیر:

بلوک ۱

```
var$ = "My computer is P4"
PlugInSet("Plug-In" , "var$")
PlugInRun("Plug-In" , "upper case")
PlugInGet("Plug-In" , "text$")
```

بلوک ۲

```
PlugInSet("Plug-In" , "text$")
PlugInRun ("Plug-In" , "Show message box")
```

در مثال قبل ابتدا رشته در بلوک ۱ به حالت بزرگ تبدیل شده و به بلوک ۲ ارسال می‌شود تا در کادر پیغام نشان داده شود.

رابطه‌ای Plug-In و متغیر تنها به یک انتساب ساده ازقبیل var$=myname منتهی نمی‌شود.بلکه گاهی اوقات پیش می‌آید که این رشته‌ها توسط کاراکتری از هم جدا می‌شوند مثلاً برای تعیین مقدار رنگ:

color$ = "220,150,105"

این گونه‌ها بر اساس تعاریفی که از پیش در Plug-In به وسیله‌ی مؤلف آن آمده است پردازش و تجزیه می‌شود.حالت‌های دیگری از این قبیل انتساب‌ها هم وجود دارند. مثال:

Regkey $ = 'software\program\section\keyname'

Flag$ = 'con|yes no cancel|caption'

کنترل رویداد (Event Handing)

وقایع (Event): اتفاقات و توسعه‌هایی هستند که در Plug-In رخ می‌دهند و به برنامه شما اطلاع داده می شوند.

کنترل کننده رویداد: شیءهای اسکریپت MMB هستند که همانند نوعی شبکه وقایع را دریافت می‌کند.

همانطور که قبلاً خواندید بعضی از فرامین مقادیر خروجی تولید می کنند اما برخی از Plug-In ها بلافاصله پس از دستور PlugInRun مقدار خروجی تولید نمی کنند. وظایفی از قبیل پردازش داده‌ها ، وقفه‌های طولانی، رویدادهای دوره‌ای و ... سبب بروز چنین حالتی می شوند. خب در این صورت سؤال اینجاست که Plug-In برنامه MMB چگونه در این حالت با MMB به تعامل می‌پردازد؟ جواب اینجاست که خود Plug-In فشردن کلید یا کلیدهایی را جهت ارسال اطلاعات به MMB شبیه‌سازی می کند. به شکلهای زیرتوجه کنید:

شکل ۱: فرستاده شدن رویداد از Plug-In را بیان می‌کند.

شکل ۲: شبیه‌سازی کلیدها را تداعی می‌کند.

شکل ۳: نحوه در جریان قرار گرفتن برنامه ما را نشان می‌دهد.

شکل ٤: دریافت واقعه را توسط شیء اسکریپتی که برای این واقعه تنظیم شده است را نشان می‌دهد.

می‌بینیم که شیء اسکریپت برای حالت فشردن کلیدهای Alt + Control + Shift + D تنظیم شده است و چنانچه ما از پیش برای Plug-In این ترکیب را تعریف کرده باشیم هنگام بروز یک رویداد برای حالت‌های خاص گفته شده شیء اسکریپت فراخوانی می‌شود و فرامین داخل شیء اسکریپت اجرا می‌شود. این نکته را باید دانست که کلیدهای ترکیبی نبایستی کلیدهایی باشند که از پیش برای اعمال مختلف در سیستم عامل در نظر گرفته شده‌اند باشند.

همچنین بایستی به بزرگی و کوچکی حروف و روشن بودن کلید Caps Lock توجه داشت.اما برای استفاده از Event Handling بایستی استفاده از راهنمای Plug-In را فراموش نکرد.

توجه: چنانچه در هنگام ارسال کلیدها برنامه شما در حالت فعال نباشند ممکن است که این شبیه‌سازی فشردن کلیدها به برنامه‌ای دیگر فرستاده شود. به عنوان مثال هنگامی که برنامه یک کار تبادلی را احضار می‌کند دیگر کانون(Focus) روی برنامه اصلی نیست بلکه بر روی کادر تبادلی است، راه‌حل این مشکل غیر فعال‌سازی عمل Event Handling در حین نمایان بودن کادر تبادلی است. البته اگر ویژگی غیرفعال سازی برای Plug-In تعریف شده باشد.همچنین استفاده مکرر ازکنترل رویداد ممکن است تداخل ایجاد کند که می‌تواند سبب بروز اشکال در اجرای برنامه شود. بر همین اساس گفته می‌شود که در نسخه‌های بعدی MMB این مشکل حل خواهد شد.

خلاصه‌ای از چگونگی Plug-In های MMB:

- Plug-In های MMB همانند Plug-In های برنامه‌های مختلف دیگر نیستند.

- یک پنجره ۲ بعدی است که گاهی بصری است و گاهی نیست.

- برای نوع اول در صورت نیاز به چندین پنجره و چندین نسخه نیازداریم.

- برای نوع اول تمام تنظیمات و پردازش‌ها از طریق اسکریپت‌های MMB صورت می‌گیرد.

- حالت‌های مختلفی جهت استفاده از ۳ دستورPlugInGet, PlugInRun PlugInSet وجود دارد که برحسب نیاز ممکن است یکی یا دو تا از این فرامین حذف شوند.

- عمل Event Handling هنگامی که به کار می‌آید که یکسری عملیات‌های تأخیری رخ دهد.

- فایل‌های Plug-In برنامه MMB از سایتهای مربوطه یا سایت مؤلف قابل دریافت هستند.

حال که به انتهای این بحث رسیدید دنیایی باور نکردنی و جالب را در پیش روی خود خواهید دید،دنیایی که روز به روز در حال توسعه و تکامل است و آن چیزی نیست جز دنیای کاربران MMB.

Tweak Plug-In

ایـن PlugIn یکـی از قـدیمی‌تـرین و در عـین حـال کـاراترین PlugIn بـرای MMB مـی‌باشـد کـه در ادامـه مـی‌خـواهیم تنها به شرح ویژگی‌ها و فرامین آن بپردازیم.

- Change video mode

فرامین ارائه شده در این قسمت برای تغییر پارامترهای « ویدوئی » هستند.

VideoModeChange-

جهـت تعـویض مـد « ویـدوئی » مـی‌باشـد کـه ایـن مـد را مـی‌تـوانیم توسـط یـک متغیـر بـه PlugIn ارائـه کنـیم. فرمـت محتوی متغیر باید به گونه‌ی زیر باشد.

' WIDTH | HEIGHT | BPP '

(بیت برای هر پیکسل ، ارتفاع ، عرض)

مثال:

```
Vidmode$ = ' 640 | 480 | 8 '
PlugInSet(''Tweak'' , ''Vidmode$'')
PlugInRun( ''Tweak'' , ''VideoModeChange'')
```

مثال بالا سبب تغییر مد به حالتی می‌شود که عرض 640 ،ارتفاع 480 و BBP 6 می‌شود.

VideoModeRestore-

این فرمان سبب بازگشت مد ویدئویی به تنظیمات پیش فرض می‌شود.

مثال:

```
PlugInRun( ''Tweak'' , ''VideoModeRestore'')
```

Dialoge Box –

این قسمت برای نمایان ساختن کادرهای تبادلی می باشد.

FDSetDefualt –

این فرمان پوشه پیش فرش را برای کادر بازگشایی معین می‌کند.

FDSetFilter –

این فرمان فیلتر را برای کادر بازگشایی معین می‌کند.

FDOpenFileDialoge -

این فرمان کادر بازگشایی را نشان می‌دهد.

FDSaveFileDialoge-

این فرمان کادر ذخیره فایل را نشان می‌دهد.

FDGetFileName-

این فرمان مسیر کامل فایل انتخاب شده را نشان می‌دهد.

FDGetFilePath-

این فرمان فقط نام فایل انتخاب شده بدون احتساب مسیر را نشان می‌دهد.

مثال:

```
Filter$='Text Files (*.txt)|*.All Files|*.*||'
PlugInSet("Tweak","Filter$")
PlugInRun("Tweak","FDSetFilter")
DefualtFolder$='C:\My ocuments\\'
```

```
PlugInSet("Tweak","DefualtFolder$")
PlugInRun("Tweak","FDOpenFileDialog")
PlugInRun("Tweak","FDGetFileName")
PlugInGet("Tweak","Path$")
Message("Selected File is: ","Path$")
```

Message Box –

این ویژگی این امکان را فراهم می‌کند تا بتوانیم کادر پیغام داشته باشیم. البته MMB خود کادر پیغام دارد،اما شما نمی‌توانید کنترل کاملی بر آن داشته باشید. این PlugIn کادر پیغام را با جزئیات کامل که تحت کنترل شماست را ارائه می‌دهد.

MsgBox-

ابتدا بایستی مقدار ورودی را با متغیری جهت ارائه به PlugIn تعیین کرد. برای این کار متغیری رشته‌ای را انتخاب و عمل انتساب را به صورت زیر انجام می‌دهیم:

Mes$ = ' Message text | Title bar text | Flags '

Message text : متن پیغام را مشخص می‌کند. چنانچه بخواهید متن را در چند خط بگنجانید می‌توانید با استفاده از عبارت \n خط جدیدی تولید و ادامه متن را پس از این عبارت بیاورید. اما عبارت \t هم فاصله‌ای به اندازه ۸ کاراکتر قبل از کلمه بعد از خود ایجاد می‌کند.

Title bar text: این قسمت برای معین ساختن عنوان کادر پیغام می‌باشد.

Flags : این قسمت جهت مشخص ساختن دکمه‌های کادر پیغام و نوع آنها است.

نکته: توجه کنید که چنانچه در متون خود در هنگام استفاده از این فرمان نیاز به استفاده از کاراکتر " \ " را دارید بایستی از " \\ " استفاده کنید برای مثال‘Path is C:\\ test’ به ‘Path is C:\ test’ ترجمه خواهد شد. اما جدول زیر Flag های موجود را ارائه می‌دهد:

B Flag سبب ایجاد دکمه می‌شود:	
دکمه‌ی Ok (به صورت پیش فرض)	Bo
دکمه‌های Cancel,Ok	Boc

دکمه‌های No.Yes	Byn
دکمه‌های Cancel,No,Yes	Bync
دکمه‌های Cancel,Retry	Brc
دکمه‌های Ignore,Retry,Abort	Bari
I Flag سبب قرار گیری آیکن بر روی کادر می‌شود	
سبب ظاهر شدن آیکن تعجب می‌شود	Ie
یک آیکن حاوی حرف I انگلیسی در کادر ظاهر می‌شود.	Ii
آیکن پرسش در کادر نمایان می‌شود	Iq
آیکن ایست را در کادر نمایان می‌سازد.	is
D Flag کنترل می‌کند در هنگام ظهور کادر کدام دکمه به صورت پیش فرض فعال باشد.	
کانون (Focus) به دکمه اول منتقل می‌شود. (حالت پیش فرض)	D1
کانون به دکمه دوم منتقل می‌شود.	D2
کانون به دکمه سوم منتقل می‌شود.	D3
M Flag رفتار کادر پیغام را مشخص می‌کند.	
این حالت کادر پیغام را نمایان می‌سازد و کاربر بایستی به آن پاسخ دهد اما کابر می‌تواند در حین فعال بودن کادر به سراغ سایر پنجره‌ها هم برود.	ma
این حالت کادر پیغام را نمایان می‌سازد که کاربر باید به آن واکنش نشان دهد و در حین نمایان بودن این کادر تمامی پنجره‌های دیگر غیر فعال و غیر قابل دسترس است. از این کادر هنگامی باید استفاده کنیم که بخواهیم به کاربر پیغامی جدی و حیاتی را منتقل کنیم.	Ms

پس از نمایش کادر پیغام می‌توانیم با استفاده از دستور PlugInGet بفهمیم که کاربر چه کلمه‌ای را فشار داده است. در جدول زیر مقدارهای بازگشتی به همراه مفاهیم آنها آمده است.

مفهوم	مقدار بازگشتی
Ok	O
Cancel	C
Yes	Y
No	N
Abort	A
Retry	R
Ignore	I

مثال:

```
Message$ = ' Would you like to register? | My App | byn + ig '
PlugInSet(''Tweak'' , ''Message$'')
PlugInRun(''Tweak'' ,''MsgBox'')
PlugInGet(''Tweak'' , ''Selected$'')
If ( Selected$= 'y'  ) then
Message(''You pressed yes ''  , '' '')
Else
Selected$ = ' n '
Massage ( ''You pressed no ''  , '' '' )
End
```

مثال بالا یک کادر پیغام با ۲ دکمه Yes و No را نشان می‌دهد و در ادامه دکمه انتخاب شده را باز می‌گرداند.

Registry I\O

این قسمت قابلیت ایجاد،خواندن و پاک کردن کلیدهای رجستری را فراهم می‌کند.

RegistrySetRootKey –

این فرمان جهت مشخص کردن ریشه به کار می‌رود. ابتدا، بایستی یکی از عبارت‌های زیر را بر حسب نیاز به متغیری رشته‌ای جهت تعیین ریشه نسبت دهیم:

-HKEY – CLASSES – ROOT
-HKEY – CURRENT – CONFIG
-HKEY – CURRENT – USER
-HKEY – LOCAL – MACHINE
-HKEY – USERS

سپس توسط فرمان PlugInSet عبارت مورد نظر را به PlugIn ارسال می‌کنیم به صورت زیر:

```
rootkey$ = ' HKEY – LOCAL – MACHINE '
PlugInSet(''Tweak'' , ''rootkey$'')
PlugInRun(''Tweak'' , ''RegistrySetRootKey'')
```

نکته: ریشه پیش فرض در هنگام بارگذاری فایل PlugIn ، HKEY- CURRENT-USER است.

RegistryCreateKey–

این فرمان برای ایجاد کلید در رجستری مورد استفاده قرار می‌گیرد. و این قابلیت را دارد تا همزمان چند کلید تولید نماید. به عنوان مثال چنانچه محتوی متغیر ورودی به گونه‌ی زیر باشد:

Software\\MyMMBApp\\Setting

PlugIn همزمان دو کلید Setting و MyMMBApp را تولید می‌نماید. جهت استفاده از این فرمان ابتدا مسیر و نام کلید را با استفاده از PlugInSet به PlugIn ارسال و توسط فرمان PlugInRun کلید را تولید می‌کنیم. حال جهت درک بیشتر به مثال زیر توجه کنید:

```
Newkey$= 'Software\MyMMBApp\MySetting
PlugInSet( "Tweak" , "Newkeys$")
PlugInRun("Tweak" , "RegistryCreateKey")
```

مثال بالای تنها یک کلید تولید می‌نماید. باید توجه داشت که قبل از به کارگیری دستور RegistryCreateKey بایستی ریشه را توسط فرمان RegistrySetRootKey معین کرد.

RegistryDeleteKey-

برای پاک کردن کلید از رجستری از این فرمان استفاده می‌شود. البته در استفاده از این فرمان باید احتیاط کرد،چرا که استفاده نا به جا و غلط ممکن است به رجستری آسیب رساند و سبب خرابی و ایجاد اشکال جدی در ویندوز شود.برای استفاده ابتدا باید مسیر را توسط متغیری با فرمان PlugInSet به PlugIn ارسال کنیم. در ادامه با دستور PlugInRun کلید را حذف می‌کنیم.

مثال:

```
delkey$= 'Softwear\MyAppkey\MySetting'
PlugInSet("Tweak" , "delkey$")
PlugInRun("Tweak ", "RegistryDeleteKey")
```

مثال بالا کلید MySetting را از مسیر مربوطه حذف می‌کند. البته در این قسمت هم بایستی از دستور RegistrySetRootKey استفاده کرد.

RegistrySetValue-

این فرمان مقدار یک کلید را تنظیم می‌کند چنانچه کلید داده شده موجود نباشد این فرمان آن کلید را ایجاد می‌کند.ابتدا بایستی با استفاده از حالت زیر محتوی متغیر را مشخص کرد:

' Full path to the registry key | Value name | Value to set '

قسمت اول سمت چپ همان مسیر کلید در رجستری است. قسمت دوم نام متغیر در رجستری است و قسمت سوم مقدار جهت تنظیم در رجستری است. چنانچه بخواهیم مقدار متغیرپیش فرض(Default) را در رجستری تنظیم کنیم تنها کافیست تا قسمت دوم را خالی بگذاریم و از نوشتن آن صرف نظر کنیم.

```
rootkey $ = 'HKEY - CURRENT - USER '
PlugInSet("Tweak" , "rootkey$")
PlugInRun("Tweak", "RegistrySetRootKey")
SetValCmd$='ControlPanel\Desktop\wallpaper|C:\Windows\Bubbls.BMP'
PlugInSet("Tweak" , "SetValCmd$")
PlugInRun ("Tweak", "RegistrySetVal")
```

مثال بالا عکس پس زمینه دسکتاپ را به عکس مورد نظر تغییر می‌دهد توجه کنید که در اینجا نام متغیر wallpaper است که با کاراکتر "|" جدا شده است و قسمت بعدی محتوی متغیر در رجستری است.

RegistryGetValue-

این فرمان محتوی یک کلید رجستری یا یک متغیر رجستری را دریافت می‌کند. جهت استفاده از این فرمان ابتدا بایستی با استفاده از قالب زیر مسیر کلید را به PlugIn ارسال داشت.

' Full path to the registry key | Value name '

قسمت اول مسیر کامل کلید رجستری را معین می‌کند و قسمت دوم نام متغیر رجستری را.چنانچه بخواهید مقدار متغیر پیش فرض یک کلید را بگیرید کافیست از نوشتن قسمت دوم پرهیز کنید.

مثال:

```
rootkey$ = ' HEKY - CURRENT - USER '
PlugInSet("Tweak" , "rootkey$")
PlugInRun("Tweak", "RegistrySetRootKey")
SetValCmd$ = 'ControlPanel\Desktop|wallpaper'
PlugInSet( "Tweak" , "SetValCmd$")
PlugInRun("Tweak", "RegistryGetValue")
PlugInGet("Tweak", "wallpaper$")
Message("","wallpaper$")
```

مثال بالا مسیر عکس پس زمینه ویندوز را که در یک متغیر در رجستری ویندوز است را باز می‌گرداند.

Tray Icon –

این قسمت این امکـان را فـراهم مـی‌کنـد تـا برنامـه در حـال اجـرا بـهSystemTrayمنتقـل شـود.همچنـین مـی‌تـوان آیکنـی رابرای برنامـه بـه هنگـام رفـتن بـه SysTray تعیـین کـرد و یـا آن را بـه‌روزرسـانی کـرد. البتـه شـایان ذکـر اسـت کـه ایـن فرمان در نسخه 4.9.5 برنامه MMBبه درستی عمل نمی‌کند.

MinimizeToTrayIcon-

این دستور آیکن برنامه هنگامی که به SysTray می‌رود را تعیین می‌کند.

مثال:

```
IconPath$ = ' < Emhedded >\Globe.Ico'
PlugInSet( "Tweak" , "IconPath$")
PlugInRun("Tweak", "MinimizeToTrayIcon")
```

UpdateTrayIconTooltip-

این فرمان Tooltip برنامه را هنگامی که به Systray منتقل شده است را مشخص می‌کند.

مثال:

```
Tooltip$=' My MMB App'
PlugInSet("Tweak" , "Tooltip$")
PlugInRun("Tweak", "UpdateTrayIconTooltip")
```

UpdateTrayIcon-

برای به‌روزرسانی آیکن برنامه در حالت Minimize در Systray بایستی از این دستور استفاده کرد.

مثال:

```
IconPath$ = ' < Embedded > \NewIcon.ico'
PlugInSet("Tweak" , "IconPath$")
PlugInRun("Tweak", "UpdateTrayIcon")
```

TrayIconIsMinimize-

این دستور کنترل می‌کند که آیا برنامه به حالت Minimize شده در Systray است یا خیر.

مثال:

```
PlugInRun("Tweak", "TrayIconIsMinimize")
PlugInGet("Tweak", "IsMinimize")
```

مثال بالا ۲ مقدار ۱ به معنی Minimize بودن و صفر را به معنی عدم Minimize بودن باز می‌گرداند.

Chang Shape -

یکی از قسمتهای جالب Tweak همین ویژگی است که به ما امکان می‌دهد شکل صفحه و پروژه را در زمان اجرا تغییر دهیم. البته از قبل بایستی با برنامه‌ای به نام CreateRegion که در CD همراه کتاب است،شکل را ایجاد و در فایلی مخصوص ذخیره نماییم و سپس با دادن مسیر آن فایل به PlugIn می‌توانیم شمایل پنجره پروژه را تغییر دهیم.

SetRegion-

فرمانی است که عمل تغییر شکل را صورت می‌دهد.

ResetRegion-

شکل پنجره را به حالت اول باز می‌گرداند.

مثال:

```
Region$ = ' < Embedded >\MyRegion.rgn'
PlugInSet("Tweak" , "Region$")
PlugInRun("Tweak", "SetRegion")
```

Detect App Instances -

چنانچه بخواهیم بدانیم که چند نسخه از برنامه ما در حال اجراست بایستی از این فرمان استفاده کرد. تنها کافیست که مسیر فایل اجرایی برنامه مورد نظر را به PlugIn ارسال کنیم.

GetNumInstances-

تعداد نسخه‌های در حال اجرا را باز می‌گرداند.

مثال:

```
appPath$ = '< SrcDir>\AutoRun.exe'
PlugInSet("Tweak" , "appPath$")
PlugInRun("Tweak", "GetNumInstances")
PlugInGet("Tweak" , "Instances")
Message("" , "Instances")
```

نکته: یکی از مواقعی که ممکن است بخواهیم از این تابع استفاده کنیم هنگامی است که تنها می‌خواهیم یک نسخه از برنامه ما در آن واحد اجرا شود.

Detect CPU Vender and Speed

این قسمت نام کارخانه سازنده پردازشگر و سرعت آن را باز می‌گرداند.

CPUGetName-

این تابع نام سازنده پردازشگر را باز می‌گرداند.

CPUGetSpeed-

این تابع سرعت CPU را باز می‌گرداند.

مثال:

```
PlugInRun("Tweak", "CPUGetName")
PlugInGet( "Tweak" , "CPUName$")
PlugInRun("Tweak", "CPUGetSpeed")
PlugInGet("Tweak" , "CPUSpeed$")
Info$='Your CPU is: '+CPUName$ + 'Running at '+ CPUSpeed$
Message("" , "Info$")
```

Manipulate INI Files

ایـن فرمـان هـم یکـی از قـسمت‌هـای پرکـاربرد Tweak اسـت کـه بـرای خوانـدن و نوشـتن در فایـل‌هـای INI بـه کـار می‌رود.

IniSetFile-

این فرمان فایل ini را که باید خوانده و یا درآن نوشته شودرا مشخص می‌کند.

IniRead-

این فرمان اطلاعات را از فایل ini می‌خواند.

IniWrite-

این فرمان اطلاعات را در فایل ini می‌نویسد.

IniDelete-

این فرمان یک قسمت از اطلاعات فایل ini را حذف می‌کند.

توجه کنید که فرمـت ini یـک پـسوند اسـتاندارد ویـنـدوز اسـت کـه نحـوه ذخیـره‌سـازی اطلاعـات در آن بـه صـورت زیـر است:

[Section Name] ⟵ نام قسمت

Name = MMB ⟵ اطلاعات که مقادیر آنها به

Ver=4.9.5 نسبت داده شده است.

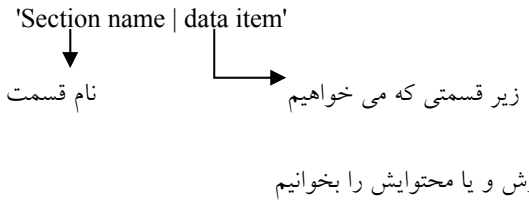
توجه: مزیت فایل‌های INI در این اسـت کـه عملیـات‌هـای خوانـدن، نوشـتن و حـذف تمامـاً در حافظـه صـورت مـی‌گیـرد. و بدین ترتیـب سـبب بهبـودی و توسـعه عملیـات‌هـای خوانـدن و نوشـتن و حـذف داده‌هـا مـی‌شـود و مـادامی کـه برنامـه شما در حال اجـرا اسـت اطلاعـات جدیـد در فایـل ذخیـره نمـی‌شـود و بـه محـض خـروج از برنامـه و بـسته شـدن فایـل ini اطلاعات در فایل ذخیـره مـی‌شـود. خـوبی دیگـر ایـن نـوع فایـل هـا ایـن اسـت کـه بـه سـادگی امکـان ویـرایش آنهـا توسط برنامه Notepad ویندوز فراهم است.

مثال:

```
iniFilePath$ = '< Embedded >\MyApp.ini'
PlugInSet( "Tweak" , "iniFilePath$")
PlugInRun("Tweak", "IniSetFile")
```

باید توجه داشت که چنانچه فایل مسیر دهی شده وجود خارجی نداشته باشد این دستور آن را ایجاد می‌کند.

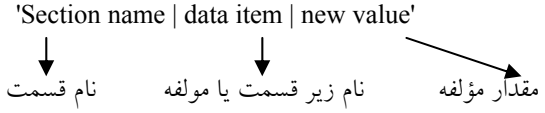
ابتدا بایستی بر طبق فرمت زیر قسمتی را که مایلیم اطلاعات ازآن خوانده شود را مشخص کنیم.



مثال:

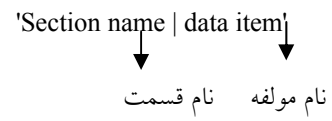
```
args$ = 'name|ver'
PlugInSet("Tweak" , "args$")
PlugInRun("Tweak", "IniRead")
PlugInGet("Tweak" , "data$")
Message("", "data$")
```

اما برای نوشتن در فایل ini ابتدا بایستی همانند فرمت زیر مقدار و نام مؤلفه و نام قسمت را به PlugIn ارسال کنیم.




```
args$ = 'name|ver|4.9.5'
PlugInSet("Tweak" , "args$")
PlugInRun("Tweak", "IniWrite")
```

مجدداً برای حذف یک قسمت فایل ini هم بایستی به گونه‌ی زیر عمل کرد:



و اگر بخواهیم تنها یک زیر قسمت را حذف کنیم تنها کافیست تا نام آن را به PlugIn ارسال کنیم.

```
args$ = 'name |ver'
PlugInSet("Tweak" ,"args$")
PlugInRun("Tweak", "IniDelete")
```

و برای حذف یک قسمت (Section):

```
args$ = 'name '
PlugInSet("Tweak" , "args$")
PlugInRun ("Tweak", "IniDelete")
```

Pop-up Menu –

این قسمت برای تولید منوهای جهنده (Pop-up) می‌باشد. باید گفت که هر برنامه تنها می‌تواند یک منوی جهنده داشته باشد. ممکن است این منوهای جهنده در هنگام طراحی به درستی عمل نکنند اما خوشبختانه در فایل کامپایل شده نهایی به درستی عمل می‌کنند.

MenuReset-

این دستور تمام مؤلفه‌های منوی جهنده جاری را از بین می‌برد. این فرمان هنگامی مفید است که بخواهیم منوی جهنده قبلی را از بین ببریم و یک منوی جهنده جدید تولید کنیم.

مثال:

```
PlugInRun("Tweak" , "MenuReset")
```

MenuAddItem-

برای افزودن مؤلفه‌ها به یک منـوی جهنـده از ایـن فرمـان اسـتفاده مـی‌کنـیم و مکـرراً بـرای افـزودن هـر مؤلفـه بایـد از ایـن دستور استفاده کرد. همچنین علاوه بـر مؤلفـه‌هـا مـی‌تـوان جـدا کننـده‌هـا (Seprators) را بـا اسـتفاده از کـاراکتر "- " بـه منوها اضاف کرد.

مثال:

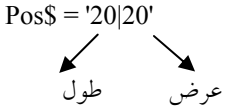
```
MenuItem$ = 'This is Item # 1'
PlugInSet("Tweak" , "MenuItem$")
PlugInRun("Tweak", "MenuAddItem")
MenuItem$ = 'This is Item # 2'
PlugInSet("Tweak" , "MenuItem$")
PlugInRun("Tweak", "MenuAddItem ")
MenuItem$ = ' - '
PlugInSet("Tweak" , "MenuItem$")
PlugInRun("Tweak", "MenuAddItem")
MenuItem$ = 'Exit'
PlugInSet( "Tweak" , "MenuItem$")
PlugInRun("Tweak", "MenuAddItem")
```

مثال بالا ابتدا ۲ مؤلفه ایجاد می‌کند،سپس یک جدا کننده تولید و نهایتاً گزینه Exit را به منو اضاف می‌کند.

MenuShow-

وقتی که مولفه‌های یک منو را اضاف مـی‌کنـیم بایـستی بـا اسـتفاده از ایـن دسـتور جهـت نـشان دادن منـو عمـل کنـیم. البتـه می‌توان بـه PlugIn مختـصات نقطـه‌ای را کـه مـی‌خـواهیم منـو از آنجـا ظـاهر شـودرا بـدهیم. جهـت اینکـه دریـابیم کـدام مؤلفه‌ی منـو کلیـک شـده مـی‌تـوان بـا PlugInGet("Tweak", "Selected") شـماره مؤلفـه را دریافـت کـرد. اگـر شـماره دریافتی صفرباشد به معنای عـدم کلیـک اسـت و بـاقی اعـداد بـین ۱ تـا شـماره آخـرین مؤلفـه بـه معنـای مؤلفـه‌ی مربوطـه است.باید توجه کنیم که جداکننده‌ها در محدوده شماره قرار نمی‌گیرند چرا که قابل کلیک کردن نیستند.

مثال:




```
PlugInSet( "Tweak" , "Pos$")
PlugInRun("Tweak", "MenuShow")
```

مثال بالا باعث نمایان شدن منو در مختصات (20,20) می شود.

مثال:

```
PlugInRun("Tweak" , "MenuShow$")
PlugInGet("Tweak", "Selected")
```

مثال بیان شده شماره قسمت کلیک شده را باز می‌گرداند.

MenuSetHandler-

اما حالتی است که می‌خواهیم منوی ما با عمل راست کلیک نمایان شود و یا به هنگام انتخاب مؤلفه‌ای شی اسکریپت مربوطه فراخوانی شود. ابتدا منو را ایجاد کنید و سپس کنترل کننده آن را تنظیم کنید. سپس کاربر با انتخاب یک مولفه از منوی جهنده سبب فراخوانی شی اسکریپت می‌شود. البته بایدبه PlugIn گفت که چه کلیدهایی سبب فعال شدن شی اسکریپت می‌شود.

```
Msg$ = 'CONTROL + SHIFT + ALT + A'
PlugInSet("Tweak" , "Msg$")
PlugInRun("Tweak","MenuSetHandler")
```

یادآوری می‌کنم که بایستی همان کلید یا کلیدهایی که برای PlugIn تعریف می‌شود برای شی اسکریپت هم تعریف شود.

نکته: جهت اضافه کردن یک منوی جهنده به آیکن برنامه هنگامی که درSystray قرار گرفته بایستی قبل از Minimize شدن از دستور MenuSetHandler استفاده کرد.

MenuClearHandler-

این دستور برای پاک کردن کنترل منوی جهنده است و با اعمال این دستور منوی جهنده با عمل راست کلیک ویا فشردن کلیدهای معین شده به عنوان کنترل کننده نمایان نخواهد شد.

مثال:

```
PlugInRun("Tweak","MenuClearHandler")
```

Client Area Drag

حتماً توجه کرده‌اید که برخی از برنامه‌ها بـرای عمـل درگ کـردن تنهـا از نـوار عنـوان خـود اسـتفاده مـی‌کننـد.بـا اسـتفاده از این فرمان می‌توان کل سطح پنجره را قابل درگ کرد.

EnableClientAreaDrag-

مثال:

```
PlugInRun("Tweak","EnableClientAreaDrag")
```

Change the wallpaper-

برای تعویض عکس پس زمینه ویندوز، بایـستی از ایـن قـسمت اسـتفاده کـرد. عـلاوه بـر تغییـر عکـس زمینـه ایـن ویژگـی Tweakایـن قابلیـت را فـراهم مـی کنـد تـا حالـت عکـس اعـم از کاشـی وار(Tile)،کـشیده(Stretch)ومرکـزی(Center) را تنظیم کنید.

SetWallpaper-

همانند برخی دیگر از فرامین ابتدا باید توسط قالب زیر اطلاعات را به PlugIn ارسال کنیم.

'Bitmap for wallpaper | Bitmap display style | Commit'

Bitmap for wallpaper :مسیر عکس جدید
Bitmap display style :حالت عکس (مرکز = 0 کاشی وار=1 کشیده =2)
Commit:دائمی بودن یا موقتی بودن زمینه را تنظیم می کند.عدد 1 برای دائمی و0 برای موقتی بودن

اگر از نوشتن Commit صرف نظر کنیم پس زمینه به طور دائمی نمایان می‌شود.

مثال:

```
BitmapPath$ =' C:\Windows\Bubbles.bmp'
BitmapStyle$ = '|1'
BitmapPermanet$ ='|1'
WallpaperStuff$=BitmapPath$+BitmapStyle$+BitmapPermanet$
PlugInSet("Tweak","WallpaperStuff$")
PlugInRun("Tweak","SetWallpaper")
```

Snap an MMB application to the desktop's edge's

با به کارگیری این ویژگی برنامه‌ها به هنگام نزدیک شدن به لبه‌های Desktop به سمت لبه‌ها جذب می‌شوند(پرش می کنند).

EnableEdgeSnap-

با فراخوانی این فرمان،عمل جهش صورت می‌گیرد.

DisableEdgeSnap-

با فراخوانی این فرمان عمل جهش غیر فعال می‌شود.

SetEdgeSnapDistance-

به طور پیش فرض هنگامی که لبه‌های برنامه از لبه‌های دسکتاپ فاصله داشته باشد عمل جذب رخ می‌دهد. حال با استفاده ازفرمان بالا می‌توان این فاصله را تنظیم کرد. البته این دستور اختیاری است . فاصله در واحد پیکسل تعریف می شود.

مثال:

```
Distance=30
PlugInSet("Tweak" , "Distance")
PlugInRun("Tweak","SetEdgeSnapDistance")
PlugInRun("Tweak","SetEnableEdgeSnap")
```

GetEdgeSnapDistance-

این دستور مقدار فاصله تنظیم شده برای عمل جهش را باز می‌گرداند.

مثال:

```
PlugInRun("Tweak","GetEdgeSnapDistance")
PlugInGet("Tweak","Distance")
Message("","Distance")
```

Memory Stats -

جهت اطلاع از وضعیت حافظه دستگاه می‌توان از این ویژگی استفاده کرد.

GetMemoryLoad-

این فرمان درصدی از حافظه را که در حال استفاده است را باز می‌گرداند.

GetTotalPhys-

این فرمان تعداد بایت‌های حافظه فیزیکی را باز می‌گرداند. (همان RAM)

GetAvailPhys-

مقدار کل حافظه فیزیکی (RAM)را باز می‌گرداند.

GetTotalPageFile-

مقدارکل PageFile را باز می گرداند.

GetAvailPageFile-

مقدار در دسترس PageFileرا باز می گرداند.

GetTotalVirtual-

مقدار کل حافظه مجازی را باز می‌گرداند.

GetAvailVirtual-

مقدار در دسترس حافظه‌های مجازی را باز می‌گرداند.

مثال:

```
PlugInRun(“Tweak”,”GetTotalPhys”)
PlugInGet(“Tweak”,”T-Phys”)
Message(“Your system physical memory is: “,”T-Phys”)
```

نکته: جهت تبدیل بایت به مگابایت کافیست بایت را ۲ بار بر 1024 تقسیم کنیم.

FreePlay PlugIn

این Plugln که اخیرا به مجموعه ی Plugln های MMB اضافه شده است قابلیتهای منحصر به فردو در عـین حـال پر کاربردی را به بدنه برنامه MMB اضافه می کند.همانند قـسمت Tweak قـصد داریـم فـرامین ایـن Plugln را فهرست وار شرح دهیم.همانطور که اطلاع دارید حـق اسـتفاده از تکنولوژیهـای Mp3 اخیـرا از حالـت رایگـان بـه صورت اشتراکی تغییر کرده و محدودیت هایی را ایجاد نموده است.در همین راستا استفاده از این Plugln می تواند راه حلی جهت این محدودیتها باشد.

SetItemHandler

توسط این فرمان،Plugln فشردن کلید هایی را شبیه سازی و آنها را به MMB ارسال می کند.بنا- براین از قبل باید اینکه را که میانبرهایش عینا همانند کلید هایی باشد که Plugln شبیه سازی می کند.

مثال:

```
Var$='CTRL,SHIFT,ALT,U'
PluglnSet("Plugln","var$")
PluglnRun("Plugln","SetItemHandler")
```

فرمان بالا از ترکیب ۳کلید shift وctrl وalt به همراه یک کلید دیگر از صفحه کلید استفاده می -کنند. چنانچه قصد نداشته باشید از کلید های سیستمی گفته شده استفاده کنید تنها کافیست تا نام آن کلید را خالی بگذارید،همانند مثال زیر:

مثال:

```
Var$='CTRL,,ALT,R'
PluglnSet("Plugln","var$")
PluglnRun("Plugln","SetItemHandler")
```

SendEvents

این فرمان ارسال رویداد از Plugln را فعال یا غیر فعال می سازد.در حالت پیش فرض این ویژگی فعال می باشد.توصیه می شود که این ویژگیقبل از نمایان شدن کادرهای محاوره ای غیر فعال شود چرا که از انتخاب اشتباه کانون توسط برنامه ها جلوگیری می شود.

مثال:

```
Var=1
PluglnSet ("Plugln","var$")
PluglnRun("Plugln","SendEvents")
```

Load

از این فرمان برای بارگذاری فایل های صوتی پشتیبانی شده استفاده می شود.فرمت های پشتیبانی شده WMA، mt3 وwar می باشند.

مثال:

```
Var$='<srcDir>\Music\Back ground,wma'
PluglnSet("Plugln","var$")
PluglnRun("Plugln","Load")
```

Play

این فرمان فایل یا فایل های بارگذاری شده را اجرا می کند.چنانچه ویژگی Send Event فعال باشد Plugln موقعیت اجرای فایل صوتی در حال اجراء را باز می گرداند.

مثال:

```
PluglnRun("Plugln","Play")
```

Pause

سبب ایجاد وقفه موقت در اجرای فایل صوتی می شود.فشردن مجدد این فرمان سبب لغو وقفه می شود.

مثال:

```
PluglnRun("Plugln","Pause")
```

Stop

اجرای فایل صوتی رابه طور کامل متوقف می کند.

مثال:

```
lulnRun("PlPugln","Stop")
```

SetPosition

موقعیت اجرای فایل صوتی را بیان می کند.مقدار دریافتی برحسب درصد می باشد.

مثال:

```
Var=40
PluglnSet("Plugln","var")
PluglnRun("Plugln","SetPosition")
```

GetPosition

فرمان بالا موقعیت اجرای فایل صوتی را بر حسب ۲ مقدار درصد و دقیقه باز می گرداند.

مثال:

```
Plugln Run("Plugln","Get Position")
PluglnGet("Plugln","var")
PluglnGet("Plugln","var$")
```

GetDuration

این فرمان زمان کل فایل صوتی را باز می گرداند.به همین ترتیب این فرمان ۲مقدار را یکی برحسب ثانیه و دیگری برحسب دقیقه باز می گرداند.

مثال:

```
Plugln Run("Plugln","GetDuration")
Plugln Get("Plugln","var)
PluglnGet("Plugln",'var$")
```

SetVolume

این فرمان سبب تنظیم حجم صدا می شود.البته تغییر حجم صدا توسط این فرمان هیچ تاثیری بر حجم صدای اصلی ویندوز نمی گذارد.در این فرمان مقدار ورودی برحسب درصد بیان می شود.

مثال:

```
Var=70
PluglnSet("Plugln","Var")
PluglnGet("Plugln","Setvolume")
```

Getvolume

این فرمان حجم صدا را باز می گرداند.البته این مقدار همان مقدار تنظیم شده توسط فرمانsetvolum می باشد.

مثال:

```
PluglnRun("Plugln","Getvolume")
PluglnGet("Plugln","var")
```

UseEqualizer

این فرمان ویژگی Equalizerرافعال یا غیر فعال می سازد.در حالت پیش فرض این ویژگی غیر فعال می باشد.پس از فعال کردن این ویژگی بایستی از فرمانSetEQvalue جهت دریافت مقدار باند های Equalizer استفاده کنیم.جهت فعال کردن ویژگی از پارامتر1وغیر فعال کردن از پارامتر 0 استفاده می کنیم.

مثال:

```
Var=1
Plugln Set(Plugln","var")
Plugln Run(Plugln","UseEqualizer")
```

SetEQvalue

از این فرمان جهت تعیین مقدار باند های Equalizer استفاده می شود.ابتدا توسط یک متغیر عددی شماره باند را مشخص می کنیمو سپس توسط یک متغیر رشته ای مقدار رابهPluglng ارسال می کنیم.ترتیب باندها به صورت زیر است:

محدوده ی متغیر عددی بین1 تا10 می باشدو محدوده ی متغیر رشته ای بین 0 تا30 است،مقدار پیش فرض

1-80H2
2-170H2
3-310H2
4-600H2
5-1000H2
6-3000H2
7-6000H2
8-10000H2
9-12000H2
10-14000H2

مثال:

```
VAR=1
VAR$'30'
PluglnSet("Plugln","var")
PluglnSet("Plugln","var$")
PluglnRun("Plugln","SetEQvalue")
```

SetEchoEffect

از این فرمان برای فعال یا غیر فعال کردن جلوهEcho استفاده می شود.جهت فعال کردن این ویژگی از پارامتر1 و برای غیر فعال کردن آن از پارامتر0 استفاده می شود.به طور پیش فرض این جلوه غیر فعال می باشد.

مثال:

```
Var=1
PluglnSet(Plugln","var")
PluglnRun(Plugln","SetEchoEffect")
```

Setflanger Effect

فرمان مذکور جلوهflarger را فعال یا غیر فعال می سازد.به طور پیش فرض این ویژگی غیر فعال است.توسط۲پارامتر 1 و0می توان این ویژگی را فعال یا غیر فعال کرد.

مثال:

```
Var=1
PluglnSet("Plugln","var")
PluglnRun("Plugln","SetflangerEffect")
```

SetReverbEffect

این دستور جلوهReverb را فعال یا غیر فعال می سازد.در حالت پیش فرض این جلوه غیر فعال است.توسط مقادیر 1 و 0 به ترتیب می توانیم این ویژگی را فعال یا غیر فعال سازیم.

مثال:

```
Var=1
PluglnSet("Plugln","var")
PluglnRun("Plugln","SetReverbEffect")
```

ShowAnalogMeter

این فرمان صوت سنج آنالوگ را فعال می سازد.منظور از صوت سنج قسمتی است که به صورت بصری مقدار پیشرفت آهنگ را نمایان می کند.در حالت پیش فرض این قابلیت مخفی می باشد.

مثال:

```
PluglnRun("Plugln","ShowAnalogMeter")
```

HideAnalogMeter

این فرمان صوت سنج آنالوگ را مخفی می کند.

مثال:

```
PluglnRun("Plugln","HideAnalogMeter")
```

MeterBackImage

فرمان بالا برای تعیین زمینه صوت سنج آنالوگ می باشد.تنها فرمت پشتیبانی شدهBmp می باشد.

توصیه می شود که حتی الامکان از تصویری استفاده کنید که اندازه اش با اندازه پنجره Plugln مطابقت داشته باشد.

مثال:

```
Var$='My-back-image.bmp'
PluglnSet("Plugln","var$")
PluglnRun("Plugln","MeterBackImage")
```

ShowAnalogKnob

فرمان مذکور دستگیره ی گردی را جهت تغییر حجم صدا نمایان می سازد. این دستگیره همانند پیچ تنظیم صدا در دستگاه های پخش صوت می باشد.در حالت پیش فرض این ویژگی غیر فعال است.

مثال:

```
PluglnRun("Plugln","ShowAnalogKnob")
```

HideAnalogKnob

این فرمان عکس فرمان قبل عمل می کند.

مثال:

```
PluglnRun("Plugln"," HideAnalogKnob")
```

Knobcolor

از این فرمان جهت تعیین رنگ دستگیره گرد استفاده می شود.ابتا بایستی همانند مثال مقدار رنگ را در کانال RGB به Plugln ارسال کنیم.

مثال:

```
Var$='200,80,100'
PluglnSet("Plugln","var$")
PluglnRun("Plugln","Knobcolor")
```

KnobTickcolor

جهت تعیین شاخص دستگیره گرد می باشد.در این فرمان نیز بایستی مقدار رنگ رادر کانال RGB به Plugln ارسال کنیم.

مثال:

```
Var$='10,80,200'
PluglnSet("Plugln","var$")
pluglnRun("Plugln","KnobTickcolor")
```

GetKnobposition

فرمان بالا موقعیت دستگیره گرد را در واحددرصد باز می گرداند.

مثال:

```
PluglnRun("Plugln","GeTKnobPosition")
PluglnGet("Plugln","var")
```

SetKnobPosition

این فرمان بر اساس مقدار ورودی که در واحد درصد می باشد مو قعیت دستگیره گرد را مشخص می کند.

مثال:

```
Var=40
PluglnSet("Plugln","var")
PluglnRun("Plugln","SetKnobPosition")
```

MMBDLL

فراخوانی فایل های DLLدیگر فرایندی است که تقریبا تمام نرم افزارها از آن بهره می برند.به واسطه ایـن Pluglnارزشمند،MMB نیز از این پس قادر است با فایل هایDLL ارتباط برقرار کند.برای استفاده از DLL های دیگرتنهـا کافیست به راهنمای همراه آنDLL مراجعه کنیم و نحوه فراخوانی توابع از آن را استخراج کنیم.بدین ترتیب برخی از فرامینی را که ممکن است هر کاربر MMB به هنگام طراحی برنامه خود آرزوی داشتن آن راداشته باشد از ایـن پـس در دسترس شماست.همان طور که در ابتدای مبحث Plugln شـرح داده شـد DLL (Dynamic Link Library)مجموعه ای از توابع است که مورد فراخوانی قرار می گیرد.هر فایلDLL رویه ای خاص خود دارد و این معین مـی کند چگونه بایستی از این فایل DLL استفاده شود.امر فراخوانی و برقراری ارتباط بـاDLL هـا تحـت عنـوان APL (Application Programming Interface) شناخته می شود.فراخوانی فایل های DLL و استفاده از توابع ارائه شده توسطDLL عمده مطلب API می باشدو این قسمت در زبان های برنامه نویسی از قسمت های اساسی برنامـه نویسی تحت ویندوز می باشد.بنابراین می توانیم توسط محیط های برنامه نویسی از قبیل VBوDelphi و هم اکنـون MMB به توابع فایلDLL دسترسی داشته باشیم.اصلی ترین توابعAPI در سه فایل زیر که درون هر سیـستم مبتنـی بر ویندوز می باشد موجوداست:

۱- User32.dll برای کنترل جزئیات بصری و رابط کاربر می باشد.

۲- Kernel32.dll برای کنترل فایل ها و مدیریت حافظه می باشد.

۳- Gdi32.dll فرامین گرافیکی را ارائه می دهد.

اما در عین حال تعداد بیشتری DLL درون سیستم عامل ویندوز نهفته است که می توانیم با دانستن نحوه اسـتفاده بـه فراخوانی آنها بپردازیم.یک مزیت دیگر استفاده از DLL اینست که این فایل ها قبلا توسـط زبانهـایی نظیـرC++وC تولید شده اند و تا حد امکان بهینه سازی شده می باشند. بنابراین این قبیلDLL ها سریعتر و قابل اطمینان تر از سایر DLL ها می باشند.با فراخوانیDLL ها تقریبا می توان تمام اعمالی را کـه ویـنـدوز انجـام مـی دهـد را انجـام داد،از خاموش کردن کامپیوتر گرفته تا نصب یک چابگر .به خاطر طبیعت ویندوز همزمان چندین برنامه می توانند فایل های DLL ویندوز را فرا خوانی نمایند. جهت بررسی بیشتر WinAPI می توانید به پایگاهای زیر مراجعه کنید:

- http//Msdn.Microsoft.comilibrary/
- http//www.mentalis.org/apilist/apilist.php/
- http//www.mentalis.org/

فرامین

تنها چهار فرمان توسطMMBDLL ارائه می شود که می توانید از آنها برای فراخوانی DLL ها استفاده کنید.

- **Path**

این فرمان مشخص می کند که فایلDLL مورد نظر در کجا واقع شده است.چنانچه قصد فراخوانی DLLهای ویندوز را داشته باشیدذکر تنها نام فایل کافی می باشد.

مثال:

```
Path$='user32.dll'
Plugln("plugln","path$")
Plugln("plugln","path")
```

- **Functions**

این فرمان مشخص می کند که چه تابعی از DLL مد نظر ماست.

مثال:

```
Function$='MyDLLFunction'
PluglnSet("Plugln","Function$")
PluglnRun("Plugln","Function")
```

- **Parameters**

جهت ارسال پارامتر های مورد نیازDLL ازMMB می باشد.پارامترها می توانندهم از نوع متغیرهای عددی باشند و هم به صورت رشته ای.چنانچه احتیاج به ارسال بیشتراز یک پارامتر باشدبین پارامتر از علامت"|" استفاده می کنیم.در هنگام عدم نیاز به یک پارامتر می توان جای آن را خالی گذاشت.

مثال:

```
Parameters$='0|Test message|TestTitle'
Pluglnset("plugln","Parameters$")
PluglnRun("Plugln","parameters")
```

- **Registration**

از آنجاییکه اینPlugln یک برنامه اشتراکی می باشد از این دستور برای ثبت برنامه استفاده می شود.

مثال:

```
Reg$='RegGde'
Pluglnset("Plugln","Reg$")
PluglnRun("plugln","Registration")
```

ممکن است فراخوانی تابعی از DLL خروجی به همراه داشته باشدکه برای دریافت خروجی می توانیم از دستور PluglnGet استفاده کنیم . مقدار خروجی همیشه به صورت متغیر رشته ای می باشد.

مثال:

```
PluglnGet("plugln","Result$")
Message("The DLL Sent these results: ","Result$")
```

SlideShowPlugIn

این Plugln نیز یکی دیگر از Plugln های پر کاربردMMB است. هدف از استفاده از این Plugln نشان دادن خودکار تعدادی عکس با وقفه های زمانی معین است.این Plugln از تمامی ابزارهایی که ممکن است در تولید یک SlideShow مورد نیاز واقع شود برخوردار است.بر همین اساس در ادامه قسمت شرح Plugln هایMMB قصد داریم تا به شرح توابع این Plugln قدرتمند بپردازیم.

SetPictureDirectory

این فرمان پوشه های حاوی عکس ها را مشخص می کند.در استفاده از این فرمان می توان زیر شاخه ها را نیز معین کرد ،برای اینکار بایستی شماره ی زیر شاخه را توسط متغیری بهPlugln ارسال نماییم.این Plugln تنها از فرمت های Ico، Bmp،Jpg ،Wmf حمایت می کند.چنانچه شماره ی زیر شاخه را صفر قرار دهیم هیچ زیر شاخه ای جستجو نمی شود.

SelectPictureDirectory

فرمان گفته شده یک کادر باز گشایی فایل را جهت انتخاب پوشه عکسهای مورد نظر نمایان می –سازد.در این قسمت نیز می توانیم زیر شاخه ها را نیز مشمول جستجو کنیم.

LoadPictureList

این تابع لیست عکس ها را از یک فایل متنی خارجی بار گذاری می کند و به درون یک لیست درونی انتقال می دهد.فایل خارجی می تواند هر پسوندی داشته باشد.چنانچه عکس مشخص شده در لیست موجود نباشد،Plugln از بار گذاری آن به درون لیست درونی صرف نظر می کند.این تابع ۲ مقدار 1به معنی پیدا کردن لیست و0 به معنی عدم یافتن لیست را باز می کند.

SetFullScreenDirectory

این فرمان محتوی پوشه عکس را به صورت تمام صفحه (FullScreen)نشان می دهد.شایان ذکر است که در بزرگنمایی نسبت وجوه رعایت می شود.

StartSlideShow

توسط این فرمان،بر اساس تنظیم های انجام شده نمایش عکس ها شروع می شود.

StopSlideShow

سبب توقف نمایش عکس ها می شود.

EnableToolTip

این فرمان ویژگی ToolTip را برای پنجره Plugln فعال می سازد.

DisableToolTip

فرمان مذکور ویژگی ToolTip را غیر فعال می کند.

SetToolTipText

این فرمان متن ToolTip را مشخص می کند.

EnableMouseControl

توسط این فرمان می توان از دکمه های موس جهت مشاهده عکسها استفاده کرد.به طور پیش فرض این ویژگی فعال است،اما از آنجاییکه فرمان DisableMouseControl موجود می باشد ضرورت وجود این فرمان آشکار می شود.

DisableMouseControl

قابلیت کنترل SlideShow توسط موس را غیر فعال می کند.

ExitProject

به خاطر برخی از مشکلات موجود در MMB برای کنترل Plugln ها،ممکن است برخی اوقات فرمانExit به درستی عمل نکند ، بر همین اساس این فرمان به لیست فرامین این Plugln اضافه شده است.

SetBackGroundColor

از این دستود برای تعیین رنگ زمینه Pluglnاستفاده می شود.ابتدا بایستی توسط متغیری مقدار رنگ را در کانال RGBبهPlugln ارسال کنیم.

SetBackGroundPicture

از این دستور جهت مشخص کردن عکس زمینه Plugln استفاده می شود.

EraseBackGroundPicture

فرمان بالا عکس زمینه پنجره Plugln را پاک می کند.

SetTransParentColor

از این فرمان جهت برداشتن رنگ پس زمینه عکس زمینه استفاده می شود.توسط این فرمان می –توانید شمایل پنجره Plugln را تغییر دهید.توصیه می شود که حتی الامکان از تصاویر با فرمت Bmp استفاده نمایید چرا که تشخیص رنگ زمینه توسط Plugln در فرمتBmp بهتر و راحتر صورت می گیرد.در فرمت Jpg به خاطر فشردگی ممکن است نتیجه مطلوب حاصل نشود.

SetStrechBackground

این فرمان عکس پس زمینه را به فرمی در می آورد که تمام پنجره Plugln را بپوشاند.

SetTileBackground

چنانچه بخواهیم عکس پس زمینه Pluglnرا به حالت کاشی وار درآوریم از این دستور استفاده می کنیم.

SetCenterBackground

این فرمان عکس پس زمینه را در وسط پنجره Plugln قرار می دهد.چنانچه اندازه عکس مربوطه از اندازه پنجرهPlugln بزرگتر باشد تنها قسمتی که مطابق پنجره Plugln است نمایان می شود.

SetScalePicture

از این فرمان جهت تغییر مقیاس عکس ها به هنگام نمایش در پنجره Plugln استفاده می شود.در این حالت نیز نسبت وجوه رعایت می شود.

SetOriginalPicture

اگر بخواهیم عکس ها در اندازه اصلی خود در پنجره Plugln نمایان شوند از این دستور استفاده می کنیم.

SetFillPicture

توسط این فرمان می توان طوری عمل کرد که عکس پس زمینه کاملا پنجرهPlugln را بپوشاند.البته در این حالت نسبت وجوه رعایت نمی شود.

InitSlideShow

این فرمان برای بنیان نهادن SlideShowمی باشد.چنانچه در حین اجرای SlideShow از این فرمان استفاده کنیم نمایش عکسها مجددا از اولین عکس از سر گرفته می شود.

HidePlugln

پنجره Plugln را مخفی می کند.

ShowPlugln

پنجره Plugln را نمایان می سازد.

ShowNextPicture

برای نشان دادن عکس بعدی نسبت به عکس جاری از این دستور استفاده می شود.

ShowPreviousPicture

برای نشان دادن عکس قبلی نسبت به عکس جاری از این فرمان استفاده می شود.

ShowPicture

همان طور که گفته شد عکسهای مورد نظر جهت نمایش در SlideShow در یک لیست درونی و به ترتیب قرار می گیرند.حال چنانچه بخواهیم یک عکس خاص را از این لیست نمایش دهیم از فرمان بالا استفاده می کنیم.

ShowFullScreenPicture

این فرمان عکس جاری را به صورت تمام صفحه نشان می دهد.ضمنا در این حالت نسبت وجوه رعایت می شود.همچنین کلیک کردن بر روی عکس در حالت تمام صفحه سبب لغو حالت تمام صفحه می شود.

GetCurrentPicture

این فرمان مسیر عکس جاری را باز می گرداند.همچنین این فرمان شماره عکس جاری را باز می گرداند.

GetAllPicture

این فرمان مسیر تمامی عکسها به همراه تعداد عکسها باز می گرداند.

GetPictureDiscription

این فرمان خصوصیت یک عکس را از یک فایل متنی باز می گرداند.چنانچه یک فایل متنی هم نام با عکس مورد نظر در پوشه عکسها باشد این فرمان محتویات این فایل متنی را نمایان می کند. چنانچه هیچ فایل متنی برای نگهداری خصوصیت عکس موجود نباشد این دستور عدد صفر را باز می گرداند.

PrintCurrentPicture

این فرمان عکس جاری را چاپ می کند.به هنگام استفاده از این فرمانPlugln کادر چاپ را نمایان می سازد.

SetScriptKey-PicDone

این فرمان همان EventHandle است.توسط این فرمان می توانیم کلید هایی را مشخص کنیم تا به هنگام بارگذاری هر عکس توسط فشرده شدن مجازی این کلیدها شی اسکریپتی از برنامه فرا خوانی می شود.

فصل پنجم

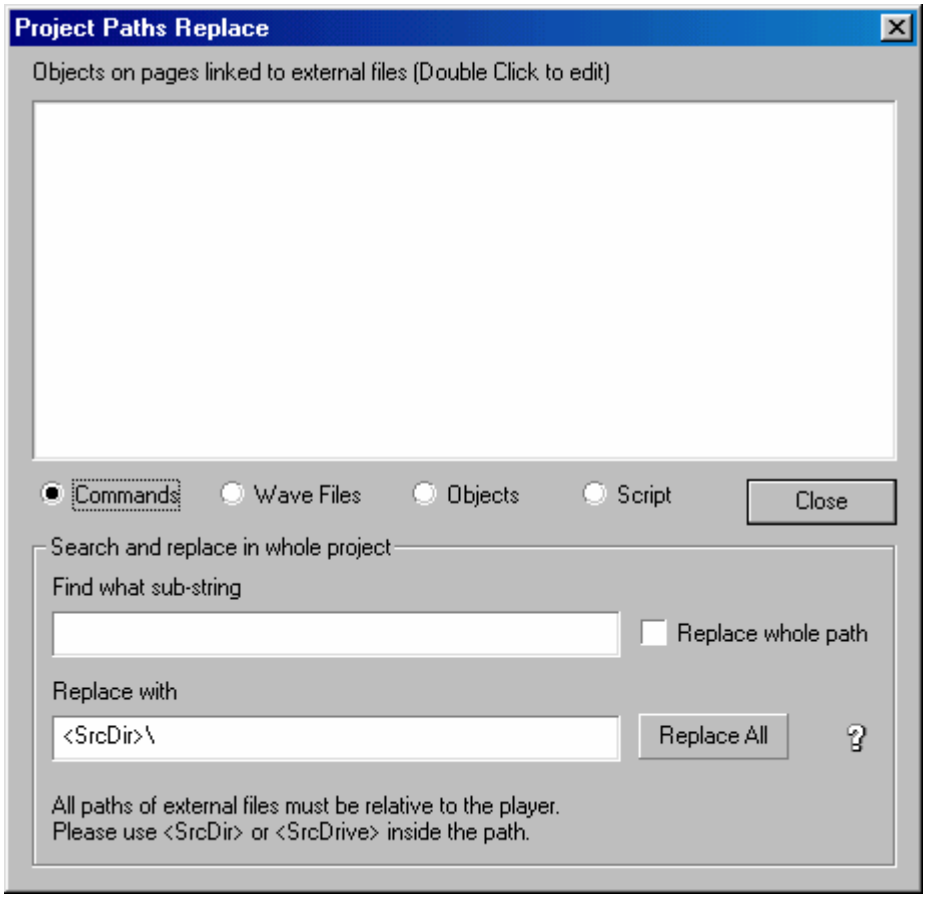
تولید نهایی

Multimedia Builder

- Path Replace
- Text Replace
- Compile

تولید نهایی

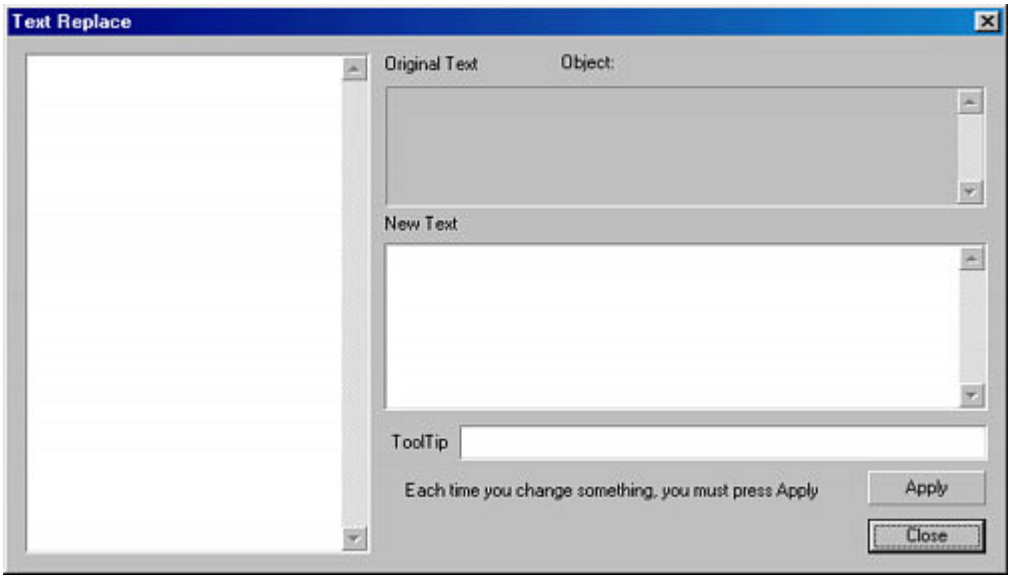
هنگـام تولیـد نهـایی محـصول فـرا رسـیده،پس از مـدتها تـلاش و برنامـه‌نویـسی منتظریـد تـا صـورت نهـایی پـروژه‌تـان را ببینیـد. امـا قبـل ازاینکـه دکمـه complie را فـشار دهیـد لازم اسـت تـا بـرای آخـرین بـار بـه بررسـی برخـی از قـسمتها و احتمـالاً تنظـیم صـحیح آنهـا بپردازیـد. اولـین قـسمتی کـه بایـستی بـه سـراغ آن بـرویم path replace اسـت.فایـل‌هـای ویدئویی ، موزیک‌هـاو... کـه فایـل‌هـایی خـارجی محـسوب مـی‌شـوند از بیـرون بـا پـروژه مـا در ارتبـاط هـستند و طبیعتـاً جزء فایل MBD نمی‌باشند چرا کـه باعـث بـالا بـردن حجـم فایـل و در نتیجـه کنـدی اجـرای پـروژه مـی شـوند. بـا توجـه به آن چه گفتـیم ایـن بـسیارمهم اسـت کـه مـسیر ایـن فایـل‌هـای خـارجی بـه گونـه‌ای تنظـیم شـود کـه کـاربران نهـایی در هنگـام اسـتفاده از پـروژه مـا بـا هـیچ مـشکلی روبـرو نـشوند، بـه همـین منظـور بـرای تنظـیم ایـن مـسیرها ابـزار Path replace طراحی شده اسـت. در ادامـه بـه شـرح قـسمتهای مختلـف ایـن ابـزار مـی‌پـردازیم. شـکل زیـر پنجـره ابـزار path replace را نشان می‌دهد.



دربـالای ایـن پنجـره کـادری موجـود اسـت کـه اقـلام خـارجی را نـشان مـی‌دهـد. در وسـط ایـن کـادر چهـار گزینـه Commands (فـرامین)، files ،Wave ،Object (اشـیاء) و Script موجـود اسـت.کـه هـر کـدام بـسته بـه نـوع خـود فایلهای خارجی اتصال یافته از همان نوع را نـشان مـی‌دهـد. حـال جهـت تنظـیم درسـت مـسیر بایـستی بـر روی هـر مؤلفـه کلیک کنـیم بلافاصـله پـس ازایـن عمـل قـسمتی از مـسیر کـه بایـستی تـصحیح شـود در کـادر ورودی کـه عنـوانش Find what sub- string است ظاهرمی‌شود. در زیرهمین کادر هم کادر ورودی

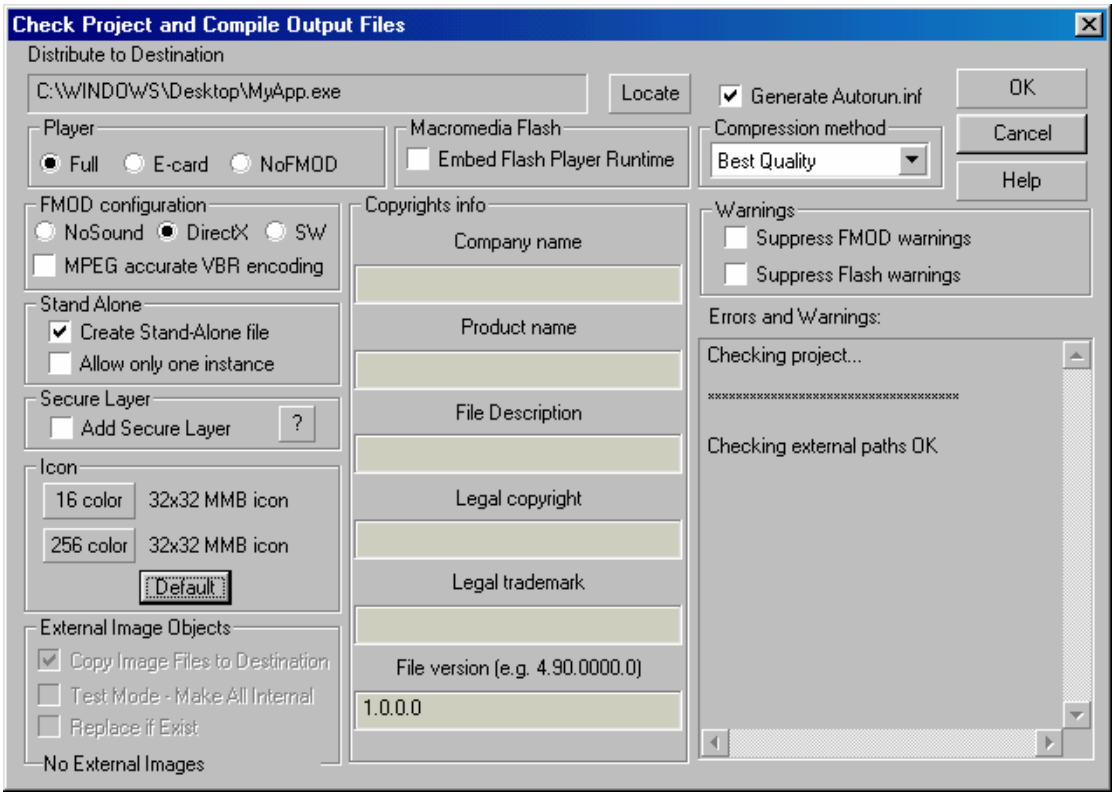
دیگـری موجـود اسـت کـه عنـوان آن Replace with مـی‌باشـد و کـاربرد آن ایـن اسـت کـه مـی‌تـوانیم اصـلاح مـسیر را دراین کادر انجام دهیم. با توجـه بـه توضـیحاتی کـه در بخـش‌هـای قبـل راجـع بـه ماکروهـای مـسیر گفتـه شـد،اکنون بایـد بدانید که بر حـسب نیازتـان مـسیر را چگونـه تـصحیح کنیـد. هنگامیکـه تغییـرات لازم را صـورت دادیـد تنهـا کافیـست تـا دکمـه Replace All را بـرای اعمـال تغییـرات فـشار دهـیم.امـا یـک گزینـه دیگـر هـم مانـده کـه بایـستی راجـع بـه آن توضـیحاتی گفتـه شـود.درکنـار کـادر ورودی اول یـک گزینـه بـا عنـوان Replace whole path قـرار دارد و کـار آن اینست که تصحیحی را که ما درکادر دوم صـورت دادیـم بـر همـه‌ی مـسیرها اعمـال کنـد، نـه فقـط یـک مـسیر تنهـا کـه مـا بر روی آن کلیک کـرده‌ایـم. البتـه بایـستی در اسـتفاده از ایـن گزینـه دقـت کنـیم چـرا کـه یـک سـهل‌انگـاری ممکـن اسـت ساعت‌ها وقتمان را هدر دهد. بـرای مثـال چنانچـه پـروژه مـا متـشکل از ۳۰۰ فایـل خـارجی باشـد و نـوع فایـل‌هـا مختلـف باشد وهمگی در پوشه هـای مختلـف نهـاده شـده باشـند ممکـن اسـت نتـوانیم مـسیرها را یکجـا تـصحیح کنـیم. همـین جـا اگر با یک اشـتباه کوچـک تمـام مـسیرها را یکـسان تغییـردهیم دیگـر راه بازگـشت سـریعی نمـی‌یـابیم و بایـستی تـک تـک مجدداً مسیرها را تصحیح کنیم و این خود مستلزم صرف وقت زیادی می‌باشد.

پس ازاینکه خیالمان از بابت مـسیرها راحـت شـد بایـد بـه سـراغ قـسمت Text replace بـرویم کـه بـه کمـک آن میتـوانیم متون به کار رفته در پروژه را درصورت لزوم تصحیح کنیم.



در قسمت سمت چپ پنجره این ابزار لیستی ازاشیاء حاوی متن ارائه شده که با کلیک بر روی آنها می توان آنها را تغییر داد. درسمت راست و بالا کادری قرار دارد که محتوی اصلی را نشان می‌دهد و در صورت تغییر نادرست متن‌ها می توانیم از این قسمت برای بازگشتن به حالت اصلی استفاده کنیم. در زیر این کادر پنجره‌ای جهت انجام تغییرات تعبیه شده که متون جدید را بایستی درآن با متون قدیمی جایگزین کنیم. ضمناً در صورت لزوم می توانیم هنگام تغییر متون tooltip آنها را هم تنظیم نماییم.در نهایت با فشردن دکمه‌ی Apply تغییرات صورت گرفته به پروژه اعمال می‌شود ،اما توجه کنید که برای هر تغییر مجزا باید دکمه‌ی Apply را فشار دهیم و نمی‌توانیم اول همه‌ی متن‌ها را تغییر دهیم و این کلید را فشار دهیم.

هنگامی که از درستی مسیرها و متون پروژه اطمینان حاصل کردیم باید سری به فایل‌های خارجی بزنیم و از سالم بودن آنها و همچنین محل قرارگیری آن‌ها بر اساس مسیرهای تنظیم شده اطمینان حاصل کنیم. خسته شدید؟! نگران نباشید دیگر نوبت قسمت Compile است.به هنگام کامپایل MMB بر اساس تنظیمات انجام شده فایل exe نهایی را تولید می‌کند.بر اساس شواهد موجود MMB فایل اجرایی نهایی را با استفاده از برنامه upx که یک فشرده‌ساز می باشد فشرده می‌کند که پس از تشریح قسمت کامپایل در رابطه به آن مطالبی را خواهیم گفت.اما با فشردن دکمه‌ی Compile کادری همانند شکل زیر نمایان می‌شود.



اما قسمت‌های این کادر به چه صورت است:

- Distribute to destination

این قسمت برای مشخص کردن مسیر فایل اجرایی نهایی می‌باشد با فشردن دکمه Locate کادر محاوره‌ای جهت تعیین کردن نام فایل و مسیرجهت ذخیره آن ظاهر می‌شود.

- Generate AutoRun.inf

این گزینه باعث می‌شود تا فایل inf که لازمه CD های AutoRun (خود اجرا) می‌باشد به همراه فایل نهایی تولید شود.در قسمت ضمیمه می‌توانید توضیحات مفصلی در رابطه با فایل AutoRun.inf بیابید.

- Player

این قسمت نوع Player (اجراکننده)رامشخص می کند که براساس نیاز می‌توانیم آن را تعیین کنیم.

- Full

یک اجرا کننده کامل می‌باشد و تمامی کتابخانه‌ها و Resource ها در آن موجود است.شما در اکثر مواقع از این قسمت استفاده می‌نمایید.

- E- card

یک فایل اجرایی با حجم کمتر تولید می‌کند که برای ارسال توسط پست الکتریکی مناسب می‌باشد اما باید توجه داشت که این نوع محدودیتهایی را دربردارد. محدودیت‌هایی از قبیل:

۱- عدم حمایت از MP3

۲- عدم داشتن زمینه تمام صفحه و آهنگ زمینه

۳- و دیگر محدویتهای جزئی

- No FMOD

این گزینه از به الحاق در آوردن موتور FMOD به همراه فایل اجرایی جلوگیری می‌کند.این گزینه زمانی کار آمده است که صدا وصوت نقشی در پروژه ما نداشته باشد. FMOD یک موتور اجرا کننده صوتی بسیار قوی با کیفیت صوتی بسیار بالا می‌باشد.

Maromedia flash –

اعمال این گزینه اجرا کننده فلش (Flash player) را به همراه فایل اجرایی صادر می‌کند.

FMOD Configuration–

این قسمت جهت تنظیم موتور FMOD می‌باشد در این قسمت می‌توان مشخص کرد موتور FMOD چگونه فایلهای صوتی را اجرا کند.

Copyrights Info –

دراین قسمت مؤلف می‌تواند اطلاعاتی ازقبیل: company name (نام شرکت)، product name (نام محصول)، file discription (توصیف فایل)، Legal CopyRight (کپی رایت قانونی)،Legal trade mark (مارک تجاری قانونی)، File version (شماره نسخه فایل) را در کادرهای مربوطه حداکثر به طول ۲۵ حرف برای هر قسمت وارد نماید.

Stand Alone –

خروجی استاندارد MMB یک اجرا کننده (Player) و فایل MBD مربوطه می‌باشد که به اجرا کننده اتصال یافته است. با انتخاب گزینه Create stand alone file کامپایلر تنها یک فایل اجرایی (Exe .*)حاوی اجرا کننده و فایل mbd.* را تولید خواهد کرد. گزینه Allow only one instance باعث می‌شود تا تنها یک نسخه از فایل اجرایی در آن واحد اجرا شود.

Secure layer –

با فعـال نمـودن ایـن قابلیـت مـی‌تـوان از مـشاهده کـردن اسـکریپت هـا و متـون بـه کـار رفتـه در برنامـه توسـط ویرایـشگر hex جلوگیری نمود. باید توجه کرد کـه بـا اعمـال ایـن گزینـه ازسـرعت فایـل اجرایـی تـا حـدودی کاسـته مـی‌شـود. پـس تاکید می‌کنیم تنها در صورت لزوم از این ویژگی استفاده کنید.

lcon –

در ایـن قـسمت مـی‌تـــوانیم lcon فـــایل اجرایـــی را مـشخص کنـــیم.برحـسب نیـاز مـی‌تـوانیم یـا از lcon هـای color 16 (١٦ رنگ) استفاده کنیم یا ازcolor 256 (٢٥٦رنگ).

External lmage objects –

چنانچه عکس‌های موجـود در پـروژه مـا بـه صـورت خـارجی باشـند ایـن قـسمت فعـال مـی‌شـود. Copy lmage file to destination تصاویـــر را با تـوجه بـه مسیر فایـل اجرایی به مقصد کپی می‌کند. گزینه

Test mode- Make all internal بـرای حـالتی اسـت کـه بخـواهیم جهـت امتحـان پـروژه فایـل اجرایـی نهـایی را تولیـد کنیم .و نهایتاً Replace if exist برای جایگزینی عکسها در صورت موجود بودن آنها در پوشه مقصد است.

Compression method –

برای نشر فایل‌ها در اینترنت و یا داشتن مـسئله در رابطـه بـا حجـم فایـل نهـایی مـی‌تـوان از گزینـه‌هـای ارائـه شـده در ایـن قسمت که هرکـدام سـطحی خـاص ازفـشردگی را ارائـه مـی کننـد اسـتفاده کـرد. کیفیـت فایـل تولیـد شـده بـر اسـاس ایـن گزینه‌ها از گزینه پایین به بالا بهتر می‌شود.

Warnings –

ممکـن اسـت موتـور FMOD وهمچنـین Flash player در هنگـام اجـرای پـروژه اخطارهـایی را متـذکر شـوند کـه توسـط این قسمت می توان از نمایان شدن آنها جلوگیری کرد.

Errors and warning –

ایـن قـسمت اشـکالات و خطاهـایی را کـه بـه خـاطر عـدم توجـه مـا هنـوز درپـروژه موجـود اسـت را بـازگو مـی کنـد. اشـکالاتی نظیـر مـسیرهای اشـتباه و ... ، چنانچـه مـشکلی نباشـد در ایـن کـادر عبـارت checking external path ok ظاهر میشود.

فصل ششم

ضمیمه

Multimedia Builder

- نکته ها
- پسوندهای معروف
- خروجی MMB چگونه فایلی است؟
- مقدمه ای بر AutoRun.Inf
- RunDLL32.exe
- میانبرهای پیش فرض MMB
- واژه نامه

نکته ها:

در این قسمت قصد داریم تا چند نکته ی کلیدی را جهت تولید یک پروژه ی هر بهتر ارائه کنیم.

- رفع عیب (Debug)

احتمالا تولید هیچ پروژه ای خالی از اشتباه نیست وهمین مطلب اشکال زدایی(Debug) پروژه را اجتناب ناپذیر ساخته است.عدم اشکال زدایی ممکن است منجر به هدر رفتن سرمایه وساعت ها کار شود. اولین اشکال زدایی توسط خود مؤلف صورت می گیرد و چنانچه مولف در آزمایشات خود اشکالی را نیابد نسخه آزمایشی(بتا)را در اختیار چند نفر جهت افزایش دامنه آزمایشات قرار می دهدتا احیانا اشکالات دیگر احتمالی بروز یابند.چنانچه هیچ خطایی یافت نشود مؤلف برنامه نسخه اصلی برنامه (آلفا)را ارائه می دهد.MMB نیز ازاین قاعده مستثنا نمی باشد و ابزار رفع عیب را در اختیار مؤلف قرار می دهد.مد رفع عیب از منوی Debug<- Setting قابل دسترس است.در هنگام رفع عیب MMBلیستی از متغیرها و محتوی آنها را جهت بررسی بهتر ارائه می دهد.همچنین جزئیات دیگر برنامه در لیستی دیگر در پایین مد رفع عیب ثبت می شود.البته باید گفت که سرعت اجرای برنامه در حالت رفع عیب تا حدودی کم می شود.

- توزیع برنامه(Distribution)

هنگامیکه مراحل رفع عیب برنامه به پایان رسید هنگام عرضه محصول است بدین منظور باید روشی را برای انتقال برنامه به سیستم کاربران نهایی بر گزینیم.به ۲ طریق می توان برنامه را در اختیار کاربر قرار داد:

۱.قراردادن آن در اینترنت برای دریافت:

این روش بیشتر برای فایلهای کم حجم استفاده می شود .این روش از امنیت و کارایی بیشتری برخوردار است.

۲.قرار دادن آن برروی CD:

این روش برای فایلهای با حجم بالاو عرضه محصول در فروشگاه ها مناسب می باشد.

اما گذشته از روش انتقال بایستی ملزومات را جهت اجرای برنامه مشخص نمود و سپس آنها را به اطلاع کاربر رسانید.منظور از ملزومات همان مقدار حافظه مورد نیاز،فضای دیسک سخت مورد نیاز،پردازشگر مورد نیاز و سازگارکننده گرافیکی مورد نیاز به همراه سایر ملزومات نرم افزاری از قبیل DirectX

و ...است.

- افزایش سرعت

سرعت اجرای بالا یکی از دغدغه های اصلی تولید کنندگان و کاربران می باشد،به همین منظور باید تا حد امکان طوری عمل کرد که سرعت اجرای برنامه افزایش یابد.در زیر چند توصیه برای بهبود سرعت آمده است:

۱–حتی الامکان سعی کنید از فایلها به خصوص تصاویر به صورت خارجی استفاده نمایید .

۲– در طراحی پروژه های بزرگ سعی نمایید که پروژه را به قسمت های ریز تری تقسیم نمایید و طوری برنامه ریزی کنید که به هنگام اجرای برنامه تنها قسمت های مورد نیاز فراخوانی شوند.

۳– به هنگام کامپایل پروژه در صورت امکان از قسمت فشرده سازی استفاده نکنید چرا که فشردگی کاهش سرعت را به همراه دارد.اگر قصد انتشار محصول خود بر روی CD را دارید در صورت امکان از فشرده سازی فایل اجرایی نهایی صرف نظر کنید تا سرعت اجرای برنامه بالاتر رود.

٤–استفاده صحیح از حلقه ها و وقفه ها سبب بهبود و بهینه سازی سرعت می شود.

۵– از الحاق کردن(Embedding) فایل ها تا حد امکان پرهیز کنید.

٦–در طراحی پروژه از تکلف و زیاده روی در استفاده از تصاویر و قسمتهای گرافیکی پرهیز نمایید.زیرا کارایی پروژه مهمتر از ظاهر آن است.

۷–چنانچه در پروژه از صوت و موسیقی استفاده نمی کنید از به انضمام در آوردن موتور صوتی FMOD خود داری کنید.

۸–حق تقدم (Priority) استفاده از CPU را متناسب با عملکرد پروژه تنظیم نمایید.

- هنگام شروع برنامه(Initialization)

برخـی از Plug-Inهـای MMB و فـرامین کنتـرل اشـیا را نمـی تـوان در هنگـام شـروع برنامـه مـورد اسـتفاده قـرار داد.بـه عنوان مثال در Tweak فرامین INIدر هنگام شروع به درستی عمل نمی کنند.

- استفاده از برنامهRealDRAW

توصیه می شود کـه بـرای طراحـی قالـب هـا و دکمـه مـورد نیـاز پـروژه از ایـن برنامـه اسـتفاده نماییـد،چرا کـه ایـن برنامـه کاملا با محیط MMB سازکار می باشد.همچنین این برنامه در عین سادکی از قدرت مناسبی برخوردار است.

پسوندهای معروف

ANI

این پسوند معرف فایل Animated Cursor است که برای اولین بار در ویندوز ۹۵ معرفی شد. پس از آن در نسخه‌های بعدی ویندوز مورد استفاده قرار گرفت.

PSD

فایل طرح مورد استفاده در برنامه Photoshop با پسوند PSD

نشان داده می‌شود.

DLL

این پسوند معرف فایل‌هایی است که توانایی برقراری ارتباط با کتابخانه‌های اینتراکتیو (تعاملی) را دارند. این فایل‌ها معمولاً در سیستم عامل‌های ویندوز و OS/2 وجود دارند و تنها وقتی که برنامه‌های مربوط به آنها اجرا می‌گردند در حافظه بارگذاری می‌شوند.

HTML , HTM

این پسوند معرف فایل HyperText Mark up Language است. HTML زبانی است که از آن برای طراحی محتوی وب، استفاده می‌شود. بنابراین فایلهای HTML اساساً با صفحات وب در ارتباط هستند.

INI

پسوند فوق معرف فایل راه اندازی است که در برنامه‌های کاربردی مختلف وجود دارد. این نوع فایل وقتی اجرا می‌شود که کامپیوتر راه اندازی شده یا برنامه فعال گردد این فایل برای ذخیره پیکربندی استفاده می‌شود.

JPG

این پسوند معرف فشرده سازی Joint Photographic Experts Group می‌باشد. از این فرمت برای تصاویر طرح بیتی و رنگی استفاده می‌شود این فرمت بعد از تعیین استاندار فشرده سازی پدید آمده و نامگذاری شده است. از این پسوند گرافیکی در اکثر برنامه‌های کاربردی و گرافیکی استفاده می‌شود.

EXE

این پسوند نشان دهنده‌ی فایل اجرایی است که همراه با برنامه‌های کاربردی برای کامپیوترهای مبتنی بر Dos اجرا می‌شود. یک فایل اجرایی فرمتی است که کامپیوتر می‌تواند بدون کمک کاربر اجرا نماید.

GIF

این پسوند معرف فایل Graphic Interchang Format می باشد. پسوند مذکور اولین بار توسط شرکت Compuserve ارائه شده است. در حقیقت GIF یک فرمت فشرده‌سازی است که اطلاعات دیجیتالی را فشرده نموده و به تصاویر گرافیکی ثابت و یا متحرک تبدیل می‌کند.

AVI

این پسوند نشان دهنده‌ی یکی از فرمت‌های فایل تصویری به نام Audio Video Inter Leave می‌باشد. در ویندوز از این فرمت همراه با برنامه‌ی Microsoft Video استفاده می‌شود.

AVI یا MPEG

فرمت AVI در تمام نسخه های سیستم عامل ویندوز شناسایی و حمایت می شود.اما فرمت MPEG مستقیما توسط سیستم عامل حمایت نمی شود و لازم است تا اجرا کننده و راه انداز آن در سیستم عامل نصب شود.اما با این وجود تمام نسخه های ویندوز ۹۸ به بعد از فرمت MPEG پشتیبانی می کنند و این امر به واسطه برنامه Windows Media Player صورت می گیرد.بنابر این بهترین راه برای اجرای فرمت MPEG بر روی سیستم های قدیمی نصب Media Player جدید و سازگار با آن نسخه از ویندوز می باشد.در مقایسه بین این دو فرمت، MPEG از کیفیت بالاتری برخوردار است.اما با این حال احتیاج به استفاده بیشتری از CPU نسبت به فرمت AVI دارد.اگر به بررسی بیشتر فرمت AVI بپردازیم متوجه خواهیم شد که این فرمت نیز گونه های مختلفی دارد و مسبب این حالات Codec ها می باشند.

Codec چیست؟

در حالت طبیعی می توانیم از فایل های AVI بدون فشرده سازی استفاده کنیم ، اما با یک مشکل رویرو خواهیم شد و آن چیزی جز مشکل کمبود فضا نیست.تنها چند دقیقه کوتاه از یک فیلم با فرمت AVI در این حالت به چندین مگابایت فضا نیاز دارد.بنابراین پای نرم افزارهای فشرده ساز ی تصویری به میان می آید که با نام کلی Codec شناخته می شوند.موارد استفاده از Codec تنها برای فشرده سازی فایل تصویری نیست ، بلکه به هنگام اجرای فایل تصویری فشرده سازی شده Codec مورد نیاز بایستی از قبل بر روی سیستم نصب شده باشد.همانند سایر شاخه های نرم افزار هم اکنون Codec های متعددی در دسترس می باشند که بسته به نوع هر کدام سطح خاصی از فشرده سازی امکان پذیر می شود.چنانچه قصد استفاده از فشرده ساز خاصی را دارید در صورتی که تردید داشته باشید که آیا آن فشرده ساز بر روی سیستم کاربر نهایی

موجود است یا خیر بایستی آن Codec را به همراه بسته نرم افزاری خود ارائه دهید.در حالت پیش فرض چند Codec در تمام ویندوز های 9x,2000 و XP از قبل نصب می شود.Codec هایی از نظیر Cinepack،Indeo ، Microsoft Video از این دست هستند.در ادامه به شرح مختصر Codec های موجود می پردازیم.

Microsoft Video1

Codec اصلی که به همراه نسخه های پیشین ویندوز ارائه شده است.از نظر کیفیت این Codec کیفیت بدی را ارائه می دهد،چرا که این Codec در زمانی که وضوح 256 رنگ حالت استاندارد بوده است تولید شده است.بهتر است از این نوع فشرده ساز صرف نظر کنیم.

Indeo 4, 5

این دو نسخه نیز محصول شرکت اینتل هستند اما تفاوتشان با نسخه قبلی در تکنولوژی به کار رفته در هر کدام می باشد.این دو Codec کیفیت بهتری را نسبت به موارد قبلی ارائه می دهند.اما در عین حال به سخت افزارهای قوی تری نیاز دارند.تقریبا تمام ویندوزهای از ۹۵ به بالا داری این Codec می باشند.

MPEG-1

این فرمت کیفیت مطلوبی را به ارمغان می آورد.این فرمت بسیار مشهور است و فرمتی است که برای سی دی های تصویری(VideoCD) استفاده می شود، MPEG-1 است.

MPEG-2

کیفیتی که این فرمت ارائه می دهد بسیار عالی است.از این فرمت در DVD ها استفاده می شود.برای اجرای این فرمت به سخت افزارهای نسبتا سریع و پیشرفته ای نیاز است.

MPEG-4

اگر سیر صعودی کیفیت فایل های ویدویی را بررسی کنید هنگامی که به این فرمت برسید در می یابید که این فرمت کیفیت بسیار عالی را فراهم می آورد.شرکت مایکروسافت نیز بر اساس همین فرمت، فرمت ASF را ارائه داده است.

حال که به این قسمت رسیده اید،بدون شک از خود سوال می کنید کدام فرمت مناسب است؟پاسخ این سوال بستگی به خودتان دارد.ابتدا بایستی بررسی کنید که ملزومات پروژه شما چیست.آیا نیاز است که کیفیت بالای ویدویی استفاده شود یا خیر؟آیا می خواهید که پروژه تان در تمام سیستم ها اجرا شود؟ اگر کیفیت مهم است بایستی از فرمت MPEG استفاده شود.اگر اجرای صحیح برایتان اهمیت دارد AVI می تواند انتخاب خوبی باشد.البته توصیه می شود که Codec های مربوطه را به لحاظ اطمینان بیشتر به همراه بسته نرم افزاری خود به کاربران ارائه دهید.

« خروجی MMB چگونه فایلی است ؟ »

قبلاً خواندیم که MMB با استفاده از Upx فایل اجرایی نهایی را فشرده و سپس تولید می‌کند،البته در صورتی که به هنگام تولید نهایی از فشرده سازی استفاده کنیم.

Upx برنامه‌ایست که برای فشرده سازی فایل‌هایی از قبیل exe, dll ,ocx و... به کار می‌رود. مزیت این فشرده‌سازی علاوه بر ایجاد فشردگی مطلوب عدم نیاز به استخراج فایلها از بسته فشرده شده به هنگام اجرای فایل فشرده می باشد. به عنوان مثال همان طور که می‌بینید فایل‌های اجرایی که توسط MMB تولید شده‌اند هیچ پیغامی را در هنگام اجرا مبنی بر استخراج فایلها ارائه نمی‌دهند. اکثر توسعه‌دهندگان Plug-In هم از این برنامه جهت فشرده سازی فایلهایdll تولید شده استفاده می‌کنند. لازم به ذکر است که Upx یک برنامه به صورت Command Line می‌باشد. جهت بررسی این برنامه می‌توانید آن را از روی سی دی ضمیمه این کتاب پیدا کنید.

اما اگر پا را فراتر بگذاریم می‌توانیم از همین Upx برای ایجاد تغییراتی جزئی در فایل اجرایی نهایی ایجاد کنیم. همه‌ی برنامه‌های اجرایی دارای منابعی (Resource) هستند که شمایل برنامه‌ها را تشکیل می‌دهند منظور ما از منابع همان تصاویر، آیکن‌ها، متون، مکان نماها و.. به کار رفته در فایل اجرایی است. در همین زمینه برنامه‌هایی از قبیل Resource Hacker موجودند که کارشان تغییر همین منابع است.بنابراین جهت ایجاد تغییرات در فایل اجرایی نهایی می‌توانیم از Upx برای بازگرداندن فایل اجرایی تولید شد از حالت فشرده به حالت عادی استفاده کنیم و در نهایت با استفاده از برنامه‌های تغییر منابع تغییرات دلخواه را صورت دهیم. البته نکته مهمی را که در اینجا باید متذکر شد اینست که عمل تغییر منابع را تنها برای فایل‌هایی می‌توانیم انجام دهیم که مجاز به تغییر آنها می‌باشیم.

مقدمه‌ای برفایل AUTORUN. INF

درمحیط Windows 32- BIT تکنولوژی بنام AUTOPLAY ظهورکرد، که بوسیله آن سیستم عامل می‌تواند وجود دیسک را در درایو تشخیص دهد. در محیط windows ، هنگامی که دیسک در درایو قرار می‌گیرد، سیستم عامل به دنبال فایل AUTORUN. INF میگردد، این فایل که یک فایل متنی است، توانایی‌هایی را به شرح زیر برای سیستم عامل فراهم می‌کند.

بوسیله این فایل میتوان:

– مسیر و نام برنامه اجرایی را برای سیستم عامل مشخص نمود تا هنگام قرار دادن دیسک در درایو، این برنامه اجرایی به صورت خودکار به اجرا در آید.

– آیکون مورد نظر خود را جایگزین آیکون پیش فرض درایو در My Computer نمود.

– متن مورد نظر خود را جایگزین متن پیش فرض (Label) درایو در windows explorer نمود.

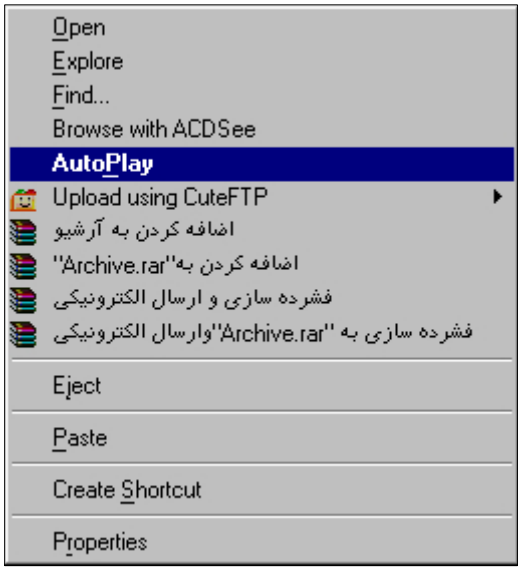
در ابتدا توجه شما را به چند نکته در مورد ایجاد فایل AUTORUN. INF جلب می‌کنیم.

۱– برای ایجاد این فایل می‌توانید از برنامه Notepad کمک بگیرید و سپس پسوند فایل ایجاد شده را از TXT به INF تغییردهید.

۲– همانطور که در مثال زیر مشاهده می‌کنید، در خط اول از فایل، نوشتن "[Autorun]" الزامی است.

۳– هر دستور را در یک خط مجزا بنویسید.

٤- بـرای اطـلاع ازایـن کـه دیـسکی کـه در درایـو قـرارداده‌ایـد محتـوی فایـل AUTORUN . INF اسـت یـا خیـر، میتوانیـد بر روی آیکون درایو مذکور راست کلیک کنید سپس منـویی کـه بـه آن منـوی میـان‌بـر گفتـه مـی شـود ماننـد شـکل زیـر بـاز می‌گردد.



درشـکل بـالا اگـر گزینـه Open را انتخـاب کنیـد میتوانیـد بـدون اجـرای AutoRun محتویـات درون درایـو را مـشاهـده کنیـد، امـا اگـر گزیـــــنه Autoplay را انـــــتخاب کنیـد سیـــــستم عامـل بـا اسـتفاده از برنامـه . AUTORUN EXE ، فایل AUTORUN . INF را می خواند و محتویات آن را همانطور که در آن تعریف شده اجرا می‌کند.

دستورات فایل AUTORUN . INF

ایـن مجموعـه، دایـره‌المعـارفی بـرای رجـوع بـه دسـتوراتی اسـت کـه در فایـل AUTORUN . INF بکـار مـی‌رود و شـامل shell/ verb open, label, icon می‌باشد.

Icon

دستور Icon آیکون درایوی را مشخص میکند که قابلیت Autoplay را درwindows دارد. (منظور درایوی است که Disk در آن قرارمی گیرد). و این دستور همانطور که دربالا گفته شد آیکون این درایو را تعیین می‌کند شکل کلی دستور به صورت زیر است:

incon= inconfilename [, index]

جزییات دستور

Iconfilename : نام یک فایل با پسوند bmp, exe, dll یا ico می‌باشد که شامل اطلاعات آیکون مورد نظر باشد. اگر فایل بیش از یک آیکون داشته باشد میتوانید با استفاده از پارامتر index (از0 تا N) آیکون مورد نظر را مشخص کنید.

توضیحات بیشتر

دستور Icon باعث می‌شود که در Windows Explorer به جای آیکونی که به صورت پیش فرض برای دستگاه دیسک خوان انتخاب شده از آیکون مورد نظر شما استفاده شود.مثال زیر، آیکون دوم را از فایل Myprog. Exe مشخص می‌کند:

icon= Myprog. Exe 1

Label

دستور Label متن Label دستگاه دیسک خوانی را مشخص می‌کند که قابلیت Autoplay دارد. شکل کلی دستور به صورت زیر است:

label = Level Text

جزییات دستور

Label Text: متنی است که برای جایگزینی بجای Label پیش فرض درایو مورد نظر درسیستم عامل انتخاب میشود و میتواند فواصل را نیز شامل شود.

توضیحات بیشتر

Label در زیر؛ آیکون درایوی که قابلیت Autoplay دارد به نمایش در می‌آید و درواقع متن، زیر آیکون می باشد. مثال زیر عبارت "Mydrivelabel" رابجای Label پیش فرض سیستم عامل قرارمی‌دهد.

Label= My Drive Label

Open

دستور Open، مسیر و نام فایل برنامه‌ای را که باید هنگام قراردادن دیسک درون درایو توسط کاربر به صورت خودکار اجرا شود. مشخص می‌کند. شکل کلی دستور به صورت زیر است:

open= [exepath/] exefile [param1 [param2]…]

جزییات دستور

Exefile: مسیرکامل فایل اجرایی است که قراراست هنگام قرار گرفتن دیسک در درایو، به صورت خودکار اجرا شود. اگر فایل اجرایی درشاخه اصلی درایو باشد میتوانید فقط نام آن را بنویسید، درغیر این صورت نوشتن مسیر کامل فایل الزامی است.

Shell / Verb

Shell/ Verb: یک دستور به منوی میان‌بر اضافه می‌نماید که فرم کلی آن به صورت زیراست:

Shell/Verb/command= filename. Exe

Shell/Verb= MenuText

جزییات دستور

Verb: به صورت پیش فرض، متنی است که درمنوی میان بر نمایش داده می‌شود. دستور Verb, Shell /Verb/Command رابا یک فایل اجرایی پیوند میزند.

Filename . exe

مسیر و نام فایل برنامه‌ای است که با Verb پیوند می خورد.

Menutext

متنی را مشخص می‌کند که درمنوی میان‌بر نمایش داده می‌شود که اگر حذف گردد یا نوشته نشود، بجای آن Verb نمایش داده می‌شود.بین کلمه‌ها در Meun Text میتوان فضای خالی داشت (در مورد Verb چنین نیست) همچنین با قراردادن کلمه "&" درابتدای کلمه مورد نظر درMenu Text میتوانید برای آن کلید shortcut درست نمایید.

توضیحات بیشتر

هنگامی که کاربر بر روی آیکون درایو با قابلیت Autoplay راست کلیک می‌کند منوی میان بر ظاهر می‌گردد. افزودن دستورات Shell/Verb به فایل Autorun.inf به شما اجازه می‌دهد که گزینه‌هایی را به این منو اضافه کنید.

این دستور دو قسمت دارد که باید در دو خط جداگانه باشند. قسمت اول shell/ Verb/ Command می‌باشد که وجود آن الزامی است این دستور یک رشته متنی را که Verb نام دارد به برنامه‌ای که هنگام اجرای دستور باید اجرا شود پیوند می‌زند. قسمت دوم، دستور Shell/Verb می باشدکه انتخابی است و شامل متنی است که در منوی میان بر نمایش داده میشود.

این قطعه از فایل Readit, AutoRun, inf را (که در واقع همان Verb است) با دستور "Notepad ABC/ Readm. Txt" پیوند میزند. متن منوی میانبر (Menu Text) , "Read Me"می‌باشد و "M" به عنوان کلید Shortcut انتخاب شده است. وقتی کاربر گزینه را از منوی میان بر انتخاب می‌کنند فایل ABC/ Readme.Txt بوسیله برنامه Notepad باز میشود.

Shell/readit/ command= notpad abc/readme. Txt

Shell/readit= Read&Me

چگونه فایلهای AutoRun.inf را امتحان کنیم.

همانطورکه گفته شد درمحیط windows هنگامی که دیسک در درایو قرار می‌گیرد سیستم عامل به دنبال فایل‌های سیستم و فایل AutoRun. Inf میگردد. حال اگر بخواهید این فایل را قبل ازنصب بر روی CD امتحان کنید لازم است در Windows Registry کامپیوتر خود تغییرات زیر را اعمال کنید تا قابلیت Autoplay به ابزارهای دیگر نظیر Floppydisk یا Network Drives منتقل شود.

برنامه Regedit را اجرا کرده و تغییرات لازم را درکلید زیراعمال کنید:

HKEY- CURRENT- USER\

Software\

Microsft\

Windows\

CurrentVersion\

Policies\

Explorer\

"NoDrive Type AutoRun"

ایـن کلیـد ازنـوع Reg- Binary میباشـد و از ٤ بایـت تـشکیل شـده اسـت. بـه بایـت اول آن Bitmask گفتـه مـی‌شـود Bitmask تعیین می‌کند که کـدام درایـو یـا ابـزار بایـد بـه صـورت خودکـار اجـرا شـود. ۳ بایـت دیگـر بایـد مقدارشـان (0) باشد. هر بیت در Bitmask بیانگر توضیحی برای Registry میباشد که شرح آنها در زیر آمده است.

توضیحات	بیت	نوع
-	0	Drive - Unknown
-	1	Drive - No-Root-Dir
Floppy Disk	2	Drive - Removable
-	3	Drive - Fixed
Network Drive	4	Drive - Remote
-	5	Drive - CD ROM
-	6	Drive - Ramdisk

بـه صـورت پـیش فـرض ارزش ایـن کلیـد Registry، 0×95 مـی‌باشـد بـه عنـوان مثـال بـرای ایـن کـه بتوانیـد قابلیـت Autoplay را برای Flopydisk تنظیم کنید ارزش Bitmask را به 0×91 تغییر (Modify) دهید.

پیغام خطای:Windows cannot find autorun. Exe

هنگامی که بـه لیـست درایوهـا در Windows explorer نگـاه مـی‌کنیـد ممکـن اسـت یـک یـا چنـد درایـو، آیکـون اشـتباه و نامربوط به خـود را داشـته باشـند، هنگـامی کـه بـرای مـشاهده محتویـات آنهـا بـر روی آیکـون درایـو دابـل کلیـک مـی‌کنیـد، ممکن است پیغام خطای زیر نشان داده شود.

Program Not Found Windows Cannot Find Autorun.exe

This Program is needed for opening files of type “file”

سپس از شما آدرس و مسیر فایل Autorun.exe خواسته می‌شود.

توضیحات بیشتر

فایـل Autorun. inf همـراه بـا درایـوی کـه قابلیـت Autoplay را دارد (ماننـد CD- Drive) بـرای اجـرای خودکـار یـک برنامه مشخص به کار گرفتـه مـی‌شـود. هنگـامی کـه یـک درایـو کـه قابلیـت Autoplay نـدارد محتـوی فایـل Autorun. inf باشـد بعـد از دابـل کلیـک کـردن کـاربر بـر روی آیکـون آن درایـو، برنامـهAutoplay سـعی میکنـد فایـل Autorun.exe را (کـه درآن درایـو وجـود نـدارد) بـرای خوانـدن فایـل Autorun. inf اجـرا کنـد، کـه باعـث روی دادن خطـا درسیـستم عامـل می‌شود.

راه‌حل

برای رفع این مشکل بـر روی درایـو فـوق‌الـذکر راسـت کلیـک کنیـد، منـوی میـانبر بـاز میـشود (مـشاهده میکنیـد کـه گزینـه Autoplay. پیش فـرض مـی‌باشـد). گزینـه open را انتخـاب نماییـد و پـس از بـاز شـدن درایـو، فایـل Autorun. inf را کـه در شاخه اصلی درایو قراردارد پاک نمایید و سپس کامپیوتر را Restart کنید.

چگونه میتوان به CD- ROM اجازه اجرای Auto Runداد یا از آن سلب اجازه نمود.

اطلاعات این قسمت فقط درسیستم عامل‌های زیرصدق می‌کند:

Microsoft windows NT server*

Microsoft windows NT workstation*

اگریک تعویض کننده CD- ROM (CD- ROM Changer) به کامپیوتر شما متصل باشد، ممکن است بخواهید امکان AutoRun را از آن سلب کنید. اگر تعویض کننده CD- ROM ، امکان AutoRun داشته باشد، هر زمان که شما CD را دریکی از طبقات آن قرارمیدهید، سیستم عامل تمام طبقات تعویض کننده CD را مرور میکند.

توضیحات بیشتر

گزینه‌ای در Control panel برای سلب اختیار نمودن سیستم عامل از اجرای AutoRun وجود ندارد، برای اینکار شما باید Registry را ویرایش کنید. برای انجام این کار مراحل زیر را دنبال کنید.

اخطار: اگر از برنامه Registry Editor درست استفاده نکنید، میتواند سبب بوجود آمدن مشکلات جدی برای سیستم عامل کامپیوتر شما شود، که شما را مجبور به دوباره نصب کردن windows میکند.Registry را با مسئولیت خود تغییر دهید.

۱- مقدار AutoRun را درکلید Registry زیربه شرحی که آمده است تغییر دهید.

HKEY- LOCAL-

Machine\ System\ Current Control Set\ Services \ CDRom

اگر مقدار AutoRun رابه0 تغییر دهید ازتعویض کننده CD- ROM برای اجرای AutoRun سلب اختیار میشود و اگر ارزش آن را به 1 تغییر دهید، دوباره اختیار AutoRun به تعویض کننده CD- ROM داده میشود.

۲-کامپیوتر را Restart کنید.

نکته: روش بالا به طورکامل از انجام پروسه AutoRun سلب اختیار میکند. اگر میخواهید بر روی هر دیسک به طور مجزا برای اجرای AutoRun نظارت کنید بعد از قراردادن دیسک در درایو، کلید shift را پایین نگه‌دارید.

میانبرهای پیش فرض MMB

New	Ctrl+N
Open	Ctrl+O
Print	Ctrl+P
Save	Ctrl+S
Copy	Ctrl+C
Delete	Del
Hide/Show	Ctrl+T
Paste	Ctrl+V
Paste Bitmap	Ctrl+B
Redo	Ctrl+Y
Select All	Ctrl+A
Undo	Ctrl+Z
Test CurrentPage	F5
10 Pixel Down	Ctrl+Down
10 Pixel Left	Ctrl+Left
10 Pixel Right	Ctrl+Right
10 Pixel Up	Ctrl+Up
Bring Forward	Ctrl+PageUp
Bring to Front	Shift+PageUp
Group	Ctrl+G
Pixel Down	Down
Pixel Left	Left
Pixel Right	Right
Pixel Up	Up
Send Backward	Ctrl+Page down
Send to Back	Shift+Page down
Ungroup	Ctrl+U

توجه:با راست کلیک بر روی رابط کاربر MMB منویی ظاهر می شود که می توانید از آن جهت سفارشی سازی رابط کاربر MMB استفاده کنید.

منابع :

- راهنمای برنامه MMB
- سایت www.mmbforums.com
- ماهنامه کامپیوتر جوان

واژه نامه

((A))

Alpha Button = دکمه آلفا

Animated GIF= متحرک GIF

Audio visualization= بصری ساز صوتی،رقص نور

Align =ترازکردن

Arrange =مرتب کردن

Antialias =هموار سازی

Apply =فعال کردن

Aspect =وجه

Actual =حقیقی

Adjust =تنظیم کردن

Animate =متحرک سازی

Advanced=پیشرفته

Additional =اضافی

Array = آرایه

Accurate=دقیق،صحیح

Artist=هنرمند

Absolute=مطلق

((B))

Bitmap Button=دکمه نقشه بیتی

Bitmap=نقشه بیتی

Binder=موظف کننده،دراینجا برقرا کننده اتصال

Background=پس زمینه

Backward=به عقب

Blur=بلور

Bevel=برجسته سازی،اریب کردن،تراشیدن

Button=دکمه

Border=حاشیه،لبه

Bumpy=دارای برآمدگی،نا هموار

Bubble=حباب

Buffer=حافظه میانجی

Blend=مخلوطی،آمیخته

Boost=بالا بردن،ترقی دادن

Balance=تعادل

((C))

Circle=دایره

Customize=سفارشی سازی

Comment=توضیح

Convert=تبدیل کردن

Clone=زاییدن

Combine=ترکیب کردن

Crop=چیدن

Crayon=مداد رنگی مومی

Cube=مکعب

Cascade=آبشاری

Calculation=محاسبه

Circularity=مدور بودن

Chunks=تکه بزرگ وکلفت

Column=ستون

Caption=عنوان

Capture=ربودن،گرفتن

Constant=ثابت،دائم

Clipboard=تخته برش

Curl=پیچاندن،حلقه کردن

Compression=فشردگی

Configuration=پیکربندی

Current=جاری،کنونی

Content=مندرجات،محتوی

Counter=شمارشگر

((D))

Dynamic fx=پویا fx

Dimension=بعد

Debug=اشکال زدایی

Distribute=نشردادن

Designer=طراح

Default=پیش فرض

Disposal=در معرض گذاری

Define=معین کردن،تعریف کردن

Drag=کشیدن

Drop=انداختن

Disable=غیر فعال کردن

Display=نمایش دادن،نمایان ساختن

Decimal=دهدهی،اعشاری

Delay=به تاخیر انداختن

Destination=مسیر،مقصد

Description=توصیف،توضیح

((E))

Embedded=جاسازی شده

Effect=جلوه

Emboss=برجسته کردن،اندودن

Expand=گسترش دادن

Export=صادر کردن

Enable=فعال سازی

External=خارجی

Edge=لبه،یال

Extension=بسط،پسوند نام فایل

Engine=موتور

Else=دیگر،جز این

Editor=ویرایشگر

((F))

Flash=روشنایی آنی و مختصر

Forward=به جلو

Flip=چرخاندن،تلنگرزدن

Fixed=ثابت وماندنی

Fade out=محو شدن

Float=معلق

Flat=تخت

Fill=پرکننده

Fit=مناسب،درخور

Frame=فریم،قاب

Factor=فاکتور،عامل

Flare=روشنایی خیره کننده و نامنظم

Function=تابع

((G))

Grid=شبکه،چهارخانه بندی

Glow=درخشش

Glass=شیشه

Grip=چنگ زدن،گرفتن

Gradient=شیب

Generate=تولید کردن

((H))

HTML=زبان نشانه گذاری ابر متن

Horizontally=افقی

Height=ارتفاع

Highlight=منور کردن،پر رنگ کردن

HotSpot=منطقه داغ

Handle=دسته

((I))

Input Text=متن ورودی

Image Matrix=ماتریس تصویر

Interaction=تعامل

Impressionist=

Import=واردکردن

Invert=معکوس کردن

Integer=عدد صحیح

Internal=درونی

ID=عبارت شناسایی

Interference=تداخل

Interval=فاصله،خلال

Idle=بیکار

If=اگر

Initialize=آغازکردن

((J))

((K))

Keep=نگهداری کردن

((L))

ListBox=کادر لیست

Line=خط

Label=برچسب

Library=کتابخانه

Loop=حلقه تکرار

Lumi=روشن

Lens=عدسی،به شکل عدسی

Layer=لایه

Legal=قانونی

((M))

Mosaic=موزاییک،تکه تک بهم پیوستن

Metal=آهن

Menu=منو

Modal=مقید

Mask=ماسک

Media=رسانه

Movement=جنبش،تکان

Master=قسمت اصلی،صاحب،مدیر

Mixed=آمیخته شده

Message=پیغام

Milliseconds=میلی ثانیه

Mute=صامت،بی صدا

Magnify=بزرگ کردن

MMB=مولتی میدیا بیلدر

Macromedia Flash=برنامه فلش شرکت ماکرو مدیا

Method=روش

Macro=ماکرو

((N))

Nudge=اشاره کردن،در اینجا تکان دادن مختصر

Numeric=عددی

Null=پوچ

((O))

Opacity=غلظت

Origin=مبدا،منشا

((P))

Paragraph Text=پاراگراف متنی

Panorama=چشم انداز

Polygon=چند ضلعی

Ploygonal Hotspot=منطقه داغ چند ضلعی

Plug-In=در اینجا یعنی قسمتهایی که سبب افزایش قابلیت های برنامه می شوند

Property=خصوصیت،مشخصه

Previous=قبلی

Preview=بازبینی

Parameter=پارامتر

Pixel=پیکسل

Primitive=اصلی،بدوی

Plasma=پلاسما

Preset=از پیش نشاندن

Progress=پیشرفت

Preserve=حفظ کردن

Priority=حق تقدم

Period=دوره

Picker=چیننده،برداشت کننده

Pitch=زیر وبمب صدا

Predefined=از پیش تعریف شده

Particle=ذره

Poor=نا مرغوب،ناچیز

Path=مسیر

Primary=نخستین

Page=صفحه

((Q))

Quality=کیفیت

Quantize=کمیت گذاری

((R))

Rectangle=مسطتیل

Replace=جایگزین کردن

Ripple=موج دارشدن

Restore=بازسازی

Ratio=نسبت

Refresh=تازه کردن

Row=طبقه،ردیف

Raised=برافراشته شده،برجسته شده

Return=بازگشت

Resume=از سر گیری

Rotate=چرخاندن

Relative=وابسته،نسبی

Randomize=تصادفی کردن

Random=تصادفی،رندم

Reverse=معکوس

Redo=دوباره انجام دادن

Reflectivity=بازتاب پذیری

Real-DRAW pro=برنامه ریل درا برای طراحی ۲بعدی

((S))

Script=اسکریپت

Snap=پرش

Sound=صدا،صوت

Shadow=سایه

Sketch=طرح

Setting=تنظیمات

Skin=پوسته

String=رشته

Solid=یکدست،جامد

Style=سبک

Safe=امن

Stage=مرحله

Scroll=پیمایش

Storage=انبارش،ذخیره سازی

Stretch=کشیدن،امتداد دادن

Static=ایستا،ساکن

Sunken=فرورفته

Still=هنوز،خاموش،بی حرکت

Spectrum=طیف

Saturation=اشباع

Smoke=دود

Status=وضعیت

Shape=شکل،قیافه

Sparkle=تلالوداشتن

Separator=جدا کننده

Speech=سخنرانی

Syntax=نحو

Scale=مقیاس گذاری

Shortcut=میانبر

Smart=باهوش

Suppress=مانع شدن

Solarize=در معرض آفتاب قرار دادن

Sample=نمونه،مثال

Source=منبع،منشا

Statement=اظهار

((T))

Text=متن

Text Button=دکمه متنی

Template=قالب

Tile=کاشی

Twirl=چرخش،گردش

Transparent=پشت نما،شفاف

Type=سنخ،نوع

Terminate=پایان دادن

Transition=واسط،مرحله تغییر

Title=عنوان

Tolerance=تحمل

Then=سپس

Total=مجموع

Trademark=علامت تجاری

Tweak=نیشگون گرفتن،پیچاندن

((U))

URL=آدرس اینترنتی

Undo=خنثی کردن

UI=رابط کاربر

Urban=شهری

((V))

Video=ویدئو

Vertically=عمودی

Variable=متغیر

Vibration=ارتعاش

Visualization=تجسم

Volume=حجم

Visible=قابل دید

((W))

Wrap=پیچیدن

Width=عرض

Wizard=جادوگر

((Z))

Zero=صفر

منتظر ارائه راهنمای بعدی در این زمینه باشید...