



Controllo numerico per Foratrici / Fresatrici Multimandrino

EDICAD

Configurazione e personalizzazioni

24/11/2011



tecnologie e prodotti per l'automazione

20099 Sesto S. Giovanni (MI) - Via Carducci, 221

tel. (02) 26223139

fax (02) 2481008

1.	CONDIZIONI AL CONTORNO	4
1.1.	Risorse	4
1.2.	Percorsi notevoli	4
1.3.	Files di Help	5
1.4.	Avvio di EDICAD.EXE	5
2.	CONFIGURAZIONI DI EDICAD (MENU IMPOSTA)	6
2.1.	Imposta: Menu	6
2.2.	Imposta: Settaggi	7
2.3.	Imposta: Abilitazioni	13
2.4.	Imposta: Importazioni	16
2.5.	Imposta: Ottimizzatore	17
2.6.	Impostazioni dirette	22
3.	DATABASE DELLE LAVORAZIONI (EDICADWR.DEF)	26
3.1.	Gestione dei files database delle lavorazioni	27
3.2.	Assegnazione di lavorazione	28
3.3.	Assegnazione di parametro di lavorazione	32
3.4.	Gestione delle immagini di help delle lavorazioni	36
3.5.	Lavorazioni con codice operativo in intervallo [1 – 1000]	37
3.5.1.	Punto di applicazione	37
3.5.2.	Codice di lavorazione e codice esecutivo	37
3.5.3.	Sottotipi di lavorazioni	38
3.5.4.	Come configurare una lavorazione custom in intervallo [1 - 1000]	38
3.6.	Prospetti riassuntivi di lavorazioni puntuali e di setup	39
3.6.1.	Prospetto di lavorazione puntuale	39
3.6.2.	Prospetto di lavorazione setup	40
3.7.	Lavorazioni custom di profilo [2300 – 2400]	41
3.8.	Codici di macro [1001 – 2000]	41
3.8.1.	Macro-programma	42
3.8.2.	Come configurare un codice di macro	44
3.8.3.	Creazione di modello per PASTUDIO	47
3.8.4.	Parametri di assegnazione ausiliaria (parametri di appoggio)	48
4.	LAVORAZIONI RISERVATE AI MACROPROGRAMMI	49
4.1.	Istruzioni logiche	49
4.1.1.	IF / ENDIF	49
4.1.2.	FOR – ENDFOR / FOREVER - ENDFOR	50
4.1.3.	BREAK	50
4.1.4.	ERROR	52
4.1.5.	CONTINUE	52
4.2.	Istruzioni di assegnazioni ausiliarie	53
4.2.1.	Istruzione: Punto di applicazione	53
4.2.2.	Istruzioni di assegnazione delle variabili J	53
4.2.3.	Istruzioni di assegnazione delle variabili \$	54
4.2.4.	Istruzione di assegnazione ternaria delle variabili \$	54
5.	TIPOLOGIE DI PROGRAMMA	56
5.1.	Sottociclo	56
5.2.	Sottoprogramma	56
5.3.	Come configurare un codice di chiamata sottoprogramma	57
5.4.	Sviluppo di sottoprogramma	58
6.	IL LINGUAGGIO MLC	60
6.1.	Elenco delle istruzioni	60
6.2.	Convenzioni	60
6.3.	Struttura di una generica istruzione MLC	61
6.4.	Elenco delle parole chiave (costanti predefinite) e loro valore	61
6.5.	Elenco delle parole chiave (variabili predefinite) e loro significato	61
6.6.	Funzionalità: SCELTA DI COLORE	61
6.7.	Funzionalità: DISEGNO DI MARCATORE	62
6.8.	Funzionalità: DISEGNO DI LINEA	62
6.9.	Funzionalità: DISEGNO DI ARCO	62
6.10.	Funzionalità: DISEGNO DI CERCHIO	65
6.11.	Funzionalità: DISEGNO DI ELLISSE	65
6.12.	Funzionalità: DISEGNO DI RETTANGOLO	66
6.13.	Funzionalità: CONDIZIONAMENTI	66
7.	OPTIMIZE	68
7.1.	Comunicazione con Ottimizzatore custom	68

7.1.1.	File di matrice pezzo: MXPZIN.BIN	68
7.1.2.	Files di parametriche	71
7.2.	Esposizione di oggetti	72
7.3.	Descrizione errori	76
8.	PARAMETRICHE	78
8.1.	Formato testo	78
8.2.	Formati binari	78
8.2.1.	Formato binario tipo 1	79
8.2.2.	Formato binario tipo 2	79
8.3.	Assegnazione di matrice MxCnfGru	80
8.3.1.	Assegnazione di gruppi in aggancio	82
8.3.2.	Assegnazione di utensili ruotati	82
8.3.3.	Assegnazione dell'utensile speculare	83
8.3.4.	Assegnazione di testine multipunta per lavorazioni puntuali	83
8.3.5.	Assegnazione e gestione doppi pistoni	85
8.3.6.	Gestione setup con utensili multipli	86
9.	MATRICE PEZZO	87

1. CONDIZIONI AL CONTORNO

1.1. Risorse

Si elencano le **risorse utilizzate** dal programma EDICAD, in modo da definire le *condizioni al contorno* richieste:

- come ogni applicativo Albatros, EDICAD.EXE deve essere installato ed avviato in un ambiente configurato per il prodotto Albatros (cartella principale e secondarie). Esempio: "c:\albatros\bin";
- legge i settaggi di ambiente da file TPA.INI;
- legge e scrive i settaggi dell'applicativo in PEZZO.INI (in cartella *cadcfg*. Esempio: C:\ALBATROS\CADCFG\);
- legge i files di messaggi: EDICAD.LNG e DLGCLASS.LNG (cercati in directory di messaggi di Albatros. Esempio: C:\ALBATROS\LNG\); EDICADWR.LNG ed eventuale CUSTOM.LNG (cercati in cartella *cadcfg*); eventuale CSTBREAK.LNG (cercato in cartella *cadcfg*);
- legge i files di immagine: HELPEDI.GRF ed eventuale CUSTOM.GRF (cercati in cartella *cadcfg*);
- legge il database delle lavorazioni da file EDICADWR.DEF ed eventuale CUSTOM.DEF (in cartella *cadcfg*);
- estrae la tabella dei diametri utensili con lettura dei files di assegnazioni tecnologiche in directory di configurazione modulo: PARTEC.PAR, PARTOOL.PAR (Esempio: C:\ALBATROS\MOD.0\CONFIG\ per il modulo di macchina 1).

ATTENZIONE:

- EDICAD.EXE può funzionare fino alle versione 2.5.xxx di Albatros;
- EDICAD.EXE non funziona in sistemi attuali a 64bit.

1.2. Percorsi notevoli

Nell'ipotesi di installare l'ambiente Albatros in cartella *bin* ="C:\ALBATROS", la tabella elenca i percorsi notevoli per l'applicativo:

\$ se il nome non è fisso ↓

W crea se non esiste ↓
R solo in lettura

X se deve esistere ↓
() altrimenti

Voce in TPA.INI	Esempio	Files	Descrizione dei files	
bin + "\BIN"	C:\ALBATROS\BIN\	EDICAD.EXE EDICAD.HLP EDICAD.CNT	applicativo files di help	X () R () R
bin + "\LNG"	C:\ALBATROS\LNG\	EDICAD.LNG DLGCLASS.LNG	messaggi dell'applicativo messaggi ausiliari in finestre delle lavorazioni	X R () R
DirCadCfg	C:\ALBATROS\CADCFG\	EDICADWR.DEF EDICADWR.LNG HELPEDI.GRF CUSTOM.DEF CUSTOM.LNG CUSTOM.GRF CUSTOM.HLP CUSTOM.CNT CSTBREAK.LNG PALWS.BMP PALWS0.BMP PALWS1.BMP PALWS2.BMP PALWB.BMP PALWB0.BMP PALWB1.BMP PALWB2.BMP PEZZO.INI	DataBase di lavorazioni Messaggi del Database Bitmaps delle lavorazioni DataBase Custom Messaggi di Database Custom Bitmaps del Database Custom Files di help per lavorazioni custom Messaggi custom (errori, palette lavorazioni); Bitmap custom per lavorazioni Bitmap custom per lavorazioni	X W X R X R () R () R \$ () R \$ () R \$ () R \$ () R \$ () R () R \$ () R \$
DirCadCfg+"\MCR"	C:\ALBATROS\CADCFG\MCR\	*.MCR	Macroprogrammi	() W
DirCadCfg+"\SUB"	C:\ALBATROS\CADCFG\SUB\	*.*	Sottocicli	() W

DirCadCfg+ "MODELS"	C:\ALBATROS\CADCFG\MOD ELS\	*.*	Modelli per PASTUDIO	() W
Dirprod	C:\ALBATROS\PROD	*.*	Programmi	() W
Dirprod + "\SUB"	C:\ALBATROS\PROD\SUB	*.*	sottoprogrammi	() W
Dirprod+"IMPORT"	C:\ALBATROS\PROD\IMPORT	*.*	ausiliario in conversione di formato	() W
system	C:\ALBATROS\SYSTEM	LOGOCUST.BMP	logo cliente	() R
rootmod+"MOD.n\ CONFIG"	C:\ALBATROS\MOD.0\CONFIG	PARTEC.PAR PARTOOL.PAR	Parametri di assegnazione gruppi ed utensili di macchina 1	() R
	C:\ALBATROS\MOD.1\CONFIG	PARTEC.PAR PARTOOL.PAR	Parametri di assegnazione gruppi ed utensili di macchina 2	() R
		HEAD.BMP	bitmap della testa	() R \$

1.3. Files di Help

EDICAD.EXE gestisce i files di Help con le specifiche di seguito riportate:

- EDICAD.HLP è il default (gestisce lingua: inglese). È gestita l'assegnazione di files differenti per la lingua con la regola: nome del file = "ECAD_" + (stringa di lingua), estensione = "HLP" (Esempio: ECAD_ITA.HLP);
- EDICAD.HLP (oppure: ECAD_ITA.HLP, ECAD_ENG.HLP,...): forniscono l'help per i comandi di EDICAD.EXE e per le lavorazioni lette da EDICADWR.DEF;
- Per le lavorazioni custom (assegnate da CUSTOM.DEF): CUSTOM.HLP è il default. È gestita anche in questo caso l'assegnazione di files differenti per la lingua, con la regola: nome file = "CUST_" + (stringa di lingua), estensione = "HLP" (Esempio: CUST_ITA.HLP);
- I files di default (EDICAD.HLP e CUSTOM.HLP) devono essere installati anche nel caso di installazione dei files differenziati per la lingua: costituiscono infatti il default, in caso di file in lingua non trovato;
- localizzazione files HLP:
 - ◊ i files di help dell'applicativo sono cercati secondo gli usuali criteri Windows (di norma: installati in BIN);
 - ◊ i files di help delle lavorazioni custom sono prima cercati in CADCFG e, se non trovati, in BIN.

1.4. Avvio di EDICAD.EXE

In avvio di programma, possono comparire una o più segnalazioni:

- finestra di indicazione errori derivanti dal controllo delle descrizioni grafiche delle lavorazioni: sono indicati gli errori formali riscontrati nella stringa di rappresentazione grafica della lavorazioni, come indicate nel database. La rappresentazione grafica viene automaticamente annullata;
- messaggio "*Tabella diametri non impostata*": la tabella di assegnazione dei diametri utensili non è stata impostata per errore di lettura (file non trovati o non validi);
- messaggio: "*Errore: impossibile aprire il file EDICADWR.DEF*" oppure "*File o gruppo lavorazioni non trovato*": segnalano che non è stato possibile leggere il database della lavorazioni, per errore di lettura sul file o perché non vi risultano assegnate lavorazioni. Si tratta di errori gravi: con database non assegnato non è possibile aprire o creare alcun programma;
- messaggio "*Formato file inaspettato*": compare nel caso in cui è attiva la impostazione di caricamento automatico dell'ultimo programma aperto in precedente istanza dell'eseguibile. Il messaggio indica che il programma non è trovato o non può essere letto.

2. CONFIGURAZIONI DI EDICAD (MENU IMPOSTA)

Si esaminano qui tutte le impostazioni protette di EDICAD.EXE: modificabili cioè a password costruttore. Tutte le impostazioni indicate sono archiviate nel file di inizializzazioni PEZZO.INI, in directory *cadcfg*. Dopo avere verificate e modificate le impostazioni protette è buona norma riavviare il programma EDICAD (ad esempio i Menu sono assegnati solo all'avvio di EDICAD e le voci eventualmente eliminate, se abilitate dopo l'avvio non vengono ripristinate: l'unico sistema per farlo è chiudere e riavviare EDICAD).

2.1. Imposta: Menu

La finestra si compone di tre pagine.

_____ Pagina 1: Menu

Legge MRU[0]	<input checked="" type="checkbox"/> abilita l'apertura automatica del primo programma in lista dei file ultimi letti. Se abilitato: <ul style="list-style-type: none"> ◇ in avvio EDICAD apre direttamente il programma ultimo aperto da EDICAD stesso; ◇ se EDICAD viene avviato a password utente e l'ultimo programma aperto è un macro-programma oppure un ciclo: segnala il messaggio "Formato file non valido" ed annulla l'apertura.
Mantiene direttori	<input checked="" type="checkbox"/> i comandi <i>File>>Open</i> e <i>File>>Save As..</i> : mantengono e memorizzano l'ultimo path. Se non abilitato: <ul style="list-style-type: none"> ◇ <i>File>>Open</i> si predispose con apertura della cartella dei programmi; ◇ <i>File>>Save As..</i> si predispose con apertura della cartella dei programmi o dei macroprogrammi, in base alla tipologia del programma.
Importa	<input checked="" type="checkbox"/> abilita la voce corrispondente del menu File
Annulla	<input checked="" type="checkbox"/> abilita la voce corrispondente del menu Edita
Proprietà	
Informazioni geometriche	
Assegnazione tecnologica	
Assegnazioni complessive	
Trova	
Selezione di	<input checked="" type="checkbox"/> abilita la voce corrispondente del menu Selezioni. Consigliato: <input type="checkbox"/> (OFF).
Profilo	<input checked="" type="checkbox"/> abilita la voce corrispondente del menu Applica Consigliato: <input type="checkbox"/> (OFF).
Correzione utensile	<input checked="" type="checkbox"/> abilita le voci a menu relative alla Correzione utensile
Misura	<input checked="" type="checkbox"/> abilita la voce corrispondente del menu Strumenti
Punto con mouse	<input checked="" type="checkbox"/> abilita la voce corrispondente del menu Imposta, voce: Posizionamento. Il tipo di posizionamento di <i>Punto con mouse</i> è utilizzabile in strumenti generali (Trasla, Ruota, Specula). Consigliato: <input type="checkbox"/> (OFF).
Punto progressivo	<input checked="" type="checkbox"/> abilita la voce corrispondente del menu Imposta, voce: Posizionamento. Il tipo di posizionamento di <i>Punto progressivo</i> è utilizzabile in strumenti generali (Trasla, Ruota, Specula). Consigliato: <input type="checkbox"/> (OFF).
Vista tridimensionale	<input checked="" type="checkbox"/> abilita la voce corrispondente del menu Vista. Il comando risulta comunque non abilitato se è dichiarato disabilitato l'Asse z (Imposta>> Abilitazioni>> Abilitazioni pezzo)
Vista macchine	<input checked="" type="checkbox"/> abilita la voce corrispondente del menu Vista (attivare solo con abilitate più macchine) Consigliato: <input type="checkbox"/> (OFF).
Mostra vista	<input checked="" type="checkbox"/> abilita la voce corrispondente del menu Vista Il comando è utile in caso di gestione abituale di programmi lunghi (esempio: 10000-30000 linee) che solitamente sono creati per importazione da altro formato: in questo caso è utile potere disattivare la grafica della faccia, in fase di modifica del programma (esempio per: Assegnazione tecnologiche, Assegnazioni complessive,..). Consigliato: <input type="checkbox"/> (OFF).
Menu strumenti	<input checked="" type="checkbox"/> abilita il menu Strumenti <input type="checkbox"/> elimina il menu Strumenti
Menu Costruisce	<input checked="" type="checkbox"/> abilita il menu Costruisce <input type="checkbox"/> elimina il menu Costruisce

_____ Pagina 1: Menu Strumenti

Trasla	<input checked="" type="checkbox"/> abilita la voce corrispondente del menu Strumenti
Centratura Orizzontale	
Centratura Verticale	
Ruota	
Simmetria Orizzontale	
Simmetria Verticale	
Simmetria Generica	
Ripete singolo	
Ripete multiplo	
Spezza profilo	
Inverte direzione	
Sposta setup	
Unisce profilo	
Linearizza arco	
Arco per 3 punti	
Raccorda spigolo	
Smussa spigolo	
Applica entrata	
Applica uscita	
Linearizza Z	
Trascina	

Pagina 3: Palette Strumenti

Permette di cambiare la composizione della palette Strumenti: il comando permette ad esempio di assegnare la palette con gli strumenti di uso più comune, lasciando invece comunque completa la lista delle voci presenti a menu. È possibile: eliminare e/o spostare i pulsanti della palette.

Pagina 4: Menu Costruisce

Percorso ricordato	<input checked="" type="checkbox"/> abilita la voce corrispondente del menu Costruisce
Profilo frammentato	
Profilo minimizzato	
Profilo con attacchi	

Pagina 5: Disegno

(assegnare solo se è abilitata voce Profilo di menu Applica)

Punto isolato	<input checked="" type="checkbox"/> abilita la voce corrispondente della palette di Disegno
Punto spline	Lasciare: <input type="checkbox"/> (OFF).
Assegna setup per i punti isolati	<input checked="" type="checkbox"/> in operatività di Disegno: i punti isolati sono inseriti come setup. Abilitare nel caso in cui non sono disponibili utensili di foratura.
Assegna tecnologia	<input checked="" type="checkbox"/> in operatività di Disegno: per i setup ed i punti isolati utilizza direttamente i codici tecnologici. Se abilitato: <ul style="list-style-type: none"> • per i punti isolati assegna: <ul style="list-style-type: none"> ◊ se non è abilitata la voce "Assegna setup per i punti isolati": la lavorazione impostata come "Codice puntuale di default" (vedi: Imposta>> Settaggi>> Database lavorazioni), con le relative assegnazioni tecnologiche; ◊ se è abilitata la voce "Assegna setup per i punti isolati": la lavorazione impostata come "Codice setup di default" (vedi: Imposta>> Settaggi>> Database lavorazioni), con le relative assegnazioni tecnologiche; • per i punti iniziali assegna: <ul style="list-style-type: none"> ◊ la lavorazione impostata come "Codice setup di default" (vedi: Imposta>> Settaggi>> Database lavorazioni), con le relative assegnazioni tecnologiche;
Livelli per menu Disegna	Assegna il livello di difficoltà del menu di Disegno. Posizionare le barre di scorrimento da valore 1 (livello minimo di difficoltà) a valore 4 (livello massimo di difficoltà)

2.2. Imposta: Settaggi

La finestra si compone di quattro pagine.

Pagina 1: Database lavorazioni

File e gruppo "db-client"	Sono impostati tre campi, a configurazione di un gruppo custom di lavorazioni: <ul style="list-style-type: none"> ➤ controllo on/off di abilitazione; ➤ campo per assegnare il nome dell'eseguibile (utilizzare: CUSTOM);
---------------------------	---

	<p>➤ numero del gruppo custom di lavorazioni (consigliato: 0).</p> <p>Se non è assegnato gruppo custom: EDICAD utilizza le sole lavorazioni assegnate nel file EDICADWR.DEF. Se è assegnato gruppo custom: EDICAD utilizza anche le lavorazioni assegnate nel file custom CUSTOM.DEF.</p> <p>Per maggiori dettagli si rimanda a documentazione dedicata.</p>
Help- Client	Nome del file di help per le lavorazioni custom. Utilizzare: CUSTOM.
Dimensione bitmap x-y	<p>Dimensioni x ed y dei bitmaps delle lavorazioni (unità: pixels). Valori validi per ogni dimensione: da 0 (non visualizza i bitmaps) a 250. Default: 80 * 80.</p> <p>I valori sono utilizzati per assegnare la dimensione dei pulsanti in lista grafica di selezione delle lavorazioni:</p> <ul style="list-style-type: none"> ◇ con bitmap più piccolo: è riportato centrato sul pulsante; ◇ con bitmap più grande: è visualizzato con riduzione alla dimensione del pulsante. <p>In finestra di impostazione della lavorazione, il bitmap della lavorazione è invece comunque visualizzato in dimensione reale.</p>
Visualizza bitmaps	<input checked="" type="checkbox"/> abilita la visualizzazione dei bitmap delle lavorazioni in finestre di: <ul style="list-style-type: none"> ◇ lavorazione; ◇ selezione lavorazione da lista di testo; ◇ assegnazione tecnologica.
Abilita codici di profilo	<input checked="" type="checkbox"/> abilita la gestione di profili (codici di arco, linea, setup, macro di profilo). <input type="checkbox"/> elimina la gestione dei codici di lavorazioni assegnati di tipologia: setup, arco o linea (non sono disponibili).
Codice puntuale di default	<p>Assegna il codice la lavorazione puntuale utilizzato nei casi:</p> <ul style="list-style-type: none"> ◇ inserimento di punti isolati in operatività di Profilo, ◇ assegnazione tecnologica di punti geometrici in fase di ottimizzazione. <p>Assegna anche i parametri di tecnologia.</p>
Codice di setup di default	<p>Il codice viene utilizzato nei casi:</p> <ul style="list-style-type: none"> ◇ per intestare profili non aperti da setup, ◇ inserimento di punti di inizio in operatività di Profilo, ◇ assegnazione tecnologica di setup geometrici in fase di ottimizzazione. <p>Assegna anche i parametri di tecnologia.</p>

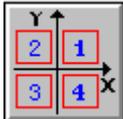
Pagina 2: Variabili ri

(la pagina di impostazioni è eliminata a partire dalla versione EDICAD-2.5.001)

Range fisso per variabili riassegnabili	Lasciare <input type="checkbox"/> (OFF).
Variabili riassegnabili	Impostare 100
Range fisso per variabili run-time	Lasciare <input type="checkbox"/> (OFF).
Variabili riassegnabili run-time	Impostare 100

Pagina 3: Riferimenti

Moltiplicatore di Epsilon	<p>Fattore moltiplicativo per la risoluzione geometrica. EDICAD gestisce un valore di risoluzione geometrica non nullo, indicato come <i>epsilon</i> ed assegnato per default al valore 0.001 mm: un punto P2 non è distinto da P1 se cade entro la circonferenza con centro in P1 e raggio pari ad <i>epsilon</i>. Il parametro qui impostato può cambiare il valore assegnato ad <i>epsilon</i>: impostare valore 100 equivale ad esempio ad assumere $\epsilon = (0.001 * 100) \text{ mm} = 0.1 \text{ mm}$.</p> <p>Valori validi: da 0.001 e 1000. Default: 1. Consigliato: 5 o 10 (impostare un valore tale da assegnare <i>epsilon</i> almeno 10 volte più piccolo della risoluzione degli assi di macchina). Il fattore <i>epsilon</i> non influenza la precisione di calcolo con cui EDICAD lavora, ma interviene per valutazioni di confronto, quali ad esempio:</p> <ul style="list-style-type: none"> ◇ soluzione di un arco con esecuzione di arco o cerchio: se i punti di inizio e di fine di un arco stanno ad una distanza inferiore ad <i>epsilon</i>, l'arco risolve un cerchio; ◇ test di validità geometria di arco: un arco è corretto se i raggi calcolati sul punto di inizio (distanza tra punto di inizio e centro) e sul punto di fine (distanza tra punto di fine e centro) differiscono per una quantità inferiore ad <i>epsilon</i>.
Origini in trasparenza	<input checked="" type="checkbox"/> abilita la gestione delle origini in trasparenza per le facce 2,5,e 6: gli assi x ed y sono in trasparenza rispetto alle 1, 3 e 4, mentre l'asse z rimane positivo in uscita dal pezzo; <input type="checkbox"/> ogni faccia assegna l'origine con terne cartesiane tra loro omogenee. Consigliato: <input checked="" type="checkbox"/>

Origine faccia 2	<p>Imposta l'origine adottata in faccia 2: 0= default; 1= xy in trasparenza alla faccia 1 2=opposto in x alla faccia 1. Valori validi: da 0 a 2. Interpreta valore 0 in caso di attivazione delle Origini in trasparenza.</p> <p>Consigliato: 0.</p>
Quadrante operativo	<p>Imposta il quadrante di lavoro della macchina (indica come sono orientati gli assi x ed y della macchina):</p>  <p>Il valore impostato è utilizzato dal programma OPTIMIZE e passato in matrice pezzo.</p> <p>Valori validi: da 1 e 4. Default: 1.</p>
Facce abilitate	<p>Maschera delle facce abilitate del parallelepipedo di assegnazione pezzo. La maschera è applicata in caso di pezzo non sagomato. Esempio: "135" → abilita le facce: 1, 3, 5. Per abilitare tutte le facce, impostare "123456". Se nessuna faccia viene indicata: sono tutte gestite (imposta in modo automatico: "123456").</p> <p>Una faccia non abilitata:</p> <ul style="list-style-type: none"> ◇ ha lista di programma vuoto, ◇ non è abilitato l'accesso alla vista di faccia, ◇ rimane però possibile assegnare facce fittizie simili a facce non abilitate. <p>In caso di macro-programma: EDICAD gestisce sempre tutte le facce del parallelepipedo.</p>

Pagina 4: Codici di trascodifica

(la pagina di impostazioni è eliminata a partire dalla versione EDICAD-2.5.001 e la solo voce "Codici CNC" è spostata in pagina "Database lavorazioni")

Codici CNC	<input checked="" type="checkbox"/> mantiene la lavorazioni associate alla conversione di programmi da CNC90. Consigliato: <input type="checkbox"/> se non è richiesta conversione da CNC90
Codici DXF	Lasciare <input type="checkbox"/> (OFF).
Codici DGT	Lasciare <input type="checkbox"/> (OFF).
Codici ISO	Lasciare <input type="checkbox"/> (OFF).

Pagina 5: Generali

File lungo [kbytes]	<p>Imposta la dimensione del file registrato (in Kbytes) per la quale vengono attivate procedure particolari per file lungo. Valori validi: da 0 e 1000. Default: 15. Valore 0 disattiva i controlli associati alla gestione di file lungo.</p> <p>In caso di file lungo:</p> <ul style="list-style-type: none"> ◇ in caricamento automatico di file MRU[0]: chiede conferma per avviare la lettura del file; ◇ dopo salvataggio del file : chiede conferma per avviare l'ottimizzazione.
Faccia lunga [linee]	<p>Imposta il numero di linee programmate per le quali vengono attivate procedure particolari per facce lunghe. Valori validi: da 0 e 10000. Default: 800. Valore 0 disattiva i controlli associati alla gestione di faccia lunga.</p> <p>In caso di faccia lunga:</p> <ul style="list-style-type: none"> ◇ in caso di comando che prevede Undo (Annulla): chiede conferma per eseguire copia della faccia (per permettere successivo annullamento del comando);
File di grafica testa	<input checked="" type="checkbox"/> abilita il caricamento del file HEAD.BMP di grafica della testa. Il file è assegnato per ogni macchina abilitata, in cartella: MOD.n\CONFIG (dove n= numero di modulo: 0 per macchina 1, 1 per macchina 2,..).
Altezza controllo statico di lavorazione	Imposta l'altezza dei controlli statici (leggi: scritte) visualizzati in finestra delle lavorazioni. Valori validi: da 5 e 20 (in pixels). Default: 10. Il settaggio è assegnato al fine di garantire che tutti i controlli statici assegnati in finestra di lavorazione con applicativo PASTUDIO.EXE siano visualizzati con la stessa altezza.
Altezza controllo di edit di lavorazione	Imposta l'altezza dei controlli di edit visualizzati in finestra delle lavorazioni. Valori validi: da 5 e 20 (in pixels). Default: 10.
Altezza pulsante di lavorazione	Imposta l'altezza dei pulsanti visualizzati in finestra delle lavorazioni. Valori validi: da 5 e 20 (in pixels). Default: 15.

Pagina 6: Palette lavorazioni

Viene qui definita la configurazione della palette delle lavorazioni.

La palette delle lavorazioni permette di accedere al database delle lavorazioni caricate da EDICAD.EXE, con lettura dai files:

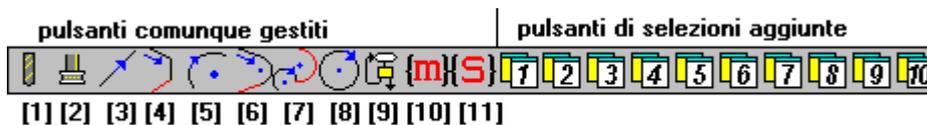
- EDICADWR.DEF: lavorazioni di base (assegnate in TPA e distribuite con la installazione di ALBATROS);
- ed eventualmente da CUSTOM.DEF: lavorazioni custom (assegnate dal costruttore).

Le lavorazioni gestite in EDICAD.EXE sono suddivise in "gruppi" (secondo criteri assunti di default oppure definiti dal costruttore) ed ogni gruppo "risponde" alla selezione di un pulsante della palette. La suddivisione delle lavorazioni in gruppi utilizza:

- ◇ la tipologia della lavorazione (puntuale, setup,...)
- ◇ parametri aggiunti.

EDICAD.EXE ha una palette di default per la selezione delle lavorazioni, che viene utilizzata nel caso in cui non è assegnata nessuna palette esterna e comunque in caso di edit di testo di macro-programma. La palette è gestita in due dimensioni:

- ◇ palette piccola (dimensione di ogni pulsante: 22 * 22 pixels)
- ◇ palette grande (dimensione di ogni pulsante: 33 * 33 pixels).



[1] gruppo delle lavorazioni puntuali (forature, inserimenti)

[2] gruppo delle lavorazioni di setup

[3] gruppo delle lavorazioni di tipo linea singola

[4] gruppo delle lavorazioni di smusso

[5] gruppo delle lavorazioni di tipo arco singolo

[6] gruppo delle lavorazioni di raccordo

[7] gruppo delle lavorazioni di tipo archi multipli (doppi archi, ovale)

[8] gruppo delle lavorazioni di tipo cerchio

[9] gruppo delle lavorazioni logiche (IF, logiche custom)

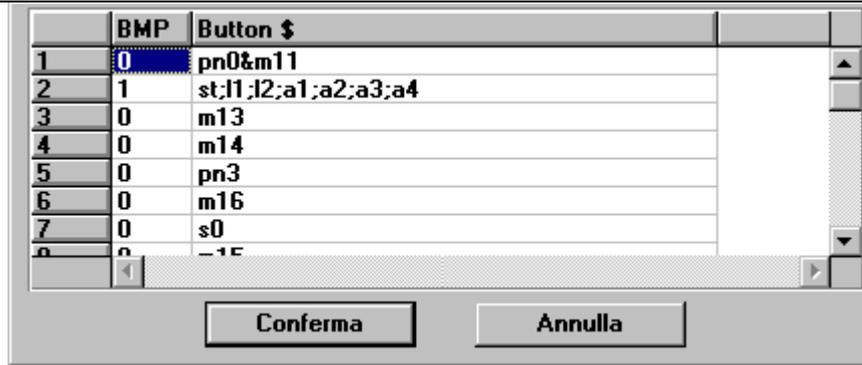
[10] sotto-gruppo delle lavorazioni di applicazione macro-programma (codici operativi in intervallo: 1001-2000) e personalizzati di chiamata sottoprogramma (codici operativi in intervallo: 4001-5000)

[11] gruppo delle lavorazioni di applicazione generica di sottoprogramma (SUB0,...).

Selezioni aggiunte in palette	Imposta il numero di pulsanti di "selezioni aggiunte" visualizzati nella palette di default. Ogni pulsante di "selezione aggiunta" raggruppa parte delle lavorazioni di applicazione di un macro-programma (codici operativi in intervallo: 1001-2000) o applicazione dedicata di sottoprogramma (codici operativi in intervallo: 4001-5000)
-------------------------------	--

Maggiori dettagli sulla palette di default sono dati più avanti.

Abilita palette esterna	<input checked="" type="checkbox"/> abilita il caricamento di una palette esterna
File di palette (small - big)	<p>Sono gestiti due campi ad assegnazione del nome del bitmap da caricare (estensione è aggiunta fissa = BMP), in caso di utilizzo di:</p> <ul style="list-style-type: none"> ◇ palette piccole (nome consigliato: PALWS) ◇ palette grandi (nome consigliato: PALWB). <p>I files PALWS.BMP, PALWB.BMP sono cercati in cartella <i>cadcfg</i>.</p> <p>La dimensione della palette è comunque libera e letta automaticamente dal bitmap: assegnare il bitmap con una sola riga di pulsanti di dimensioni quadrate (dimensione di larghezza = dimensione di altezza * numero di pulsanti). È possibile assegnare un unico file (impostare lo stesso nome nei due campi, esempio: PALWS).</p> <p>La figura è un esempio di palette esterna, con assegnati 10 pulsanti:</p>
Bitmap esterno	Il pulsante apre una pagina di configurazione del bitmap esterno:



Viene aperta una tabella di 25 righe (numerata da 1 a 25):

- ◇ ogni riga corrisponde ad un pulsante della palette (PALWS.BMP)
- ◇ la tabella lascia righe non assegnate in coda.

Campi per ogni riga:

BMP colonna di valore numerico:

0 = il pulsante apre direttamente la selezione di un gruppo di lavorazioni;

1-3 = il pulsante rinvia ad una palette secondaria (sempre assegnata esterna).

È consentito utilizzare fino a tre palette secondarie, con nome assegnato a partire dal nome della palette primaria. Nel nostro esempio (palette primaria: PALWS.BMP e PALWB.BMP):

- PALWS0.BMP e PALWB0.BMP per palette di numero 1
- PALWS1.BMP e PALWB1.BMP per palette di numero 2
- PALWS2.BMP e PALWB2.BMP per palette di numero 3;

BUTTON \$ colonna di impostazione di stringa, ad indicazione dei tipi di lavorazione gestiti sul pulsante. In caso di valore BMP significativo (1,2,3): caratterizza l'intera palette secondaria.

Esempio di palette complessa:

BMP	Button\$	Gruppo di lavorazioni																
1	0	pn0&pn1&m11 puntuali di tipologia 0 + puntuali di tipologia 1 + cicli (sub e/o macro) di numero 11																
2	0	pn3 puntuali di tipologia 3;																
3	1	st;l1;l2;a1;a2;a3;a4; apre palette secondaria PALWS0.BMP. In questa palette assegna i pulsanti: <table border="0" style="margin-left: 20px;"> <tr> <td>1</td><td>setup (tutti)</td> <td>3</td><td>lineari l2 (smussi)</td> </tr> <tr> <td>2</td><td>linee l1 (linee singole)</td> <td>5</td><td>archi a2 (raccordi)</td> </tr> <tr> <td>4</td><td>archi a1 (archi singoli)</td> <td>7</td><td>archi a4 (cerchi)</td> </tr> <tr> <td>6</td><td>archi a3 (archi multipli)</td> <td></td><td></td> </tr> </table>	1	setup (tutti)	3	lineari l2 (smussi)	2	linee l1 (linee singole)	5	archi a2 (raccordi)	4	archi a1 (archi singoli)	7	archi a4 (cerchi)	6	archi a3 (archi multipli)		
1	setup (tutti)	3	lineari l2 (smussi)															
2	linee l1 (linee singole)	5	archi a2 (raccordi)															
4	archi a1 (archi singoli)	7	archi a4 (cerchi)															
6	archi a3 (archi multipli)																	
4	0	m14 codici di applicazione macro-programma e/o dedicata di sottoprogramma di tipologia 14;																
5	0	m12&m13 codici di applicazione macro-programma e/o dedicata di sottoprogramma di tipologie 12 e 13;																
6	0	lg1 logiche di tipologia 1;																
7	0	lg0 logiche di tipologia 0;																
8	0	s0 codici di applicazione generica di sottoprogramma;																
9	0	m15 codici di applicazione macro-programma e/o dedicata di sottoprogramma di tipologia 15;																

La sintassi del campo "BUTTON \$" utilizza:

- ◇ sigle di tipologie gruppi di lavorazioni (pn0, st, l1, m12,..)
- ◇ carattere & di somma tra tipologie (pn0&pn1): al massimo si possono sommare 5 sigle di tipologie. Si raccomanda: tutte le tipologie sommate con carattere & devono avere stesse possibilità di applicazione e questo per determinare lo stato "attivo / non attivo" del pulsante (sarebbe sbagliato assegnare un pulsante come: a1&pn0). Comunque: **la prima tipologia indicata determina le possibilità di applicazione del pulsante;**
- ◇ carattere ; separatore dei pulsanti di palette secondaria (st;l1;l2;a1;a2;a3;a4;).

Lavorazioni	Sigle di tipologie	Gruppo
Puntuali	pn pn0;pn1;pn2;pn3;pn4	tutte le lavorazioni di tipologia puntuale; lavorazioni puntuali con parametro [8112] di valore rispettivamente: 0, 1, 2, 3 4;
Setup	st st0;st1;st2;st3;st4	tutte le lavorazioni di tipologia setup; lavorazioni di setup con parametro [8112]

		di valore rispettivamente: 0, 1, 2, 3 4;
Lineari	l0 l1 l2	tutte le lavorazioni di tipologia lineare; codici lineare con esecuzione di 1 tratto; codici lineare di smusso;
Archi	a0 a1 a2 a3 a4	tutte le lavorazioni di tipologia arco; codici arco con esecuzione di 1 tratto; codici arco di raccordo; codici di archi multipli; codici di cerchio
Codici di applicazione Sottoprogrammi		
	s0	codici generici di sottoprogrammi (SUB0, SUB1,..) + codici in range [4001-5000], senza parametro [8117] o =0;
Codici di applicazione Macro-programmi o Sottoprogrammi		
	m0	codici in range [1001-2000], senza parametro [8117] o =0;
	m1;m2;...;m10	codici in range [1001-2000] e [4001-5000] con valore di parametro [8117] da 1 a 10;
	m11;m12;...;m20	codici in range [1001-2000] e [4001-5000] con valore di parametro [8117] da 11 a 20;
Logiche	lg lg0;lg1;lg2;lg3;lg4	tutte le lavorazioni di tipologia logica; lavorazioni logiche con parametro [8112] di valore rispettivamente: 0, 1, 2, 3 4

La palette lavorazioni di default corrisponde alle impostazioni:

	BMP	Button\$	Descrizione
1	0	pn	tutte le lavorazioni puntuali (forature, inserimenti,..)
2	0	st	tutte le lavorazioni di setup
3	0	l1	linee singole
4	0	l2	smussi
5	0	a1	archi singoli
6	0	a2	raccordi
7	0	a3	archi multipli
8	0	a4	cerchi
9	0	lg	tutte le lavorazioni logiche
10	0	m0	codici in range [1001-2000], senza parametro [8117]; codici in range [1001-2000], con valori di parametro [8117] da 11 a 20
11	0	s0	codici generici di applicazione sottoprogrammi (SUB0, SUB1,..) + codici in range [4001-5000], senza parametro [8117]; codici in range [4001-5000], con valori di parametro [8117] da 11 a 20;
12	0	m1	codici in range [1001-2000] e [4001-5000] con parametro [8117] = 1;
..	0
21	0	m10	codici in range [1001-2000] e [4001-5000] con parametro [8117] = 10;

In caso di assegnazione di una palette esterna di lavorazione, è necessario provvedere alla assegnazione dei [tooltiptext](#) di ogni pulsante della palette: i messaggi che vengono visualizzati quando il mouse è posizionato per un breve intervallo di tempo sul pulsante.

I messaggi sono letti da file CSTBREAK.LNG, registrato in cartella *cadcfg*

Numero di messaggio	Pulsante
10000	palette principale (PALWS.BMP, PALWB.BMP), pulsante 1
10001	palette principale, pulsante 2
..	
10024	palette principale, pulsante 25
10025	palette secondaria 1 (PALWS0.BMP, PALWB0.BMP), pulsante 1
.. 10049	palette secondaria 1, pulsante 25
10050	palette secondaria 2 (PALWS1.BMP, PALWB1.BMP), pulsante 1
.. 10074	palette secondaria 2, pulsante 25
10075	palette secondaria 3 (PALWS2.BMP, PALWB2.BMP), pulsante 1
.. 10099	palette secondaria 3, pulsante 25

2.3. Imposta: Abilitazioni

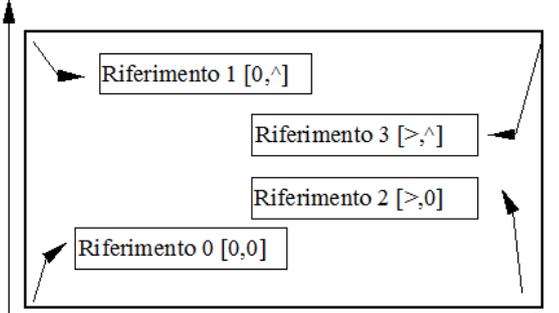
La finestra si compone di tre pagine.

_____ Pagina 1: Abilitazioni pezzo

Pezzo sagomato	<input checked="" type="checkbox"/> abilita la gestione di pezzi sagomati con profilo programmato in faccia 1
Pezzo sagomato laterale	<input checked="" type="checkbox"/> abilita la gestione di pezzi sagomati con profilo programmato in faccia 3 o 4
Facce fittizie	<input checked="" type="checkbox"/> abilita la gestione di facce fittizie
Sequenza facce	<input checked="" type="checkbox"/> abilita la impostazione del campo sequenza di faccia <input type="checkbox"/> la sequenza di ogni faccia rimane fissa a 0. Consigliato: <input type="checkbox"/> (OFF).
Imposta ottimizzazioni	<input checked="" type="checkbox"/> abilita la personalizzazione dei settaggi di ottimizzazione localmente ad un programma; <input type="checkbox"/> Valgono per ogni programma le stesse impostazioni di ottimizzazione (come assegnate in: <i>Imposta, Ottimizzatore</i>)
Variabili ciclo	<input checked="" type="checkbox"/> abilita la gestione delle variabili di ciclo
Offsets pezzo	<input checked="" type="checkbox"/> abilita la gestione degli offsets
Controllo vincoli	Lasciare: <input type="checkbox"/> (OFF). (la voce è eliminata a partire dalla versione EDICAD-2.5.001, con gestione automatica di valore 0=OFF)
Correzione utensile in contornatura (pezzo non sagomato)	Applica a pezzo non sagomato o a faccia non di proiezione di pezzo sagomato: <input checked="" type="checkbox"/> abilita la soluzione di raccordi di correzione utensile con intersezioni; <input type="checkbox"/> risolve raccordi di correzione [default].
Correzione utensile in contornatura (pezzo sagomato)	Applica a faccia di proiezione di pezzo sagomato: <input checked="" type="checkbox"/> abilita la soluzione di raccordi di correzione utensile con intersezioni; <input type="checkbox"/> risolve raccordi di correzione [default].
Correzione utensile a Zp	<input checked="" type="checkbox"/> abilita la gestione della correzione utensile con partenza dal setup originale dei profili (opzione utile per la lavorazione del marmo). Consigliato: <input type="checkbox"/> (OFF).
Asse z	<input checked="" type="checkbox"/> abilita la gestione dell'asse di profondità delle facce In caso di voce non abilitata: <ul style="list-style-type: none"> ◇ gli strumenti eliminano la gestione sull'asse di profondità (esempio: la traslazione non propone il campo di traslazione in z); ◇ viene disabilitata la gestione della Vista tridimensionale; ◇ viene disabilitata la gestione delle lavorazioni con codici di arco in piano # xy. Consigliato: <input checked="" type="checkbox"/> (ON).
Variabili tipo j	<input checked="" type="checkbox"/> abilita la gestione delle variabili globali di tipo j Consigliato: <input type="checkbox"/> (OFF).
Utilizzo esteso delle variabili j	<input checked="" type="checkbox"/> abilita la gestione delle variabili in assegnazione qualunque (non solo in test di condizioni logiche). Consigliato: <input type="checkbox"/> (OFF).
Assegna codici geometrici in macro	Lasciare: <input checked="" type="checkbox"/> (ON). (la voce è eliminata a partire dalla versione EDICAD-2.5.001, con gestione automatica di valore 1=ON)
Setup multipli	<input checked="" type="checkbox"/> abilita la gestione dei setup multipli. Consigliato: <input checked="" type="checkbox"/> (ON). In caso di voce abilitata, viene gestita la possibilità di eseguire più volte un profilo, con differenti setup assegnati. È riconosciuto il formalismo: <ul style="list-style-type: none"> ◇ 1° setup: programmazione assoluta, aggancio di punto non richiesto, non di costruito; ◇ 2° setup: aggancio di punto richiesto, non di costruito; ◇ setup successivi: aggancio di punto richiesto, non di costruito; ◇ profilo di seguito definito con lavorazioni elementari (archi, linee); ◇ a valle del profilo: aggancio di punto non richiesto; se sono verificate le condizioni sopra poste: in fase di ottimizzazione vengono realmente sviluppati tanti profili quanti sono i setup specificati. Lo sviluppo avviene anche in caso di profilo assegnato (<u>interamente</u>) in sottoprogramma o macro. In una faccia possono essere assegnati fino a 20 profili con setup multipli. Per ogni profilo non viene invece gestito un numero massimo di setups. Per ogni setup è possibile assegnare: una profondità differente (ogni profilo aggiunto viene traslato - rispetto a quello originale - della differenza di profondità risultante rispetto al primo setup), propria correzione utensile (lato, raggio fresa,..), tecnologia (utensile, velocità,..).

	Le condizioni di setup multipli sono valutate a programma non modificato (esempio: dopo apertura di un programma o con comando Compila). In queste condizioni, in finestra di lavorazioni e di proprietà: per il setup di apertura viene indicato il messaggio "Setup di apertura profilo multiplo" ed un bitmap.
Sub-Macro multipli	<input checked="" type="checkbox"/> abilita la applicazione delle chiamate indotte di sottoprogramma o macro Consigliato: <input checked="" type="checkbox"/> (ON).
Riduce lista di sottoprogramma	Lasciare: <input checked="" type="checkbox"/> (ON). (la voce è eliminata a partire dalla versione EDICAD-2.5.001, con gestione automatica di valore 1=ON)
Riduce compilazione di sottoprogramma	<input checked="" type="checkbox"/> abilita la applicazione diretta dei condizionamenti logici in applicazione di sottoprogramma (vengono compilate solo le lavorazioni che risultano verificate dai condizionamenti logici); <input type="checkbox"/> in applicazione di sottoprogramma, i condizionamenti logici sono valutati ed applicati dopo avere comunque compilato tutte le linee del sottoprogramma. Consigliato: <input checked="" type="checkbox"/> (ON). Per maggiori dettagli si rimanda a documentazione dedicata.

Pagina 2: Abilita in lavorazioni

Origine alta [0,^] Origine destra [>,0] Origine alta, destra [>,^]	<input checked="" type="checkbox"/> abilitano ogni campo la gestione del campo O di lavorazione, con la selezione indicata; <input type="checkbox"/> selezione non gestita [default] <p>Il campo O è un attributo numerico di lavorazione, con numerazione valida da 0 a 3 (valore 0 corrisponde alla assegnazione implicita). La assegnazione può essere effettuata:</p> <ul style="list-style-type: none"> ✓ per la singola lavorazione; ✓ con selezione multipla di lavorazioni. <p>È qui possibile abilitare selettivamente ogni singolo valore del campo O: ad esempio abilitare solo il valore 2 e non i valori 1 e 3. Per disabilitare completamente la gestione del campo O: selezionare non gestiti tutti i tre campi.</p> <p>L'utilizzo del campo O è finalizzato alla fase esecutiva: un uso tipico è quello di indicare lo spigolo della faccia attribuito come riferimento per la lavorazione applicata. Non si tratta in nessun modo di programmazione su un sistema di riferimento locale alla faccia, ma solo di una attribuzione di riferimento. L'informazione di campo O può comunque essere trattata con modalità specifiche per ogni singola applicazione. Considerando come prevalente l'utilizzo del campo O ad assegnazione dello spigolo di riferimento della faccia, sono indicate disponibili le selezioni indicate in figura:</p>  <p>Si definiscono alcune specifiche:</p> <ul style="list-style-type: none"> ✓ un profilo (dal setup fino alla risalita) è definito interamente con il valore di campo O come impostato per il setup; ✓ in chiamata di macro o sottoprogramma: a tutte le lavorazioni derivanti viene attribuito il valore assegnato in chiamata alla macro o sottoprogramma, ma solo in caso di assegnazione non nulla. Nel caso in cui il sottoprogramma (o macro) è utilizzato valore nullo: ogni lavorazione espansa mantiene la propria assegnazione.
Livelli	<input checked="" type="checkbox"/> abilita la gestione del campo Livello di lavorazione; <input type="checkbox"/> campo non gestito [default]. <p>Il livello è un attributo numerico di lavorazione. L'utilizzo del livello è finalizzato alla sola fase di programmazione e come tale può essere definito un ausilio alla programmazione e visualizzazione del programma. Sono utilizzabili 8 differenti livelli (numerazione da 1 a 8) oltre al livello 0 che corrisponde a</p>

	<p>nessun livello imposto. La assegnazione può essere effettuata:</p> <ul style="list-style-type: none"> ✓ per la singola lavorazione; ✓ con selezione multipla di lavorazioni. <p>Si definiscono alcune specifiche:</p> <ul style="list-style-type: none"> ✓ un profilo è definito interamente sul livello imposto per il setup; ✓ in chiamata di macro o sottoprogramma: a tutte le lavorazioni derivanti viene attribuito il livello assegnato in chiamata alla macro o sottoprogramma. <p>È possibile assegnare un colore ad ogni livello, utilizzato per rappresentare le lavorazioni. Per il livello 0:</p> <ul style="list-style-type: none"> ➤ non è assegnato colore ➤ ogni lavorazione ha la rappresentazione grafica di default o come configurata dal database delle lavorazioni; <p>I menu di selezione e vista prevedono comandi di selezione e visualizzazione selettivi in base ai livelli attribuiti.</p> <p>La assegnazione di livelli deve essere considerata alternativa alla abilitazione di <i>Vista macchine</i> (Imposta, Menu). Si tratta infatti di gestioni simili e tra loro conflittuali: in caso di conflitto, prevale il colore impostato per la Macchina.</p>
Legami	<p>Lasciare: <input type="checkbox"/> (OFF). (la voce è eliminata a partire dalla versione EDICAD-2.5.001, con gestione automatica di valore 0=OFF)</p>
Campo M	<p><input checked="" type="checkbox"/> abilita la gestione del campo M di lavorazione; <input type="checkbox"/> campo non gestito [default].</p> <p>Il campo M è un attributo numerico di lavorazione (valore da 0 a 65000). La assegnazione può essere effettuata:</p> <ul style="list-style-type: none"> ✓ per la singola lavorazione; ✓ con selezione multipla di lavorazioni. <p>Si definiscono alcune specifiche:</p> <ul style="list-style-type: none"> ✓ un profilo è definito interamente sul valore imposto per il setup; ✓ in chiamata di macro o sottoprogramma: a tutte le lavorazioni derivanti viene attribuito il valore assegnato in chiamata alla macro o sottoprogramma, ma solo in caso di assegnazione non nulla. Nel caso in cui il sottoprogramma (o macro) è utilizzato con valore di campo M nullo: ogni lavorazione espansa mantiene la propria impostazione.

Filtri in aggancio di fresata:

Applica filtri	<p><input checked="" type="checkbox"/> abilita la gestione dei filtri attivati per valutare la effettiva giunzione di fresate (leggi: con eliminazione del setup intermedio), in aggancio di punto programmato; <input type="checkbox"/> filtri non gestiti [default]. Consigliato: <input type="checkbox"/> (OFF).</p>
Codice	<input checked="" type="checkbox"/> abilita filtro: i codici operativi dei setup devono essere uguali
Sottotipo	<input checked="" type="checkbox"/> abilita filtro: i sotto-tipi dei codici devono essere uguali
Priorità tecnologica	<p>Lasciare: <input type="checkbox"/> (OFF). (la voce è eliminata a partire dalla versione EDICAD-2.5.001, con gestione automatica di valore 0=OFF)</p>
Utensile	<p><input checked="" type="checkbox"/> abilita filtro: l'utensile deve essere lo stesso, inteso come (macchina, gruppo, utensile). Nel caso di filtro attivo anche per il diametro utensile, la corrispondenza tra gli utensili è verificata solo tra i campi effettivamente validi in entrambe le lavorazioni (valori #0). A tale proposito, si rammentano le assunzioni di default applicati ai parametri di macchina, gruppo ed utensile:</p> <ul style="list-style-type: none"> ◇ macchina: assume il valore impostato (0 se campo vuoto) se il parametro è esiste nella lavorazione; valore 1 altrimenti; ◇ gruppo: come macchina; ◇ utensile: assume il valore impostato (0 se campo vuoto) se il parametro è esiste nella lavorazione; valore 0 altrimenti (NON ESISTE UTENSILE DI DEFAULT).
Diametro utensile	<input checked="" type="checkbox"/> abilita filtro: il diametro utensile deve essere lo stesso
Tipologia utensile	<input checked="" type="checkbox"/> abilita filtro: la tipologia dell'utensile deve essere uguale
Correzione utensile	<input checked="" type="checkbox"/> abilita filtro: la correzione utensile (Off/ Destra/ Sinistra) deve essere la stessa
Raggio di correzione	<input checked="" type="checkbox"/> abilita filtro: il raggio di correzione utensile deve essere lo stesso
Origine	<input checked="" type="checkbox"/> abilita filtro: l'origine deve essere la stessa
Legame	<p>Lasciare: <input type="checkbox"/> (OFF). (la voce è eliminata a partire dalla versione EDICAD-2.5.001, con gestione automatica di valore</p>

	0=OFF)
Campo M	<input checked="" type="checkbox"/> abilita filtro: il valore di campo M deve essere lo stesso

Pagina 3: Abilitazioni pezzo run-time e di testo

Abilita la elaborazione dei codici geometrici	<input checked="" type="checkbox"/> abilita la esecuzione dei blocchi (puntuali) assegnati con codice geometrico COPGPNT [default] <input type="checkbox"/> i blocchi indicati non saranno eseguiti Lasciare: <input checked="" type="checkbox"/> (ON). (la voce è eliminata a partire dalla versione EDICAD-2.5.001, con gestione automatica di valore 1=ON).
Calcolo ingombro profili	<input checked="" type="checkbox"/> abilita il calcolo degli ingombri dei singoli profili <input type="checkbox"/> gli ingombri non sono calcolati Lasciare: <input checked="" type="checkbox"/> (ON). (la voce è eliminata a partire dalla versione EDICAD-2.5.001, con gestione automatica di valore 1=ON).
Scrittura di codici formali in testo ASCII	Lasciare: <input type="checkbox"/> (OFF). (la voce è eliminata a partire dalla versione EDICAD-2.5.001, con gestione automatica di valore 0=OFF).
Scrittura report di errori in testo ASCII	Lasciare: <input type="checkbox"/> (OFF). (la voce è eliminata a partire dalla versione EDICAD-2.5.001, con gestione automatica di valore 0=OFF).
Componi testo ASCII ridotto	<input checked="" type="checkbox"/> abilita la registrazione compatta delle linee di lavorazioni (se possibile: registra la lavorazione su una sola riga). Se abilitata: questa opzione non è compatibile con versioni di EDICAD precedenti la 2.0.000 (leggi: versioni precedenti non leggono correttamente programmi registrati con questa opzione).
Archivia macro con testo ASCII	<input checked="" type="checkbox"/> abilita la registrazione dei testi di macro-programma in formato ASCII. Se abilitata: questa opzione non è compatibile con versioni di EDICAD precedenti la 2.2.001 (leggi: versioni precedenti non leggono macro-programmi registrati con questa opzione).

Pagina 4: Abilitazioni in lettura programma

(la pagina di impostazioni è eliminata a partire dalla versione EDICAD-2.5.001 e le voci spostate in pagina precedente)

Attiva in edit	<input checked="" type="checkbox"/> abilita la applicazione delle procedure indicate (riduzione di profili) in esecuzione di EDICAD.EXE
Attiva run-time	<input checked="" type="checkbox"/> abilita la applicazione delle procedure indicate (riduzione di profili) in esecuzione di OPTIMIZE.EXE
Angolo di riduzione profili	Valore dell'angolo di riduzione dei profili in fase di lettura di programma (valori: da 0° a 20°). Se attivo: in lettura di programma è automaticamente applicata una riduzione dei profili, in base alla valutazione dell'orientamento di tratti consecutivi (con procedura simile allo strumento di <i>Minimizzazione profilo</i>). Attenzione: abilitare solo in caso di programmi originali generati con procedure esterne di apprendimento (scanner) e conseguente generazione di programmi molto lunghe (numero di linee superiori a quelle gestite da EDICAD: 32500), in programmazione rigorosamente assoluta e numerica (in caso contrario: non applica).

2.4. Imposta: Importazioni

Sono previste 10 codifiche di conversione di formato.

Ogni codifica è assegnata da :

Linea di comando	Cammino di ricerca dell'eseguibile di conversione. Esempio: c:\albatros\bin\cnvcnc90.exe.
Menu text	Testo di selezione della conversione. Esempio: "Converte da CNC90".
Argomenti	Argomenti da passare all'eseguibile di conversione. L'impostazione non è obbligatoria. In caso di assegnazione di un eseguibile di conversione a 32Bit: è necessario impostare l'argomento "/z" (scatena un meccanismo di interlock automatico tra EDICAD e convertitore, necessario per un corretto dialogo tra i due).
Tipi di File	Testo descrittivo dei file da convertire. Esempio: "Files CNC90", "DXF Files "
Nomi di file	Estensioni valide per i programmi. Esempi: "*. * ", "*.DXF "
Direttorio iniziale	Cartella di apertura della finestra di <i>File Apri</i> per la selezione del programma da convertire. L'impostazione non è obbligatoria. Esempi: "C:\CNC90\MOD1\SORG\SRG ", "C:\FORMAT\DXF"
Tipo automatico	Sigla di tipologia di programmi per la quale vale una conversione automatica (è possibile

	<p>avviare la conversione direttamente da comando: <i>File>>Apri</i>).</p> <p>Sono valide le sigle:</p> <p>CNC per conversione di testi in formato Cnc90-Tpa DGT per conversione di testi in formato DGT (formato Tpa) DXF per conversione di testi in formato DXF ISO per conversione di testi in formato ISO.</p> <p>L'impostazione non è obbligatoria.</p>
--	--

2.5. Imposta: Ottimizzatore

Le assegnazioni sono prevalentemente relative al programma di ottimizzazione OPTIMIZE.EXE.

Pagina 1: Settaggi di ambiente

Si tratta di settaggi generali, validi per tutti i programmi e non modificabili localmente al programma medesimo.

Optimize in Edicad	<input checked="" type="checkbox"/> abilita la corrispondente voce in menu File
Optimize dopo Salva	<input checked="" type="checkbox"/> abilita l'ottimizzazione dopo un salvataggio di programma
Macchine abilitate	Assegnare una stringa con indicazione dei numeri di macchine abilitate (per queste macchine i programmi EDICAD ed OPTIMIZE leggono le parametriche). È ammesso abilitare fino ad 8 macchine, con numeri validi da '1' ad '8'. Esempio: "124" per abilitare le macchine: 1, 2 e 4.
Nome matrice di macchina	Imposta il nome della matrice di configurazione dei gruppi macchina. Il nome di default è: "MxCnfMac". Il numero massimo di righe lette in matrice di configurazione dei gruppi macchina è: ◇ 100 fino alla versione 2.5.002a ◇ 200 a partire dalla versione 2.5.003. Il numero massimo delle colonne è 64.
Nome matrice dei gruppi	Imposta il nome della matrice di configurazione degli utensili. Il nome di default è: "MxCnfGru".
Numero di mandrini per gruppo	Imposta il numero di mandrini più alto di tutti i gruppi. Il numero massimo è 1000. Default: 200.
Numero di colonne in matrice pezzo	Imposta il numero di colonne che compongono la matrice pezzo. Default: 64 (=valore massimo).
Sequenza facce	<p>Assegnare una stringa con indicazione dell'ordine di esecuzione imposto alle facce, se differente da quella di default (progressiva sul numero della faccia, a partire da faccia 1). Esempio: "5;3;4".</p> <p>Questa assegnazione può sommarsi a quella impostata su ogni singola faccia (Settaggio in Imposta>> Abilitazioni>> Abilitazione pezzo), il cui uso è ora sconsigliato.</p> <p>Si riporta in dettaglio come le differenti assegnazioni di sequenza sono risolte:</p> <p>caso 1 tutte le facce del pezzo hanno sequenza non impostata (=0):</p> <ul style="list-style-type: none"> ◇ si assume la sequenza di default (settaggio dell'Ottimizzatore); ◇ le facce non in elenco sono lavorate dopo, in ordine di numerazione progressiva; ◇ in caso di pezzo sagomato: la faccia generatrice e la faccia di lavoro sono risolte per prime; <p>caso 2 esistono facce del pezzo con sequenza assegnata:</p> <ul style="list-style-type: none"> ◇ prima esegue le facce con sequenza #0 (come assegnato); ◇ se ci sono facce con sequenza non assegnata: sono lavorate dopo, in ordine di numerazione progressiva. <p>Si supponga di impostare un ordine di default = 4;1;2;3;5;6. Il pezzo ha assegnate:</p> <ul style="list-style-type: none"> ◇ le sei facce di pezzo parallelepipedo, con sequenza assegnata non nulla solo per la faccia 3 (sequenza 1); ◇ 2 facce fittizie (di numero: 7 e 8), con sequenza nulla. <p>La sequenza di ottimizzazione sarà: 3; 4; 1; 2; 5; 6; 7; 8.</p>
Ottimizzatore custom	<p>Sono impostati quattro campi, a configurazione di un eseguibile di ottimizzazione custom:</p> <ul style="list-style-type: none"> • un controllo on/off di abilitazione; • campo per assegnare il nome dell'eseguibile (massimo 8 caratteri, senza estensione) (Esempio: CUSTOPTI); • <i>Ordina custom</i>: <input checked="" type="checkbox"/> sono delegati all'ottimizzatore custom: <ul style="list-style-type: none"> ✓ i controlli relativi agli aggregati – frese, lame -; ✓ l'ordinamento della matrice pezzo;

	<ul style="list-style-type: none"> ✓ la sequenza di esecuzione utensili; ✓ la gestione delle sequenze di esecuzione; <ul style="list-style-type: none"> • <i>Traduttore custom</i>: <input checked="" type="checkbox"/> se l'ottimizzatore custom genera il file finale secondo criteri differenti da quelli standard di creazione della matrice pezzo; <p>Le modalità di avvio dell'ottimizzatore custom sono definite oltre.</p>
Assegna sequenze	<p><input checked="" type="checkbox"/> abilita la gestione delle sequenze di esecuzione (deve essere assegnato un Ottimizzatore custom con abilitato flag "<i>Ordina custom</i>": in caso contrario la abilitazione è ignorata). Si tratta di una abilitazione generale:</p> <ul style="list-style-type: none"> ◇ se attiva, consente poi di assegnare la gestione on/off a livello del singolo programma; ◇ se non attiva, comunque non è attivabile a livello del singolo programma. <p>=====</p> <p>Le sequenze di esecuzione sono assegnate in automatico a livello dell'intero pezzo e sono modificabili tramite procedura dedicata di EDICAD.EXE: il significato è di assegnare l'ordine con cui sono eseguite le lavorazioni di un pezzo, in modo anche trasversale sulle facce. Attenzione:</p> <ul style="list-style-type: none"> ◇ la sequenza di esecuzione non modifica l'ordine in lista di programma; ◇ il rispetto della sequenza in esecuzione è delegato completamente all'ottimizzatore custom. <p>A livello di programma, la sequenza è definibile come una proprietà delle lavorazioni. Solo: la gestione e la relativa assegnazione non è locale alla lavorazione o alla faccia di applicazione della lavorazione, ma complessiva per l'intero programma. L'assegnazione della sequenza di una lavorazione comporta la gestione di due informazioni:</p> <ol style="list-style-type: none"> 1) il progressivo di sequenza (univoco per l'intero pezzo, appunto) 2) un flag di ottimizzazione. <p>In automatico viene gestito il progressivo di sequenze: incrementato ad ogni nuovo inserimento di lavorazione. Occorre fare alcune precisazioni:</p> <ul style="list-style-type: none"> ◇ in caso di profilo: la sequenza è assegnata solo per il setup; ◇ in caso di lavorazione logica: la sequenza è assegnata solo per i codici in range custom; ◇ in database lavorazioni: è possibile escludere la gestione della sequenza per una singola lavorazione (tipicamente: per le lavorazioni custom). <p>In caso di lettura di un programma che non ha assegnati i progressivi di sequenza: viene assegnata una sequenzialità che rispetta il testo originale (in particolare: rispetta anche il frazionamento di una faccia in più sezioni).</p> <p>La assegnazione diretta delle sequenze di un pezzo consente di definire una struttura <i>massima</i> di programma:</p> <ol style="list-style-type: none"> a) in testa: blocchi in sequenza fissa (flag di ottimizzazione associato al campo sequenza non attivo); b) in mezzo: blocchi ottimizzabili (flag di ottimizzazione associato al campo sequenza attivo); c) in coda: blocchi in sequenza fissa (flag di ottimizzazione associato al campo sequenza non attivo). <p>Non è necessaria la presenza di tutti i tre segmenti di programma. È gestita una dimensione massima di programma su cui poter attivare le sequenze: numero complessivo di linee del pezzo = 32500.</p> <p>In caso di sequenza abilitata (sia a livello generale sia per il singolo programma) il modulo OPTIMIZE:</p> <ul style="list-style-type: none"> ✓ in caso di profilo: è significativo solo il progressivo di sequenza del setup; ✓ in caso di applicazione di sottoprogramma, in matrice pezzo sono riportati: il progressivo di sequenza primario (relativo alla chiamata del sottoprogramma) e quello secondario (corrisponde alla sequenza assegnata nel testo del sottoprogramma). In caso di chiamata indotta su altre facce: se il sottoprogramma non è stato registrato con sequenza attiva, per tutte le facce utilizzate il progressivo attribuito parte da 1; ✓ in caso di sviluppo di macro: viene riportato significativo solo il progressivo primario (relativo alla chiamata della macro), mentre il progressivo secondario è sempre 0; <ul style="list-style-type: none"> ✓ matcha solo le lavorazioni che hanno attivo il flag di ottimizzazione ed a queste, se richiesto, applica i criteri di ordinamento; ✓ i blocchi ottimizzati arrivano in matrice pezzo tutti con lo stesso progressivo primario di sequenza (ad esempio: il primo della lista) e progressivo secondario =-1; ✓ la sequenza dei cambi utensile viene scritta rispettando l'ordine delle lavorazioni riportate in matrice pezzo. È compito dell'ottimizzatore custom modificare l'ordine di esecuzione delle lavorazioni e quindi anche del cambio utensile.

Pagina 2: Settaggi di ottimizzatore

Le assegnazioni corrispondono alle condizioni di ottimizzazioni assunte per default per un programma, in assenza di assegnazioni dirette, localmente al programma stesso.

Non ottimizza	<input checked="" type="checkbox"/> le lavorazioni vengono riportate nella matrice pezzo senza eseguire alcuna elaborazione o ottimizzazione. Sono applicati: la sequenza imposta per le facce, le conversioni di unità.
Ottimizzazione minima	<input checked="" type="checkbox"/> esegue in sequenza: <ul style="list-style-type: none"> ◇ la verifica delle lavorazioni programmate per utensile (la verifica si limita alle lavorazioni puntuali se sono assegnati: <input checked="" type="checkbox"/> <i>Ottimizzatore custom</i> + <input checked="" type="checkbox"/> <i>Ordina custom</i>); ◇ la selezione degli utensili per le lavorazioni programmate per diametro, senza attivare procedure di match battute (la selezione degli utensili si limita alle lavorazioni puntuali se sono assegnati: <input checked="" type="checkbox"/> <i>Ottimizzatore custom</i> + <input checked="" type="checkbox"/> <i>Ordina custom</i>); ◇ non esegue ordinamenti: né di codici né di quote.
Gestione codici speciali	<input checked="" type="checkbox"/> l'opzione è applicata <i>solo se</i> nel programma sono impostati dei codici definiti alla voce "Codici Speciali". Effettua: <ul style="list-style-type: none"> ◇ la verifica delle lavorazioni programmate per utensile; ◇ la selezione degli utensili per le lavorazioni programmate per diametro con attivazione delle procedure di match battute solo localmente a gruppi di lavorazioni puntuali non separate da codici speciali; ◇ esegue ordinamenti (di codici e di percorso) solo tra lavorazioni non separate da codici speciali.
Ottimizza multigruppo	<input checked="" type="checkbox"/> esegue un ordinamento in X crescente per tutte le lavorazioni del pannello, anziché sulle singole facce. La selezione è: <ul style="list-style-type: none"> ◇ obbligatoria in caso di macchina con più gruppi; ◇ può essere attivata anche per macchina ad un solo gruppo. <input type="checkbox"/> applica ottimizzazioni con criteri di "monogruppo". Il settaggio condiziona anche il formato della matrice pezzo: <input checked="" type="checkbox"/> genera la matrice in formato di "multigruppo"; <input type="checkbox"/> genera la matrice in formato di "monogruppo".
Ordinamento codici	<input checked="" type="checkbox"/> abilita l'ordinamento delle lavorazioni per codice. L'ordine dei codici è assegnato in pagina 5. <u>Ordinamento Codici</u> . Il settaggio è utilizzato solo se non è assegnato: <input checked="" type="checkbox"/> <i>Ottimizzatore multigruppo</i> .
Ottimizza il tempo di esecuzione	<input checked="" type="checkbox"/> ottimizza il tempo di esecuzione del programma (può comportare un allungamento dei tempi di match delle battute di foratura, per la ricerca della condizione di match migliore); <input type="checkbox"/> ottimizza il tempo di ottimizzazione (in fase di match delle battute di foratura può operare delle scelte abbreviate, a scapito di una scelta di match non completamente ottimizzata).
Test lunghezza utensili per profili	<input checked="" type="checkbox"/> controlla che la profondità programmata per una lavorazione di setup sia inferiore o uguale alla lunghezza dell'utensile; <input type="checkbox"/> non esegue il controllo.
Test lunghezza utensili per puntuali	<input checked="" type="checkbox"/> controlla che la profondità programmata per una lavorazione puntuale sia inferiore o uguale alla lunghezza dell'utensile; <input type="checkbox"/> non esegue il controllo.
Utensili Tn in Matrice pezzo	<input checked="" type="checkbox"/> abilita la compilazione in matrice pezzo delle sequenze di esecuzione battute per gli utensili in lavorazioni di setup indicati oltre il primo (per un massimo di 4).
Ottimizza per utensile	<input checked="" type="checkbox"/> abilita il raggruppamento dei setup programmati con lo stesso utensile. Il settaggio rispetta i criteri assegnati per : <ul style="list-style-type: none"> • l'ordinamento per codici; • la gestione di codici speciali (il raggruppamento per utensile è fatto solo tra gruppi di setup non interrotti da codici speciali). Questo criterio è utilizzato: <ul style="list-style-type: none"> ◇ se non sono assegnati: <input checked="" type="checkbox"/> <i>Ottimizzatore custom</i> + <input checked="" type="checkbox"/> <i>Ordina custom</i>;
Quota q1	Il significato può cambiare: <ul style="list-style-type: none"> ➤ con attivo Ottimizzazione "monogruppo", definisce un'altezza di suddivisione del campo Y: <ul style="list-style-type: none"> ◇ nella prima zona le lavorazioni vengono ordinate in X crescente; ◇ nella seconda zona vengono ordinate in X decrescente. ➤ se Ottimizzazione "multigruppo", definisce il valore limite di Y al di sotto del quale le

	lavorazioni vengono eseguite solo con i gruppi bassi.
Ordinamento x	<input checked="" type="checkbox"/> ordina le lavorazioni in X crescente o decrescente se è selezionata la voce "Ordinamento decrescente". Il criterio è dominante sul criterio "Ordinamento y". Se è abilitato l'ordinamento in X: 1) con ordinamento in Y attivo: è gestito a parità di X (in Y crescente); 2) con ordinamento in Y non attivo: la posizione in Y non influisce sull'ordinamento; In caso di ordinamento in X può essere attivato un funzionamento con suddivisione del campo y in due zone (vedi: <i>Quota q1</i>): ◇ X crescente (o decrescente) per la prima zona ◇ X decrescente (o crescente) per la seconda zona. Per ognuna delle due zone è applicato anche un eventuale ordinamento in Y. Gli ordinamenti di percorso sono applicati alle sole lavorazioni puntuali (forature, inserimenti).
Ordinamento y	<input checked="" type="checkbox"/> ordina le lavorazioni in Y crescente o decrescente se è selezionata la voce "Ordinamento decrescente". Questo criterio è utilizzato: ◇ solo con Ottimizzazione "monogruppo"; ◇ solo se non richiesto anche "Ordinamento x". L'ordinamento in Y non viene influenzato dalle posizioni in X.
Ordinamento decrescente	<input checked="" type="checkbox"/> ordina le lavorazioni in Y o X decrescente. Per macchine che richiedono Ottimizzazione multigruppo l'ordinamento è sempre in X crescente.
Ordinamento a greca	<input checked="" type="checkbox"/> ordina le lavorazioni puntuali secondo il seguente criterio: ◇ ordinamento per X crescente; ◇ tra lavorazioni con uguale X si sceglie quella ad Y più prossima alla Y del punto precedente; ◇ se in sequenza ad una lavorazione puntuale è programmata una serie di lavorazioni puntuali ad X costante ed Y differente: si considerano solo le due lavorazioni estreme, a confronto con la Y del punto precedente. Tutte le altre lavorazioni vengono poi ordinate per Y crescente o decrescente, a seconda della lavorazione scelta. Non è applicato per macchine che richiedono Ottimizzazione multigruppo.
Ordinamento battute	<input checked="" type="checkbox"/> ordina le lavorazioni in X crescente prima di effettuare qualsiasi operazione di ottimizzazione (influenza il match delle battute).

Pagina 3: Codici speciali

Le assegnazioni corrispondono alle condizioni di ottimizzazioni assunte per default per un programma, in assenza di assegnazioni dirette, localmente al programma stesso.

La pagina consente di assegnare le lavorazioni custom di tipologia logica (codice operativo in range: 1 –1000) che devono essere riconosciute come "codici speciali".

La assegnazione dei Codici speciali non è valutata per macchine che richiedono Ottimizzazione multigruppo.

È possibile assegnare un massimo di 10 codici.

Pagina 4: Settaggi speciali

Le assegnazioni corrispondono alle condizioni di ottimizzazioni assunte per default per un programma, in assenza di assegnazioni dirette, localmente al programma stesso.

Assegna sequenze	<input checked="" type="checkbox"/> abilita la gestione delle sequenze in creazione di programma
(....)	<p>Nella pagina possono essere proposti dei campi di assegnazione personalizzati, sulla base di ogni applicazione. Si tratta comunque di assegnazioni che possono modificare solo il comportamento di un ottimizzatore custom e devono essere concordate con il responsabile della applicazione.</p> <p>La configurazione avviene tramite il file CSTBREAK.LNG (in cartella "cadcfg"): sono riservati i messaggi dal 101 al 116, per un totale di 16 settaggi disponibili:</p> <ul style="list-style-type: none"> • utilizzare i primi 8 messaggi (dal 101 al 108) per selezioni a due valori (0/1; On/Off); • utilizzare i restanti 8 messaggi (dal 109 al 116) per campi con valori assegnati in intervallo più generico, anche con decimali (intervallo massimo di valori è comunque assegnato: da -

- 32768 a 32767);
- i messaggi dal 117 al 200 sono disponibile per i messaggi in caso di selezione in lista.

Esempio di messaggio "Ordina;0;3;117" dove:

- ';' è carattere separatore tra i campi;
- 1° campo: descrizione ("Ordina");
- 2° campo: valore minimo (nell'esempio: 0) {maggiore o uguale di: -32768};
- 3° campo: valore massimo (nell'esempio: 3) {minore o uguale di: 32767};
- 4° campo: messaggio cui fare riferimento per una selezione in lista (nell'esempio: 117). È possibile assegnare:
 - ◊ un unico messaggio, con le voci separate da carattere '|',
 - ◊ oppure singoli messaggi in sequenza di numerazione.

Per mantenere disabilitato un settaggio: non assegnare i campi successivi al primo (Esempio: "Ordina").

Per gestire una impostazione da tastiera non assegnare il 4° campo (Esempio: "Ordina;-12;120").

Per gestire un settaggio di *selezione* (On/Off) indicare: valore minimo=0, valore massimo=1 ed opzionalmente il quarto campo (Esempio: "Ordina;0;1" → propone scelta tra OFF ed ON; "Ordina;0;1;117" → propone scelta tra 2 voci lette da messaggio 117).

Per gestire un settaggio di tipo lista indicare: valore minimo=0, valore massimo <10 ed obbligatoriamente il quarto campo (Esempio: "Ordina;0;4;117" → propone scelta tra 5 voci lette da messaggio 117).

La figura riporta un esempio di assegnazione dei settaggi speciali di tipo custom:

Fase A	Attiva	▼
Fase B	Off	
Scarico	Off	
Q. scarico avanti	1000	
Q. scarico indietro	800	

il file CSTBREAK.LNG può essere assegnato come di seguito:

```

101: "Fase A;0;1;118"           // 101-108: Flags custom
102: "Fase B;0;1;"
103: "Flag custom 3"
104: "Flag custom 4"
105: "Flag custom 5"
106: "Flag custom 6"
107: "Flag custom 7"
108: "Flag custom 8"
109: "Scarico;0;2;117"         // 109-116: Settaggi custom
110: "Q. scarico avanti;0;1000"
111: "Q. scarico indietro;0;2000"
112: "Settaggio custom 4"
113: "Settaggio custom 5"
114: "Settaggio custom 6"
115: "Settaggio custom 7"
116: "Settaggio custom 8"
117: "Off|Avanti|Indietro"    // Messaggi ausiliari
118: "Attiva|Non Attiva"

```

Pagina 5: Ordinamento codici

La pagina consente di assegnare una lista di priorità di codici: questa lista di priorità definisce i criteri dell'ordinamento per codici applicato dal programma OPTIMIZE.

Sono gestite 10 livelli di ordinamento e per ognuno di essi si possono inserire 10 codici di lavorazione.

Ad ogni livello di priorità assegnato per i codici sono applicati gli ordinamenti di percorso assegnati (ordinamento in x, in y,...), con alcune limitazioni: gli ordinamenti di percorso sono applicate solo ai codici di tipo puntuale.

Nel caso di livello assegnato con tipologie di lavorazioni differente:

- 1) i codici custom di tipo logico sono estratti per primi, nella sequenza programmata;
- 2) seguono i codici di setup, nella sequenza programmata;
- 3) seguono i codici puntuali, nella sequenza imposta dai settaggi di ordinamento di percorso.

È possibile assegnare come codice di ordinamento anche un codice generico di lavorazione:

COPGPNT : tutte le lavorazioni puntuali;

COPGPNT0: forature normali (lavorazioni puntuali di sotto-tipologia =0 o non assegnata)

COPGPNT1: forature con utensile orientato (lavorazioni puntuali di sotto-tipologia =1)

COPGPNT2: forature speciali (lavorazioni puntuali di sotto-tipologia =2)

COPGPNT3: lavorazioni puntuali di inserimento (lavorazioni puntuali di sotto-tipologia =3)

COPGSET : tutte le lavorazioni di setup;

COPGSET0: setup fresa normali (lavorazioni di setup di sotto-tipologia =0 o non assegnata)

COPGSET1: setup fresa con utensile orientato (lavorazioni di setup di sotto-tipologia =1)

COPGSET2: setup fresa speciali (lavorazioni di setup di sotto-tipologia =2)

COPGSET3: setup lama (lavorazioni di setup di sotto-tipologia =3)

COPGGEN1: tutte per lavorazioni logiche (con codice operativo in intervallo: 1 –1000).

È possibile attivare le impostazioni di default, per le quali vengano ordinate prima tutte le lavorazioni logiche, poi tutte le lavorazioni setup ed infine tutte le lavorazioni puntuali.

2.6. Impostazioni dirette

Alcune impostazioni utilizzate da EDICAD e/o OPTIMIZE devono essere assegnate direttamente nei files di configurazione "cadcfg\pezzo.ini, oppure "tpa.ini ": si tratta di impostazioni generalmente gestite dal tecnico TPA responsabile dell'applicazione.

Sezione in file.INI	Voce	Versione
[EDICAD] in PEZZO.INI	NameOptimize	2.4.007
<p>Con la versione [2.4.007] di EDICAD ed OPTIMIZE è possibile installare due ambienti distinti di programmazione: due installazioni di Albatros, ma duplicati solo per la programmazione.</p> <p>Il caso trova applicazione con la gestione di una linea: in questo caso un personal gestisce sia una macchina (tipico: la prima) sia la programmazione della linea. In questi casi è usuale gestire una programmazione centralizzata: un unico programma (scritto in ambiente di linea) viene smistato in esecuzione sulle singole macchine.</p> <p><u>L'installazione di gestione della linea</u> predispone un ambiente Albatros di gestione della sola programmazione. Per l'ottimizzazione non è possibile installare il programma OPTIMIZE (che è già utilizzato nell'installazione di macchina), ma deve funzionare l'eseguibile alternativo OPTIALBY: il settaggio "NameOptimize=OPTIALBY" indirizza EDICAD all'utilizzo di questo eseguibile.</p> <p>In gestione della linea è inoltre tipico istanziare l'oggetto che il programma ottimizzatore (OPTIALBY o OPTIMIZE) espone: vedi paragrafo successivo.</p> <p><u>L'installazione di gestione della macchina</u> (sullo stesso computer) predispone un ambiente Albatros completo. In particolare: la voce "NameOptimize" non deve comparire nel file PEZZO.INI di questa installazione.</p>		
[EDICAD] in PEZZO.INI	ModeOptiN ModeOptiX ModeOptiY ModeOptiXY	2.4.010
<p>L'impostazione riguarda l'eseguibile EDICAD.</p> <p>Le voci permettono di personalizzare l'ottimizzazione di programma in esecuzione di "File, Salva" (se è abilitata nei settaggi di Ottimizzatore).</p> <p>Fino alla versione [2.4.009]: in esecuzione di "File, Salva" viene richiesta una ottimizzazione minima (l'ottimizzatore valuta solo se il programma è eseguibile, ma non crea il file di matrice pezzo) in esecuzione normale, con offsets delle battute tutti nulli (0.0), senza : il funzionamento rimane valido nel caso in cui nessuna delle voci indicate è impostata, o nessuna è significativa.</p> <ul style="list-style-type: none"> • Significato di ogni settaggio: <ul style="list-style-type: none"> ◇ "ModeOptiN": personalizza l'ottimizzazione Normale; ◇ "ModeOptiX": personalizza l'ottimizzazione per Speculare X; ◇ "ModeOptiY": personalizza l'ottimizzazione per Speculare Y; ◇ "ModeOptiXY": personalizza l'ottimizzazione per Speculare XY; • Formato di ogni settaggio (Esempio: "1;2700.8;0;0"): <ul style="list-style-type: none"> ◇ 1° campo: <ul style="list-style-type: none"> ✓ minore o uguale a zero (<=0) disabilita la voce corrispondente; ✓ =1 abilita la voce per ottimizzazione minima. <p>Se EDICAD gestisce l'Ottimizzatore con invio di messaggio: l'Ottimizzatore non crea la matrice pezzo in memoria e non crea file di matrice in formato ASCII (TXN,...). Se EDICAD gestisce l'Ottimizzatore con metodo OLE : l'Ottimizzatore non crea file di matrice pezzo in formato binario e non crea file di matrice in formato ASCII</p> 		

(TXN,..);

- ✓ =2 abilita la voce per ottimizzazione completa.

Se EDICAD gestisce l'Ottimizzatore con invio di messaggio: l'Ottimizzatore non crea la matrice pezzo in memoria e crea file di matrice in formato ASCII (TXN,..). Se EDICAD gestisce l'Ottimizzatore con metodo OLE : l'Ottimizzatore non crea file di matrice pezzo in formato binario e crea file di matrice in formato ASCII (TXN,..);

- ◇ 2° campo: offsets x battuta (se non impostato interpreta: 0.0);
- ◇ 3° campo: offsets y battuta (se non impostato interpreta: 0.0);
- ◇ 4° campo: offsets z battuta (se non impostato interpreta: 0.0);

Se almeno una di queste impostazioni è assegnata con il 1° campo maggiore di zero (>0): dopo l'archiviazione del programma sono eseguite richieste di ottimizzazione in numero pari alle impostazioni dichiarate attive, ognuna per tipo di ottimizzazione corrispondente (Normale, Speculare X,...) e con gli offsets di battuta dichiarati.

In caso di errore in ottimizzazione, nella finestra che segnala il fallimento viene indicato il tipo di ottimizzazione (Normale, Speculare X,...).

[EDICAD] in PEZZO.INI	CustomParams	2.6.10
------------------------------	---------------------	---------------

L'impostazione riguarda l'eseguibile EDICAD.

Esempio: *CustomParams=9700;9750;9754*

Elenca fino ad un massimo di 20 tipologie di parametro custom di utensile. I parametri in elenco sono letti dalla parametrica utensile e testabili in programmazione parametrica (esempio: funzione *prtool[...]*).

I parametri sono assegnati di tipologia float ed è possibile assegnare più colonne con la stessa tipologia (i valori sono sommati).

[LISTER] in TPA.INI	ObjectOptimize	2.5.002a
----------------------------	-----------------------	-----------------

L'impostazione riguarda l'eseguibile EDICAD ed il LISTER (da ALBATROS.EXE).

Con "*ObjectOptimize=1*", EDICAD e LISTER istanziano l'eseguibile ottimizzatore come oggetto (vedi paragrafo successivo).

Altrimenti: ottimizzatore è avviato come normale eseguibile (a 16 bit) e la comunicazione con esso avviene tramite "messaggi".

[DEBUG] in TPA.INI	Optimize	
---------------------------	-----------------	--

L'impostazione riguarda l'eseguibile OPTIMIZE.

Con "*Optimize=1*" OPTIMIZE registra dei file ausiliari di debug, tutti in formato testo e tutti archiviati in direttorio assegnato in file TPA.INI dalla variabile *dirreport*:

- ◇ PARAMIN.TXT: dati in ingresso all'ottimizzatore da SendMessage (caso di comunicazione tramite messaggio). Un esempio per il file PARAMIN.TXT:

```
"bCommOk=0
nError=0
szPathFileName=
Dimx=0.000000
Dimy=0.000000
Dimz=0.000000
SzVarList=
NTipoStore=2
NumColonneMat=0
NumRigheMat=0
HGlobal=0
N_exetype=0
pos battutaX=0.000000 Y=0.000000 Z=0.000000"
```

- ◇ PARAMOUT.TXT: dati in uscita dall'ottimizzatore da SendMessage (caso di comunicazione tramite messaggio). Un esempio per il file PARAMOUT.TXT:

```
"NumColonneMat=0
NumRigheMat=0
HGlobal=0
n_exetype=0
pos battutaX=0.000000 Y=0.000000 Z=0.000000
Errore: "
```

- ◇ TABMAC(numero macchina).TXT: matrice MxCnfMac letta da parametrica Albatros. Un esempio per il file:

```
"Ut1: t6v2.0c1 t0v1c6 t0v1.0c7 t108v-200.0c12 t107v80.0c13
Ut2: t6v0.0c1 t108v0.0c12 t107v0.0c13
Ut3: t6v0.0c1 t108v0.0c12 t107v0.0c13
Ut4: t6v0.0c1 t108v0.0c12 t107v0.0c13
....."
```

<ul style="list-style-type: none"> ◇ TABUTE(n.o macchina).TXT: matrici MxCnfGru e MxUtensili lette da parametrica Albatros; ◇ CUSTMAC(n.o macchina).TXT: corrisponde al file binario MxMac.bin; ◇ CUSTTUT(n.o macchina).TXT: corrisponde al file binario MxGru.bin. 		
[DEBUG] in TPA.INI	OptiAlby	
L'impostazione riguarda l'eseguibile OPTIALBY: il significato è uguale a quello del settaggio <i>Optimize</i> per l'eseguibile OPTIMIZE.		
[OPTIMIZE] in PEZZO.INI	DebugCust	
<p>L'impostazione riguarda l'eseguibile OPTIMIZE (OPTIALBY).</p> <p>Con "<i>DebugCust=1</i>" e solo se è abilitato un ottimizzatore custom, OPTIMIZE registra dei file ausiliari in formato testo:</p> <ul style="list-style-type: none"> ◇ MXPEZZO.DBG (in direttorio assegnato in file TPA.INI dalla variabile <i>dirreport</i>): è la traduzione in formato testo della matrice pezzo che viene letta dall'ottimizzatore custom in formato binario in file MXPZIN.BIN; ◇ PARCST.DBG (in direttorio assegnato in file TPA.INI dalla variabile <i>dirreport</i>): è la traduzione in formato testo delle strutture di intestazione del file MXPZIN.BIN. <p>Funzionamento di default: 0.</p> <p>Per il formato del file MXPZIN.BIN si rimanda alla documentazione specifica.</p> <p>L'utilizzo del settaggio è tipico per una fase di test del funzionamento dell'ottimizzatore custom.</p>		
[OPTIMIZE] in PEZZO.INI	CfrPerUt	
<p>L'impostazione riguarda l'eseguibile OPTIMIZE (OPTIALBY).</p> <p>Il valore del settaggio condiziona il controllo relativo ai parametri di corrispondenza tra utensile e lavorazione per verifica di uguaglianza (tipologia parametro: da 1001 a 2000) nel caso di <u>programmazione per utensile</u> (forature o fresate). Solo nel caso in cui il parametro risulta gestito sia per la lavorazione che per l'utensile, ma è impostato significativo (leggi: valore non nullo) o per la lavorazione o per l'utensile (e non per entrambi):</p> <ul style="list-style-type: none"> ◇ "<i>CfrPerUt=0</i>": il controllo non è eseguito; ◇ "<i>CfrPerUt=1</i>": il controllo è comunque eseguito. <p>Funzionamento di default: 0.</p> <p>Se uno dei parametri in questione risulta gestito ed impostato significativo (leggi: valore non nullo) sia per la lavorazione che per l'utensile: il controllo è comunque eseguito.</p> <p>Se uno dei parametri in questione risulta gestito o per la lavorazione o per l'utensile (e non per entrambi): il controllo non è comunque eseguito.</p> <p>Un esempio può chiarire l'utilità del settaggio. Si considera il parametro di assegnazione del tipologia utensile (tipologia parametro: 1002):</p> <ul style="list-style-type: none"> ✓ il parametro è assegnato in lavorazione "FORO PER UTENSILE": si chiama la lavorazione con utensile 65 e tipologia utensile non assegnata (campo vuoto); ✓ il parametro è assegnato anche per l'utensile 65, con valore 1. <p>In questo caso:</p> <ul style="list-style-type: none"> ◇ "<i>CfrPerUt=0</i>": il controllo non è eseguito e la programmazione è accettata; ◇ "<i>CfrPerUt=1</i>": il controllo è eseguito e la programmazione non è accettata (l'Ottimizzatore segnala un errore). 		
[OPTIMIZE] in PEZZO.INI	CfrNotPerUt	
<p>L'impostazione riguarda l'eseguibile OPTIMIZE (OPTIALBY).</p> <p>Il valore del settaggio è analogo al precedente: riguarda ora il caso di <u>programmazione per diametro</u> (forature o fresate):</p> <ul style="list-style-type: none"> ◇ "<i>CfrNotPerUt=0</i>": il controllo non è eseguito; ◇ "<i>CfrNotPerUt=1</i>": il controllo è comunque eseguito. <p>Funzionamento di default: 1.</p>		
[OPTIMIZE] in PEZZO.INI	CustomObject	2.5.002a
<p>L'impostazione riguarda l'eseguibile OPTIMIZE (OPTIALBY).</p> <p>"<i>CustomObject=nomeOggetto</i>" indica che la chiamata dell'Ottimizzatore custom deve avvenire tramite OLE: "<i>nomeOggetto</i>" è il nome dell'oggetto che OPTIMIZE deve instanziare; "<i>Elabora</i>" è il nome (FISSO) del metodo che "<i>nomeOggetto</i>" deve esporre.</p> <p>Tutti i dati vengono passati come in funzionamento precedente, attraverso file (MXPZIN.BIN).</p> <p>Se non è instanziato come oggetto, l'Ottimizzatore custom viene avviato tramite l'istruzione <i>WinExec</i>.</p> <p>ATTENZIONE: L'ottimizzatore custom deve essere un ESEGUIBILE (no DLL) perché non è possibile alcuna comunicazione tra un eseguibile a 16 bit (OPTIMIZE) ed una DLL a 32bit.</p> <p>ATTENZIONE: L'oggetto "<i>nomeOggetto</i>" è allocato e de-allocato ad ogni ottimizzazione, per permettere una veloce trasformazione degli ottimizzatori custom già funzionanti in oggetti (senza problemi di inizializzazioni).</p> <p>ATTENZIONE:</p>		

Nel caso di chiamata via OLE dell'Ottimizzatore custom, anche l'ottimizzatore standard (OPTIMIZE.EXE, OPTIALBY.EXE) devono essere chiamati via OLE (vedi paragrafo successivo).

[OPTIMIZE] in PEZZO.INI**CustomModeExec****2.6.14**

L'impostazione riguarda l'eseguibile OPTIMIZE (OPTIALBY), in modalità di chiamata dell'Ottimizzatore custom NON tramite OLE:

◇ “*CustomModeExec=0*”: l'Ottimizzatore custom è avviato tramite istruzione *WinExec* (come versioni precedenti; è la situazione di default);

“*CustomModeExec=1*”: l'Ottimizzatore custom non è avviato dall'eseguibile OPTIMIZE (OPTIALBY), ma il lancio deve essere fatto da un eseguibile (a 32bit) avviato a parte. OPTIMIZE (OPTIALBY) si limita a creare un file (dirreport\CUSTVIA.TXT): l'eseguibile a 32bit deve intercettare la presenza di questo file, lanciare l'ottimizzatore custom e quindi rimuovere il file ausiliario.

3. DATABASE DELLE LAVORAZIONI (EDICADWR.DEF)

Condizione necessaria affinché EDICAD possa gestire dei programmi è la disponibilità di un file di assegnazione delle lavorazioni.

L'applicativo EDICAD legge i prototipi delle lavorazioni in file EDICADWR.DEF, che viene gestito in creazione e modifica dall'applicativo PASTUDIO.EXE.

Il file EDICADWR.DEF assegna ogni lavorazione in tutti i suoi aspetti:

- ◇ caratterizzazioni generali (codice operativo, tipologia, nome,...)
- ◇ lista parametri (tipologie parametri,...)
- ◇ finestre di assegnazione (disposizione dei parametri su una o più pagine, disposizione dei controlli,...).

Il file EDICADWR.DEF è cercato in cartella *cadcfg* e deve esistere definito correttamente (EDICAD deve trovarvi assegnato un gruppo non vuoto di lavorazioni).

Nella medesima cartella devono essere presenti i files ausiliari:

- EDICADWR.LNG messaggi utilizzati in configurazione delle lavorazioni (nomi della lavorazioni, messaggi associati ai parametri,...);
- HELPEDI.GRF immagini di help per le lavorazioni;
- EDICADWR.GRF immagini ausiliarie per uso locale in PASTUDIO.EXE.

Dal file EDICADWR.DEF viene letto il gruppo di lavorazioni indirizzato con il primo disponibile (solitamente: è anche l'unico): denominiamo questo come gruppo lavorazioni di sistema (o **gruppo base**).

In aggiunta al gruppo base è possibile utilizzare anche un secondo gruppo di lavorazioni (**gruppo custom**) per lavorazioni personalizzate ed in generale differenti per ogni macchina.

Il gruppo custom è letto da un file separato (CUSTOM.DEF. Anche il file CUSTOM.DEF è cercato in cartella *cadcfg*).

Per creare il file di lavorazioni custom:

- copiare EDICADWR.DEF in CUSTOM.DEF;
- copiare EDICADWR.LNG in CUSTOM.LNG;
- avviare PASTUDIO.EXE ed aprire CUSTOM.DEF:
 - ◇ lasciare le sole lavorazioni che devono essere personalizzate;
 - ◇ eliminare i restanti codici;
 - ◇ aggiungere le lavorazioni necessarie.

Il **gruppo custom** è previsto per assegnare i codici specifici di una macchina:

- ◇ lavorazioni custom di tipologia puntuale, setup o logiche (codici operativi in intervallo: 201 - 1000);
- ◇ lavorazioni di profilo personalizzate (codici operativi in intervallo: 2300 - 2400);
- ◇ codici di chiamata macro-programmi (codici operativi in intervallo: 1501 - 1899);
- ◇ codici di chiamata diretta di sottoprogramma (codici operativi in intervallo: 4501- 5000).

Con caricamento di un gruppo custom, le lavorazioni in esso impostate sono inserite (in aggiunta o in sostituzione) a quelle lette dal gruppo base. In particolare:

- sono comunque eliminate tutte le lavorazioni: con codice operativo non valido (non compreso nell'intervallo: 1-5000), disabilitate o con nessuna pagina assegnata;
- in caso di codice operativo duplicato: viene tenuta la lavorazione del gruppo custom;
- questa fase di eliminazione, inserimento o sostituzione di lavorazioni avviene solo in memoria: nessuna modifica riguarda i files EDICADWR.DEF e/o CUSTOM.DEF, che rimangono gestiti solo dal programma PASTUDIO.

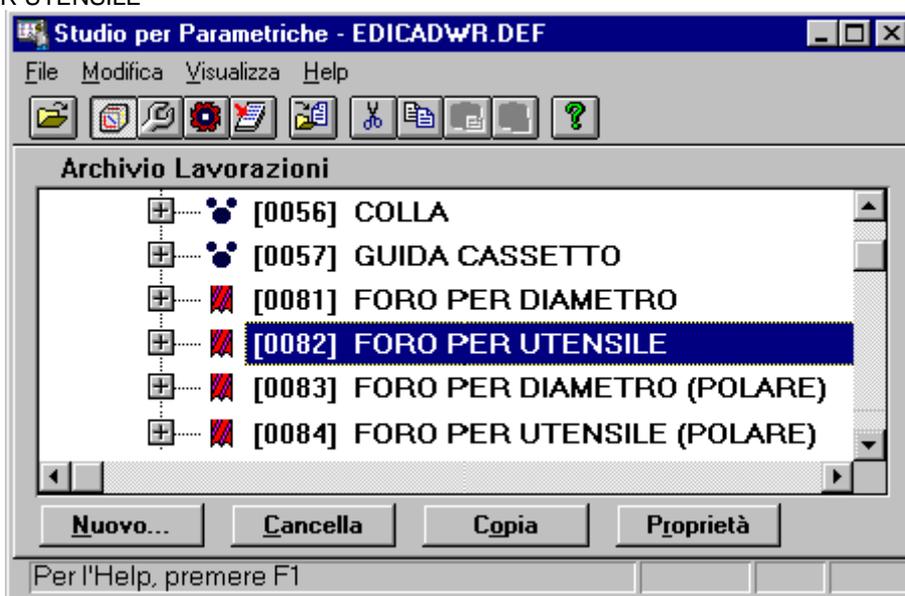
3.1. Gestione dei files database delle lavorazioni

La figura indica come si presenta all'avvio l'applicativo PASTUDIO: in modo automatico si predispone in gestione delle lavorazioni, con apertura del file EDICADWR.DEF.



ATTENZIONE: avviare PASTUDIO solo dopo avere chiuso gli applicativi EDICAD ed OPTIMIZE.

Il nodo *Lavorazioni di macchina* permette di espandere su tutte le lavorazioni configurate nel file. La figura riporta il nodo espanso, con banda di scorrimento posizionata sulla lavorazione: [0082] FORO PER UTENSILE

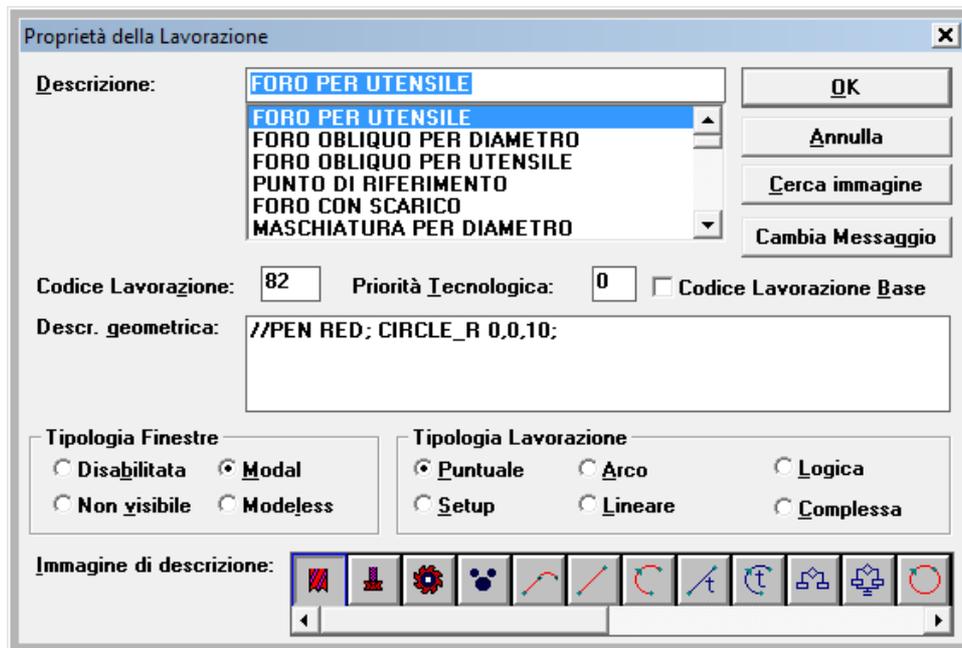


Si ipotizza qui di aprire il file di database EDICADWR.DEF, ma le considerazioni fatte rimangono valide aprendo un file di lavorazioni custom (CUSTOM.DEF). Per aprire il file CUSTOM.DEF selezione il comando "Apre file di definizione" in menu File.

ATTENZIONE: un file EDICADWR.DEF non può essere aperto con il comando "Apre file di definizione", anche se da cartella differente da *cadcfg*.

3.2. Assegnazione di lavorazione

Si rimanga posizionati sulla lavorazione "[0082] FORO PER UTENSILE" e si selezioni il pulsante **Proprietà** (in basso a destra). Si apre la finestra di assegnazione delle proprietà generali della lavorazione:



- **Descrizione:** campo descrittivo della lavorazione (Nell'esempio: "FORO PER UTENSILE")
Si tratta di una informazione univoca tra le lavorazioni: in PASTUDIO non è possibile assegnare lo stesso messaggio di descrizione per più di una lavorazione.
Per assegnare la descrizione:
 - ◇ scegliere tra uno dei messaggi disponibili a lista;
 - ◇ oppure inserire un nuovo messaggio.
- **Codice lavorazione (codice operativo):** campo numerico (Nell'esempio: 82)
Si tratta di una assegnazione univoca: in PASTUDIO non è possibile assegnare lo stesso codice per più di una lavorazione.

Per EDICAD: il codice operativo costituisce l'elemento di assegnazione univoca di una lavorazione.

Vengono date delle regole per la assegnazione dei codici di lavorazioni. Sono di seguito riportati gli intervalli di valori che interessa conoscere per una personalizzazione delle lavorazioni di macchina:

[1; 1000] codici di lavorazioni elementari (custom)

In questo intervallo sono assegnate le lavorazioni personalizzate di tipologia:

- ◇ puntuale (forature, inserimenti);
- ◇ setup (setup fresa, setup lama);
- ◇ logiche custom (rapido, stop programmato,...).

In questo intervallo:

- i codici [1-200] sono riservati per codici di interesse generale e sono presenti in file EDICADWR.DEF;
- i codici [201-1000] sono disponibili per lavorazioni custom e devono essere aggiunte in file CUSTOM.DEF.

[1001; 2000] codici di macro

In questo intervallo sono assegnate le lavorazioni personalizzate di tipologia complessa che comportano l'utilizzo di un macro-programma.

Assegnare tipologia: complessa. In questo intervallo:

- i codici [1001-1500] sono riservati per codici di interesse generale e sono presenti in file EDICADWR.DEF;
- i codici [1501-1899] sono disponibili per lavorazioni custom e devono essere aggiunte in file CUSTOM.DEF;
- i codici [1900-2000] sono riservati per i codici di conversione e sono presenti in file EDICADWR.DEF (esempio: i codici per conversioni da CNC90).

[2300; 2400] codici di lavorazioni custom di profilo

In questo intervallo devono essere assegnate le lavorazioni di tipologia arco o linea che comportano personalizzazioni:

- ◇ la tipologia deve essere: arco o linea;

- ◇ l'utilizzo del parametro [8110] individua il codice di riferimento per la compilazione geometrica del codice.

Un codice così definito permette di specializzare tratti di profilo: possono essere aggiunti parametri d'uso custom ma devono essere mantenuti i parametri necessari alla corretta interpretazione geometrica del tratto.

[4001; 5000] codici di sottoprogramma con chiamata diretta

In questo intervallo sono assegnate le lavorazioni personalizzate di tipologia complessa che comportano l'utilizzo di sottoprogrammi o sottocicli.

Assegnare tipologie: complessa. In questo intervallo:

- i codici [4001 -4500] sono riservati per codici di interesse generale e sono presenti in file EDICADWR.DEF;
- i codici [4501-5000] sono disponibili per lavorazioni custom e devono essere aggiunte in file CUSTOM.DEF.

- **Priorità tecnologica:** lasciare valore 0 (l'informazione viene ignorata da EDICAD a partire dalla versione EDICAD-2.5.001)

- **Codice lavorazione di base:** si tratta di una impostazione On/Off (default: Off=0) ad assegnazione di condizione di match battute per codici puntuali che gestiscono programmazione per diametro.

È un'informazione significativa per le lavorazioni di tipologia puntuale e con codice operativo in intervallo: [1 -1000]. Se la lavorazione è letta dal database di lavorazioni custom, in generale non si dovrebbe applicare match delle battute in fase di ottimizzazione (se infatti hanno assegnati parametri custom da passare in matrice pezzo, con il match sarebbero persi). L'informazione non è significativa in caso di funzionamento con *Ottimizzatore custom* e in modalità *Non ottimizza*.

- ◇ OFF abilita il match
- ◇ ON match non abilitato.

- **Descrizione geometrica:** campo di tipo stringa che assegna informazioni aggiunte (numero massimo di caratteri editabili: 1024).

Viene qui riportato un prospetto completo dell'utilizzo delle descrizione grafica di lavorazione. È possibile assegnare sotto-stringhe con utilizzo distinto ed ogni sotto-stringa è individuata da una intestazione:

- **"\..." assegna descrizione grafica valida (vedere documentazione dedicata: *Linguaggio MLC*):**
in assenza di una descrizione grafica personalizzata, EDICAD rappresenta ogni tipologia di lavorazione semplice con un colore ed una grafica di default (colori impostati in menu *Imposta*; cerchio di dimensione dell'utensile per puntuali e setup; archi e linee per tipologie di profilo).
In file EDICADWR.DEF: una descrizione grafica è ad esempio assegnata per le lavorazioni di inserimento.
- **"//..." rende la descrizione grafica commentata:**
- **"|bs|..." assegnazioni personalizzate in barra di stato di EDICAD (nella zona riservata alle quote):**
Esempio: "|bs|L=%1512 ... H=%1523 "
%1512 viene sostituito con il valore assegnato al parametro [1512]
%1523 viene sostituito con il valore assegnato al parametro [1523]
un risultato può essere: "L=50 ..H=123"
Può essere utilizzata in caso di codice di chiamata macro o sottoprogramma.
- **"|aux|..." assegna settaggi ausiliari:**
"\$op=..." nome breve di lavorazione. Esempio: "|aux|...\$op=foro"
(vedi: EDICAD, menu *Imposta*, voce *Editor*, pagina *Applica lavorazioni*).
"\$face=..." facce di applicazione. Esempio: "|aux|...\$face=135"
(omette la lavorazione in selezione di palette lavorazioni).
Esempi:
"\$face=1" la lavorazione è applicabile solo in faccia 1;
"\$face=135" la lavorazione è applicabile in facce 1,3,5;
"\$face=*" la lavorazione è applicabile in tutte le facce (default);
"\$face=17" la lavorazione è applicabile in faccia 1 ed in tutte le facce fittizie.
Se il campo non è assegnata: la lavorazione è applicabile in tutte le facce.
"\$mouse=..." assegna per "piazzamento diretto". Esempio: "|aux|...\$mouse=1c3"
con selezione di lavorazione in palette lavorazioni, il piazzamento ed inserimento avviene con il mouse (con i settaggi di snap e griglia come attivati).
Sintassi: sotto-stringa "\$mouse=1". È possibile selezionare anche un particolare cursore (disponibili 5, di numero da 1 a 5). L'impostazione è ignorata se lavorazione: logica, tratto di profilo (arco o linea).
Esempio: "\$mouse=1c2", seleziona il secondo cursore disponibile.
"\$snap=..." trascina il punto di applicazione di lavorazione complessa. Esempio: "|aux|...\$snap=1"
In caso di *Trascinamento* (menu: *Strumenti*) di lavorazione attuale (NB. non di lavorazione/i selezionate) e snap attivo: lo snap è applicato sul punto di applicazione e non sull'ingombro totale

(l'ingombro stesso è visualizzato sul punto di applicazione). Esempio: nel caso di macro di applicazione di elementi di bloccaggio e/o di ingombro: l'ingombro dell'elemento può essere visualizzato con profili di costruito, ma serve piazzare il punto di applicazione. Per lavorazioni con abilitato il piazzamento con il mouse è assunto di default anche il trascinamento del punto di applicazione.

“\$seq=...” per escludere la gestione della sequenza per la lavorazione. Esempio: “[aux]...\$seq=0” configura se la lavorazione viene esclusa dalla sequenzializzazione.

“\$seq=1” → sequenza abilitata (default);

“\$seq=0” → sequenza non abilitata.

“\$prop=...” assegna le proprietà di default in inserimento. Esempi: “\$prop=l1m12”, “\$prop=b1”.

Queste impostazioni vengono applicate in fase di inserimento di lavorazione.

L'impostazione viene ignorata in caso di lavorazione: logica non in intervallo custom, tratto di profilo (arco o linea). Non è necessario impostare tutte le proprietà.

Sintassi: sotto-stringa “\$prop=l..o..b..m..”:

‘l’ assegna il livello (l0 - l8)

‘o’ assegna il campo o (o0 - o3)

‘b’ assegna il costruito (b1= costruito on, b0=costruito off)

‘m’ assegna il campo m (m0 - m65000).

- (altrimenti) assegna descrizione grafica.

➤ tipologia finestra assegna lo stato di visibilità della lavorazione.

È possibile selezionare tra quattro modalità:

- **disabilitata:** la lavorazione non viene in alcun modo considerata dal programma EDICAD;
- **non visibile:** la lavorazione è protetta
 - ◊ in testo di programma: non è gestita per inserimento diretto, può risultare da sviluppo di macro o sotto-ciclo;
 - ◊ in testo di macro e sotto-ciclo: è gestita per inserimento diretto;
- **modale:** la lavorazione è sempre abilitata;
- **modeless** { da non usare}.

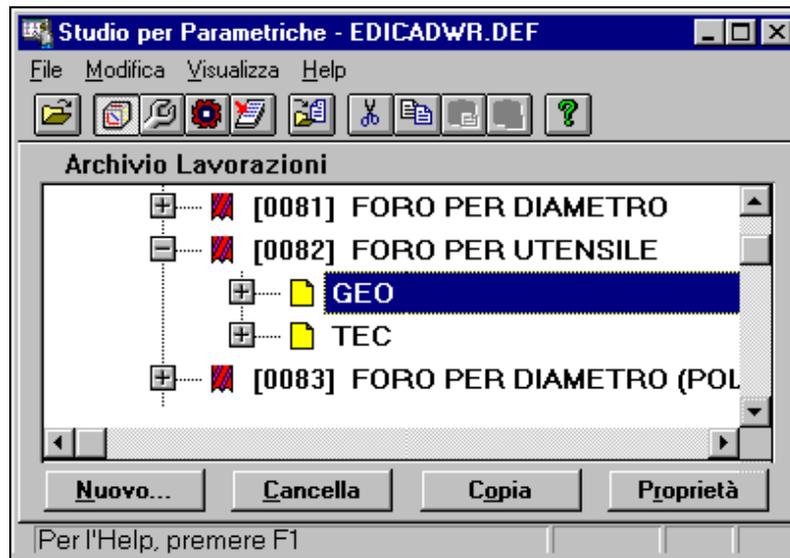
➤ tipologia di lavorazione

- puntuale (da usare per: forature, inserimenti)
- setup (lavorazione di inizio profilo o lamata)
- arco
- linea
- logica (non hanno assegnazioni geometriche)
- complessa (applicazione di sottoprogramma o macro-programma).

➤ Immagine di descrizione

Scegliere una tra le immagini in lista (l'uso è solo locale a PASTUDIO.EXE).

Uscendo dalla finestra di **Proprietà** della lavorazione si torna sulla lista delle lavorazioni, con banda di scorrimento posizionata sulla lavorazione "[0082] FORO PER UTENSILE". Apriamo il nodo della lavorazione (tasto ENTER):



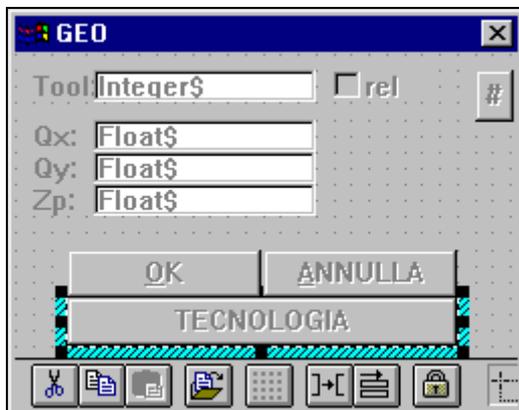
il nodo apre sulla lista delle pagine (finestre) che sono assegnate per l'impostazione interattiva della lavorazione (il numero massimo di pagine è 3): nel nostro caso abbiamo 2 pagine, con intestazione "GEO" e "TEC".

La lista delle pagine di una lavorazione può essere modificata con i pulsanti posti sulla parte inferiore della finestra:

- Nuovo** per inserire una nuova pagina
- Cancella** per eliminare una pagina
- Copia** per inserire una nuova pagina con copia da una già assegnata
- Proprietà** per cambiare l'intestazione della pagina.

L'ordine sulla lista rispetta l'ordine di presentazione su pagine successive.

Apriamo il nodo della pagina GEO della lavorazione "[0082] FORO PER UTENSILE" (tasto ENTER):



viene visualizzata la pagina di apertura per l'impostazione della lavorazione. Vi sono posizionati controlli di edit, di testo, di tipo check box e pulsanti.

Ogni elemento posizionato in una pagina di lavorazione corrisponde ad un parametro di lavorazione.

3.3. Assegnazione di parametro di lavorazione

Un parametro di lavorazione può avere significato di:

- ✓ parametro effettivo di assegnazione della lavorazione (dato geometrico, o tecnologico): è ad esempio il caso di una quota o di una velocità;
- ✓ parametro ausiliario, assegnato per la impostazione interattiva della lavorazione: è il caso dei pulsanti di conferma o di cambio pagina.

Per modificare un parametro di lavorazione già posizionato in finestra occorre selezionare con il mouse il controllo corrispondente e confermare con ENTER. Viene aperta una finestra di proprietà composta da due pagine, con riportate le caratteristiche del parametro:

➤ nome

Il nome del parametro è una stringa univoca nella lista dei parametri di una pagina di lavorazione: nessun controllo di univocità è eseguito in PASTUDIO.EXE su tutte le pagine della lavorazione ma deve essere assunto come regola necessaria. PASTUDIO richiede che il nome del parametro sia assegnato prima di passare alla seconda pagina della finestra: è quindi necessario assegnarlo subito, in inserimento di un parametro.

Relativamente al programma EDICAD il nome di un parametro assume significato solo nella definizione delle regole di rappresentazione grafica della lavorazione (vedi linguaggio: MLC).

➤ descrizione

Il messaggio viene utilizzato come testo per il controllo e può non essere assegnato.

➤ livelli di password

un livello per la Visualizzazione ed uno per la Modifica.

Si tratta di livelli d'accesso da confrontare con il livello attuale, secondo le modalità assegnate in ambiente Albatros. Lasciare normalmente impostati al valore minimo (Operatore).

➤ range

assegnato come valore minimo e massimo, ma con significato che può cambiare in base al tipo di controllo assegnato per il parametro.

➤ definizione in Matrice

Sono proposti un campo per il nome della matrice pezzo (dove necessario: impostare MXPEZZO) ed il numero della colonna per il deposito del parametro: impostare i campi solo per i parametri che devono essere scritti in matrice pezzo.

Il campo di Info aggiunte è utilizzato per assegnazioni aggiunte, specificate dove occorre.

➤ Modo di visualizzazione

È possibile selezionare tra tre modalità:

- ◇ View & Modify : il parametro è visualizzato e modificabile (se la tipologia del parametro lo prevede);
- ◇ Only View : il parametro è visualizzato ma non modificabile (in questo caso: assegnare un *Valore di default* significativo);
- ◇ Hide : il parametro non è visualizzato: si tratta di un parametro di uso riservato (in questo caso: assegnare un *Valore di default* significativo).

Si tratta di livelli d'accesso indipendenti dal livello d'accesso assegnato in ambiente Albatros: l'accesso reale ad ogni parametro viene comunque determinato in base alla risultanza logica di tutti i livelli di accesso assegnati (Livelli di password e Modo di visualizzazione).

➤ Tipologia parametro (tipologia Albatros)

Per EDICAD: la tipologia Albatros costituisce l'elemento di assegnazione univoca di un parametro di lavorazione.

Tutti i parametri utilizzati in progetto Albatros sono classificati in gruppi fondamentali, distinti in base ad una numerazione (tipologia) dei parametri stessi.

Per un certo numero di tipologie, EDICAD assume una interpretazione definita (esempio: quote di assegnazione del punto di applicazione di un foro, tipologie di assegnazione di utensile o diametro utensile,...).

Tra le restanti tipologie: è assegnato l'intervallo di valori 9501-9999 disponibili per la assegnazione di parametri custom di lavorazioni (parametri d'uso custom, che devono essere scritti in matrice pezzo).

Un aspetto particolare riguarda la assegnazione di parametri che riguardano la tecnologia di un utensile e che sono testati dal programma di ottimizzazione OPTIMIZE.EXE. Interessa qui esaminare due particolari gruppi di parametri, distinti in altrettanti intervalli:

- [1001-2000] parametri per i quali l'ottimizzatore verifica una condizione di uguaglianza tra il parametro assegnato in lavorazione e quello assegnato in parametrica utensile;

- [2001-3000] parametri per i quali l'ottimizzatore verifica l'esistenza nel range.

Parametri che appartengono al primo gruppo sono ad esempio: tipologia dell'utensile (foratore, inseritore, fresa, lama,...), diametro, tipologie accessorie.

Per selezionare un utensile di diametro 5mm: nella lavorazione si deve assegnare una tipologia di parametro nell'intervallo [1001-2000] ed un parametro con stessa tipologia deve comparire in parametrica degli utensili. Per ogni parametro assegnato in lavorazione con tipologia nell'intervallo [1001-2000] è ragionevole ipotizzare che sia assegnato un parametro di stessa tipologia in parametrica utensile.

Alcune tipologie notevoli sono definite a priori: 1001 per il tipo utensile, 1002 per il diametro.

Parametri che appartengono al secondo gruppo sono ad esempio le velocità di avanzamento e di rotazione utensile. Per selezionare un utensile con velocità di rotazione di 5000 giri/min: nella lavorazione si deve assegnare una tipologia di parametro nell'intervallo [2002-3000], su gruppi di 3 unità. Si assegni ad esempio un parametro di tipologia 2002: due parametri devono comparire in parametrica degli utensili, un primo che assegna il valore minimo ed ha la tipologia precedente (nell'esempio: 2001=2002-1) ed un secondo che assegna il valore massimo ed ha la tipologia successiva (nell'esempio: 2003=2002+1).

Per ogni parametro assegnato in lavorazione con tipologia nell'intervallo [2001-3000] è ragionevole ipotizzare che sia assegnato una coppia di parametri in parametrica utensile, con tipologia precedente e successiva.

➤ valore di default

Si tratta di un campo di impostazione di tipo stringa, che generalmente è lasciato vuoto.

In caso di campo assegnato, viene pre-impostato ad ogni richiesta di inserimento della lavorazione: si tratta di una pratica utile specialmente per i parametri di un codice di macro, per i quali possono essere individuate assegnazioni di uso tipico (si pensi al passo di foratura di una macro di fitting: un valore di default tipico può essere 32).

Il campo deve essere impostato se il parametro non è editabile (vedi: *Modo di visualizzazione*).

➤ tipologia di controllo

seleziona la tipologia di controllo che rappresenta il parametro.

Anche se qui è riportato come ultimo campo di assegnazione di parametro, in effetti è il primo: per inserire un parametro si sceglie una tipologia di controllo (tra quelle disponibili nella palette apposita) e si piazza il controllo corrispondente in finestra (il piazzamento del controllo comporta assegnare: la posizione e l'ingombro). Solo successivamente è possibile accedere alla finestra di proprietà e qui non è comunque più possibile cambiare la tipologia del controllo (se necessario: bisogna cancellare il parametro ed assegnarlo da capo).

Sono previsti i seguenti codici di formato, distinti in base ai tipi di controlli gestiti:

Controlli di tipo edit

Integer 	Impostazione di valore numerico intero. I valori estremi impostabili sono assunti in <u>range</u> e considerati sulla sola parte intera (impostare valore minimo = valore massimo =0 per disabilitare un controllo sul range). ATTENZIONE: non ammette programmazioni parametriche (solo un numero): si consiglia di evitarne l'uso (utilizzare il controllo (Integer)).
Float 	Impostazione di valore numerico con virgola. I valori estremi impostabili sono assunti in <u>range</u> (impostare valore minimo = valore massimo =0 per disabilitare un controllo sul range). ATTENZIONE: non ammette programmazioni parametriche (solo un numero): si consiglia di evitarne l'uso (utilizzare il controllo (Float)).
Edit 	Impostazione generica di stringa. Il valore massimo impostato in <u>range</u> indica il numero massimo di caratteri programmabili (impostare valore massimo =0 per abilitare il controllo di default sulla lunghezza di stringhe = 250 caratteri). Il controllo può assegnare il parametro corrispondente al nome di sottoprogramma.
(Integer) 	Impostazione parametrica e cast finale su valore intero. Il valore massimo impostato in <u>range</u> indica il numero massimo di caratteri programmabili (impostare valore massimo =0 per abilitare il controllo di default sulla lunghezza = 50 caratteri). Il controllo può assegnare parametri corrispondenti a: velocità di rotazione di un mandrino, numero di macchina o gruppo o utensile, contatore.
(Float) 	Impostazione parametrica e cast finale su numerico con virgola. Il valore massimo impostato in <u>range</u> indica il numero massimo di caratteri programmabili (impostare valore massimo =0 per abilitare il controllo di default sulla lunghezza = 50 caratteri). Il controllo può assegnare parametri corrispondenti a: quote, distanze, velocità di movimento.

Su tutti i controlli di tipo edit sono assegnate in automatico le proprietà:

- ◇ allineamento del testo a sinistra (se è assegnato il campo descrizione);
- ◇ immissione su singola linea;
- ◇ scorrimento orizzontale automatico;
- ◇ visualizzazione del bordo;
- ◇ immissione in formato minuscolo.

Per un controllo di tipo edit è generalmente assegnato il campo descrizione. In tal caso è effettuata la gestione combinata di due controlli: uno di tipo testo (TextBox) ed uno di tipo edit (del tipo assegnato tra quelli sopra indicati). In questo caso:

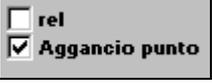
- ◇ il testo appare alla sinistra del controllo di edit;
- ◇ il controllo di edit è allineato alla destra dell'ingombro assegnato ed ha dimensioni fisse, in base alla tipologia del controllo.

La figura riporta un esempio di controllo di edit con assegnata una descrizione:



- ◇ la scritta **Float\$** riportata a riempimento della casella di edit indica che il controllo ha tipologia (Float);
- ◇ la descrizione è "X1".

Caselle combinate

<p>Check Box</p> 	<p>controllo di tipo casella di selezione singola. La programmazione comporta selezione tra stato on/off:</p> <ul style="list-style-type: none"> ◇ stato ON corrisponde a valore 1; ◇ stato OFF corrisponde a valore 0. <p>Il testo è allineato alla destra del controllo. Il controllo si presenta come da figura:</p>  <p>Il primo controllo in figura corrisponde a casella in stato OFF; il secondo controllo corrisponde a casella in stato ON.</p>
<p>Radio Button</p> 	<p>gruppo di controlli di tipo pulsante di opzione: la selezione è esclusiva su un gruppo di controlli simili. La programmazione comporta selezione attiva su uno solo dei pulsanti ed il valore attribuito al controllo corrisponde all'indice del pulsante attivo (significativo da 0).</p> <p>Si utilizzano i campi di valore minimo e massimo in <u>range</u> del parametro per assegnare:</p> <ul style="list-style-type: none"> ◇ il numero totale dei controlli del gruppo ◇ una visualizzazione dei controlli in riga o in colonna. <p>In particolare: il valore minimo indica il numero di righe, il valore massimo indica il numero di colonne. Il numero totale dei controlli risulta dalla <u>moltiplicazione delle righe per le colonne</u>. Di norma: la organizzazione dei controlli Radio Button avviene su una riga o su una colonna.</p> <p>I campi descrittivi dei singoli controlli possono essere:</p> <ul style="list-style-type: none"> ◇ gestiti distinti: devono essere consecutivi, sul file di messaggi; ◇ unico: il messaggio deve rispettare un assegnato formalismo: "radio1 radio2 radio3" il carattere ' ' è utilizzato come separatore tra i messaggi dei singoli controlli (nell'esempio: il messaggio gestisce tre controlli) ◇ essere assegnati in modo combinato sui due casi precedenti: "radio1 radio2 radio3" il messaggio gestisce tre controlli "radio4" messaggio successivo: quarto controllo "radio5" messaggio successivo: quinto controllo. <p>Proprietà gestite su ogni controllo:</p> <ul style="list-style-type: none"> ◇ testo alla destra del pulsante; ◇ tutti i controlli hanno proprietà di impostazione di default. <p>In figura viene riportato il caso di controllo assegnato con 3 possibili stati ed allineamento dei singoli pulsanti in riga:</p>  <p>la selezione attiva è indicata con un pallino nero a sfondo del cerchio del pulsante (in figura è selezionato il primo pulsante).</p>
<p>Combo Box, List Box</p> 	<p>controllo di tipo casella di riepilogo o lista.</p> <p>Si utilizza il campo di valore massimo per indicare il numero totale delle voci in lista: i campi descrittivi delle voci in lista devono essere consecutivi, su file dei messaggi, con formalismo assegnato come per la tipologia Radio Button.</p> <p>La programmazione:</p> <ul style="list-style-type: none"> ◇ corrisponde alla selezione tra una delle voci in lista; ◇ comporta selezione attiva su una delle voci in lista: il valore attribuito al controllo corrisponde all'indice della voce attiva (significativo da 0). <p>Proprietà gestite sul controllo:</p>

	<ul style="list-style-type: none"> ◇ opzione DropDownList per il tipo di lista; ◇ lista di stringhe; ◇ presentazione delle voci in lista nell'ordine come assegnato ◇ scorrimento verticale della lista. <p>In figura viene riportato il caso di controllo assegnato con 2 selezioni in lista:</p> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <p>Utensile 48 x 6</p> <p style="background-color: #e0e0e0;">48 x 6</p> <p>49 x 9.5</p> </div> <p>la lista viene visualizzata selezionando il pulsante riportato sul controllo (opzione DropDownList).</p>
--	--

Controlli di edit statico

 <p>TextBox</p>	Controllo di tipo testo (il testo è allineato a sinistra).
 <p>Group Box</p>	Controllo di tipo casella di gruppo: disegna un contorno (è possibile assegnare un testo sulla casella di gruppo) entro cui è possibile posizionare altri controlli.
 <p>Button</p>	<p>Controllo di tipo pulsante.</p> <p>La risposta alla selezione di un pulsante viene assegnata in base alla tipologia assegnata al parametro (con selezione su una lista di voci). In EDICAD la gestione dei pulsanti è personalizzata. In particolare:</p> <p>BUTTON_OK [tipologia Albatros: 8001]: pulsante di conferma. Conferma le impostazioni assegnate per i parametri ed esegue:</p> <ul style="list-style-type: none"> ◇ chiusura della finestra di assegnazione lavorazione (su tutte le finestre gestite), in caso di acquisizione dati corretta; ◇ gestione diagnostiche e rientro in fase di edit, in caso di acquisizione dati non corretta. <p>Il pulsante deve essere messo solo nella prima pagina di lavorazione: in caso contrario, viene interpretato come pulsante di "pagina precedente" (apertura e passaggio alla pagina precedente). Stessa procedura viene gestita in caso di uscita con tasto ENTER, se la pagina attuale ha il pulsante di BUTTON_OK come pulsante di default (primo pulsante in ordine di tabulazione).</p> <p>BUTTON_CANCEL [tipologia Albatros: 8002]: pulsante di annulla. Chiude la finestra di impostazione senza acquisizione dati:</p> <ul style="list-style-type: none"> ◇ annulla dato in prima pagina, chiude la fase di impostazione per la lavorazione; ◇ annulla dato in pagina successiva alla prima, chiude la sola pagina in questione. <p>Stessa procedura viene gestita in caso di uscita su tasto ESCAPE.</p> <p>BUTTON_PGDN [tipologia Albatros: 8004]: pulsante di passaggio alla pagina successiva BUTTON_PGUP [tipologia Albatros: 8003]: pulsante di passaggio alla pagina precedente La successione delle pagine è imposta dalla successione in lista di assegnazione delle finestre stesse. Il pulsante BUTTON_PGUP non deve essere messo in prima pagina. Il pulsante BUTTON_PGDN non deve essere messo in ultima pagina.</p> <p>BUTTON_HELP [tipologia Albatros: 8005]: pulsante di Help (richiama l'Help per la lavorazione). Il pulsante può essere messo in una qualunque delle pagine della lavorazione. Il pulsante non è mai utilizzato in EDICADWR.DEF: il richiamo dell'Help è infatti di norma gestito con pulsante F1.</p> <p>BUTTON_PAGE [tipologia Albatros: 8006] *** NON USARE</p> <p>BUTTON_FILE [tipologia Albatros: 8008] Pulsante da utilizzare in codici generici di sottoprogramma per aprire la finestra di <i>File>> Apre</i>, per la selezione del sottoprogramma direttamente da archivio. Il pulsante deve essere messo in prima pagina di lavorazioni di chiamata generica di sottoprogramma (codici: SUBn).</p> <p>BUTTON_VARS [tipologia Albatros: 8009] pulsante da utilizzare in codici di sottoprogramma per aprire la finestra delle variabili riassegnabili del sottoprogramma. Il pulsante deve essere messo in prima pagina di lavorazioni di chiamata generica di sottoprogramma (codici: SUBn).</p> <p>BUTTON_GEN1 [tipologia Albatros: 8010]: pulsante di visualizzazione del bitmap di help della lavorazione. Il pulsante può essere messo in una qualunque delle pagine della lavorazione. Deve essere prestata attenzione al posizionamento del pulsante in finestra: la finestra deve</p>

	potere essere espansa sui lati destro e basso, in modo da rappresentare l'immagine in dimensioni reali, senza coprire altri controlli. BUTTON_GEN2 [tipologia Albatros: 8011] *** NON USARE
Bitmap Button Image 	*** NON USARE

3.4. Gestione delle immagini di help delle lavorazioni

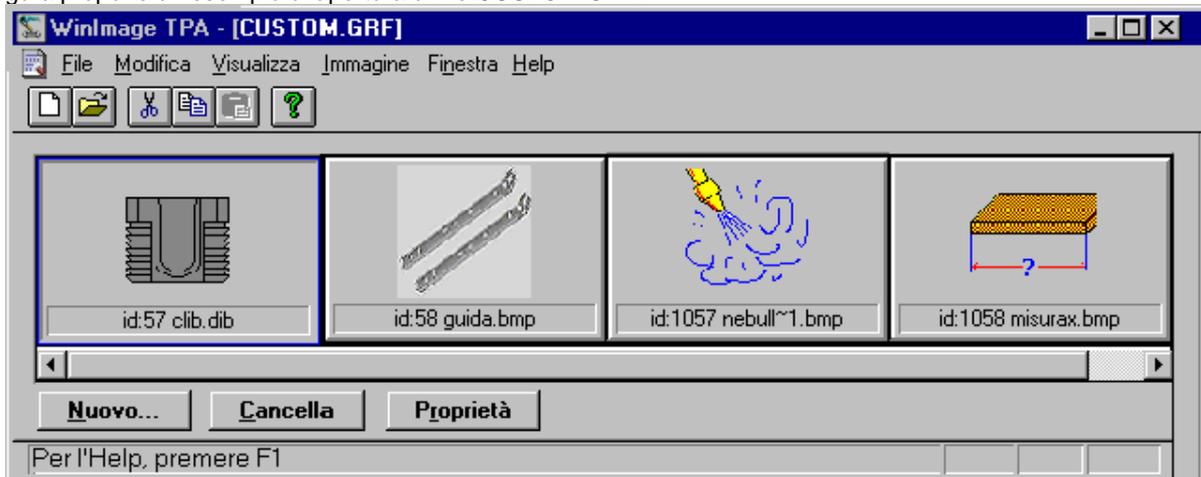
Per ogni lavorazione del gruppo base è assegnata una immagine di rappresentazione: si tratta di un elemento in file HELPEDI.GRF, con identificativo di ogni immagine di valore uguale al codice operativo della lavorazione.

Per le lavorazioni custom è possibile assegnare una immagine in file CUSTOM.GRF: assegnare l'identificativo dell'immagine di valore uguale al codice operativo della lavorazione.

ATTENZIONE: se l'immagine di una lavorazione custom non è disponibile in file CUSTOM.GRF, è cercata anche in file HELPEDI.GRF.

I files "GRF" sono gestiti dall'applicativo WINIMAGE.EXE, sempre a password costruttore.

La figura propone un esempio di apertura di file CUSTOM.GRF:



Sono visibili quattro immagini, con identificatori:

- [57] e [58]: sono immagini relativi a due lavorazioni custom elementari (puntuali, setup o logiche);
- [1057] e [1058]: sono immagini relativi a due lavorazioni complesse (codici di macro).

Per inserire una nuova immagine si sceglie una tra le possibilità:

- pulsante **Nuovo** in barra pulsanti posizionata in basso nella finestra. Si apre una pagina di proprietà dell'immagine:
 - **IMAGE ID**: assegnare valore numerico uguale al codice operativo della lavorazione;
 - **Commento**: è un campo opzionale di descrizione dell'immagine;
 - Pulsante "Cerca File..": permette di cercare un file archiviato su disco (le tipologie valide sono riportate in casella di finestra: bitmap (*.BMP), Metafile (*.WMF), TIFF files (*.TIF),...);
 - Confermando la scelta di un file, al ritorno in finestra di Proprietà viene riportata la rappresentazione dell'immagine;
 - Uscendo con Conferma dalla pagina di Proprietà l'immagine è inserita in file CUSTOM.GRF.
- Con immagine in *Clipboard* (esempio: ho selezionato e Copiato una immagine in programma PAINT): attivando il comando *Incolla* (disponibile in *menu Modifica* ed in *barra Strumenti*) viene attivata la creazione di una nuova immagine (assegnata da clipboard) ed aperta la pagina di Proprietà : assegnare il campo "IMAGE ID" ed uscire con conferma.

Per modificare una immagine già inserita:

- Per selezionare l'immagine da un file esterno: posizionarsi sull'immagine ed aprire la pagina di Proprietà. A questo punto scegliere il file con "Cerca File..";
- per modificare un'immagine già inserita: posizionarsi sull'immagine e copiarla con comando di *Copia* (disponibile in *menu Modifica* ed in *barra Strumenti*). Eseguire quindi *Incolla* ad esempio in applicativo PAINT e modificare l'immagine. Al termine:
 - Registrare l'immagine in file e procedere come al punto precedente; oppure:
 - mettere l'immagine in *Clipboard* (con: *selezione* e *Copia*), eliminare l'immagine da file CUSTOM.GRF (pulsante **Cancella** in banda pulsanti posizionata in basso nella finestra) ed inserirlo nuovamente con comando *Incolla*.

3.5. Lavorazioni con codice operativo in intervallo [1 – 1000]

Si tratta di lavorazioni generalmente definite secondo le esigenze di ogni macchina e sono gestite in base alla tipologia della lavorazione, che deve essere: puntuale, setup o logica.

In file EDICADWR.DEF sono assegnate lavorazioni ritenute di interesse generale (con valore di codice operativo in intervallo [1-200]) ed eventuali lavorazioni che si ritenesse di aggiungere in file EDICADWR.DEF avrebbero codice operativo comunque compreso in intervallo [1–200]. Si riportano ad esempio:

- “[82] FORO PER UTENSILE” esempio di lavorazione puntuale;
- “[89] SETUP FRESA PER UTENSILE” esempio di lavorazione setup fresa;
- “[95] SETUP LAMA” esempio di lavorazione setup lama;
- “[102] RAPIDO” esempio di lavorazione logica.

I codici operativi compresi in intervallo [201–1000] sono invece riservati per applicazioni custom: ogni client assegna in questo intervallo i propri codici operativi e ne mantiene documentazione.

3.5.1. Punto di applicazione

Con tipologie puntuale e di setup deve essere definito un punto di applicazione, che coincide con il punto di lavoro:

- il punto di applicazione è definito in due coordinate di piano (x ed y, tipologia dei parametri: [1] e [2]) ed una coordinata di profondità (z, tipologia del parametro [3]);
- la posizione xy del punto di applicazione (posizione sul piano della faccia) può essere assegnata in coordinate cartesiane (default) o polari:
 - ✓ sistema cartesiano: sono valide le programmazioni date per le coordinate x ed y;
 - ✓ sistema polare: sono valide le programmazioni date per le coordinate del polo (tipologia dei parametri: [31] per la coordinata x e [32] per la coordinata y), l'angolo (tipologia del parametro [8044]) ed il vettore (tipologia del parametro [8017]);
- la selezione della programmazione xy in coordinate polari avviene tramite il parametro di tipologia [8133], con impostazione di valore non nullo (1);
- per la selezione tra sistema cartesiano e polare:
 - ✓ prevedere due lavorazioni distinte:
 - ◇ una lavorazione per programmazione cartesiana: parametri [1] e [2] visibili e modificabili;
 - ◇ una lavorazione per programmazione polare: parametro [8133] nascosto (Hide), con valore di default =1; parametri [31], [32], [8044], [8017] visibili e modificabili; parametri [1] e [2] non assegnati.

3.5.2. Codice di lavorazione e codice esecutivo

Come visto, per il programma EDICAD ogni lavorazione è individuata da un codice numerico (codice operativo).

Anche a valle di EDICAD.EXE la lavorazione rimane comunque identificata con un codice numerico, che però può essere diverso (cioè: la lavorazione arriva in Matrice pezzo con un codice diverso).

Per il gruppo delle lavorazioni che stiamo esaminando e solo per le tipologie puntuale e setup è possibile assegnare un diverso codice operativo di deposito in matrice pezzo aggiungendo alla lavorazione il parametro di tipologia [8111]:

- ✓ Tipologia di controllo: integer;
- ✓ Modo di visualizzazione: nascosto (hide);
- ✓ Valore di default = codice di assegnazione in matrice pezzo (compreso in intervallo [1–1000]).

In file EDICADWR.DEF sono assegnati casi di riassegnazione di codice operativo. Alcuni esempi sono:

- la lavorazione “[84] FORO PER UTENSILE (POLARE)” arriva in Matrice pezzo con il codice [82] (lavorazione: “FORO PER UTENSILE”). Non interessa infatti sapere con quale geometria il foro è stato programmato (cartesiana o polare), ma serve mantenere l'informazione di scelta di tecnologia;
- la lavorazione “[83] FORO PER DIAMETRO (POLARE)” arriva in Matrice pezzo con il codice [81] (lavorazione: “FORO PER DIAMETRO”);
- la lavorazione “[53] INSERIMENTO DI CERNIERA” arriva in Matrice pezzo con il codice [50] (lavorazione: “INSERIMENTO”). In particolare: tutti i codici di inserimento arrivano in Matrice pezzo con il codice [50] di inserimento generico ed il riconoscimento del tipo di inserimento è affidato al parametro corrispondente di scelta della tipologia utensile (parametro di tipologia [1001]);
- la lavorazione “[93] SETUP FRESA PER UTENSILE (POLARE)” arriva in Matrice pezzo con il codice [89] (lavorazione: “SETUP FRESA PER UTENSILE”).

ATTENZIONE:

in caso di assegnazione di un Ottimizzatore custom (ormai è il caso generale), in Matrice pezzo devono arrivare solo codici operativi che questo Ottimizzatore è in grado di esaminare. Mentre l'Ottimizzatore generale (OPTIMIZE.EXE) interpreta le lavorazioni in modo dinamico, in base alla assegnazione in database lavorazioni (ma appunto perché dinamico anche meno specializzato !!), un Ottimizzatore custom è specifico di ogni applicazione: e della applicazione conosce in modo definito le specifiche di funzionamento. Tra queste: i codici operativi che la/le macchine possono mettere in esecuzione.

3.5.3. Sottotipi di lavorazioni

Il parametro [8112] permette di qualificare un sotto-tipo per lavorazione in intervallo [1-1000], di tipologie quindi: puntuale, setup o logica custom.

Impostare il parametro di tipologia [8112]:

- ✓ Tipologia di controllo: integer;
- ✓ Modo di visualizzazione: nascosto (hide);
- ✓ Valore di default = numero da 1 a 4.

Valore non impostato o non valido o parametro non presente corrisponde a sottotipo = 0.

La qualifica di un sottotipo di lavorazione è utilizzata:

- ✓ in assegnazione della palette delle lavorazioni (sigle: pn0, pn1, ...);
- ✓ in fase di ottimizzazione (permette ad esempio di differenziare l'ordinamento delle lavorazioni).

È assegnata una convenzione per il significato delle varie sotto-tipologie:

Sotto-tipo	Lavorazione puntuale	Lavorazione setup	Lavorazione logica
0	Forature normali	Setup fresa normale	Riservato per codici logici interni (IF,..)
1	Forature con utensile orientato	Setup fresa con utensile orientato	Libera assegnazione di logiche custom
2	Forature speciali (con scarico,..)	Setup fresa speciale	Libera assegnazione di logiche custom
3	Inserimenti	Setup lama	Libera assegnazione di logiche custom
4	Libera assegnazione	Libera assegnazione	Libera assegnazione di logiche custom

3.5.4. Come configurare una lavorazione custom in intervallo [1 - 1000]

Si deve ad esempio assegnare una lavorazione custom di tipologia puntuale.

È possibile copiare una tra le lavorazioni di EDICADWR.DEF e poi effettuare le personalizzazioni necessarie.

In file EDICADWR.DEF troviamo ad esempio le lavorazioni:

- "[82] FORO PER UTENSILE": foratura verticale con programmazione in coordinate cartesiane;
- "[84] FORO PER UTENSILE (POLARE)": foratura verticale con programmazione in coordinate polari.

Per copiare una lavorazione da file EDICADWR.DEF a file CUSTOM.DEF:

- avviare PASTUDIO.EXE;
- aprire il nodo delle lavorazioni di macchina e posizionare la banda di selezione sulla lavorazione " [82] FORO PER UTENSILE"



- senza espandere il nodo della lavorazione, selezionare il pulsante *Copia* della Barra strumenti;
- aprire il file CUSTOM.DEF (in menu di File, selezionare "Apri file di definizione" e cercare il file in cartella *cadcfg*;
- espandere il nodo delle lavorazioni custom di macchina;
- inserire la lavorazione copiata selezionando il corrispondente pulsante della Barra strumenti (attenzione: la lavorazione è inserita solo se non c'è già una lavorazione con stesso *Codice operativo* o *Descrizione* !!!);
- modificare la lavorazione:
 - ✓ se si vuole modificare la lavorazione semplificandola (ad esempio: eliminare la assegnazione di macchina, gruppo) o aggiungendo dei parametri custom: è possibile mantenere il codice operativo originale;
 - ✓ se si vuole assegnare un ciclo esecutivo nuovo (esempio: una foratura speciale):
 - ◇ cambiare il Codice operativo (scegliere un valore libero in intervallo [201-1000]);
 - ◇ cambiare la Descrizione;
 - ◇ in caso di tipologia puntuale e programmazione per diametro, valutare la assegnazione della informazione di Codice lavorazione base;
 - ◇ valutare la necessità di una Descrizione grafica dedicata (ad esempio: per una lavorazione di inserimento);

- ◇ valutare se impostare il parametro [8111] di assegnazione del codice di deposito in Matrice pezzo;
- ◇ assegnare il parametro [8112] di sotto-tipo;
- ◇ assegnare parametri custom:
 - tipologia in intervallo 9501-9999
 - tipologia di controllo: scegliere secondo necessità;
 - assegnare il Modo di visualizzazione ed eventuale Valore di default;
 - assegnare deposito in matrice:
 - nome di matrice: "MXPEZZO",
 - numero di colonna: tra quelli disponibili (vedere la documentazione relativa alle assegnazioni predefinite per la matrice pezzo),
 - info aggiunte: imposta il significato **dimensionale** del parametro (serve in caso di utilizzo della lavorazione in programma scritto in [inch]). Valori significativi:
 - 1 se il parametro non ha dimensione
 - 2 se il parametro è assegnato in unità di misura: [mm] o [inch] (esempi: quota, offset)
 - 3 se il parametro è assegnato in unità di misura: [m/min] o [inch/sec].

3.6. Prospetti riassuntivi di lavorazioni puntuali e di setup

Di seguito sono riportati i prospetti complessivi dei parametri che è possibile assegnare per una lavorazione puntuale o di setup.

3.6.1. Prospetto di lavorazione puntuale

Caso di programmazione in sistema cartesiano:

Parametro	Tipo	Note (*)	Significato
PRQX [1] PRQY [2] PRQZ [3]	Double	S	punto di applicazione x, y, z. PRQZ: non configurare se non gestito.
PROFFZ [8056]	Int	N	Flags di abilitazione per l'asse z: OFF(1)/ON(0). Se non presente o se in stato ON (0): abilita il movimento dell'asse.
PRASSINC [8015]	Int	N	Assoluto(0)/ relativo(1). Se non presente: applica assoluto. Applicato a: PRQX, PRQY, PRQZ
PRESR1 [8070] PRESR2 [8073] PRCONDIZ1 [8060]	Double Double Int	N	Parametri di assegnazione <i>IF diretto</i>
Parametri per assegnazione della tecnologia (utensile):			
PRFI_TOOL [1002]	Double	N	Diametro utensile
PRN_MAC [201]	Int	N	Numero macchina
PRN_GRP [203]	Int	N	Numero gruppo
PRN_TOOL [205]	Int	N	Numero utensile
PRTIPO_TOOL [1001]	Int	N	Tipologia utensile
[9001-9999]		S/H	Parametri custom (devono arrivare in matrice pezzo)
PRMTXNAME [8111]		N/H	Codice operativo di deposito in matrice pezzo
PRPNTSET [8112]		N/H	Sottotipo di lavorazione

(*) stato di configurazione del parametro:

S= Visibile e Modificabile

H= Nascosto (con assegnato valore di default)

N= non obbligatorio (può essere: S, H oppure non configurato)

Caso di programmazione in sistema polare:

Parametro	Tipo	Note	Significato
PRQZ [3]	Double	S/N	punto applicazione z: non configurare se non gestito.
PROFFZ [8056]	Int	N	Flags di abilitazione per l'asse z: OFF(1)/ON(0). Se non presente o se in stato ON (0): abilita il movimento dell'asse.
PRASSINC [8015]	Int	N	Assoluto(0)/ relativo(1). Se non presente: applica assoluto. Applicato a: PRQI, PRQJ, PRQZ
PRESR1 [8070] PRESR2 [8073] PRCONDIZ1 [8060]	Double Double Int	N	Parametri di assegnazione <i>IF diretto</i>

PRPOLARE [8133]	Int	H	Sistema polare: valore di default =1
PRQI [31] PRQJ [32]		N	Coordinate x/y del centro di sistema polare (polo): assoluto o relativo, come da parametro PRASSINC.
PRANG [8044]	Double	N	Angolo di sistema polare: in unità di gradi e decimali di gradi.
PRRADIUS [8017]	Double	N	Vettore di sistema polare.
Parametri per assegnazione della tecnologia (utensile):			
PRFI_TOOL [1002] ..PRTIPO_TOOL[1001]			Vedi caso di sistema cartesiano
[9001-9999]		S/H	Parametri custom (devono arrivare in matrice pezzo)
PRMTXNAME [8111]		N/H	Codice operativo di deposito in matrice pezzo
PRPNTSET [8112]		N/H	Sottotipo di lavorazione

3.6.2. Prospetto di lavorazione setup

Caso di programmazione in sistema cartesiano:

Parametro	Tipo	Note	Significato
PRQX [1] PRQY [2] PRQZ [3]	Double	S /N	Punto applicazione x, y, z: PRQZ: non configurare se non gestito.
PROFFZ [8056]	Int	N	Flags di abilitazione per l'asse z: OFF(1)/ON(0). Se non presente o se in stato ON (0), abilita il movimento dell'asse.
PRASSINC [8015]	Int	N	Assoluto(0)/ relativo(1). Se non presente: assoluto. Applica a: PRQX, PRQY, PRQZ
PRLKPROFILE [8101]	Int	N	=1: richiede aggancio a punto di applicazione precedente.
Parametri per assegnazione della correzione utensile:			
PRCORR_RF[40]	Int	N	Flag di correzione utensile: 0=Off, 1=Sx, 2=Dx
PRCORR_RF2 [8134]	Int	N	Flag di correzione utensile, da usare in alternativa a PRCORR_RF: 0=Off, 1=Sx, 2=Dx. Comunque: non applica correzione in fase esecutiva.
PRRADIUS_RF [9020]	Double	N	Raggio di correzione utensile
PRCORR_CONT [39]	Int	N	Flag di contornatura in correzione utensile: = 1: inserisce raccordi in correzione = 2: applica contornatura (riduce raccordi ad intersezioni) non presente o valore #: non influisce sul settaggio generale (default).
PRCORR_TOZP [38]	Int	N	flag di correzione utensile a Zp: = 1: applica correzione normale; = 2: applica correzione a Zp. Cioè: (con PRCRF_INIT (RF)→) non attivo, aggiunge tratto lineare di inizio correzione al setup, in modo da portarsi a Zp sul punto (x,y) programmato ed al punto corretto nel pezzo; Se non presente o valore #: non influisce sul settaggio generale (default).
PRCRF_INIT [8135]	Int	N	(RF)→ flag di avvio lineare in correzione utensile
PRCRF_END [8136]	Int	N	(RF)→ flag di chiusura lineare in correzione utensile
Parametri per assegnazione della tecnologia (utensile):			
PRFI_TOOL [1002] ..PRTIPO_TOOL[1001]			Vedi caso di <i>Lavorazione puntuale</i>
[9001-9999]		S/H	Parametri custom (devono arrivare in matrice pezzo)
PRMTXNAME [8111]		N/H	Codice operativo di deposito in matrice pezzo
PRPNTSET [8112]		N/H	Sottotipo di lavorazione

Caso di programmazione in sistema polare:

Parametro	Tipo	Note	Significato
PRQZ [3]	Double	S /N	punto applicazione z: PRQZ: non configurare se non gestito.
PROFFZ [8056]	Int	N	Flags di abilitazione per l'asse z: OFF(1)/ON(0). Se non presente o se in stato ON (0), abilita il movimento dell'asse.
PRASSINC [8015]	Int	N	Assoluto(0)/ relativo(1). Se non presente: assoluto. Applica a: PRQI, PRQJ, PRQZ
PRPOLARE [8133]	Int	H	Sistema polare: valore di default =1.

PRQI [31] PRQJ [32]	Double	N	Coordinate x/y del centro di sistema polare (polo): assoluto o relativo, come da parametro PRASSINC.
PRANG [8044]	Double	N	Angolo di sistema polare: in unità di gradi e decimali di gradi.
PRRADIUS [8017]	Double	N	Vettore di sistema polare.
PRLKPROFILE [8101]	Int	N	=1: richiede aggancio a punto di applicazione precedente.
Parametri per assegnazione della correzione utensile:			
PRCORR_RF[40] ...PRCRF_END [8136]			Vedi caso di sistema cartesiano
Parametri per assegnazione della tecnologia (utensile):			
PRFI_TOOL [1002] ..PRTIPO_TOOL[1001]			Vedi caso di <i>Lavorazione puntuale</i>
[9001-9999]		S/H	Parametri custom (devono arrivare in matrice pezzo)
PRMTXNAME [8111]		N/H	Codice operativo di deposito in matrice pezzo
PRPNTSET [8112]		N/H	Sottotipo di lavorazione

3.7. Lavorazioni custom di profilo [2300 – 2400]

Si tratta di lavorazioni di tipologia arco o linea che necessitano di personalizzazioni:

- inserire una lavorazione in file CUSTOM.DEF copiando la lavorazione di riferimento geometrico da file EDICADWR.DEF. Si copi ad esempio la lavorazione “[2201] L01”;
- modificare la lavorazione in file CUSTOM.DEF:
 - ◊ cambiare il Codice operativo (scegliere un valore libero in intervallo [2300-2400]);
 - ◊ lasciare invariata la tipologia (nell'esempio: *Lineare*);
 - ◊ assegnare il parametro di tipologia [8110]:
 - Tipologia di controllo: integer;
 - Modo di visualizzazione: nascosto (hide);
 - Valore di default = 2201.

Il parametro [8110] permette di individuare il codice di riferimento per la compilazione del nuovo codice operativo, che impone le regole di interpretazione geometrica.

- ◊ lasciare **invariati** i parametri che assegnano la geometria della lavorazione di riferimento (nell'esempio: “[2201] L01”)
- ◊ assegnare parametri custom: tipologie in intervallo 9501-9999

3.8. Codici di macro [1001 – 2000]

Si tratta di lavorazioni generalmente definite secondo le esigenze di ogni macchina e che comportano l'esecuzione in macchina di un ciclo di lavorazioni elementari (tipologie: puntuale, setup, arco, linea, logica).

La tipologia delle lavorazioni deve essere: **complessa**.

In file EDICADWR.DEF sono assegnati codici di macro ritenuti di interesse generale (con valore di codice operativo in intervallo [1001-1500]) ed eventuali lavorazioni che si ritenesse di aggiungere in file EDICADWR.DEF avrebbero codice operativo comunque compreso in intervallo [1001-1500]. Si riportano ad esempio:

- “[1001] FITTING X”
- “[1025] ELLISSE X”
- “[1050] LAMA X”.

Sempre in file EDICADWR.DEF sono assegnati codici in range [1900-2000]: si tratta di codici utilizzati in programmi di trascodifica da altro formato (esempio: i codici per conversioni da CNC90).

I codici operativi compresi in intervallo [1501-1899] sono invece riservati per applicazioni custom: ogni client assegna in questo intervallo i propri codici e ne mantiene documentazione.

Anche per assegnare una lavorazione custom di macro si consiglia di partire copiando una lavorazione di macro tra quelle assegnate in file EDICADWR.DEF e poi effettuare le personalizzazioni necessarie.

Prima di arrivare ad assegnare un codice di macro è però necessario scrivere la materia prima:

l'assegnazione ed utilizzo di un codice di macro comporta infatti l'esistenza di un macro-programma.

3.8.1. Macro-programma

Un macro-programma assegna un ciclo di lavorazioni definito secondo regole generalmente custom.

È quindi paragonabile ad un generico programma gestito da EDICAD: anche un programma assegna un ciclo di lavorazioni, definito secondo regole custom, ma ci sono differenze sostanziali:

- un programma generico viene scritto per essere direttamente eseguito;
- un macro-programma viene invece scritto per essere “applicato” in un programma, esattamente come una lavorazione elementare.

L'operatore sceglie un codice di macro tra quelli disponibili da palette lavorazioni allo stesso modo di come sceglie una lavorazione di foratura o un setup fresa: l'operatore sa come la macchina lavora quando incontra una lavorazione di foratura e sa anche come la macchina lavora quando “*incontra*” un codice di macro (di fatto in macchina non arriva il codice di macro, ma le lavorazioni elementari che ne conseguono dal suo sviluppo).

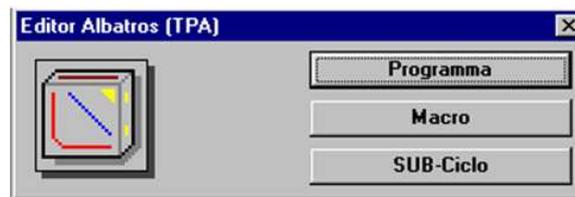
Dal punto di vista dell'utilizzo non esiste differenza tra una lavorazione di foratura o un setup fresa o un codice di macro: sono tutte paragonabili a mattoni di costruzione, solo di forma o colore diverso.

Un macro-programma:

- è archiviato in direttorio dedicato (a partire da: cadcfg\mcr\);
- non è eseguibile direttamente;
- è gestibile (creazione e/o modifica e/o salvataggio) a partire da password costruttore e la gestione è effettuata con programma EDICAD;
- è utilizzabile in un programma **solo** utilizzando un codice di macro.

È possibile assegnare la tipologia di macro-programma in uno dei due modi indicati:

- con richiesta di nuovo file (a password costruttore) viene presentata la finestra di selezione della tipologia da assegnare al programma:



scegliere la voce **Macro** (a password inferiore la finestra non compare e viene avviata comunque la creazione di un Programma);

- con aperto un programma di differente tipologia (programma o sottociclo): scegliere la voce “*Cambia tipologia di programma*” in menu di File (la voce è gestita solo a password costruttore). Anche in questo caso viene presentata una finestra di selezione della tipologia da assegnare al programma: scegliere la voce **Macro**.

Attenzione: il cambiamento di tipologia è eseguito solo se possibile. Cioè:

- il programma non può avere assegnata una geometria variabile (pezzo sagomato o facce fittizie);
- non devono essere utilizzati codici complessi (di macro o di chiamate a sottoprogramma);
- non devono essere utilizzati codici non assegnati nel database delle lavorazioni.

Dal punto di vista della programmazione:

- avviene come per un programma normale: con definizione di un pezzo ed applicazione di lavorazioni in una o più facce;
- sono utilizzabili le variabili r (da r0 ad r299);
- non sono gestiti: offsets, variabili v, sequenze;
- in applicazione di lavorazioni in una faccia: sono utilizzabili variabili ausiliarie (variabili di tipo \$), normalmente non disponibili in edit di programma;
- sono utilizzabili le lavorazioni elementare: anche quelle protette (esempio: “[95] SETUP LAMA”);
- sono utilizzabili particolari lavorazioni logiche, normalmente non disponibili in edit di programma:
 - istruzioni di ciclo (FOR, FOREVER, ENDFOR);
 - istruzioni di assegnazione variabili ausiliarie (variabili di tipo \$);
 - istruzioni speciali di controllo (BREAK, CONTINUE);
- **non** sono utilizzabili codici complessi (codici di chiamata sottoprogramma, codici di macro).

I codici di lavorazioni logiche riservate alla stesura dei macro-programmi sono esaminati in un capitolo successivo.

Definire un macro-programma ed il (o i) codice corrispondente di applicazione non presenta difficoltà particolari, ma occorre definire bene la sequenza dei passaggi che è necessario percorrere:

- analizzare il problema sulla “carta”: è una regola fondamentale avere una idea chiara del problema, prima da partire a scrivere il macro-programma. In particolare: occorre valutare le variabili che servono a definire il problema. Si consideri un esempio tra quelli già implementati in EDICADWR.DEF: l'esecuzione di una fila di fori a passo costante lungo l'asse x (lavorazione: “REPEAT X”). Il problema è definito nei punti che seguono:

- ✓ parametri che rimangono uguali per tutti i fori: posizione y, profondità di foratura, assegnazione di tecnologia (scelta diretta di utensile oppure di diametro), velocità e rallentamenti di foratura;
- ✓ parametri di definiscono lo sviluppo delle forature: posizione x del primo foro, numero totale di forature, interasse x tra forature consecutive (passo x).

Lo schema logico di sviluppo è: esegue un numero assegnato di fori, a passo costante in x, a partire da una posizione assegnata.

Le caratteristiche che abbiamo indicato (posizione y, profondità di foratura, assegnazione di tecnologia, velocità e rallentamenti di foratura, posizione x del primo foro, numero totale di forature, interasse x tra forature consecutive) sono grandezze che devono essere assegnate al momento dell'inserimento della lavorazione di "REPEAT X". Come vedremo: alcuni tra questi parametri (posizione y, profondità di foratura, posizione x del primo foro) possono essere assegnati con modalità generali, comunque valide per ogni codice di macro, mentre gli altri devono essere assegnati come variabili del macro-programma;

- si inizia a scrivere un macro-programma solo dopo avere valutato che è necessario farlo:
 - ✓ se l'esecuzione richiede l'utilizzo di cicli FOR. Esempi: l'esecuzione di una cava di svuotamento oppure del "REPEAT X" (il caso del nostro esempio);
 - ✓ se ne è richiesto l'utilizzo in sottoprogrammi (si ricorda che le chiamate di sottoprogramma non possono essere innestate);
 - ✓ è opzionale scrivere un macro-programma o un sottoprogramma (o un sotto-ciclo) in tutti gli altri casi;
- avviare il programma EDICAD da password costruttore: condizione necessaria per la gestione di programmi di tipologia "macro";
- assegnare un nome significativo al programma di macro: un nome che richiami il codice della macro (esempio: mcr1560) o che sia di suggerimento per le lavorazioni richieste (esempio: cava);
- di preferenza, per l'archiviazione dei macro-programmi custom creare una cartella a partire da "cadcfg\mcr", in modo da distinguere i macro-programmi utilizzati in EDICADWR.DEF da quelli utilizzati in CUSTOM.DEF;
- si può iniziare la scrittura del macro-programma avviando la funzionalità di programma nuovo: in finestra di selezione della tipologia da assegnare al programma scegliere la voce **Macro**;
- definire subito la lista delle *grandezze variabili* che devono essere assegnate in inserimento del codice di macro. Le abbiamo già individuate: assegnazione di tecnologia, velocità e rallentamenti di foratura, numero totale di forature, interasse x tra forature consecutive.

Ciò avviene assegnando le variabili r con le stesse regole valide per un normale programma. Solamente ora la selezione di variabile *riassegnabile* indica che la variabile può essere assegnata al momento dell'applicazione del codice di macro.

Torniamo al nostro esempio di macro-programma associato alla esecuzione di un "REPEAT X", come definito in file EDICADWR.DEF. Il macro-programma in questione è archiviato in cartella "cadcfg\mcr" con nome REPEATX.MCR. Proviamo ad aprirlo (con EDICAD):

- ✓ innanzitutto vediamo che ha lavorazioni su due facce: faccia 1 (che ha nome: "fitting x") e faccia 2 (che ha nome: "repeat x"). Diciamo subito che il macro-programma è utilizzato per assegnare due codici di macro in EDICADWR.DEF: il codice di "Fitting x" (esegue il programma della faccia 1) e quello di "Repeat x" (esegue il programma della faccia 2). È un fatto normale, che esamineremo più avanti: adesso possiamo comunque anticipare che un macro-programma può essere utilizzato per assegnare anche più di un codice di macro;
- ✓ apriamo la lista delle variabili r ed esaminiamone alcune (qui sono assegnate le variabili utili sia ad assegnare il repeat x che il fitting x):

◇ r12: tipologia *float*, *riassegnabile*, valore =32, commento "passo x";

◇ r17: tipologia *intero*, *riassegnabile*, valore =5, commento "numero di forature".

Se entriamo in faccia 2, possiamo facilmente intuire (e per ora ci basta) che r12 è utilizzato per assegnare il passo tra forature consecutive, mentre r17 è utilizzato per assegnare il numero totale delle forature.

È anche superfluo sottolineare l'importanza che può avere il campo commento delle variabili r, soprattutto in caso di macro-programmi che richiedono molte assegnazioni.

Possiamo volere limitare alcune riassegnazioni, in applicazione di un "REPEAT X": ad esempio per il momento non vogliamo dare la possibilità di cambiare le quote di rallentamento. Abbiamo detto *per il momento*: potrebbe comunque essere utile modificare il macro-programma in una seconda fase e rendere riassegnabili anche le quote di rallentamento. In questo caso è di certo conveniente assegnare anche le quote di rallentamento con delle variabili r (per ora lasciate: non riassegnabili): una revisione futura del macro-programma può metterle riassegnabili o solo cambiarne il valore senza che nessuna altra modifica deve essere apportata al macro-programma.

- valutare le condizioni di validità delle variabili riassegnabili. Nel nostro esempio:
 - ◇ la variabile r12 può essere positiva o negativa, ma definiamo che il suo valore assoluto debba essere almeno uguale ad 1 (1 mm);
 - ◇ la variabile r17 può avere valori (interi) compresi tra 1 e 300 (definiamo che al massimo il macro-programma sviluppa 300 forature).

Per ogni condizione di non validità delle variabili occorre:

- ◊ assegnare un default di funzionamento. Nel nostro esempio: se è $r17 < 1$ ($r17$ è minore di 1) possiamo comunque assumere di eseguire un foro;
- ◊ oppure definire una segnalazione di errore (con codice BREAK o ERRORE). Nel nostro esempio: se è $r17 < 1$, il macro-programma riconosce una situazione di errore e non esegue nessuna lavorazione.

➤ varianti dello stesso problema possono trovare soluzione:

- ◊ sviluppando i vari sviluppi su più facce del macro-programma;
- ◊ oppure: sviluppando i vari aspetti su una sola faccia con verifica di condizioni differenti.

I due punti possono inoltre essere entrambi presenti, se necessario.

Nel nostro esempio: poniamo il caso di dover scrivere un macro-programma che ripete lavorazioni di foratura in direzione x, con imposti:

- ◊ gli estremi di sviluppo ed il passo tra due lavorazioni successive;
- ◊ oppure: la posizione di partenza, il numero di lavorazioni ed il passo.

Il problema qui posto corrisponde esattamente al caso del macro-programma REPEATX.MCR:

- ◊ il primo caso corrisponde alla soluzione di "fitting x": programmato in faccia 1;
- ◊ il secondo caso corrisponde alla soluzione di "repeat x": programmato in faccia 2.

Soluzioni alternative sarebbero state:

- ◊ scrivere due macro-programmi: uno per il "fitting x" ed uno per il "repeat x";
- ◊ scrivere entrambi i programmi nella stessa faccia (esempio: faccia 1), in due cicli IF posti in esecuzione alternativa testando il valore di una variabile r (riassegnabile);
- ◊ è anche possibile avere un macro-programma con più facce programmate e due o più rami in esecuzione alternativa in ogni faccia.

La scelta di quale soluzione adottare non è definibile a priori ed è abbastanza ininfluente in caso di programmi di bassa complessità. Nel caso di programmi di complessità medio/alta è preferibile dividere i casi in facce differenti:

- ◊ facilita la stesura del programma di macro (si esaminano aspetti differenti del problema in momenti separati);
- ◊ riduce la dimensione del programma di ogni faccia.

Nelle scelte della soluzione di stesura del macro-programma deve anche essere considerata la possibilità o necessità di applicare il codice di macro con sviluppo di chiamate indotte: una tale eventualità obbliga a sviluppare tutte e solo le facce eventualmente interessate. In questo caso il macro-programma viene visto come una sorta di "pezzo campione", piuttosto che come un "contenitore" di procedure elementari.

Si può ora procedere a scrivere il testo della macro.

A tale proposito è necessario segnalare che le variabili locali (tipo \$) ed i condizionamenti logici, compresi i cicli FOR, sono assegnati e risolti solo in fase di utilizzo del codice di macro. Questi aspetti determinano alcuni fatti specifici della funzionalità di scrittura di un testo di macro:

- innanzitutto la rappresentazione grafica non è sempre significativa. Si pensi ad un profilo costruito con sviluppo di un ciclo FOR: in stesura del testo di macro si vede rappresentato solo ciò che è direttamente interpretato, senza interpretazione o sviluppo di assegnazioni di variabili locali, condizionamenti logici o cicli;
- in inserimento o modifica di una lavorazione possono essere segnalate situazioni di errore fasulle, da considerare non significative. Si esaminano due esempi tipici:
 - ◊ impostazione di parametro come "(r5+r10)/\$5": in edit di macro-programma la variabile locale \$5 ha comunque valore 0 e ciò determina un errore di programmazione parametrica (denominatore nullo);
 - ◊ se poi il parametro imposta il raggio di un arco: viene anche segnalato errore geometrico in codice di interpolazione circolare.

I casi di errori qui proposti non possono costituire un filtro per la assegnazione della lavorazione e/o la registrazione del testo: quello che interessa è infatti lo sviluppo della macro in un programma.

Per tale motivo, in caso di errori di interpretazione di una lavorazione:

- ◆ viene comunque proposta la lista degli errori rilevati;
- ◆ chiudendo la finestra degli errori con pulsante RETRY si rientra in fase di modifica della lavorazione;
- ◆ chiudendo la finestra degli errori con pulsante CHIUDI si prosegue: con successiva conferma delle impostazioni effettuate viene ignorata la condizione di errore e la lavorazione è aggiornata come da impostazioni.

3.8.2. Come configurare un codice di macro

A questo punto possiamo modificare il file delle lavorazioni custom CUSTOM.DEF.

Avviare il programma PASTUDIO.EXE, aprire il file CUSTOM.DEF, espandere il nodo delle lavorazioni custom e richiedere una nuova lavorazione (pulsante **NUOVO** in parte bassa della finestra):

- Assegnare il Codice operativo (scegliere un valore in intervallo [1501-1899]) e la Descrizione;
- tipologia: complessa;
- tipologia finestra: *Modal* (nel caso particolare di codice che si vuole tenere nascosto all'utente finale ed utilizzare solo in testi di sotto-ciclo: assegnare *Non Visibile*).

È anche possibile inserire la nuova lavorazione copiandola da una di assegnazione similare (ad esempio in EDICADWR.DEF).

Esaminiamo i parametri da assegnare in codice di macro:

- parametro di assegnazione macro-programma. Tipologia: [8098]
 - ◊ Tipologia di controllo: *Edit*;
 - ◊ Modo di visualizzazione: nascosto (hide);
 - ◊ Valore di default = path di lettura macro-programma (relativo a "cadcfg\mcr"). Esempi:
 - se il macro-programma è: "..\cadcfg\mcr\cava" impostare "cava";
 - se il macro-programma è: "..\cadcfg\mcr\custom\cava.uno" impostare "custom\cava.uno";
 - per il codice "[1002] RIPETI X" in file EDICADWR.DEF: è impostato "repeatx.mcr".

Il parametro [8098] è obbligatorio: dice quale macro-programma viene utilizzato per assegnare il codice di macro. È fondamentale sottolineare come non sia possibile chiamare un macro-programma archiviato fuori dalla cartella "cadcfg\mcr".

- parametro di selezione della faccia del macro-programma. Tipologia: [6]
 - ◊ Tipologia di controllo: (*Integer*);
 - ◊ Modo di visualizzazione: nascosto (hide);
 - ◊ Valore di default = numero di faccia del macro-programma (se è richiesta applicazione di chiamate indotte =-1). Per il codice "[1002] RIPETI X" in file EDICADWR.DEF: è impostato 2 (chiama la faccia 2 del macro-programma : "..\cadcfg\mcr\repeatx.mcr").

Se il parametro [6] non è assegnato: viene chiamata la faccia 1 del macro-programma. Importante: con il parametro [6] si dice quale faccia si vuole utilizzare del macro-programma e nessuna importanza ha la faccia dove il codice di macro è applicato, a meno che non sia configurato un meccanismo di chiamate indotte.

- parametri di assegnazione di condizioni logiche per l'esecuzione (se si vuole assegnare una condizione sul codice di macro).
Si tratti dei parametri utilizzati in codici generici di applicazione di sottoprogramma (l'interpretazione è uguale): aprire la prima pagina della lavorazione "[2010] SUB0" di EDICADWR.DEF e copiare i controlli

- parametro di assegnazione in palette lavorazioni. Tipologia: [8117]
 - ◊ Tipologia di controllo: (*Integer*);
 - ◊ Modo di visualizzazione: nascosto (hide);
 - ◊ Valore di default = numero di sotto-gruppo dei codici (1-20);

Non sarà possibile utilizzare il codice di macro se non è reso disponibile su un pulsante della palette.

- parametro di assegnazione immagine di help di default. Tipologia: [8010]-BUTTON_GEN1
 - ◊ Tipologia di controllo: (*Button*);

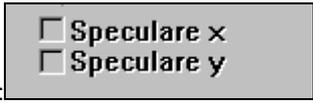
- piazzamento per traslazione in modalità assoluta o relativa:

parametro [8020] punto di applicazione lungo l'asse x
 parametro [8021] punto di applicazione lungo l'asse y
 parametro [8022] punto di applicazione lungo l'asse z
 parametro [8015] modalità assoluta o relativa.

Si tratti dei parametri X1, Y1, Z1 e rel utilizzati in codici generici di applicazione di sottoprogramma (l'interpretazione è uguale): aprire la prima pagina della lavorazione "[2010] SUB0" di EDICADWR.DEF e copiare i controlli

- ◊ assegnare solo i parametri necessari;

- ◇ eventualmente cambiare la Descrizione dei parametri;
- ◇ eliminare il parametro *ref* se non si vuole ammettere piazzamento in coordinate relative;
- parametro di aggancio punto. Tipologia [8101]:
 - ◇ il parametro è utilizzato in codici generici di applicazione di sottoprogramma (l'interpretazione è uguale);
 - ◇ assegnare il parametro solo nel caso in cui il macro-programma inizia lo sviluppo con un profilo ed il setup del profilo può essere *saltato*.
- Parametri di trasformazioni geometriche: è un sotto-insieme delle trasformazioni applicabili ad un generico codice di applicazione di sottoprogramma (l'interpretazione è uguale).
Le trasformate richieste sono applicate con l'ordine di seguito riportato:

- *rotazione* (parametro [8044]): 
- *speculare* (parametri: [8103] per speculare x, [8104] per speculare y): 

➤ **assegnare *tutti e solo* i parametri corrispondenti a variabili riassegnabili del macro-programma, con tipologia:**

- ◇ [8500] per variabile r0
- ◇ [8501] per variabile r1
- ◇ ...
- ◇ [8799] per variabile r299.

Per ogni parametro:

- ◇ Tipologia di controllo: come necessario (si raccomanda di **non** usare le tipologie puramente numeriche: Integer, Float);
- ◇ Modo di visualizzazione: Visibile e modificabile oppure Nascosto;
- ◇ Valore di default = se richiesto (necessario se il parametro non è modificabile).

Per un parametro che imposta una variabile riassegnabile del macro-programma può essere assegnata una regola di trasformata geometrica. Questa regola è applicata al parametro in esecuzione di uno strumento di trasformata geometrica (Esempio: Strumento di simmetria; Strumento di rotazione).

Per il parametro in questione, l'informazione è assegnata nel campo di Info aggiunte:

Codice	Regola	Esempio
10	cambia segno con simmetria verticale	passo di Repeat x
11	cambia segno con simmetria orizzontale	passo di Repeat y
12	cambia segno con simmetria verticale o orizzontale	
13	simmetria verticale di angolo (180- valore)	
14	simmetria orizzontale di angolo (- valore)	
15	simmetria verticale e/o orizzontale di angolo (caso 13 + caso 14)	
16	simmetria booleana con simmetria verticale o orizzontale (valore 0 diventa 1 e viceversa)	rotazione di arco
17	simmetria di valori 1 e 2 con simmetria verticale o orizzontale (valore 1 diventa 2 e viceversa).	correzione utensile

ATTENZIONE: nel caso in cui sia necessario assegnare regole di trasformata geometrica ai parametri della macro, è opportuno non assegnare i parametri di rotazione e/o speculare: le trasformate sarebbero infatti sommate, con conseguente non facile comprensione dell'effetto risultante.

Per nessun parametro di un codice di macro deve essere assegnato un deposito in matrice pezzo (nome + colonna): in matrice pezzo non arriva infatti il codici di macro, ma le lavorazioni che il codice applica.

3.8.3. Creazione di modello per PASTUDIO

Un aiuto per la creazione di un codice di macro può essere la creazione del *Modello* del macro-programma. Si tratta di un file che viene registrato in cartella "cadcfg\models" con lo stesso nome del file di registrazione del macro-programma: la registrazione avviene con selezione della voce "Salva modello per PASTUDIO", disponibile a menu di File.

La voce "Salva modello per PASTUDIO" è gestita solo in caso di tipologie di programma: macro e sottociclo.

Il file fornisce gli elementi fondamentali utili alla assegnazione di un codice di macro per il macro-programma in questione. Di seguito è riportato un ipotetico esempio:

/****** MODEL FOR PASTUDIO *****/	
//.....range advised (macro)	Range consigliato per il codice operativo:
CODE_INI=1501	◇ estremo inferiore del range
CODE_END=1899	◇ estremo superiore del range.
//.....general parameters	Parametri generali per un codice di macro:
#8098=REPEATX.MCR prname t 0 0 h	nome di macro-programma (tipologia: 8098; valore di default: "REPEATX.MCR"; nome proposto: "prname"; tipologia di controllo: t = testo; visibilità: h=hide)
#6=1 prside i 0 0 w	faccia (tipologia: 6; valore di default proposto: 1; nome proposto: "prside"; tipologia di controllo: i = intero; visibilità proposta: w = visibile e modificabile).
	Assegnare come necessario.
#8117=0 prmenu i 0 0 h	Assegnazione gruppo in palette lavorazioni (tipologia: 8117; ...): assegnare come necessario.
#8020= prqx1 f 0 0 w	Quote x/y/z di piazzamento (tipologie: 8020/8021/8022; nessun valore di default; nome proposto: "prqx1"/...; tipologia di controllo: f = float; visibilità proposta: w = visibile e modificabile).
#8021= prqy1 f 0 0 w	Assegnare come e se necessario.
#8022= prqz1 f 0 0 w	
#8015=0 prel i 0 0 w	Relativo (tipologia: 8015; ...): assegnare se necessario.
#8044= prang f 0 0 w	Angolo di rotazione (tipologia: 8044;...): assegnare come e se necessario.
#8103=0 prmirx i 0 0 w	Speculare x (tipologia: 8103; ...): assegnare come e se necessario.
#8104=0 prmiry i 0 0 w	Speculare y (tipologia: 8104; ...): assegnare come e se necessario.
#8010=0 # b 0 0 w	Pulsante di help grafico (tipologia: 8010): assegnare se necessario.
//.....variables	Parametri di variabili riassegnabili del macro-programma:
#8507=0 ri w f	R7 (tipologia: 8507; valore originale: 0; commento: "ri"; tipologia: f = float)
#8508=0 ro w f	R8 (tipologia: 8508; valore originale: 0; commento: "ro"; tipologia: f = float)
#8509=0 1=mac e gruppo con diametro w i	R9 (tipologia: 8509; valore originale: 0; commento: "mac e gruppo con diametro"; tipologia: i = intero)
#8510=32 passo x w f	R10 (tipologia: 8510; valore originale: 32; commento: "passo x"; tipologia: f = float)

3.8.4. Parametri di assegnazione ausiliaria (parametri di appoggio)

Esaminiamo qui una questione di interesse specifico nella stesura di un testo di macro (come si vedrà: si applica anche a testo di sottociclo).

È utile porre il problema partendo da un caso pratico: il macro-programma assegna un profilo e si vuole impostare il parametro di correzione utensile del setup. Definiamo di utilizzare la variabile r10 del macro-programma:

- riassegnabile;
- tipologia: intero;
- i valori validi sono: 0 = correzione non richiesta, 1 = correzione sinistra, 2 = correzione destra.

Apriamo la finestra di impostazione di un setup fresa a *password costruttore*:



vediamo il controllo di edit intestato "C. rf" che normalmente non appare (in realtà, il campo può apparire in pagina di Tecnologia oppure in altra posizione rispetto alla finestra qui riportata, ma la questione è del tutto ininfluyente).

Aperto infatti la stessa finestra a password utente è visibile solo il controllo intestato "C. rf" di assegnazione su radio pulsanti (Off/ Sx/ Dx): un controllo di questo tipo non consente di certo una assegnazione diretta.

Il controllo di edit intestato "C. rf" permette invece di editare il parametro di correzione, con formalismo anche parametrico (nel nostro caso deve essere assegnato a "r10"): il controllo di edit "C. rf" ha la funzione di *parametro di appoggio* per la assegnazione del corrispondente parametro di normale assegnazione diretta.

In EDICADWR.DEF allo stesso modo è predisposta la modalità di assegnazione diretta (su controllo di edit) per i parametri:

Doppia profondità di Z	in lavorazione "SETUP LAMA"
Calcolo corda	in lavorazione "SETUP LAMA"
Correzione raggio utensile	in lavorazioni di setup
Doppio pistone	in lavorazioni di setup
Senso di rotazione	in lavorazioni di tipo arco

Il funzionamento qui descritto è reso possibile da particolari tipologie dei parametri di appoggio. Lo schema fisso di assegnazione di un parametro di questo tipo è:

- ◇ Tipologia di controllo: (Integer);
- ◇ **Tipologia Albatros**: in range [8450 – 8499]
- ◇ **Password di visualizzazione**: Builder/Integrator
- ◇ Password di modifica: Operator
- ◇ Nome matrice: non assegnata
- ◇ **Info aggiunte**: **tipologia Albatros del parametro per il quale funziona da appoggio** (e che deve essere assegnato per la lavorazione)
- ◇ Modo di visualizzazione: View & Modify;
- ◇ Valore di default: non assegnata.

L'utilizzo dei parametri di appoggio è motivato per assegnazione di controlli di caselle: radio, check, liste, combo.

È importante sottolineare che:

- la visibilità dei parametri di appoggio è determinata dalla Password di visualizzazione (**Builder/Integrator**) ma non dalla tipologia del programma: sono visibili anche in caso di programma normale, purché sia attivato il livello di password richiesto;
- la assegnazione dei parametri per i quali fungono da appoggio avviene però **solo** in fase di sviluppo di macro-programma (o di sottociclo) in seguito a richiamo da codice complesso.

4. LAVORAZIONI RISERVATE AI MACROPROGRAMMI

4.1. Istruzioni logiche

Rientrano in questo gruppo:

- esclusioni logiche (codici: IF, ENDIF): utilizzabili anche in programma normale;
- cicli ripetitivi (codici: FOR, FOREVER, ENDFOR);
- istruzioni di controllo (codici: BREAK, CONTINUE).

I codici operativi qui esaminati non corrispondono a nessuna lavorazione esecutiva (non arrivano in matrice pezzo): è quindi più corretto indicarle come istruzioni, anche se sono assegnate nel file EDICADWR.DEF come ogni altra lavorazione.

⋘ ATTENZIONE: tutte le condizioni logiche poste in testo di un macro-programma sono valutate solo in applicazione del macro-programma, tramite codice di macro.

4.1.1. IF / ENDIF

Le istruzioni sono utilizzabili per ogni tipologia di programma.

In finestra dell'istruzione IF sono indicate tre condizioni logiche di validità, condizionate tra loro in OR oppure in AND:

- se il risultato del test risulta VERO: determina la esecuzione su una o più lavorazioni specificate a valle dell'IF;
- se il risultato del test risulta FALSO: esclude l'esecuzione delle lavorazioni interessate.

In figura è riportato un esempio di impostazione di istruzione IF su tre termini:

In termini informatici può essere tradotta come:

```
IF ( ((r1 < 1) or (r2 < r5/2)) and (lf < r5+r7) ) THEN
...
ENDIF
```

In termini discorsivi può essere tradotta come:

- ◇ se (r1 minore di 1) oppure (r2 minore di r5/2)
 - ◇ ed in aggiunta è (lf minore di (r5+57))
- allora esegue le lavorazioni che seguono, fino alla istruzione ENDIF di chiusura.

È anche assegnato un commento ("caso particolare"): è un modo di documentare l'istruzione. L'interpretazione dell'istruzione IF è riportata nel manuale di *Programmazione Lavorazioni*.

L'istruzione ENDIF delimita le lavorazioni condizionate dall'IF.

La chiusura di un IF con istruzione ENDIF è obbligatoria, a meno che non sia richiesto un "IF aperto". In questo caso l'istruzione IF condiziona la sola lavorazione che segue, che deve essere assegnata del tipo:

- ◇ puntuale;
- ◇ complesso (codice di macro oppure di chiamata sottoprogramma): è il caso di IF utilizzato in testo di programma;
- ◇ logica, ma differente da codice IF- ENDIF o di ciclo (FOR, FOREVER, ENDFOR) o punto di applicazione;
- ◇ setup: *solo in testo di IF utilizzato in testo di macro-programma*;
- ◇ logica con codice assegnato in intervallo custom (1- 1000).

Se la lavorazione che segue un IF aperto non è valida: determina una segnalazione di errore in fase di applicazione delle condizioni logiche.

In testo di macro-programma le istruzioni (IF,..., ENDIF) possono essere inserite in un profilo. Esempio:

```
setup
if
    linea
endif
..
```

In testo di macro-programma: condizioni "IF-ENDIF" possono essere innestate in cicli "FOR-ENDFOR" oppure possono condizionare l'esecuzione di cicli "FOR-ENDFOR". Esempio:

```

IF (...)
...
  FOR (..)
    ..
    IF (...)
      ...
    ENDIF
  ...
  FOREVER
  ...
ENDIF

```

NOTA: Condizioni di confronto tra numeri (uguaglianza,...) sono sempre effettuate con utilizzo di un *epsilon*=0.001: valori che si scostano per meno di *epsilon* sono assunti uguali.

4.1.2. FOR – ENDFOR / FOREVER - ENDFOR

L'istruzione FOR permette di ripetere una o più lavorazioni e/o istruzioni per un determinato numero di volte. Il corpo dell'istruzione FOR è definito dall'istruzione di chiusura (obbligatorio) ENDFOR e è eseguito per zero o più volte, fino al raggiungimento dello stato FALSO di una determinata condizione.

L'istruzione FOR è assegnata con indicazione di tre termini o espressioni:

espressione iniziale	assegna un valore iniziale ad una variabile di tipo \$ (obbligatoria)
espressione di condizione	indica la condizione che permette di proseguire nel ciclo for (obbligatoria)
espressione di loop	esegue una operazione di assegnazione ad una variabile di tipo \$, ripetuta ad ogni ripetizione del ciclo.

In figura è riportato un esempio di impostazione classica di una istruzione FOR.

- ◇ l'iteratore \$0 viene assegnato al valore iniziale r1;
- ◇ il ciclo viene ripetuto fino a quando il valore dell'iteratore \$0 rimane inferiore (<) al valore di r2. Il test di confronto è eseguito anche per la assegnazione iniziale (qui: \$0=r1): il numero di ripetizioni eseguite può quindi essere nullo se l'assegnazione iniziale non verifica il confronto;
- ◇ ad ogni ripetizione successiva alla prima, il valore dell'iteratore \$0 viene incrementato di 1.

L'utilizzo di variabili di tipo \$nn (nn di valore da 0 e 99) è riservata ai testi di macro-programmi: un iteratore di ciclo FOR costituisce senz'altro l'utilizzo primario di questo tipo di variabili. Si tratta di variabili numeriche (con virgola) e tutte partono con assegnato valore iniziale 0.

L'istruzione FOREVER apre un ciclo for senza assegnazione di alcuna espressione: in questo caso l'uscita dal ciclo deve essere gestita con istruzione/i di BREAK (vedi oltre).

Le istruzioni FOR, FOREVER ed ENDFOR possono essere inserite in un profilo. Esempio:

```

setup
FOR
  linea
ENDFOR

```

Cicli "FOR - ENDFOR" possono essere innestati in cicli "IF-ENDIF" come possono includere l'esecuzione di cicli "IF-ENDIF".

4.1.3. BREAK

In finestra dell'istruzione BREAK sono indicate tre condizioni logiche di validità, condizionate tra loro in OR oppure in AND (vedi: istruzione IF):

- se il risultato del test risulta VERO: determina l'uscita diretta dal ciclo FOR al livello di innesto più vicino. Nel caso in cui l'istruzione è eseguita fuori da un ciclo FOR, il BREAK comporta l'uscita dallo sviluppo della macro;
- se il risultato del test risulta FALSO: determina il normale proseguimento nello sviluppo della macro.

Il risultato del test è comunque VERO se non è posta nessuna condizione logica.

L'istruzione BREAK gestisce anche un parametro numerico di *Messaggio*, che funge da codice di errore. Nel caso in cui il risultato del test assegnato per il BREAK risulti VERO e se il campo *Messaggio* ha assegnato un valore strettamente positivo (maggiore di 0), determina:

- l'uscita dallo sviluppo della macro, qualunque sia il livello di innesto del BREAK;
- l'azzeramento della lista di lavorazioni sviluppate in interpretazione del macro-programma;
- una segnalazione di errore, con messaggio individuato dal codice di errore assegnato al campo *Messaggio*.

L'istruzione può quindi gestire efficacemente:

- controlli di validità di parametri assegnati in utilizzo di macro: assegnando un codice di errore al campo *Messaggio*;
- l'uscita condizionata da cicli FOR: lasciando il campo *Messaggio* non assegnato (o assegnato con valore 0). Se di ciclo è aperto con istruzione FOREVER, l'istruzione BREAK rappresenta l'unico metodo di chiusura del ciclo.

L'istruzione BREAK può essere inserita in un profilo. Esempio:

```

setup
FOR ($0=0; $0<=r12; $0=$0+1)
  Linea
  BREAK ( $0=r5 or $0=r6)  ← se le condizioni sono verificate, salta all'istruzione successiva all'ENDFOR
  linea
ENDFOR
...

```

Assegnazione del campo *Messaggio* (numero di errore)

Si è definito di impostare il campo *Messaggio* con valori:

- [1 - 200] per gestire errori di base;
- [201 -1000] per gestire errori custom.

Nel caso di valore in intervallo [1-200]: il messaggio di errore viene cercato in file di messaggi di EDICAD.EXE (file: EDICAD.LNG), in corrispondenza a segnalazioni di errore predefinite ed utilizzate ad esempio nei programmi di macro installati con il database di base (macro-programmi registrati in "cadcfg\mcr").

Nel caso di valore in intervallo [201-1000]: il messaggio di errore viene cercato in file di messaggi "cadcfg\CSTBREAK.LNG" (se il file non esiste: visualizza un generico messaggio letto da file EDICAD.LNG).

È possibile gestire il file CSTBREAK.LNG con l'applicazione WINMESS.EXE (operando sempre a password costruttore).

Se viene impostato un valore maggiore di 1000: l'istruzione esce comunque con situazione di errore visualizzando il messaggio corrispondente al numero di errore 1000 (vedi tabella seguente).

Segue un prospetto complessivo dei codici di errore disponibili:

<i>errore impostato</i>	<i>errore visualizzato versione 2.0</i>	<i>messaggio</i>
1	1001	Passo x nullo
2	1002	Passo y nullo
3	1003	Numero battute nullo
4	1004	Troppe battute
5	1005	Raggio nullo
6	1006	Vettore nullo
7	1007	Raggio non valido
8	1008	Angolo non valido
9	1009	Numero campionamenti nullo
10	1010	Troppi campionamenti
11	1011	Campionamenti insufficienti
12	1012	Ingombri figura non validi
13	1013	Asse/ ingombro x non valido
14	1014	Asse/ ingombro y non valido
15	1015	Assi/ ingombri non validi se uguali
16	1016	Numero lati nullo
17	1017	Troppi lati richiesti
18	1018	Lati insufficienti
19	1019	Geometria non valida
20	1020	Funzione selezionata non valida
21	1021	Passo nullo
...		
200	1200	

201	1201	da assegnare in file "cadcfg\CSTBREAK.LNG"
...		
1000	2000	da assegnare in file "cadcfg\CSTBREAK.LNG"

È anche possibile assegnare direttamente un messaggio di errore: se il campo *commento* ha un'impostazione che inizia con carattere di doppi apici " " è visualizzato direttamente il messaggio come assegnato.

Attenzione: sempre se è assegnato un numero di errore.



In figura è riportato un esempio di valutazione di una condizione di errore (ad un solo termine):

se è ($r5 < r6/2$), lo sviluppo del codice di macro è annullato con segnalazione di numero di errore 1001 e messaggio direttamente impostato in campo *commento*.

4.1.4. ERROR

L'istruzione è utilizzabile anche in testo di sottoprogramma o sotto-ciclo.

Anche per l'istruzione ERROR le condizioni logiche poste sono valutate solo in applicazione del macro-programma o del sottoprogramma tramite codice complesso (codice di macro o di applicazione di sottoprogramma).

L'istruzione è analoga al BREAK: solo non ha alcun effetto per l'esecuzione dei cicli FOR.

Nel caso in cui il risultato del test assegnato per l'istruzione ERROR risulta VERO e se il campo *Messaggio* ha assegnato un valore strettamente positivo (maggiore di 0), determina:

- l'uscita dallo sviluppo della macro o del sottoprogramma;
- l'azzeramento della lista di lavorazioni sviluppate;
- una segnalazione di errore, con messaggio individuato dal codice di errore assegnato al campo *Messaggio* (o direttamente in campo *commento*).

In testo di programma di tipologia non di macro ed in funzionamento non attivo di abilitazione "*Riduce compilazione di sottoprogramma*": si raccomanda l'utilizzo di istruzioni ERROR in testa al programma, fuori da cicli IF.

4.1.5. CONTINUE

In finestra dell'istruzione CONTINUE sono indicate tre condizioni logiche di validità, condizionate tra loro in OR oppure in AND (vedi: istruzione IF):

- se il risultato del test risulta VERO: determina il ritorno diretto all'inizio del ciclo FOR al livello di innesto più vicino. Se l'istruzione è eseguita fuori da un ciclo FOR, il CONTINUE comporta il ritorno all'inizio della macro;
- se il risultato del test risulta FALSO: determina il normale proseguimento nello sviluppo della macro.

Il risultato del test è comunque VERO se non è posta nessuna condizione logica.

L'istruzione CONTINUE può essere inserita in un profilo. Esempio:

```

FOR (..)
  setup
  FOR ($0=0; $0<=r12; $0=$0+1)  ←──────────────────────────────────────────|
    Linea
    CONTINUE ($0=r5 or $0=r6) ← se le condizioni sono verificate, salta all'istruzione FOR
    linea
  ENDFOR
  ...
ENDFOR

```

4.2. Istruzioni di assegnazioni ausiliarie

Anche i codici operativi qui esaminati non corrispondono a nessuna lavorazione esecutiva (leggi: non arrivano in matrice pezzo). Si tratta di istruzioni:

- di tipologia logica,
- generalmente utilizzabili in testo di macro-programma, ma non solo.

Anche in questo caso, quando utilizzate in testo di un macro-programma, sono valutate e risolte solo in applicazione del macro-programma, tramite codice di macro.

4.2.1. Istruzione: Punto di applicazione

Permette di assegnare il punto di applicazione di un macro-programma oppure di un sottoprogramma: il punto di applicazione indica il riferimento della macro (o sottoprogramma).

In applicazione di macro o sottoprogramma viene comunque sempre gestito un punto di riferimento (di default), anche se non è direttamente programmato: si tratta della prima lavorazione in lista di tipologia puntuale, setup o complessa. È necessario programmare il punto di applicazione quando il riferimento è diverso da quello di default.

Il codice ha effetto solo in chiamata da codice complesso (codice di macro o di chiamata a sottoprogramma).

La programmazione è interpretata in coordinate assolute ed è valida per *tutte le tre coordinate* riportate: valore 0 viene comunque assunto per le coordinate non assegnate.

In caso di sottoprogramma ed in funzionamento non attivo di abilitazione "*Riduce compilazione di sottoprogramma*" è significativo assegnare un solo punto di applicazione: impostazioni successive alla prima sono ignorate.

In caso di macro-programma o di sottoprogramma in funzionamento attivo di abilitazione "*Riduce compilazione di sottoprogramma*": è significativa la prima assegnazione verificata dai condizionamenti logici.

Il codice viene cercato solo nelle *prime cinque linee* di programma e ciò per evitare di rallentare la fase di elaborazione, nella ricerca dell'istruzione. Una buona norma consiste nell'indicare il punto di applicazione nelle prime righe di testo e comunque esterno a cicli FOR.

4.2.2. Istruzioni di assegnazione delle variabili J

Le istruzioni sono utilizzabili per ogni tipologia di programma.

L'interpretazione delle istruzioni è riportata nel manuale di *Programmazione Lavorazioni*.

Sono gestite due istruzioni che assegnano una o tutte le variabili globali di tipo j (j0 – j99):

- *Assegna variabile J*: assegna una variabile di tipo j, identificata tramite l'istruzione;
- *Assegna set di variabili J*: assegna tutte le variabili di tipo j (j0 – j99).

Per entrambe le istruzioni sono indicate tre condizioni logiche di validità, condizionate tra loro in OR oppure in AND (vedi: istruzione IF):

- se il risultato del test risulta VERO: l'assegnazione è effettuata;
- se il risultato del test risulta FALSO: l'assegnazione non è effettuata.

Il risultato del test è comunque VERO se non è posta nessuna condizione logica.

Se utilizzate in testo di macro:

- le istruzioni **non** possono essere inserite in un profilo;
- sono valutate solo in chiamata da codice complesso (codice di macro).

Si sottolineano alcuni particolari aspetti di gestione, in caso di utilizzo delle istruzioni in testo di programma (o sottoprogramma) oppure di macro. Tale gestione è in parte del tutto singolare, rispetto alla interpretazione di altre istruzioni condizionate direttamente:

- in ogni caso l'assegnazione di variabile è eseguita solo se risulta verificato il condizionamento espresso nella istruzione stessa. Tale comportamento:
 - è del tutto in linea con la normale applicazione dei condizionamenti logici in testo di macro-programma;
 - mentre non rispecchia i normali criteri di sviluppo dei condizionamenti logici in testo di programma: un tale comportamento è peraltro richiesto dalla natura stessa delle due istruzioni di assegnazione.
- nel caso di testo di programma o di sviluppo di sottoprogramma in funzionamento non attivo di abilitazione "*Riduce compilazione di sottoprogramma*": l'assegnazione di variabile non è in alcun modo condizionato dall'innesto in cicli IF esterni: tale comportamento è del tutto in linea con la normale applicazione dei condizionamenti logici in testo di programma;
- nel caso di sviluppo di sottoprogramma in funzionamento attivo di abilitazione "*Riduce compilazione di sottoprogramma*": l'assegnazione di variabile è invece condizionato dall'innesto in cicli IF esterni. Avviene cioè solo se risultano verificati validi i condizionamenti esterni.

- nel caso di testo di macro-programma: l'assegnazione di variabile è invece condizionato dall'innesto in cicli IF esterni. Avviene cioè solo se risultano verificati validi i condizionamenti esterni: tale comportamento è del tutto in linea con la normale applicazione dei condizionamenti logici in testo di macro-programma.

4.2.3. Istruzioni di assegnazione delle variabili \$

L'assegnazione semplice di variabili di tipo \$ è proposta gestita con 4 differenti istruzioni. Si tratta però di una differenziazione solo formale, predisposta per fornire modalità di assegnazione con differenti personalizzazioni. Le condizioni logiche poste e le assegnazioni sono valutate solo in applicazione del macro-programma tramite codice complesso (codice di macro).

La figura propone una tra le modalità disponibili:

In tutti i casi è indicata una singola condizione logica di validità:

- se il risultato del test risulta VERO: le assegnazioni indicate sono effettuate;
 - se il risultato del test risulta FALSO: nessuna tra le assegnazioni indicate è effettuata.
- Il risultato del test è comunque VERO se non è posta nessuna condizione logica.

L'esempio riportato in figura traduce lo schema logico che segue:

```
IF (r1 =1) THEN
  $0=r5+lf/2
  $1=r5-lf/2
  $12=$10
ENDIF
```

Si raccomanda di indicare una sola assegnazione per ogni variabile richiesta.

È ad esempio da considerare in generale **non prevedibile** l'esito di impostazioni corrispondenti allo schema logico che segue:

```
IF (r1 =1) THEN
  $0=r5+lf/2
  $1=r5-lf/2
  $2=-$1           ← utilizza $1, che è assegnato nello stessa istruzione
  $0=$10          ← assegna $0, che è già assegnato nello stessa istruzione
ENDIF
```

Le istruzioni possono essere inserite in un profilo.

4.2.4. Istruzione di assegnazione ternaria delle variabili \$

L'istruzione gestisce l'assegnazione di variabili di tipo \$ in condizionamento *ternario*.

Per l'istruzione è indicata una singola condizione logica, il cui utilizzo è però ora di tipo particolare:

- se il risultato del test risulta VERO: le assegnazioni indicate sono effettuate considerando i termini posti alla sinistra del carattere ?;
- se il risultato del test risulta FALSO: le assegnazioni indicate sono effettuate considerando i termini posti alla destra del carattere ?.

Il risultato del test è comunque VERO se non è posta nessuna condizione logica.

L'esempio riportato in figura traduce lo schema logico che segue:

```

IF (r1 =1) THEN
    $0=r2
    $12=lf
ELSE
    $0=r3
    $12=hf
ENDIF

```

Si raccomanda di indicare una sola assegnazione per ogni variabile richiesta.

È ad esempio da considerare in generale **non prevedibile** l'esito di impostazioni corrispondenti allo schema logico che segue:

```

IF (r1 =1) THEN
    $0=r2
    $12=$0           ← utilizza $0, che è assegnato nello stessa istruzione
ELSE
    $0=r3
    $0=$0+hf        ← assegna $0, che è già assegnato nello stessa istruzione
ENDIF

```

L'istruzione può essere inserita in un profilo.

5. TIPOLOGIE DI PROGRAMMA

5.1. Sottociclo

Un sottociclo assegna un ciclo di lavorazioni definito secondo regole generalmente custom.

È quindi paragonabile ad un generico programma gestito da EDICAD: anche un programma assegna un ciclo di lavorazioni, definito secondo regole custom, ma ci sono differenze sostanziali:

- un programma generico viene scritto per essere direttamente eseguito;
- un sottociclo viene invece scritto per essere “applicato” in un programma, come una lavorazione elementare.

L'applicazione di un sottociclo è possibile con scelta tra i codice (complessi):

- generici di chiamata a sottoprogramma (SUB0, SUB1, SUB2);
- in intervallo [4001; 5000]: codici di sottoprogramma con chiamata diretta.

La seconda scelta è da preferire.

La tipologia di sottociclo dichiara quindi che il programma che si va a definire è da utilizzare solo come sottoprogramma.

Un sottociclo:

- non è eseguibile direttamente;
- è gestibile (creazione e/o modifica e/o salvataggio) a partire da password costruttore e la gestione è effettuata con programma EDICAD;
- è gestita la possibilità (consigliata) di mantenere un archivio separato: “cadcfg\sub”.

Dal punto di vista della programmazione:

- avviene come per un programma normale: con definizione di un pezzo ed applicazione di lavorazioni in una o più facce;
- sono utilizzabili le variabili r (da r0 ad r299);
- sono utilizzabili le lavorazioni elementare: anche quelle protette (esempio: “[95] SETUP LAMA”);
- sono utilizzabili le lavorazioni logiche normalmente disponibili in edit di programma;
- può effettuare assegnazioni di controlli di caselle mediante parametri di appoggio;
- **non** sono utilizzabili codici complessi di chiamata sottoprogramma;
- le applicazioni di macro sono solo su singola faccia (non gestisce chiamate indotte)

È possibile assegnare la tipologia di sottociclo in uno dei due modi indicati:

- con richiesta di nuovo file (a password costruttore) viene presentata la finestra di selezione della tipologia da assegnare al programma (vedi caso di macro-programma): scegliere ora la voce **SUB- Ciclo**;
- con aperto un programma di differente tipologia (programma o macro-programma): scegliere la voce “*Cambia tipologia di programma*” in menu di File (la voce è gestita solo a password costruttore). Anche in questo caso viene presentata una finestra di selezione della tipologia da assegnare al programma: scegliere la voce **SUB- Ciclo**.

Attenzione: il cambiamento di tipologia viene effettuato solo se possibile. Cioè:

- non sono utilizzati codici non assegnati nel database delle lavorazioni;
- non sono utilizzati codici complessi di chiamate a sottoprogramma;
- non sono utilizzati codici di uso riservato ai macro-programmi (FOR, ...)- (possibile se è richiesta modifica di tipologia di macro);
- non sono assegnati profili interrotti da istruzioni logiche (possibile se è richiesta modifica di tipologia di macro);
- non sono assegnate lavorazioni in facce di pezzo assegnate non gestite in testo di programma (possibile se è richiesta modifica di tipologia di macro).

La gestione della tipologia di *sottociclo* consente ad un costruttore di proteggere dei sottoprogrammi considerati di base per una applicazione. L'utente finale del controllo opera a password **Operator**, per cui non può aprire e/o modificare un sottociclo, mentre potrebbe aprire e/o modificare un normale sottoprogramma, con la possibilità di comprometterne il funzionamento.

Per l'interpretazione e lo sviluppo di un sottociclo valgono le considerazioni generali poste per il caso di sottoprogramma generico.

5.2. Sottoprogramma

L'utilizzo di un sottoprogramma è generalmente motivato dalla necessità di eseguire più volte una stessa sequenza di lavorazioni, nello stesso programma o in programmi diversi.

Questa sequenza di lavorazioni può essere:

- completamente definita, senza necessità di parametrizzazioni;
- definita in base a dei settaggi modificabili (se risultano configurate delle variabili riassegnabili).

Un sottoprogramma:

- è eseguibile direttamente: non è assegnata alcuna tipologia particolare per qualificare un sottoprogramma;
- è gestibile (creazione e/o modifica e/o salvataggio) a qualsiasi livello di password;
- è gestito un archivio dedicato, a partire da cartella: "dirprod\sub".

La registrazione in archivio dedicato non è obbligatoria, ma consigliata: è il solo modo per garantire la *trasportabilità* di un archivio di programmi.

Il sottociclo è un esempio di sottoprogramma, ma con la **dichiarazione** che si intende utilizzare il programma solo come sottoprogramma.

L'applicazione di un sottoprogramma è possibile con scelta tra i codici (complessi):

- generici di chiamata a sottoprogramma (SUB0, SUB1, SUB2);
- in intervallo [4001; 5000]: codici di sottoprogramma con chiamata diretta.

I codici generici di applicazione di un sottoprogramma sono documentati nel manuale di *Programmazione Lavorazioni* e costituiscono la modalità a disposizione dell'utilizzatore finale del controllo:

- l'utilizzatore stesso scrive ed archivia i suoi sottoprogrammi;
- li utilizza in edit di programmi mediante uno tra i codici generici disponibili (SUB0, SUB1, SUB2).

Il funzionamento proposto dai codici generici SUBn prevede che l'utilizzatore assegni direttamente:

- il sottoprogramma da applicare (in controllo di edit oppure con apertura di finestra di *File Apri*);
- la faccia del sottoprogramma che deve essere applicata;
- il valore delle variabili riassegnabili;
- le assegnazioni relative a trasformate geometriche (traslazione, rotazione, speculare, aggancio di punto, inversione..) ed eventuali ripetizioni (libera, a tabella).

È peraltro possibile assegnare dei codici dedicati di applicazione di un determinato sottoprogramma (si consiglia, in questo caso, di assegnare tipologia di sottociclo), con modalità analoghe a quanto già esaminato per l'assegnazione dei codici di chiamata macro. Il paragrafo che segue esamina la questione.

5.3. Come configurare un codice di chiamata sottoprogramma

Si tratta di una procedura del tutto simile a quanto già esaminato per l'assegnazione dei codici di chiamata macro, solo:

il Codice operativo deve essere assegnato nell'intervallo [4501-5000]

È possibile inserire la nuova lavorazione copiandola da un codice di macro di assegnazione similare (ad esempio in EDICADWR.DEF).

L'assegnazione dei parametri presenta alcune differenze:

- parametro di assegnazione sottoprogramma. Tipologia: [8098]
 - ◇ Tipologia di controllo: *Edit*;
 - ◇ Modo di visualizzazione: nascosto (hide);
 - ◇ Valore di default = path di lettura sottoprogramma. Esempi:

1) se il sottoprogramma (o sotto-ciclo) è: "..\product\sub\pippo.txt "	impostare "pippo.txt";
2) se il sottoprogramma (o sotto-ciclo) è: "..\product\sub\ante\pippo.txt "	impostare "ante\pippo.txt";
3) se il sottoprogramma (o sotto-ciclo) è: "..\product\ante\pippo.txt "	impostare "..\ante\pippo.txt".
4) se il sottoprogramma (o sotto-ciclo) è: "..\cadcfg\sub\pippo.txt "	impostare "**pippo.txt";
5) se il sottoprogramma (o sotto-ciclo) è: "..\cadcfg\sub\ante\pippo.txt "	impostare "**ante\pippo.txt";

cioè :

- come condizione di default un sottoprogramma è cercato a partire dalla cartella "..\product\sub" (casi 1) e 2);
- è possibile richiamare un sottoprogramma archiviato nella cartella dei programmi (o in una cartella figlia della cartella dei programmi): è il caso 3);
- è gestita anche una seconda cartella privilegiata: "..\cadcfg\sub" (è la cartella in cui si consiglia di archiviare i sottocicli). In questo caso la stringa deve iniziare con il carattere * (asterisco) – (casi 4) e 5).

Appare evidente come in tutti i casi la indicazione del sottoprogramma è effettuata in modo relativo (rispetto ai path notevoli di Albatros): ciò rende possibile copiare un archivio di programmi (e sottoprogrammi) in una differente installazione di Albatros (su un altro computer, o su un altro disco, o semplicemente in una differente configurazione di Albatros) e tutto continua a funzionare correttamente. Cioè: i programmi trovano correttamente i sottoprogrammi, i sotto-cicli, i macro-programmi.

(si veda anche il manuale di *Programmazione Lavorazioni*, capitolo di programmazione parametrica).

Il parametro [8098] è obbligatorio: dice quale sottoprogramma viene utilizzato per assegnare il codice.

- parametro di selezione della faccia del sottoprogramma. Tipologia: [6]
 - ◊ Tipologia di controllo: (*Integer*);
 - ◊ Modo di visualizzazione: nascosto (hide);
 - ◊ Valore di default = numero di faccia del sottoprogramma (se è richiesta applicazione di chiamate indotte: =-1 oppure **0** oppure **non assegnare il parametro**);
- È possibile assegnare tutti o in parte i parametri di trasformate di applicazione che sono gestiti per i codici SUBn (quote di piazzamento, aggancio punto, angolo di rotazione, speculare, inversione, ripetizioni libere o in tabella,...): si tratta di valutare caso per caso cosa è opportuno assegnare;

➤ assegnare *tutti e solo* i parametri corrispondenti a variabili riassegnabili del sottoprogramma, con tipologie come esaminato in assegnazione di codice di macro.

5.4. Sviluppo di sottoprogramma

È opportuno esaminare le logiche di interpretazione che possono essere applicate in fase di applicazione di un sottoprogramma. Si tratta delle due abilitazioni proposte in finestra di *Abilitazioni*, da menu di *Settaggi*:

- riduce lista di sottoprogramma;
- riduce la compilazione di sottoprogramma (richiede che sia attiva la prima).

Riduce lista di sottoprogramma

- ❖ il sottoprogramma viene compilato per intero;
 - ❖ la lista di lavorazioni corrispondente al sottoprogramma viene poi ridotta alle sole lavorazioni esecutive leggi: verificate dai condizionamenti logici);
 - ❖ alla lista che ne risulta applica le trasformate indicate (traslazione, rotazione, speculare, ripetizioni, inversione,...).
- La abilitazione è eliminata a partire dalla versione 2.5.001 di EDICAD, ed assunta in modo fisso attiva.

Riduce la compilazione di sottoprogramma

- ❖ il sottoprogramma viene compilato valutando direttamente le condizioni logiche (IF,..): la lista prodotta ha solo le lavorazioni verificate esecutive;
- ❖ alla lista che ne risulta applica le trasformate indicate (traslazione, rotazione, speculare, ripetizioni, inversione,...).

Le logiche di applicazione di sottoprogramma sopra indicate sono state introdotte in modo condizionato per compatibilità con versioni precedenti: si consiglia di tenerle abilitate, a meno di grosse incompatibilità con installazioni precedenti.

È opportuno analizzare il significato delle logiche indicate con un semplice esempio.

Testo di sottoprogramma

```

1.   IF r1<1
2.       FORO] qx=r0; qy=100;
3.   ENDIF
4.   IF r1>=1
5.       FORO] qx=100; qy=r0;
6.   ENDIF
7.   IF r5>1
8.       FORO] qx=32; incrementale
9.       FORO] qx=32; incrementale
10.  ENDIF
11.  [FORO] qx=32; incrementale
12.  [FORO] qx=32; incrementale

```

il sottoprogramma è applicato con: r0=200, r1=1, r5=0.

I casi possibili sono riportati nella tabella:

Riduce lista sottoprogramma	Riduce compilazione sottoprogramma	Lista sviluppata	Linea originale	Stato di esecuzione
Off	Off	IF (falso) [FORO] qx=200; qy=100; ENDIF IF (vero) [FORO] qx=100; qy=200; ENDIF IF (falso) [FORO] qx=132;	1 2 3 4 5 6 7 8	OFF OFF OFF ON ON ON OFF OFF

		[FORO] qx=164; ENDIF	9 10	OFF OFF
		[FORO] qx=196; [FORO] qx=228;	11 12	ON ON
On	Off	[FORO] qx=100; qy=200; [FORO] qx=196; [FORO] qx=228;	5 11 12	ON ON ON
On	On	[FORO] qx=100; qy=200; [FORO] qx=132; [FORO] qx=164;	5 11 12	ON ON ON

È evidente come lo sviluppo possa cambiare anche drasticamente, nei differenti casi.

Si notino in particolare le posizioni calcolate per le forature che verranno effettivamente eseguite (ON in ultima colonna): nei primi due casi corrispondono, ma solo nell'ipotesi di non assegnare delle quote xy di piazzamento il sottoprogramma. Assegnando infatti una quota X di piazzamento = 500, le forature che risultano avere stato esecutivo ON sono traslate a:

	X del 1°foro	X del 2°foro	X del 3°foro
caso 1:	$100+300=400$	$196+300=496$	$228+300=528$
caso 2:	$100+400=500$	$196+400=596$	$228+400=628$

Nel primo caso il punto di applicazione del sottoprogramma rimane la linea [2] del testo originale, anche se non risulta eseguibile, mentre nel secondo caso il punto di applicazione corrisponde alla prima linea eseguibile.

È peraltro evidente come l'ultimo caso riportato in tabella rispecchi maggiormente lo sviluppo di un macro-programma.

6. IL LINGUAGGIO MLC

Ogni lavorazione di EDICAD può avere associato un testo ASCII di descrizione geometrica. Il linguaggio MLC è l'insieme di regole con cui la descrizione geometrica deve essere espressa per consentirne una particolare rappresentazione grafica a video.

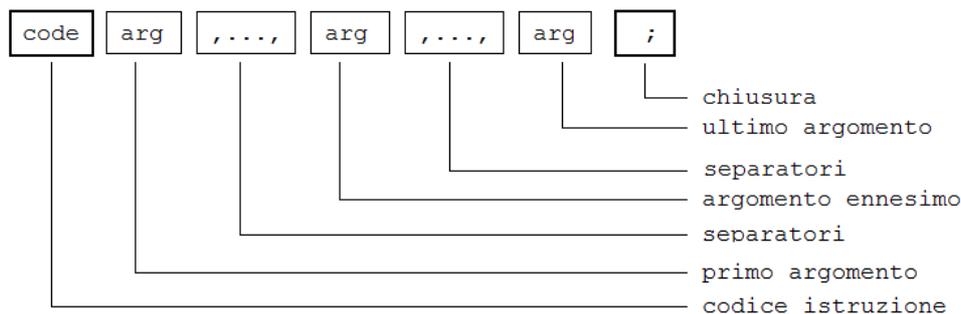
6.1. Elenco delle istruzioni

Funzionalità	Nome	Descrizione
Scelta di colore	PEN	imposta il colore della penna
	BRUSH	imposta il colore di fondo
Disegno di marcatore	MARK	disegna in modalità assoluta
	MARK_R	disegna in modalità incrementale
	MARK_D	disegna in modalità incrementale con valori di default
Disegno di linea	LINE	disegna nel piano dominante in modalità assoluta
	LINE_D	disegna nel piano dominante in modalità incrementale con valori di default
	LINE_R	disegna nel piano dominante in modalità incrementale
Disegno di arco	ARC	disegna nel piano dominante in modalità assoluta
	ARC_3D	disegna nello spazio in modalità assoluta
	ARC_3D_R	disegna nello spazio in modalità incrementale
	ARC_3P	disegna un arco per tre punti nello spazio in modalità assoluta
	ARC_3P_R	disegna per tre punti nello spazio in modalità incrementale
	ARC_D	disegna nel piano dominante in modalità incrementale con valori di default
	ARC_R	disegna nel piano dominante in modalità incrementale
	ARC_XZ	disegna nel piano XZ in modalità assoluta
	ARC_XZ_R	disegna nel piano XZ in modalità incrementale
	ARC_YZ	disegna nel piano YZ in modalità assoluta
ARC_YZ_R	disegna nel piano YZ in modalità incrementale	
Disegno di cerchio	CIRCLE	disegna nel piano dominante in modalità assoluta
	CIRCLE_R	disegna nel piano dominante in modalità incrementale
	CIRCLE_D	disegna in modalità incrementale con valori di default
Disegno di ellisse	ELLIPSE	disegna nel piano dominante in modalità assoluta
	ELLIPSE_R	disegna nel piano dominante in modalità incrementale
Disegno di rettangolo	RECT	disegna nel piano dominante in modalità assoluta
	RECT_R	disegna nel piano dominante in modalità incrementale
Condizionamenti	IF	attiva una condizione di abilitazione dei codici a seguire
	ELSE	disattiva una condizione di abilitazione ed attiva la sua negata
	ENDIF	disattiva una condizione di abilitazione

6.2. Convenzioni

costdef	costante predefinita (es. BLACK)
vardef	variabile predefinita (es. QXP)
costante	carattere, numero (intero, float)
stringa	sequenza di caratteri racchiusa tra doppi apici (es. "albatros") non deve eccedere i 50 caratteri
variabile	nome di un parametro (es. prqx)
espressione	sequenza di operazioni con operandi di tipo costante e/o variabile non deve eccedere i 50 caratteri
operatore condizionale	=, >, <, >=, <=, != operatori usati con il codice IF

6.3. Struttura di una generica istruzione MLC



6.4. Elenco delle parole chiave (costanti predefinite) e loro valore

BLACK	RGB(0,0,0)	colore nero
WHITE	RGB(255,255,255)	colore bianco
LIGHT_GRAY	RGB(192,192,192)	colore grigio luminoso
DARK_GRAY	RGB(128,128,128)	colore grigio scuro
GREEN	RGB(0,128,0)	colore verde
LIGHT_GREEN	RGB(0,255,0)	colore verde luminoso
BROWN	RGB(128,0,0)	colore marrone
RED	RGB(192,0,0)	colore rosso
LIGHT_RED	RGB(255,0,0)	colore rosso luminoso
BLUE	RGB(0,0,128)	colore blu
LIGHT_BLUE	RGB(0,0,255)	colore blu luminoso
MAGENTA	RGB(255,0,255)	colore magenta
YELLOW	RGB(128,128,0)	colore giallo
LIGHT_YELLOW	RGB(255,255,0)	colore giallo luminoso
CYAN	RGB(0,255,255)	colore azzurro
CCW	'A'	senso di rotazione antiorario
CW	'O'	senso di rotazione orario

6.5. Elenco delle parole chiave (variabili predefinite) e loro significato

QXP, QYP, QZP	quota x/y/z del punto di applicazione della lavorazione precedente
QX, QY, QZ	quota x/z del punto di applicazione della lavorazione
QI, QJ, QK	quota i/j/k del centro della lavorazione (solo lavorazioni di tipo arco)
RAD	raggio della lavorazione (solo lavorazioni di tipo arco)
WISE	senso di rotazione della lavorazione (solo lavorazioni di tipo arco)
QXn, QYn, QZn	quota x/y/z del punto di applicazione dell'ennesima lavorazione espansa
QIn, QJn, QKn	quota i/j/k del centro dell'ennesima lavorazione espansa (solo per tipo arco)
RADn	raggio dell'ennesima lavorazione espansa (solo per tipo arco)
WISEn	senso di rotazione dell'ennesima lavorazione espansa (solo per tipo arco)

6.6. Funzionalità: SCELTA DI COLORE

PEN	
Sintassi	PEN colore;
Parametri	Colore costante
Descrizione	Colore della penna con cui sono disegnati gli elementi geometrici che, all'interno della descrizione, seguono l'istruzione in oggetto. Il colori possibili sono elencati tra le costanti predefinite.

BRUSH	
Sintassi	BRUSH colore;

Parametri	Colore coste
Descrizione	Colore del fondo con cui sono disegnati gli elementi geometrici che, all'interno della descrizione, seguono l'istruzione in oggetto. Il colori possibili sono elencati tra le costanti predefinite.

6.7. Funzionalità: DISEGNO DI MARCATORE

MARK	
Sintassi	MARK quota1x, quota1x[, quota1z];
Parametri	quota1x, quota1y, quota1z vardef, costante, variabile o espressione
Descrizione	Disegna un marcatore a forma rettangolare delle dimensioni di due pixel. Le quote sono assolute rispetto all'origine della faccia sulla quale è applicata la lavorazione.

quota1x, quota1y, quota1z (facoltativo) quote x/y/z del centro del marcatore

MARK_R	
Sintassi	MARK_R quota1x, quota1x[, quota1z];
Parametri	quota1x, quota1y, quota1z vardef, costante, variabile o espressione
Descrizione	Disegna un marcatore a forma rettangolare delle dimensioni di due pixel. Le quote sono relative al punto di applicazione della lavorazione.

quota1x, quota1y, quota1z (facoltativo) quote x/y/z del centro del marcatore

MARK_D	
Sintassi	MARK_D;
Parametri	
Descrizione	Disegna un marcatore a forma rettangolare delle dimensioni di due pixel. Le quote sono ricavate direttamente da quelle della lavorazione. La lavorazione deve essere di tipo puntuale.

6.8. Funzionalità: DISEGNO DI LINEA

LINE	
Sintassi	LINE quota1x, quota1y[, quota1z], quota2x, quota2y [, quota2z];
Parametri	quota1x, quota1y, quota1z vardef, costante, variabile o espressione quota1x, quota1y, quota1z vardef, costante, variabile o espressione
Descrizione	Disegna una linea espressa con quote assolute rispetto all'origine della faccia sulla quale è applicata la lavorazione.

quota1x, quota1y, quota1z (facoltativo) quote x/y/z del punto di inizio linea
quota2x, quota2y, quota2z (facoltativo) quote x/y/z del punto di fine linea

LINE_D	
Sintassi	LINE_D;
Parametri	
Descrizione	Disegna una linea con quote di inizio e fine ricavate direttamente da quelle della lavorazione a cui l'istruzione è associata. La lavorazione deve essere di tipo lineare.

LINE_R	
Sintassi	LINE_R quota1x, quota1y[, quota1z], quota2x, quota2y [, quota2z];
Parametri	quota1x, quota1y, quota1z vardef, costante, variabile o espressione quota1x, quota1y, quota1z vardef, costante, variabile o espressione
Descrizione	Disegna una linea espressa con quote incrementali rispetto al punto di applicazione della lavorazione.
Esempio	La descrizione grafica di una lavorazione di inserimento potrebbe essere: "PEN GREEN; LINE_R 0,2*cnq,0,-2*cnq; LINE_R -2*cnq,0,2*cnq,0; CIRCLE_R 0,0,8*cnq;" si vede come per dimensioni fisse di grafica sia in generale necessario utilizzare il fattore <i>cnq</i> : questo ne garantisce il funzionamento corretto anche in caso di applicazione della lavorazione a programma in [inch]. In file EDICADWR.DEF: si veda la lavorazione di inserimento "[52] Bussola".

quota1x, quota1y, quota1z (facoltativo) quote x/y/z del punto di inizio linea
quota2x, quota2y, quota2z (facoltativo) quote x/y/z del punto di fine linea

6.9. Funzionalità: DISEGNO DI ARCO

ARC									
Sintassi	ARC quota1x, quota1x[, quota1z], quota2x,quota2y [, quota2z], quota3x,quota3y,dato;								
Parametri	<table border="0"> <tr> <td>quota1x,quota1y,quota1z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>quota2x,quota2y,quota2z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>quota3x,quota3y</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>dato</td> <td>costdef, vardef, costante, variabile o espressione</td> </tr> </table>	quota1x,quota1y,quota1z	vardef, costante, variabile o espressione	quota2x,quota2y,quota2z	vardef, costante, variabile o espressione	quota3x,quota3y	vardef, costante, variabile o espressione	dato	costdef, vardef, costante, variabile o espressione
quota1x,quota1y,quota1z	vardef, costante, variabile o espressione								
quota2x,quota2y,quota2z	vardef, costante, variabile o espressione								
quota3x,quota3y	vardef, costante, variabile o espressione								
dato	costdef, vardef, costante, variabile o espressione								
Descrizione	Disegna un arco espresso con quote assolute rispetto all'origine della faccia sulla quale è applicata la lavorazione.								
Esempio	La descrizione grafica di una lavorazione di arco singolo in piano xy (esempio: "[2101] A01") potrebbe essere: "PEN BLACK; ARC QXP,QYP,QZP,QX,QY,QZ,QI,QJ,WISE;"								

quota1x, quota1y, quota1z (facoltativo)	quote x/y/z del punto di inizio dell'arco
quota2x, quota2y, quota2z (facoltativo)	quote x/y/z del punto di fine dell'arco
quota3x, quota3y	quote x/y del punto del centro dell'arco
dato	senso di percorrenza dell'arco

ARC3D									
Sintassi	ARC3D quota1x, quota1z, quota1x, quota2x, quota2y, quota2z, quota3x, quota3y, quota3z, dato;								
Parametri	<table border="0"> <tr> <td>Quota1x,quota1y,quota1z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>Quota2x,quota2y,quota2z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>Quota3x,quota3y,quota3z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>dato</td> <td>costdef, vardef, costante, variabile o espressione</td> </tr> </table>	Quota1x,quota1y,quota1z	vardef, costante, variabile o espressione	Quota2x,quota2y,quota2z	vardef, costante, variabile o espressione	Quota3x,quota3y,quota3z	vardef, costante, variabile o espressione	dato	costdef, vardef, costante, variabile o espressione
Quota1x,quota1y,quota1z	vardef, costante, variabile o espressione								
Quota2x,quota2y,quota2z	vardef, costante, variabile o espressione								
Quota3x,quota3y,quota3z	vardef, costante, variabile o espressione								
dato	costdef, vardef, costante, variabile o espressione								
Descrizione	Disegna un arco nello spazio espresso con quote assolute rispetto all'origine della pezzo sulla quale è applicata la lavorazione.								

quota1x, quota1y, quota1z	quote x/y/z del punto di inizio dell'arco
quota2x, quota2y, quota2z	quote x/y/z del punto di fine dell'arco
quota3x, quota3y, quota3z	quote x/y/z del punto del centro dell'arco
dato	senso di percorrenza dell'arco.

I tre punti di descrizione dell'arco non devono essere allineati.

ARC3D_R									
Sintassi	ARC3D_R quota1x, quota1z, quota1x, quota2x, quota2y, quota2z, quota3x, quota3y, quota3z, dato;								
Parametri	<table border="0"> <tr> <td>quota1x,quota1y,quota1z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>quota2x,quota2y,quota2z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>quota3x,quota3y,quota3z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>dato</td> <td>costdef, vardef, costante, variabile o espressione</td> </tr> </table>	quota1x,quota1y,quota1z	vardef, costante, variabile o espressione	quota2x,quota2y,quota2z	vardef, costante, variabile o espressione	quota3x,quota3y,quota3z	vardef, costante, variabile o espressione	dato	costdef, vardef, costante, variabile o espressione
quota1x,quota1y,quota1z	vardef, costante, variabile o espressione								
quota2x,quota2y,quota2z	vardef, costante, variabile o espressione								
quota3x,quota3y,quota3z	vardef, costante, variabile o espressione								
dato	costdef, vardef, costante, variabile o espressione								
Descrizione	Disegna un arco nello spazio espresso con quote incrementali rispetto al punto di applicazione della lavorazione.								

quota1x, quota1y, quota1z	quote x/y/z del punto di inizio dell'arco
quota2x, quota2y, quota2z	quote x/y/z del punto di fine dell'arco
quota3x, quota3y, quota3z	quote x/y/z del punto del centro dell'arco
dato	senso di percorrenza dell'arco

I tre punti di descrizione dell'arco non devono essere allineati.

ARC3P									
Sintassi	ARC3P quota1x, quota1z, quota1x, quota2x, quota2y, quota2z, quota3x, quota3y, quota3z, dato;								
Parametri	<table border="0"> <tr> <td>Quota1x,quota1y,quota1z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>Quota2x,quota2y,quota2z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>Quota3x,quota3y,quota3z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>Dato</td> <td>costdef, vardef, costante, variabile o espressione</td> </tr> </table>	Quota1x,quota1y,quota1z	vardef, costante, variabile o espressione	Quota2x,quota2y,quota2z	vardef, costante, variabile o espressione	Quota3x,quota3y,quota3z	vardef, costante, variabile o espressione	Dato	costdef, vardef, costante, variabile o espressione
Quota1x,quota1y,quota1z	vardef, costante, variabile o espressione								
Quota2x,quota2y,quota2z	vardef, costante, variabile o espressione								
Quota3x,quota3y,quota3z	vardef, costante, variabile o espressione								
Dato	costdef, vardef, costante, variabile o espressione								
Descrizione	Disegna un arco nello spazio espresso con quote assolute rispetto all'origine della pezzo sulla quale è applicata la lavorazione.								

quota1x, quota1y, quota1z	quote x/y/z del punto di inizio dell'arco
quota2x, quota2y, quota2z	quota x/z di un punto intermedio all'arco
quota3x, quota3y, quota3z	quote x/y/z del punto di fine dell'arco
dato	senso di percorrenza dell'arco

I tre punti di descrizione dell'arco non devono essere allineati.

ARC3P_R									
Sintassi	ARC3P_R quota1x, quota1z, quota1x, quota2x, quota2y, quota2z, quota3x, quota3y, quota3z, dato;								
Parametri	<table> <tr> <td>quota1x,quota1y,quota1z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>quota2x,quota2y,quota2z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>quota3x,quota3y,quota3z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>dato</td> <td>costdef, vardef, costante, variabile o espressione</td> </tr> </table>	quota1x,quota1y,quota1z	vardef, costante, variabile o espressione	quota2x,quota2y,quota2z	vardef, costante, variabile o espressione	quota3x,quota3y,quota3z	vardef, costante, variabile o espressione	dato	costdef, vardef, costante, variabile o espressione
quota1x,quota1y,quota1z	vardef, costante, variabile o espressione								
quota2x,quota2y,quota2z	vardef, costante, variabile o espressione								
quota3x,quota3y,quota3z	vardef, costante, variabile o espressione								
dato	costdef, vardef, costante, variabile o espressione								
Descrizione	Disegna un arco nello spazio espresso con quote incrementali rispetto al punto di applicazione della lavorazione.								

quota1x, quota1y, quota1z quote x/y/z del punto di inizio dell'arco
 quota2x, quota2y, quota2z quota x/z di un punto intermedio all'arco
 quota3x, quota3y, quota3z quote x/y/z del punto di fine dell'arco
 dato senso di percorrenza dell'arco

I tre punti di descrizione dell'arco non devono essere allineati.

ARC_D	
Sintassi	ARC_D;
Parametri	
Descrizione	Disegna un arco le cui quote sono ricavate direttamente da quelle della lavorazione a cui la istruzione è associata. La lavorazione deve essere necessariamente di tipo arco.

ARC_R									
Sintassi	ARC_R quota1x, quota1x[, quota1z], quota2x,quota2y [, quota2z], quota3x,quota3y,dato;								
Parametri	<table> <tr> <td>quota1x,quota1y,quota1z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>quota2x,quota2y,quota2z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>quota3x,quota3y,</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>dato</td> <td>costdef, vardef, costante, variabile o espressione</td> </tr> </table>	quota1x,quota1y,quota1z	vardef, costante, variabile o espressione	quota2x,quota2y,quota2z	vardef, costante, variabile o espressione	quota3x,quota3y,	vardef, costante, variabile o espressione	dato	costdef, vardef, costante, variabile o espressione
quota1x,quota1y,quota1z	vardef, costante, variabile o espressione								
quota2x,quota2y,quota2z	vardef, costante, variabile o espressione								
quota3x,quota3y,	vardef, costante, variabile o espressione								
dato	costdef, vardef, costante, variabile o espressione								
Descrizione	Disegna un arco espresso con quote incrementali rispetto al punto di applicazione della lavorazione.								

quota1x, quota1y, quota1z (facoltativo) quote x/y/z del punto di inizio dell'arco
 quota2x, quota2y, quota2z (facoltativo) quote x/y/z del punto di fine dell'arco
 quota3x, quota3y quote x/y del punto del centro dell'arco
 dato senso di percorrenza dell'arco

ARC_XZ									
Sintassi	ARC_XZ quota1x, quota1z[, quota1y], quota2x,quota2z [, quota2y], quota3x,quota3z,dato;								
Parametri	<table> <tr> <td>quota1x,quota1z,quota1y</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>quota2x,quota2z,quota2y</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>quota3x,quota3z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>dato</td> <td>costdef, vardef, costante, variabile o espressione</td> </tr> </table>	quota1x,quota1z,quota1y	vardef, costante, variabile o espressione	quota2x,quota2z,quota2y	vardef, costante, variabile o espressione	quota3x,quota3z	vardef, costante, variabile o espressione	dato	costdef, vardef, costante, variabile o espressione
quota1x,quota1z,quota1y	vardef, costante, variabile o espressione								
quota2x,quota2z,quota2y	vardef, costante, variabile o espressione								
quota3x,quota3z	vardef, costante, variabile o espressione								
dato	costdef, vardef, costante, variabile o espressione								
Descrizione	Disegna un arco nel piano XZ espresso con quote assolute rispetto all'origine della faccia sulla quale è applicata la lavorazione.								

quota1x, quota1z, quota1y (facoltativo) quote x/z/y del punto di inizio dell'arco
 quota2x, quota2z, quota2y (facoltativo) quote x/z/y del punto di fine dell'arco
 quota3x, quota3z quote x/z del punto del centro dell'arco
 dato senso di percorrenza dell'arco

ARC_XZ_R									
Sintassi	ARC_XZ_R quota1x, quota1z[, quota1y], quota2x,quota2z [, quota2y], quota3x,quota3z,dato;								
Parametri	<table> <tr> <td>quota1x,quota1z,quota1y</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>quota2x,quota2z,quota2y</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>quota3x,quota3z</td> <td>vardef, costante, variabile o espressione</td> </tr> <tr> <td>dato</td> <td>costdef, vardef, costante, variabile o espressione</td> </tr> </table>	quota1x,quota1z,quota1y	vardef, costante, variabile o espressione	quota2x,quota2z,quota2y	vardef, costante, variabile o espressione	quota3x,quota3z	vardef, costante, variabile o espressione	dato	costdef, vardef, costante, variabile o espressione
quota1x,quota1z,quota1y	vardef, costante, variabile o espressione								
quota2x,quota2z,quota2y	vardef, costante, variabile o espressione								
quota3x,quota3z	vardef, costante, variabile o espressione								
dato	costdef, vardef, costante, variabile o espressione								
Descrizione	Disegna un arco nel piano XZ espresso con quote incrementali rispetto al punto di applicazione della lavorazione.								

quota1x, quota1z, quota1y (facoltativo) quote x/z/y del punto di inizio dell'arco
 quota2x, quota2z, quota2y (facoltativo) quote x/z/y del punto di fine dell'arco
 quota3x, quota3z quote x/z del punto del centro dell'arco
 dato senso di percorrenza dell'arco

ARC_YZ	
Sintassi	ARC_YZ quota1y, quota1z[, quota1x], quota2y,quota2z [, quota2x], quota3y,quota3z,dato;
Parametri	quota1y,quota1z,quota1x vardef, costante, variabile o espressione quota2y,quota2z,quota2x vardef, costante, variabile o espressione quota3y,quota3z vardef, costante, variabile o espressione dato costdef, vardef, costante, variabile o espressione
Descrizione	Disegna un arco nel piano YZ espresso con quote assolute rispetto all'origine della faccia sulla quale è applicata la lavorazione.

quota1y, quota1z,quota1x (facoltativo) quote y/z/x del punto di inizio dell'arco
 quota2y, quota2z, quota2x (facoltativo) quote y/z/x del punto di fine dell'arco
 quota3y,quota3z quote y/z del punto del centro dell'arco
 dato senso di percorrenza dell'arco

ARC_YZ_R	
Sintassi	ARC_YZ_R quota1y, quota1z[, quota1x], quota2y,quota2z [, quota2x], quota3y,quota3z,dato;
Parametri	quota1y,quota1z,quota1x vardef, costante, variabile o espressione quota2y,quota2z,quota2x vardef, costante, variabile o espressione quota3y,quota3z vardef, costante, variabile o espressione dato costdef, vardef, costante, variabile o espressione
Descrizione	Disegna un arco nel piano YZ espresso con quote incrementali rispetto al punto di applicazione della lavorazione.

quota1y, quota1z,quota1x (facoltativo) quote y/z/x del punto di inizio dell'arco
 quota2y, quota2z, quota2x (facoltativo) quote y/z/x del punto di fine dell'arco
 quota3y,quota3z quote y/z del punto del centro dell'arco
 dato senso di percorrenza dell'arco

6.10. Funzionalità: DISEGNO DI CERCHIO

CIRCLE	
Sintassi	CIRCLE quota1x, quota1y[, quota1z], raggio;
Parametri	Quota1x, quota1y, quota1y vardef, costante, variabile o espressione raggio vardef, costante, variabile
Descrizione	Disegna un cerchio pieno il cui centro è espresso con quote assolute rispetto all'origine della faccia sulla quale è applicata la lavorazione.

quota1x, quota1y,quota1z (facoltativo) quote x/y/z del centro
 raggio raggio del cerchio

CIRCLE_D	
Sintassi	CIRCLE_D;
Parametri	
Descrizione	Disegna un cerchio pieno il cui centro è ricavato direttamente dal punto di applicazione della lavorazione associata. Il raggio è fissato al valore 4.0. Se è la sola istruzione della lavorazione ne velocizza la rappresentazione grafica.

CIRCLE_R	
Sintassi	CIRCLE_R quota1x, quota1y[, quota1z], raggio;
Parametri	Quota1x, quota1y, quota1y vardef, costante, variabile o espressione raggio vardef, costante, variabile
Descrizione	Disegna un cerchio pieno il cui centro è espresso con quote incrementali rispetto al punto di applicazione della lavorazione.
Esempio	La descrizione grafica di una lavorazione di inserimento potrebbe essere: "PEN GREEN; LINE_R 0,2*cnq,0,-2*cnq; LINE_R -2*cnq,0,2*cnq,0; CIRCLE_R 0,0,12*cnq; CIRCLE_R -18*cnq,-12*cnq,4*cnq; CIRCLE_R 18*cnq,-12*cnq,4*cnq;" In file EDICADWR.DEF: si veda la lavorazione di inserimento "[53] Cerniera".

quota1x, quota1y,quota1z (facoltativo) quote x/y/z del centro
 raggio raggio del cerchio

6.11. Funzionalità: DISEGNO DI ELLISSE

ELLIPSE	
----------------	--

Sintassi	ELLIPSE quota1x, quota1x[, quota1z], quota2x,quota2y;
Parametri	quota1x,quota1y,quota1z vardef, costante, variabile o espressione quota2x,quota2y vardef, costante, variabile o espressione
Descrizione	Disegna una ellisse piena espressa con quote assolute rispetto all'origine della faccia sulla quale è applicata la lavorazione.
quota1x, quota1y quota1z (facoltativo) quota2x,quota2y	quote x/y minima dell'ellisse quota z dell'ellisse quote x/y massima dell'ellisse

ELLIPSE_R	
Sintassi	ELLIPSE_R quota1x, quota1x[, quota1z], quota2x,quota2y;
Parametri	quota1x,quota1y,quota1z vardef, costante, variabile o espressione quota2x,quota2y vardef, costante, variabile o espressione
Descrizione	Disegna una ellisse piena espressa con quote con quote incrementali rispetto al punto di applicazione della lavorazione.
quota1x, quota1y quota1z (facoltativo) quota2x,quota2y	quote x/y minima dell'ellisse quota z dell'ellisse quote x/y massima dell'ellisse

6.12. Funzionalità: DISEGNO DI RETTANGOLO

RECT	
Sintassi	RECT quota1x, quota1x[, quota1z], quota2x,quota2y;
Parametri	quota1x,quota1y,quota1z vardef, costante, variabile o espressione quota2x,quota2y vardef, costante, variabile o espressione
Descrizione	Disegna un rettangolo pieno espresso con quote assolute rispetto all'origine della faccia sulla quale è applicata la lavorazione.
quota1x, quota1y quota1z (facoltativo) quota2x, quota2y	quote x/y minima del rettangolo quota z dell'ellisse quote x/y massima del rettangolo

RECT_R	
Sintassi	RECT_R quota1x, quota1x[, quota1z], quota2x,quota2y;
Parametri	quota1x,quota1y,quota1z vardef, costante, variabile o espressione quota2x,quota2y vardef, costante, variabile o espressione
Descrizione	Disegna un rettangolo pieno espresso con quote incrementali rispetto al punto di applicazione della lavorazione.
Esempio	La descrizione grafica di un codice di macro potrebbe essere: "PEN GREEN;BRUSH WHITE; RECT_R -prvarw1 1/2,5,prvarw8-(prvarw11/2),prvarw11/2; BRUSH GREEN; RECT_R prvarw8-(prvarw1 1/2),5,prvarw6+prvarw8-(prvarw11/2),prvarw11/2;" ✓ vi sono riportati due istruzioni di disegno rettangolo, con cambio del colore del fondo; ✓ entrambi i rettangoli sono assegnati valutando parametri della lavorazione (nomi: <i>prvarw11</i> , <i>prvarw8</i> , <i>prvarw6</i>). È tipico che i parametri corrispondano a parametri di assegnazione delle variabili del macro-programma.
quota1x, quota1y quota1z (facoltativo) quota2x, quota2y	quote x/y minima del rettangolo quota z dell'ellisse quote x/y massima del rettangolo

6.13. Funzionalità: CONDIZIONAMENTI

IF	
Sintassi	IF par1,[operatore, par2];
Parametri	par1,par2 vardef, costante, variabile o espressione operatore operatore condizionale
Descrizione	Valuta, in base all'operatore condizionale, la relazione tra le espressioni par1 e par2. Nel caso di risultato vero abilita i codici a seguire sino ad incontrare i codici ELSE o ENDIF. Il parametro par2 è facoltativo. In questo caso l'istruzione IF valuta la validità dell'espressione par1. Non sono previsti codici IF tra loro innestati: un codice IF non può seguire un altro codice IF non chiuso.
Esempio	In file EDICADWR.DEF si veda ad esempio la lavorazione di inserimento "[57] Guida cassetto". Di seguito è riportato un estratto della rappresentazione grafica:

	<pre>"PEN LIGHT_YELLOW; IF chkdxx=,1;ARC_R 25*cnq,-25*cnq,25*cnq,25*cnq,25*cnq,0,0; ENDIF;IF chkdxx=,0;ARC_R -25*cnq,25*cnq,-25*cnq,-25*cnq,-25*cnq,0,0; ENDIF;LINE_R - 25*cnq,-25*cnq,-25*cnq,325*cnq; PEN LIGHT_BLUE; LINE_R 0,0,10*cnq,0; "</pre> <ul style="list-style-type: none"> ✓ vi sono riportati due cicli IF distinti (If; ... endif;) ✓ ogni ciclo IF testa il valore di un parametro della lavorazione, indicato per nome: <i>chkdxx</i>.
--	---

ELSE	
Sintassi	ELSE;
Descrizione	Segue ad istruzione IF e abilita, nel caso non sia verificata la condizione espressa nell'IF, i codici a seguire sino ad incontrare il codice ENDIF.

ENDIF	
Sintassi	ENDIF;
Descrizione	Chiude una istruzione IF o una istruzione ELSE.

7. OPTIMIZE

7.1. Comunicazione con Ottimizzatore custom

Il programma OPTIMIZE.EXE comunica con l'ottimizzatore custom tramite files scritti su disco.

7.1.1. File di matrice pezzo: MXPZIN.BIN

Il file ha formato binario ed è registrato in direttorio assegnato in file TPA.INI dalla variabile dirreport.

Se è attivo il flag *DebugCust*: registra anche la versione in formato testo in file MXPEZZO.DBG (vedi paragrafo precedente).

L'ottimizzatore custom scrive la matrice pezzo rielaborata in file in formato binario MXPZOUT.BIN (registrato in direttorio assegnato in file TPA.INI dalla variabile dirreport).

La tabella che segue riporta la struttura del file MXPZIN.BIN, organizzata su alcune strutture di intestazione, seguite dalla assegnazione della matrice pezzo:

Nome struttura	Campo struttura	Significato campo
IntMxPezzo	DWORD m_prog	viene aggiornato quando si cambia una delle strutture di comunicazione con l'ottimizzatore custom ◇ = 0: versione compatibile di ottimizzatore fino a 2.2.003 compresa ◇ = 1: versione compatibile di ottimizzatore da 2.2.004
	short m_nullo[2]	non utilizzato
	LONG m_inullo[10]	◇ m_inullo[0] = numero totale di macchine configurate in EDICAD.EXE; ◇ m_inullo[1] = maschera delle macchine abilitate, tra le 8 gestibili (corrisponde alla assegnazione in EDICAD.EXE)
CommInputOptimPacket	BOOL bCommOk	non utilizzato
	int nError	Viene scritto dall'ottimizzatore custom, in base all'esito dell'ottimizzazione: ◇ se l'ottimizzazione è andata a buon fine è assegnato il valore GOOD (=0); ◇ se si è verificato un errore assume valore a seconda dell'errore occorso: ERRORE (-1) errore generico ERR_NOTFILE (-2) errore in creazione o scrittura file ERR_NOTMEMORY (-3) errore in allocazione di memoria ERR_NOTELABOTT (-4) errore in elaborazione parametri;
	char szPathFileName[_MAX_PATH]	Valido in ingresso all'ottimizzatore custom: pathname del file da ottimizzare
	float dimx, dimy, dimz	Valido in ingresso all'ottimizzatore custom: dimensioni del pannello.
	char szVarsList[4000]	Valido in ingresso all'ottimizzatore custom: lista delle variabili riassegnate.
	int nTipoStore	Valido in ingresso all'ottimizzatore custom: ✓ = 0: la matrice pezzo è scritta su file testo (estensione TXN se il programma è eseguito in normale, estensione TXM se in speculare). ◇ OPTIMIZE.EXE: scrive il file binario (MXPZIN.BIN); ◇ l'ottimizzatore custom legge il file, attiva le proprie procedure di ottimizzazione e registra la matrice pezzo modificata in file binario MXPZOUT.BIN; ◇ OPTIMIZE.EXE: legge MXPZOUT.BIN e registra il file testo (estensione TXN o TXM); ✓ = 1: la matrice pezzo deve essere memorizzata in un'area di memoria globale. ◇ OPTIMIZE.EXE: scrive il file binario (MXPZIN.BIN); ◇ l'ottimizzatore custom legge il file, attiva le proprie procedure di ottimizzazione e registra la matrice pezzo modificata in file binario MXPZOUT.BIN; ◇ OPTIMIZE.EXE: legge MXPZOUT.BIN e carica la matrice pezzo in memoria globale; ✓ = 2: la matrice pezzo non è memorizzata e l'ottimizzazione deve solo verificare che le lavorazioni siano effettivamente eseguibili (non vengono attivate procedure di match utensili o di ordinamento). ◇ OPTIMIZE.EXE: scrive il file binario (MXPZIN.BIN);

	<ul style="list-style-type: none"> ◇ l'ottimizzatore custom legge il file, attiva le proprie procedure di ottimizzazione e ritorna un'eventuale diagnostica di errore; ◇ OPTIMIZE.EXE: restituisce al programma chiamante l'esito dell'ottimizzazione.
UINT NumColonneMat	numero di colonne della matrice pezzo. Viene scritto da OPTIMIZE.EXE e riassegnato in uscita dall'ottimizzatore custom.
UINT NumRigheMat	numero righe della matrice pezzo. Viene scritto da OPTIMIZE.EXE e riassegnato in uscita dall'ottimizzatore custom.
HGLOBAL hGlobalMatMemory	non utilizzato nella comunicazione con l'ottimizzatore custom
struct MsgErrore	La struttura <i>MsgErrore</i> è assegnata dall'ottimizzatore custom solo se si verifica un errore di elaborazione dati. In questo caso il campo <i>nError</i> vale -4 (ERR_NOTELABOTT). Struttura di segnalazione errore: <ul style="list-style-type: none"> ✓ <i>MsgErrore.NumSide</i> : numero identificativo della faccia ✓ <i>MsgErrore.RigaLav[4]</i> : non utilizzato ✓ <i>MsgErrore.CodErr</i> : codice di errore ✓ <i>MsgErrore.StrErr</i> : messaggio di errore da visualizzare;
struct OptiRun	Struttura valida in ingresso all'ottimizzatore custom: assegna la modalità di esecuzione del pezzo. <ul style="list-style-type: none"> ✓ <i>OptiRun.n_exetype</i> <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ modalità esecuzione del pezzo: 0 = normale, 1 = mirror X, 2 = mirror Y, 3 = mirror XY; ✓ <i>OptiRun.f_posx, OptiRun.f_posy, OptiRun.f_posz</i> <ul style="list-style-type: none"> ◇ Tipologia [float]; ◇ posizione (X,Y,Z) della battuta; ✓ <i>OptiRun.libint[10]</i> <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ libint[0] = parametro di assegnazione dell'area di lavoro (parte bassa) e delle esclusioni (parte alta). Il parametro è assegnato dalla plancia, ma non è gestito da OPTIMIZE.EXE; ◇ libint[1] = 1 abilita la lettura del parametro libfloat[1] (offset X) ◇ libint[2] = 1 abilita la lettura del parametro libfloat[2] (offset Y) ◇ libint[3] = 1 abilita la lettura del parametro libfloat[3] (offset Z) ◇ libint[4] = 1 abilita il flag di terminazione carico; ◇ libint[5-9]: non utilizzati ✓ <i>OptiRun.libfloat[10]</i> <ul style="list-style-type: none"> ◇ tipologia [float]; ◇ libfloat[0]: non utilizzato ◇ libfloat[1] = offset X ◇ libfloat[2] = offset Y ◇ libfloat[3] = offset Z ◇ libfloat[4-9]: non utilizzati
OptiFlag	Struttura valida in ingresso all'ottimizzatore custom: assegna i flag di ottimizzazione del programma. <ul style="list-style-type: none"> ✓ <i>OptiFlag.bopti_or dx</i> <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ corrisponde al settaggio: <i>Ordinamento x</i>; ✓ <i>OptiFlag.bopti_or dy</i> <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ corrisponde al settaggio: <i>Ordinamento y</i>; ✓ <i>OptiFlag.bopti_road</i> <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ corrisponde al settaggio: <i>Ordinamento decrescente</i>; ✓ <i>OptiFlag.bopti_type0</i> <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ corrisponde al settaggio: <i>Ottimizza il tempo di esecuzione</i>; ✓ <i>OptiFlag.bopti_type1</i> <ul style="list-style-type: none"> ◇ tipologia [int];

	<ul style="list-style-type: none"> ◇ corrisponde al settaggio: <i>Ottimizzazione minima</i>; ✓ OptiFlag.bopti_type2 <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ corrisponde al settaggio: <i>Gestione codici speciali</i>; ✓ OptiFlag.bopti_type3 <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ corrisponde al settaggio: <i>Ottimizza multigruppo</i>; ✓ OptiFlag.bopti_type4 <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ corrisponde al settaggio: <i>Ordinamento a greca</i>; ✓ OptiFlag.fh1 <ul style="list-style-type: none"> ◇ Tipologia [float]; ◇ corrisponde al settaggio: <i>Quota q1</i>; ✓ OptiFlag.fh2 <ul style="list-style-type: none"> ◇ Tipologia [float]; ◇ Non utilizzato; ✓ OptiFlag.fh3 <ul style="list-style-type: none"> ◇ Tipologia [float]; ◇ gestito come maschera di bit: <ul style="list-style-type: none"> ❖ bit 0 = 1 corrisponde al settaggio <i>Utensili Tn in matrice pezzo</i>; ❖ bit 1: corrisponde al settaggio <i>Test lunghezza utensile per lavorazioni puntuali</i>; ❖ bit 2 : corrisponde al settaggio <i>Ottimizza per utensile (non per ottimizzatore custom)</i>; ✓ OptiFlag.arrseq_side[32] <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ corrisponde al settaggio: <i>Sequenza facce</i>; ✓ OptiFlag.arrant_setting[16] <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ arraint_setting[0]: corrisponde al settaggio <i>Assegna sequenze</i>; ◇ arraint_setting[1-7]: disponibili come flag; ◇ arraint_setting[8-15]: disponibili come valori; ✓ OptiFlag.arrant_custom[16] <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ corrisponde a settaggi per ottimizzatore custom; ◇ arraint_custom [1-7]: disponibili come flag; ◇ arraint_custom [8-15]: disponibili come valori; ✓ OptiFlag.arrcode_engen1[10] <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ corrisponde alle assegnazioni relative ai <i>Codici speciali</i>; ✓ OptiFlag.arrcode_ord[10][10] <ul style="list-style-type: none"> ◇ tipologia [int]; ◇ corrisponde alle assegnazioni relative all' <i>Ordinamento codici</i>; ✓ OptiFlag.fh4 <ul style="list-style-type: none"> ◇ Tipologia [float]; ◇ Non utilizzato; ✓ OptiFlag.fh5 <ul style="list-style-type: none"> ◇ Tipologia [float]; ◇ Non utilizzato; ✓ OptiFlag.bopti_type5 <ul style="list-style-type: none"> ◇ Tipologia [int]; ◇ corrisponde al settaggio: <i>Test lunghezza utensile per profili</i>; ✓ OptiFlag.bopti_type6 <ul style="list-style-type: none"> ◇ Tipologia [int]; ◇ corrisponde al settaggio: <i>Ordinamento battute</i>; ✓ OptiFlag.bopti_type7 <ul style="list-style-type: none"> ◇ Tipologia [int]; ◇ corrisponde al settaggio: <i>Ordinamento codici</i>; ✓ OptiFlag.bopti_type8
--	---

		◇ Tipologia [int]; ◇ Non utilizzato; ✓ OptiFlag.bopti_type9 ◇ Tipologia [int]; ◇ Non utilizzato;
DatiProgramma	char PathFile[260]	Pathname del programma che deve essere ottimizzato
	char Commento [40]	commento del programma (troncato a 40 caratteri)
	char Data[12]	data ultima modifica del programma
	char Time[12]	ora ultima modifica del programma
	BYTE EscMac[8]	Esclusione macchine: EscMac[0]= 1 esclude la macchina 1 EscMac[1]= 1 esclude la macchina 2 .. EscMac[7]= 1 esclude la macchina 9
	BYTE EscMacGr[8][10]	Esclusione gruppi: EscMacGr[0][0] esclude il gruppo 1 della macchina 1 .. EscMacGr[7][9] esclude il gruppo 10 della macchina 8
char Vuoto[200]	Non utilizzato	
Matrice pezzo, organizzata in: <ul style="list-style-type: none"> • numero di righe = CommInputOptimPacket.NumRigheMat • numero di colonne per riga = CommInputOptimPacket.NumColonneMat Ogni cella della matrice ha tipologia [float]. La organizzazione della matrice pezzo è riportata in un paragrafo successivo.		

7.1.2. Files di parametriche

OPTIMIZE.EXE legge i files di parametrica (PARTEC.PAR, PARTOOL.PAR) e crea, nello stesso direttorio, 3 file binari con nome fisso:

MXMAC.BIN	contiene la parametrica dei limiti di macchina (da matrice MxCnfMac)
MXGRU.BIN	contiene la parametrica utensili (da matrice MxCnfGru)
MXUT.BIN	contiene la parametrica utensile (da matrice MxUtensili)
MXVEN.BIN	contiene la parametrica ventose (da matrice MxCnfVen)

I primi campi di ciascuna file rappresentano:

1° campo: [DWORD] progressivo di registrazione file

2° campo: [short] numero di righe del file

3° campo: [short] numero di colonne per riga

da 4° a 13° campo: [long] non utilizzati.

7.2. Esposizione di oggetti

Dalla versione [2.4.008] gli eseguibili di ottimizzazione (OPTIMIZE ed OPTIALBY) espongono ognuno un oggetto, che può essere istanziato direttamente da un programma client. Rispettivamente:

- Albatros.Optimize
- Albatros.Optialby.

Per registrare gli oggetti nel registro di Windows è sufficiente avviare gli eseguibili una volta direttamente.

Gli oggetti presentano i seguenti metodi:

Metodo	Ritorno	Parametri
ReadDataFile	Boolean	BSTR <i>pathname</i> BSTR* <i>szDescr</i> BSTR* <i>szDimOffs</i> BSTR* <i>szVars</i> BSTR* <i>strError</i>
<p>Il metodo interroga l'ottimizzatore in merito al file <i>pathname</i>. Se il file corrisponde ad un testo valido di programma, il metodo può restituire informazioni del file stesso (se il metodo è chiamato con argomenti significativi) :</p> <p>◇ Commento in <i>szDescr</i>;</p> <p>◇ Unità, dimensioni ed offsets in <i>szDimOffs</i>. Esempio:</p> <pre> "0;2000;750;45;50;-20;-2" 0=mm ← → offsets 1=inch → dimensioni </pre> <p>nell'esempio:</p> <ul style="list-style-type: none"> ✓ unità = mm ✓ lunghezza = 2000; altezza=750; spessore=45 ✓ offset x = 50; offset y = -20 ; offset z = -2 <p>◇ variabili riassegnabili (solo quelle con impostazione non parametrica). Esempio:</p> <pre> "#0=120 #2=45 #5=pippo" → variabile r0, impostazione "120" </pre> <p>In caso di uscita con errore (ritorno del metodo: 0) <i>strError</i> riporta il messaggio di errore. Il metodo torna un valore diverso da 0 se il file corrisponde ad un testo valido di programma. Sono riconosciuti anche formati per i quali è assegnata una conversione (esempio: file CNC90, DXF).</p>		
Optimize	Boolean	BSTR <i>pathname</i> integer <i>fgDisk</i> BSTR <i>szModeExe</i> BSTR <i>szDimOffs</i> BSTR <i>szVars</i> BSTR* <i>strError</i> BSTR* <i>strFileError</i>
<p>Il metodo richiede l'ottimizzazione del file <i>pathname</i>, con le assegnazioni date:</p> <p>◇ <u>fgDisk=0</u> non esegue archiviazione: esegue una ottimizzazione minima, non crea il file di matrice pezzo in formato binario -MXALB.BIN in directory <i>dirreport-</i> o ASCII -.TXN,.TXM- (in Edicad, corrisponde al comando File>>Salva); <u>fgDisk=1</u> esegue una ottimizzazione completa, non crea il file di matrice pezzo in formato binario, crea invece il file di matrice pezzo in formato ASCII (in Edicad, corrisponde al comando File>>Ottimizza); <u>fgDisk=2</u> esegue una ottimizzazione completa, crea il file di matrice pezzo in formato binario, mentre la creazione del file di matrice pezzo in formato ASCII è determinata dal settaggio di Debug come impostato in TPA.INI per l'eseguibile.</p> <p>◇ Modalità di esecuzione in <i>szModeExe</i>. Esempio: "1;2000;-5;0" corrisponde a: ✓ Esecuzione in speculare x (area 1); ✓ Battuta x =2000; battuta y = -5; battuta z = 0.</p> <p>Se <i>szModeExe</i> non è assegnato ottimizza per esecuzione normale, con offsets di battuta tutti nulli (0;0;0);</p> <p>◇ Dimensioni ed offsets in <i>szDimOffs</i>. Esempio:</p> <pre> "1900;0;40;1;0;0;55;200;100" Dimensioni ← </pre>		

| |
 On(1)/Off(0) per ← | | → offsets
 Riassegnazione
 di offset x, y, z

corrisponde a:

- ✓ Lunghezza = 1900; altezza = 0 (mantiene il valore originale); spessore = 40;
- ✓ offset x riassegnato con valore 55;
- ✓ offset y non riassegnato (il corrispondente flag di abilitazione dell'assegnazione è 0: il valore 200 è ignorato);
- ✓ offset z non riassegnato (il valore 100 è ignorato).

Se *szDimOffs* non è assegnato: mantiene dimensioni ed offsets originali.

◇ Riassegnazioni di variabili in *szVars*. Esempio: "#0=120|#2=45|#5=pippo" corrisponde a:

- ✓ variabile r0: assume valore 120
- ✓ variabile r2: assume valore 45
- ✓ variabile r5: assume valore "pippo" .

Assegnazioni non valide sono ignorate.

Se ritorna con errore (ritorno del metodo: 0) *strError* riporta il messaggio di errore.

Se l'errore è in fase di elaborazione del pezzo (esempi: errore di raggio fresatura o di applicazione delle condizioni logiche) è registrato un file complessivo degli errori di elaborazione ed il pathname è restituito in *strFileError* (se assegnato).

Il metodo torna un valore diverso da 0 se il file corrisponde ad un testo valido di programma e l'ottimizzazione ha avuto esito positivo.

OptimizeEx	Boolean	BSTR <i>pathname</i> integer <i>fgDisk</i> BSTR <i>szModeExe</i> BSTR <i>szDim</i> BSTR <i>szVars</i> BSTR <i>szInteger</i> BSTR <i>szFloat</i> BSTR* <i>strError</i> BSTR* <i>strFileError</i>
------------	---------	---

Il metodo è stato aggiunto in versione [2.5.002] ed è in parte analogo al metodo *Optimize*.

Il metodo richiede l'ottimizzazione del file *pathname*, con le assegnazioni date:

◇ *fgDisk=0,1*: vedi metodo *Optimize*;

fgDisk=2 gestisce una ottimizzazione completa con archiviazione di informazioni tale da permettere la chiamata OLE dell'ottimizzatore anche da LISTER. La creazione del file di matrice pezzo in formato ASCII è determinata dal settaggio di Debug come impostato in TPA.INI per l'eseguibile (se in debug: scrive nel direttorio in cui è memorizzato il file da ottimizzare il file testo TXN o TXM rappresentante la matrice pezzo).

Il passaggio dei dati da LISTER ad ottimizzatore e viceversa avviene, in funzionamento standard, via *SendMessage* ed i dati della matrice pezzo vengono passati in un'area di memoria globale.

Con la gestione via OLE la gestione dei dati deve avvenire in modo diverso.

L'ottimizzatore crea il file di matrice pezzo in formato binario -MXALB.BIN in directory *dirreport-* contenente in istanziazione la struttura di comunicazione *CommInputOptimPacket* e di seguito una sequenza di float rappresentanti la matrice pezzo. Il file MXALB.BIN viene cancellato dal programma chiamante (cioè: LISTER).

I dati della struttura *CommInputOptimPacket* che interessano in uscita il programma chiamante sono:

bCommOk TRUE = fase di ottimizzazione OK

nError se l'ottimizzazione è andata a buon fine è assegnato il valore GOOD (=0). Se si è verificato un errore assume valore a seconda dell' errore occorso.

- ERRORE (-1) errore generico
- ERR_NOTFILE (-2) errore in creazione o scrittura file
- ERR_NOTMEMORY (-3) errore in allocazione memoria
- ERR_NOTELABOTT (-4) errore in elaborazione parametri;

NumColonneMat numero di colonne in matrice pezzo

NumRigheMat numero di righe in matrice pezzo

MsgErr.CodErr codice di errore

La struttura *MsgErr* è assegnata solo se si verifica un errore di elaborazione dati. In questo caso il campo *nError* vale -4 (ERR_NOTELABOTT).

Se è (*nError*=-4, *CodErr*=0): il messaggio di errore da visualizzare è contenuto in *MsgErr.StrErr*;

MsgErr.StrErr messaggio di errore;

fgDisk=10 non esegue archiviazione e non lancia l'ottimizzatore custom pur creando tutti i file necessari all'ottimizzatore custom;

`[fgDisk=11]` crea il file di matrice pezzo in formato ASCII e non lancia l'ottimizzatore custom pur creando tutti i file necessari all'ottimizzatore custom;

`[fgDisk=12]` crea il file di matrice pezzo in formato binario -MXALB.BIN- e non lancia l'ottimizzatore custom pur creando tutti i file necessari all'ottimizzatore custom;

- ◇ Modalità di esecuzione in `szModeExe` (vedi metodo *Optimize*);
Dimensioni in `szDim`. Esempio: "1900;0;40", corrisponde a:
✓ Lunghezza = 1900; altezza = 0 (mantiene il valore originale); spessore = 40.
Se `szDimOffs` non è assegnato: mantiene dimensioni originali.

- ◇ Riassegnazioni di variabili in `szVars`. (vedi metodo *Optimize*);

Assegnazioni numeriche intere (dieci) in `szInteger`. Esempio: "0;1;0;0;0;0;0;0;0;0"
 1° campo: *** non usato dall'ottimizzatore
 2° campo: On(1)/Off(0) per riassegnazione di Offset x (valore dell'offset: nel 2° campo di `szFloat`)
 3° campo: On(1)/Off(0) per riassegnazione di Offset y (valore dell'offset: nel 3° campo di `szFloat`)
 4° campo: On(1)/Off(0) per riassegnazione di Offset z (valore dell'offset: nel 4° campo di `szFloat`)
 5° campo: *** non usato dall'ottimizzatore
 6° campo: *** non usato dall'ottimizzatore
 7° campo: *** non usato dall'ottimizzatore
 8° campo: *** non usato dall'ottimizzatore
 9° campo: *** non usato dall'ottimizzatore
 10° campo: *** non usato dall'ottimizzatore

Le assegnazioni indicate non usate dall'ottimizzatore possono assumere un significato specifico per un ottimizzatore custom.

Assegnazioni numeriche con virgola (dieci) in `szFloat`. Esempio: "0;50.5;0;0;0;0;0;0;0;0"
 1° campo: *** non usato dall'ottimizzatore
 2° campo: Offset x
 3° campo: Offset y
 4° campo: Offset z
 5° campo: *** non usato dall'ottimizzatore
 6° campo: *** non usato dall'ottimizzatore
 7° campo: *** non usato dall'ottimizzatore
 8° campo: *** non usato dall'ottimizzatore
 9° campo: *** non usato dall'ottimizzatore
 10° campo: *** non usato dall'ottimizzatore

Le assegnazioni indicate non usate dall'ottimizzatore possono assumere un significato specifico per un ottimizzatore custom.

=====

In caso di uscita con errore (ritorno del metodo: 0) `strError` riporta il messaggio di errore.
 Se l'errore è in fase di elaborazione del pezzo (esempi: errore di raggio fresatura, oppure di applicazione delle condizioni logiche) è registrato un file complessivo degli errori di elaborazione ed il pathname è restituito in `strFileError` (se assegnato).

Il metodo torna un valore diverso da 0 se il file corrisponde ad un testo valido di programma e l'ottimizzazione ha avuto esito positivo.

OptimizeInit	Boolean
--------------	---------

Il metodo è stato aggiunto in versione [2.5.002].

Il metodo richiede la rilettura dei settaggi di inizializzazione impostati in file PEZZO.INI .

Il metodo torna un valore diverso da 0 se eseguito con successo.

Esempio di programma Visual Basic:

```
Dim opti As Object
Dim NameFile As String
Dim szModeExe As String
Dim szDimOffs As String
Dim stError As String
Dim stFileError As String
Dim nRet As Integer
```

```
Set opti = CreateObject("Albatros.Optimize")
```

```
....
```

```
stError = ""
stFileError = ""
NameFile="d:\custom\mod.0\prod\pippo"      ' pathname del file da ottimizzare
szModeExe="1;2700;0;0"                    ' esecuzione speculare, battuta x a 2700 mm
szDimOffs=""                              ' dimensioni ed offsets originali
szVars = "#0=5|#2=70"                    ' variabili riassegnabili

nRet = opti.Optimize(NameFile, 1, szModeExe, szDimOffs, szVars, stError, stFileError)
If nRet <> 0 Then
    '...Ottimizzazione OK.
Else
    '...Ottimizzazione NOK.
End If
.....
Set opti = Nothing
```

7.3. Descrizione errori

Numero di errore	Messaggio	Descrizione e situazioni di errore
7	Matrice utensili non trovata o non valida	✓ La matrice dei parametri utensile (MxcnfGru) non è stata definita;
8	Matrice Macchina non trovata o non valida	✓ La matrice dei parametri macchina (MxCnfMac) non è definita; ✓ Le tipologie utilizzate per la definizione dei parametri macchina assegnano un formato binario misto tra il tipo 1 ed il tipo 2; ✓ È dichiarata l'esistenza di una macchina ma non è definito per essa alcun gruppo;
9	Fallito lancio dell'Elaboratore 4 assi	✓ L'eseguibile per l'elaborazione 4 assi non è presente nella cartella \albatros\bin; ✓ OPTIMIZE e l'elaboratore hanno problemi nel passaggio di dati della struttura di comunicazione;
10	Limiti gruppi non definiti	✓ È definito un gruppo in antiscontro, ma non sono definiti i limiti macchina
12	Codice descrizione faccia non trovato	✓ Non è definita nel data-base delle lavorazioni l'intestazione della faccia.
13	Errore nella compilazione del pezzo	✓ Richiesta esecuzione speculare diversa da X per macchina multigruppo; ✓ Errore grave durante la compilazione del pezzo, che non permette di eseguire l'ottimizzazione;
14	Utensile non trovato	✓ In programmazione per diametro non è stato trovato alcun utensile che possa eseguire la lavorazione; ✓ In programmazione per utensile possono verificarsi le seguenti situazioni: ◇ l'utensile non è definito in parametrica; ◇ l'utensile non è definito nella lavorazione; ◇ l'utensile è definito nella lavorazione, ma è impostato un valore di macchina <=0; ◇ l'utensile è definito nella lavorazione ma il gruppo a cui appartiene è disabilitato; ◇ l'utensile appartiene ad un gruppo in aggancio che ha una numerazione maggiore di 10; ◇ l'utensile appartiene ad un gruppo SLAVE diverso da quello programmato nella lavorazione.
15	Parametri di definizione utensile non trovati	✓ Non è definito alcun parametro di uguaglianza tra utensile e lavorazione (i parametri per i quali è verificata una condizione di uguaglianza appartengono al range: 1001 – 2000).
16	Utensile fuori dai limiti della macchina x+	La quota x dell'utensile supera il limite positivo dell'asse X
17	Utensile fuori dai limiti della macchina x-	La quota x dell'utensile precede il limite negativo dell'asse X
18	Utensile fuori dai limiti della macchina y+	La quota y dell'utensile supera il limite positivo dell'asse Y
19	Utensile fuori dai limiti della macchina y-	La quota y dell'utensile precede il limite negativo dell'asse Y
20	Utensile fuori dai limiti della macchina z+	La quota z dell'utensile supera il limite positivo dell'asse Z
21	Utensile fuori dai limiti della macchina z-	La quota z dell'utensile precede il limite negativo dell'asse Z
22	Operatività non ammessa in faccia	Il codice della lavorazione (puntuale o setup) è un codice geometrico e non è definita la tecnologia di default da assegnare.
23	Troppe lavorazioni sulla faccia	Il numero di lavorazioni per faccia è superiore a 32767
24	Partec.Par non trovato	Il file non esiste
27	Errore in apertura del file	✓ Errore in apertura del file .txn generato in fase di ottimizzazione; ✓ Errore in apertura del programma da ottimizzare
28	Errore in apertura parametrica utensile	Si è verificato un errore in fase di apertura del file della parametrica utensile
29	File o gruppo di lavorazioni non trovato	Errore in fase di apertura del database delle lavorazioni
30	Gruppo lavorazioni vuoto	Nel database delle lavorazioni non è definito alcun gruppo di lavorazioni
31	Creazione dialog OPTIRUN fallita	Fallita la creazione della finestra con i parametri definiti nella struttura OPTIRUN (solo per ottimizzatore interattivo)
34	Utensile fuori limiti quota aria z-	La quota aria dell'utensile è esterno al limite negativo dell'asse Z
35	Utensile fuori limiti quota aria z+	La quota aria dell'utensile è esterno al limite positivo dell'asse Z
36	Profondità lavorazione maggiore della lunghezza utensile	La profondità programmata è superiore alla lunghezza utensile
37	Incongruenza tra configurazione multigruppo e flag TYPE3	Parametrica con più gruppi con impostazione per OPTIMIZE di macchina monogruppo (questo tipo di assegnamento non è considerato di errore solo in caso di gruppi in aggancio).

38	Non è possibile l'esecuzione speculare non su asse X	Ad una macchina multigruppo è stata assegnato il flag per l'esecuzione di uno speculare diverso da X
40	Parametri lavorazione non congruenti con parametri utensile	Errore in controllo tra i parametri della lavorazione ed i corrispondenti parametri dell'utensile (viene effettuato un controllo di uguaglianza tra i parametri con tipologia compresa tra 1001 e 2000 ed un controllo di esistenza nel range tra i parametri a tipologia compresa tra 2001 e 3000).
41	La testa è in collisione con il pezzo	La quota zeta sommata all'ingombro zeta del gruppo è maggiore dello spessore del pannello
42	Nessun settaggio di pistoni è accettato	Nessuna condizione per l'utilizzo di pistoni è accettata
43	Macchina non abilitata	La macchina da utilizzare non è stata abilitata nelle impostazioni di ottimizzatore in EDICAD
44	Fallito lancio Ottimizzatore custom	Si è verificato un errore in esecuzione dell'ottimizzatore custom: 1) l'ottimizzatore custom non esiste 2) non si riesce ad istanziare l'oggetto ottimizzatore custom
45	Errore nell'apertura di parmac.bin	Si è verificato un errore in fase di apertura del file parmac.bin.
46	Errore nell'apertura di parut.bin	Si è verificato un errore in fase di apertura del file parut.bin.
47	Lettura matrice da ottimizzatore custom non corretta	La dimensione dichiarata dall'ottimizzatore custom per la matrice pezzo letta dal file mxpzout.bin non è corretta.
48	Sono stati definiti più gruppi MASTER	In parametrica dei gruppi è stato definito più di un gruppo MASTER
49	Sono stati definiti gruppi in aggancio con una configurazione multigruppo	Solo per gruppi in aggancio
50	Incongruenza tra le modalità di comunicazione	La comunicazione con l'ottimizzatore custom è stata richiesta sia in modalità OLE in sia in modalità tradizionale
53	fallita creazione file di parametrica in binario	La creazione dei file di parametrica per l'ottimizzatore custom può avere avuto uno dei seguenti problemi: ✓ Il file di parametrica PARTEC.PAR non esiste; ✓ Si è verificato un errore durante la scrittura del file binario
54	Il gruppo non può eseguire lavorazioni passanti	Si cerca di eseguire una lavorazione passante con un utensile di un gruppo che non può eseguire lavorazioni passanti
55	Dimensionamento matrice errato	È stata allocata una matrice pezzo con un numero di righe inferiore al numero di righe realmente utilizzato
57	Alcuni utensili del gruppo slave hanno la stessa numerazione del gruppo master	Solo per gruppi in aggancio
58	Non è stato definito o abilitato il gruppo master	Sono stati definiti in parametrica dei gruppi in aggancio, ma non è stato specificato quale sia il gruppo MASTER
59	Il gruppo Master lavora su un lato diverso rispetto ad uno o più gruppi slave	Il gruppo MASTER ed i gruppi SLAVE associati devono lavorare sullo stesso lato della macchina

8. PARAMETRICHE

Un primo punto da esaminare riguarda il formato delle parametriche: gli eseguibili EDICAD ed OPTIMIZE possono accedere a parametriche disponibili in formati differenti.

In questo capitolo sono poi esaminate tematiche particolari di assegnazione delle parametriche.

8.1. Formato testo

La lettura dei files in formato testo prevale su quella dei files in formato binario: se per il modulo richiesto è trovato il file MXCNFMAC.TXT, allora è caricata la parametrica da file testo.

I file testo, come quelli binari, sono memorizzati nella cartella descritta dalla variabile di ambiente CONFIG definita in tpa.ini .

I nomi dei files sono fissi:

- MXCNFMAC.TXT: assegna i parametri dei gruppi macchina (corrisponde alla matrice "MxCnfMac" dei files di parametrica binari);
- MXCNCGRU.TXT: assegna gli utensili (ogni riga rappresenta un utensile);
- MXUTEN.TXT: assegna l'archivio degli utensili (ogni riga rappresenta un utensile). Nel file MXCNFGRU troviamo il parametro con tipologia 220 che definisce il numero di riga del file MXUTEN.TXT in cui sono scritti i parametri di base dell'utensile in esame;
- MXCNFVEN.TXT: assegna i parametri dei vincoli (ventose, ..). Questo file viene semplicemente letto e riscritto da OPTIMIZE per l'ottimizzatore custom (non è necessario che il file sia assegnato). Il file con i parametri delle ventose deve essere scritto rispettando il formalismo con cui si scrive la matrice delle ventose in formato Albatros: 2 colonne per un massimo di 500 righe.

Il formato dei files è il seguente:

- ✓ la prima riga contiene delle informazioni generali separate tra loro dal carattere di ';':
 - ◊ per ora è stata definito il solo progressivo di registrazione di parametrica (se il progressivo di parametrica non è definito: OPTIMIZE assume valore 0): se il progressivo cambia, prima di eseguire una ottimizzazione OPTIMIZE ricarica la parametrica utensile;
- ✓ le righe seguenti contengono i dati di parametrica secondo il seguente formalismo: "#tipologia=valore(spazio)#tipologia=valore(spazio)".

Per l'interpretazione delle parametriche: si ricade in uno dei due casi di formato binario descritti nei paragrafi successivi.

8.2. Formati binari

Vengono letti i seguenti files o matrici:

- matrice "MxCnfMac" (il nome è configurabile), in file: "Mod0\config\PARTEC.PAR" (per il modulo 0, "Mod1\..." per il modulo 1, ecc):
 - ✓ la matrice assegna: le abilitazioni ed i limiti del modulo (macchina);
 - ✓ dimensioni lette per la matrice: numero di righe = 200 (100, fino alla versione 2.5.002a), numero di colonne = 64;
 - ✓ se il file PARTEC.PAR o la matrice non sono assegnati: per il modulo in questione non è caricata alcuna parametrica (EDICAD ed OPTIMIZE segnalano errore);
 - ✓ l'organizzazione della matrice determina l'interpretazione di uno dei due formati riconosciuti: in caso di formato non valido o di nessun gruppo assegnato non viene caricata alcuna parametrica (EDICAD ed OPTIMIZE segnalano errore);
- matrice " MxCnfGru " (il nome è configurabile), in file: "Mod0\config\PARTEC.PAR" (per il modulo 0, "Mod1\..." per il modulo 1, ecc):
 - ✓ la matrice assegna i gruppi della macchina (per un massimo di 10 gruppi);
 - ✓ dimensioni lette per la matrice: numero di righe = (numero dei gruppi abilitati in matrice "MxCnfMac") * (numero di mandrini per gruppo), numero di colonne =64. Si rammenta che il numero di mandrini per gruppo è assegnato in EDICAD (Imposta: Ottimizzatore) ed ha un valore massimo di 1000;
 - ✓ la matrice è quindi interpretata suddivisa in sezioni successive:
 - ◊ il numero delle sezioni è pari ai gruppi abilitati;
 - ◊ ogni sezione ha numero di righe = numero di mandrini per gruppo;
 - ◊ la 1°sezione assegna gli utensili del 1°gruppo a bilitato;
 - ◊ ..
 - ◊ l'ultima sezione assegna gli utensili dell'ultimo gruppo abilitato.
 - ✓ se la matrice non è assegnata: per il modulo in questione non viene caricata alcuna parametrica (EDICAD ed OPTIMIZE segnalano errore);
- file "Mod0\config\ PARTOOL.PAR" (per il modulo 0, "Mod1\..." per il modulo 1, ecc):
 - ✓ il file assegna gli attrezzaggi di macchina;
 - ✓ il file può non esistere.

8.2.1. Formato binario tipo 1

Organizzazione della matrice "MxCnfMac": una riga per ogni gruppo (per un massimo di 10 gruppi).
La 1° riga della matrice assegna il gruppo 1, la 2° riga assegna il gruppo 2, ecc.

I parametri riconosciuti per ogni riga sono:

Significato parametro	Tipologia parametro	
Abilitazione gruppo	6	Valori significativi: 0: gruppo non presente 1: gruppo abilitato in lato destro (non lavora in faccia 2) 2: gruppo abilitato in lato sinistro (non lavora in faccia 1) 3: gruppo abilitato in lato destro e sinistro (opera senza esclusione di facce).
Limite X-	151	Limiti massimi di posizionamento del gruppo, relativi agli assi gestiti
Limite X+	150	
Limite Y-	153	
Limite Y+	152	
Limite Z-	155	
Limite Z+	154	
Limite C-	157	
Limite C+	158	
Limite B-	159	
Limite B+	160	
Offset X	100	Offsets di gruppo: assegnano l'offset del gruppo rispetto allo zero di macchina, con gruppo in posizione di zero
Offset Y	101	
Offset Z	102	
Offset X speculare	138	Non utilizzato
Gruppo di antiscontro	206	Un valore positivo non nullo indica il gruppo in antiscontro (in direzione y)
Quota di antiscontro	156	Definisce il massimo avvicinamento consentito al gruppo in antiscontro, dato come valore da sommare alla propria posizione y per raggiungere il gruppo in anticollisione. Si definisce che tutti i gruppi si muovano verso Y positive nella medesima direzione: i gruppi bassi hanno quindi una quota di anticollisione positiva, i gruppi alti hanno una quota di anticollisione negativa.
Ingombro Z	161	
Abilitazione Utensili Passanti	162	La colonna è definita dalla versione 2.1.000.
Lato di riferimento	163	È significativo per macchine che gestiscono il ribaltamento del pezzo (gestione speciale per linee).

8.2.2. Formato binario tipo 2

La matrice "MxCnfMac" non associa più una riga ad un gruppo.

Ora sono riconosciuti parametri diversi per caratterizzare ogni gruppo: i parametri possono essere comunque dislocati nella matrice.

Attenzione: non devono essere mischiati parametri del formato tipo 1 con parametri del formato tipo 2: altrimenti EDICAD ed OPTIMIZE segnalano errore.

Significato parametro		Tipologia parametro in base al gruppo:									
		1	2	3	4	5	6	7	8	9	10
Abilitazione di gruppo	Valore #0: abilita il gruppo Valori significativi: <ul style="list-style-type: none"> • 0: gruppo non presente; • 1: gruppo abilitato in lato destro (non lavora in faccia 2); • 2: gruppo abilitato in lato sinistro (non lavora in faccia 1) • 3: gruppo abilitato in lato destro e sinistro (opera 	52	53	54	55	56	57	58	59	60	61

	senza esclusione di facce).										
Limite X-		301	316	331	346	361	376	391	406	421	436
Limite X+		300	315	330	345	360	375	390	405	420	435
Limite Y-		303	318	333	348	363	378	393	408	423	438
Limite Y+		302	317	332	347	362	377	392	407	422	437
Limite Z-		305	320	335	350	365	380	395	410	425	440
Limite Z+		304	319	334	349	364	379	394	409	424	439
Limite C-		307	322	337	352	367	382	397	412	427	442
Limite C+		308	323	338	353	368	383	398	413	428	443
Limite B-		309	324	339	354	369	384	399	414	429	444
Limite B+		310	325	340	355	370	385	400	415	430	445
Offset X		165	168	171	174	177	180	183	186	189	192
Offset Xm		139	140	141	142	143	144	145	146	147	148
Offset Y		166	169	172	175	178	181	184	187	190	193
Offset Z		167	170	173	176	179	182	185	188	191	194
Gruppo in Antiscontro		222	223	224	225	226	227	228	229	230	231
Quota Antiscontro		306	321	336	351	366	381	396	411	426	441
Ingombro Z		311	326	341	356	371	386	401	416	431	446
Lato di riferimento	Valori significativi: 1: lato 1; 2: lato 2; 0: entrambi. (gestione speciale per linee)	313	328	343	358	373	388	403	418	433	448
Abilitazione Utensili Passanti	Definiti dalla versione 2.1.000.	312	327	342	357	372	387	402	417	432	447

8.3. Assegnazione di matrice MxCnfGru

Una riga della matrice "MxCnfGru" assegna un utensile del gruppo.

La 1° riga della matrice assegna l'utensile 1, la 2° riga assegna l'utensile 2, ecc.

Di seguito viene dato un prospetto di massima dei parametri notevoli riconosciuti per un utensile (riga di matrice):

Tipologia parametro	Significato parametro
6	Abilitazione di utensile: assegna le facce del pezzo su cui può lavorare l'utensile. Valori significativi: 0 nessuna faccia (l'utensile non esiste) 1 faccia 1 (e facce fittizie simili o similari alla 1) 12 faccia 1 e 2 (e facce fittizie ad esse simili o similari) 46 facce 4 e 6 (e facce fittizie ad esse simili o similari) .. 123456 tutte le facce (anche tutte le facce fittizie): si tratta di un utensile universale. Il valore del parametro deve essere interpretato sul valore di ogni cifra (in ordine non obbligatorio). Per le facce fittizie si definisce di poter riconoscere se la faccia è <i>simile</i> ad una delle facce reali del parallelepipedo: in questo caso accetta programmazione per ogni utensile configurato valido per la faccia reale. La dicitura di faccia <i>simile</i> riconosce che la faccia fittizia è definita con punti che individuano una terna cartesiana traslata rispetto alla terna di assegnazione di una delle facce reali.
45	Utensile su testa rotante
46	Disabilita controlli utensile da parte dell'ottimizzatore
100	Correttore X utensile (rispetto allo zero x del gruppo)
101	Correttore Y utensile (rispetto allo zero y del gruppo)
102	Correttore Z utensile (rispetto allo zero z del gruppo)
103	Correttore Alfa (Offset tra centro mandrino e naso mandrino)
104	Correttore Beta (Offset tra centro mandrino e naso mandrino)
109	Lunghezza utensile
110	Extracorsa positivo con doppi pistoni: lunghezza pistone 3
111	Lunghezza utensile di foratura (utilizzato per confronto con la profondità programmata)
112	Usura diametro utensile espresso in millimetri su metro (visualizzato in EDICAD)
121	Quota aria
124	Utensile speculare
125	Disabilita test quota aria
126	Extracorsa positivo con doppi pistoni: lunghezza pistone 1

127	Extracorsa positivo con doppi pistoni: lunghezza pistone 2																																																				
129	Correttore y punta 1 per testina multi punta																																																				
130	Correttore x punta 2 per testina multi punta																																																				
131	Correttore y punta 2 per testina multi punta																																																				
132	Correttore x punta 3 per testina multi punta																																																				
133	Correttore y punta 3 per testina multi punta																																																				
134	Correttore x punta 4 per testina multi punta																																																				
135	Correttore y punta 4 per testina multi punta																																																				
136	Offset simmetria X per testina multi punta																																																				
137	Offset simmetria Y per testina multi punta																																																				
199	Correttore alfa per lavorazione di lama (offset tra centro di rotazione e naso mandrino)																																																				
220	Attrezzaggio: indica un utensile dell'archivio utensili (file MXUTEN.TXT o PARTOOL.PAR)																																																				
221	<p>Utensile di riferimento (in stessa matrice MxCnfGru):</p> <ul style="list-style-type: none"> ✓ valore 0: non ha alcun effetto; ✓ valore >0: assegna un utensile di riferimento per l'utensile in questione. <p>In caso di utensile di riferimento assegnato:</p> <ul style="list-style-type: none"> ✓ i primi 32 parametri dell'utensile sono assegnati prelevandoli dall'utensile di riferimento; ✓ per i successivi 32 parametri sono mantenuti quelli dell'utensile. 																																																				
1001	<p>Tipologia utensile.</p> <p>Per la verifica o scelta di un utensile attribuito o da attribuire ad una determinata lavorazione occorre in generale definire un legame univoco tra: lavorazione ed utensile.</p> <p>A tale scopo si definisce di utilizzare il parametro [1001]: si noti che appartenente al range dei parametri per i quali deve valere il test di uguaglianza tra lavorazione ed utensile.</p> <p>Con parametro [1001] presente, OPTIMIZE esegue il controllo di uguaglianza come per il caso degli altri parametri nel range [1001 - 2000]. In particolare:</p> <ul style="list-style-type: none"> ✓ Il parametro non deve essere necessariamente presente; ✓ il parametro dovrebbe essere gestito per programmazioni dirette di utensile (in caso contrario: se gli altri controlli risultano verificati è comunque possibile eseguire una lavorazione con utensile non compatibile); ✓ il parametro di tipologia utensile può non essere gestito nei casi in cui non è possibile scegliere un utensile non compatibile: ad esempio se la macchina ha un solo utensile con un determinato diametro. <p>Per i valori da attribuire al parametro si definisce una tabella di corrispondenze di massima (attenzione: è tutta una convenzione, ma è utile per non cambiare tutto da una macchina all'altra):</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">valore del parametro</th> <th style="text-align: left;">tipologia associata</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>foratore</td> </tr> <tr> <td>.</td> <td></td> </tr> <tr> <td>da 10</td> <td>foratore speciale tipo 1</td> </tr> <tr> <td>a 29</td> <td>foratore speciale tipo 20</td> </tr> <tr> <td>30</td> <td>maschiatore</td> </tr> <tr> <td>.</td> <td></td> </tr> <tr> <td>35</td> <td>foratore con scarico</td> </tr> <tr> <td>.</td> <td></td> </tr> <tr> <td>40</td> <td>tastatore tipo 1</td> </tr> <tr> <td>41</td> <td>tastatore tipo 2</td> </tr> <tr> <td>42</td> <td>tastatore tipo 3</td> </tr> <tr> <td>43</td> <td>tastatore tipo 4</td> </tr> <tr> <td>44</td> <td>tastatore tipo 5</td> </tr> <tr> <td>.</td> <td></td> </tr> <tr> <td>50</td> <td>lama rotante</td> </tr> <tr> <td>51</td> <td>lama x</td> </tr> <tr> <td>52</td> <td>lama y</td> </tr> <tr> <td>.</td> <td></td> </tr> <tr> <td>100</td> <td>fresa</td> </tr> <tr> <td>.</td> <td></td> </tr> <tr> <td>da 110</td> <td>fresa speciale tipo 1</td> </tr> <tr> <td>a 129</td> <td>fresa speciale tipo 20</td> </tr> <tr> <td>.</td> <td></td> </tr> <tr> <td>da 150</td> <td>inseritore tipo 1</td> </tr> <tr> <td>a 169</td> <td>inseritore tipo 20.</td> </tr> </tbody> </table>	valore del parametro	tipologia associata	1	foratore	.		da 10	foratore speciale tipo 1	a 29	foratore speciale tipo 20	30	maschiatore	.		35	foratore con scarico	.		40	tastatore tipo 1	41	tastatore tipo 2	42	tastatore tipo 3	43	tastatore tipo 4	44	tastatore tipo 5	.		50	lama rotante	51	lama x	52	lama y	.		100	fresa	.		da 110	fresa speciale tipo 1	a 129	fresa speciale tipo 20	.		da 150	inseritore tipo 1	a 169	inseritore tipo 20.
valore del parametro	tipologia associata																																																				
1	foratore																																																				
.																																																					
da 10	foratore speciale tipo 1																																																				
a 29	foratore speciale tipo 20																																																				
30	maschiatore																																																				
.																																																					
35	foratore con scarico																																																				
.																																																					
40	tastatore tipo 1																																																				
41	tastatore tipo 2																																																				
42	tastatore tipo 3																																																				
43	tastatore tipo 4																																																				
44	tastatore tipo 5																																																				
.																																																					
50	lama rotante																																																				
51	lama x																																																				
52	lama y																																																				
.																																																					
100	fresa																																																				
.																																																					
da 110	fresa speciale tipo 1																																																				
a 129	fresa speciale tipo 20																																																				
.																																																					
da 150	inseritore tipo 1																																																				
a 169	inseritore tipo 20.																																																				
1002	<p>Diametro utensile.</p> <p>Si tratta di un parametro notevole, assegnato anch'esso nel range dei parametri per i quali deve valere il test di uguaglianza tra lavorazione ed utensile.</p>																																																				

[1001–2000]	Parametri per i quali vale una impostazione di valore (test di uguaglianza tra lavorazione ed utensile)
[2001–3000]	Parametri per i quali vale una impostazione entro un intervallo (test di esistenza nel range tra lavorazione ed utensile)
1006	Diametro utensile punta 1 per multitestine
1007	Diametro utensile punta 2 per multitestine
1008	Diametro utensile punta 4 per multitestine
1009	Diametro utensile punta 5 per multitestine
2002	Velocità di rotazione
2005	Velocità di lavoro

ATTENZIONE:

Tutti i parametri di utensile possono essere assegnati su più colonne: in questi casi OPTIMIZE determina il valore effettivo del parametro sommando i valori di tutte le istanze.

8.3.1. Assegnazione di gruppi in aggancio**Tipologia di macchina:**

Una macchina con gruppi in aggancio solitamente è composta da due teste attrezzate nello stesso modo e distanziate in modo tale da poter eseguire le stesse lavorazioni contemporaneamente su due pezzi.

In questo caso OPTIMIZE esegue una ottimizzazione delle lavorazioni solo per un pezzo. È compito della plancia far lavorare anche la seconda testa contemporaneamente alla prima.

In alcuni casi si presenta invece la necessità di eseguire un unico pezzo con entrambe le teste, facendo in modo che i due gruppi lavorino in *aggancio*, cioè come se fossero un unico gruppo. In questo caso:

Configurazione in parametrica dei gruppi:

La parametrica deve essere configurata in modo che OPTIMIZE possa ricavare l'esistenza di un gruppo dominante, cioè quello che lavora di default, come gruppo MASTER e di uno o più gruppi in aggancio come gruppi SLAVE.

Per definire un gruppo come SLAVE di un determinato gruppo MASTER occorre impostare nella colonna di abilitazione del gruppo il valore così ricavato:

tipologia del gruppo (0 = disabilitato, 1 = superiore, 2 = inferiore, 3 = bilato) + (offset fisso pari a 10 moltiplicato per il numero del gruppo master).

Esempio: per un gruppo SLAVE superiore con gruppo master 2 si imposta: $1+2*10=21$.

Se in parametrica utensile non vengono impostati i parametri del gruppo slave (limiti assi), OPTIMIZE considera validi i parametri del gruppo master anche nel caso di analisi degli utensili del gruppo slave. Se invece vengono definiti solo alcuni parametri del gruppo slave, OPTIMIZE considera validi i parametri assegnati e per quelli non definiti utilizza quelli del gruppo MASTER.

Il solo parametro che deve necessariamente essere definito per il gruppo SLAVE è quello di abilitazione del gruppo.

Regole imposte per la definizione della parametrica:

1 - per ogni macchina può essere definito un solo gruppo MASTER e tutti gli altri gruppi devono essere definiti in aggancio (SLAVE) o disabilitati;

2 - se si disabilita un gruppo MASTER vengono disabilitati automaticamente anche tutti i suoi gruppi SLAVE;

3 - tra i diversi gruppi non deve esserci sovrapposizione nella numerazione degli utensili. In matrice pezzo, infatti, OPTIMIZE scrive gli utensili che possono lavorare contemporaneamente sulla stessa riga anche se appartengono sia al gruppo MASTER sia al gruppo SLAVE. Sulla base del numero dell'utensile, l'elaboratore è in grado di capire a quale gruppo appartiene.

Operazione che deve svolgere OPTIMIZE:

Nonostante la presenza di più gruppi sulla macchina, OPTIMIZE deve essere configurato come *ottimizzatore monogruppo* ed esso attiva tutte le funzioni specifiche di una ottimizzazione monogruppo.

Tipologia gruppo	di	Abilitazione	Azione
MASTER		Sì	Ricerca dell'utensile corretto sul gruppo MASTER e sui gruppi SLAVE abilitati
SLAVE		Sì	
MASTER		No	Vengono disabilitati anche i gruppi SLAVE del gruppo MASTER e non viene avviata nessuna ricerca o controllo sull'utensile
SLAVE		Sì	

8.3.2. Assegnazione di utensili ruotati

OPTIMIZE gestisce la possibilità di generazione automatica di utensili di foratura assegnati per le facce laterali, con rotazioni di 90°, 180°, 270°.

Questo avviene tramite l'assegnazione del parametro di tipologia [45]. Il valore del parametro determina la modalità di funzionamento:

- ✓ 0 = nessuna rotazione della testa (corrisponde anche al caso di parametro non assegnato)
- ✓ 1 = rotazione della testa di 90°, 180° e 270°
- ✓ 2 = rotazione della testa di 90°
- ✓ 3 = rotazione della testa di 180°
- ✓ 4 = rotazione della testa di 270°

Se per un utensile è assegnato il parametro [45]: OPTIMIZE crea nella propria lista di utensili uno o più nuovi utensili con correttori modificati sulla base della rotazione imposta alla testa:

- ✓ rotazione di 90°
 - ◇ correttore x = correttore y * 1.0
 - ◇ correttore y = correttore x * -1.0
- ✓ rotazione di 180°
 - ◇ correttore x = correttore x * -1.0
 - ◇ correttore y = correttore y * -1.0
- ✓ rotazione di 270°
 - ◇ correttore x = correttore y * -1.0
 - ◇ correttore y = correttore x * 1.0

La numerazione assegnata agli utensili inseriti rispecchia uno schema fisso:

- (utensile originale + 200) in caso di rotazione di 90°
- (utensile originale + 400) in caso di rotazione di 180°
- (utensile originale + 600) in caso di rotazione di 270°.

Con questa stessa numerazione sono riportati gli utensili in matrice pezzo: così l'ottimizzatore custom o l'elaboratore sanno di quanto viene ruotato l'utensile.

ATTENZIONE: in programmazione l'utente "vede" solo l'utensile originale, non quelli ruotati.

Un utensile con parametro [45] di valore assegnato diverso da zero può eseguire delle lavorazioni di foratura anche sulle facce fittizie non simili. L'utensile non deve però essere definito come utensile universale (in questo caso OPTIMIZE non gestirebbe la rotazione automatica dell'utensile).

OPTIMIZE si comporta secondo le seguenti regole:

- programmazione per diametro di lavorazione di foratura su faccia fittizia: OPTIMIZE ricerca il primo utensile con parametro [45] che possa eseguire la lavorazione escludendo il test sui limiti (che viene lasciato all'ottimizzatore custom);
- programmazione per utensile di lavorazione di foratura su faccia fittizia: OPTIMIZE verifica se l'utensile può eseguire la lavorazione escludendo il test sui limiti (che viene lasciato all'ottimizzatore custom).

8.3.3. Assegnazione dell'utensile speculare

OPTIMIZE può gestire la riassegnazione dell'utensile di lavoro in caso di esecuzione speculare:

- tramite il parametro [45], ma solo per le facce laterali;
- tramite il parametro [124], che assegna direttamente l'utensile in esecuzione speculare.

L'utilizzo del parametro [45] si basa sulla considerazione che un utensile può lavorare sulla sua faccia *mirror* se ruotato di 180° (la considerazione è limitata alle facce laterali).

Si distinguono i casi:

- ✓ in parametrica non è assegnata la colonna a tipologia [45] oppure è assegnata ma l'utensile non è ruotato di 180°
 - ◇ se è definito un utensile speculare (parametro [124]), OPTIMIZE lo utilizza, altrimenti considera che non vi sia un utensile speculare (non esegue alcuna riassegnazione dell'utensile);
- ✓ in parametrica è assegnata la colonna a tipologia [45] e l'utensile è ruotato di 180°
 - ◇ l'utensile ruotato di 180° è assegnato come utensile speculare: in caso di utilizzo dell'utensile speculare, in matrice pezzo viene riportato il numero ad esse assegnato ((utensile originale)+400);
- ✓ in parametrica è assegnata la colonna a tipologia 45 ma in essa non è definita alcuna rotazione (ha valore 0):
 - ◇ solo nel caso di foratori orizzontali, OPTIMIZE imposta automaticamente l'utensile *mirror*: in caso di utilizzo dell'utensile speculare, in matrice pezzo viene riportato il numero ad esse assegnato ((utensile originale)+400).

8.3.4. Assegnazione di testine multipunta per lavorazioni puntuali

Si pone la necessità di assegnare testine a più punte, con programmazione per utensile e conseguente grafica corretta del reale disassamento delle punte nel piano xy.

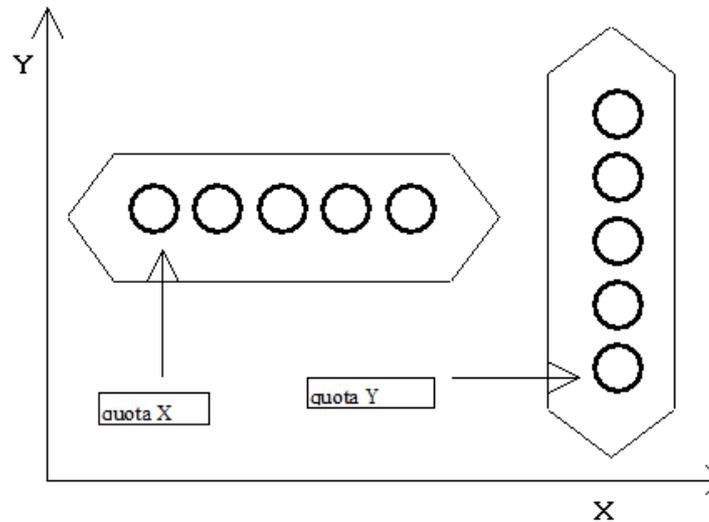
Nel caso di programmazione per utensile, la gestione delle testine a più punte riguarda esclusivamente EDICAD. Diverso è il caso di programmazione per diametro di forature singole e conseguente match con una testina a più punte, match che il programma OPTIMIZE non gestisce. In questo caso la scelta dell'utensile dovrebbe essere delegata ad un ottimizzatore custom.

La trattazione che è qui esposta riguarda la soluzione adottata per il caso di programmazione per utensile.

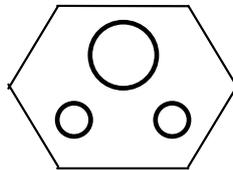
Si definiscono i criteri e limiti di configurazione:

- si tratta di testine con punte disassate in x e/o in y;
- l'utilizzo di testina multipla è solo per lavorazioni puntuali;
- la posizione programmata non corrisponde necessariamente al punto di aggancio della testina.

Un caso tipico è rappresentato da una testina di foratura multipla disassata lungo l'asse x oppure y:



Si pone anche il caso di testina di foratura per inserimento di cerniera:



La modalità di assegnazione di una testina a più punte è definita come segue:

- la testina è parametrizzata mediante assegnazione di un utensile di riferimento (attrezzaggio);
- i parametri primari della testina (correttori, diametro,...) si devono riferire alla posizione assunta di centro par la testina. In particolare: correttori (x,y,z), diametro (#0 se in centro testina è montata una punta, =0 se in centro testina non c'è punta), offset di simmetria (x ed y; vedi oltre);
- l'attrezzaggio imposta i parametri per un massimo di 4 punte aggiunte. In particolare: per ogni punta sono definiti il diametro (=0 se punta non esiste) ed i correttori (x ed y).

Parametri notevoli di assegnazione di testina a più punte:

Tipologia parametro	Significato parametro
6	Abilitazione di utensile: assegna le facce del pezzo su cui può lavorare l'utensile.
100	Correttore x utensile (rispetto allo zero x del gruppo)
101	Correttore y utensile (rispetto allo zero y del gruppo)
102	Correttore z utensile (rispetto allo zero z del gruppo)
109	Lunghezza utensile (di solito il valore è assegnato nell'utensile di riferimento)
1001	Tipologia utensile (di solito il valore è assegnato nell'utensile di riferimento). Nelle applicazioni di utensili multipunte sono assegnate tipologie notevoli per riconoscere i diversi utensili. In particolare, una prima distinzione discrimina tra 3 differenti gruppi di utensili: <ul style="list-style-type: none"> ✓ testine multipunte con disassamento delle punte in x; ✓ testine multipunte con disassamento delle punte in y; ✓ testine multipunte con disassamento delle punte in x ed in y. Non è riportata qui la classificazione iniziale delle tipologie: ogni applicazione ha infatti operato modifiche. Si rimanda quindi alle singole applicazioni, per un esame dettagliato delle tipologie.
1002	Diametro punta centrale (di solito il valore è assegnato nell'utensile di riferimento):

	=0: punta non presente #0: punta presente
221	Utensile di riferimento
Parametri assegnati per l'utensile di riferimento:	
[..]	Può assegnare tutti o in parte i parametri sopra riportati (se già assegnati: vengono sommati i valori)
136	Offset di simmetria X: assegna la posizione X del punto di applicazione della testina. =0: quota X del punto di applicazione coincide con la posizione della punta centrale #0: quota X del punto di applicazione = (posizione della punta centrale) – (offset di simmetria X). Nel caso di testina con disassamento delle punte in Y: può essere ignorato.
137	Offset di simmetria Y: ✓ nel caso di testina con disassamento delle punte in X: assegna la distanza (Y) della fila di punte rispetto al punto di aggancio della testina; ✓ altrimenti (testina con disassamento delle punte in Y oppure in X ed Y): assegna la posizione Y del punto di applicazione della testina: ◇ =0: la quota Y del punto di applicazione coincide con la posizione della punta centrale ◇ #0: quota Y del punto di applicazione = (posizione della punta centrale) – (offset di simmetria Y)
1006	Diametro punta aggiunta numero 1: =0: punta non presente → non interpreta testina multipunta; #0: punta presente
128	Correttore x punta aggiunta numero 1: relativo alla posizione della punta centrale (negativo se è alla sua "sinistra", positivo se è alla sua "destra")
129	Correttore y punta aggiunta numero 1: relativo alla posizione della punta centrale (negativo se è in basso rispetto alla punta centrale, positivo se è in alto)
1007	Diametro punta aggiunta numero 2: =0: punta non presente → non interpreta le punte aggiunte 3 e 4; #0: punta presente
130	Correttore x punta aggiunta numero 2: relativo alla posizione della punta centrale (negativo se è alla sua "sinistra", positivo se è alla sua "destra")
131	Correttore y punta aggiunta numero 2: relativo alla posizione della punta centrale (negativo se è in basso rispetto alla punta centrale, positivo se è in alto)
1008	Diametro punta aggiunta numero 3: =0: punta non presente → non interpreta la punta aggiunta 4; #0: punta presente
132	Correttore x punta aggiunta numero 3: relativo alla posizione della punta centrale (negativo se è alla sua "sinistra", positivo se è alla sua "destra")
133	Correttore y punta aggiunta numero 3: relativo alla posizione della punta centrale (negativo se è in basso rispetto alla punta centrale, positivo se è in alto)
1009	Diametro punta aggiunta numero 4: =0: punta non presente; #0: punta presente
134	Correttore x punta aggiunta numero 4: relativo alla posizione della punta centrale (negativo se è alla sua "sinistra", positivo se è alla sua "destra")
135	Correttore y punta aggiunta numero 4: relativo alla posizione della punta centrale (negativo se è in basso rispetto alla punta centrale, positivo se è in alto)

8.3.5. Assegnazione e gestione doppi pistoni

La soluzione dei doppi pistoni è adottata per aumentare la corsa effettiva di un utensile.

La parametrizzazione dell'utensile che può operare con differente settaggio dei pistoni deve essere assegnata sull'aggregato:

- rispetto alla situazione di zero con il primo pistone attivo;
- i correttori sono assegnati per l'aggregato.

In particolare, per l'aggregato di riferimento sono definiti 3 parametri di extracorsa:

- PREXTRAN [126] extra corsa per il primo pistone. Viene sottratto.
- PREXTRAP [127] extra corsa per il secondo pistone. Viene sommato.
- PREXTRAP3[110] extra corsa per il terzo pistone. Viene sommato.

In programmazione di lavorazione (EDICAD): l'operatore assegna il parametro di scelta del funzionamento dei doppi pistoni:

- ✓ tipologia parametro: 43;
- ✓ valori validi per il parametro:
 - ◇ 0 = selezione automatica dell'inserimento dei pistoni (condizione di default): è il programma OPTIMIZE (o l'ottimizzatore custom) che determina la situazione di lavoro consentita. La scelta automatica tiene in considerazione solo i pistoni 1 e 2;
 - ◇ 1 = inserisce il pistone 1;
 - ◇ 2 = inserisce il pistone 2;
 - ◇ 3 = inserisce i pistoni 1 e 2;

- ◇ 4 = non inserisce pistoni;
- ◇ 8 = inserisce il pistone 3;
- ◇ 9 = inserisce i pistoni 1 e 3;
- ◇ 10 = inserisce i pistoni 2 e 3;
- ◇ 11 = inserisce i pistoni 1, 2 e 3.

In lavorazione, il parametro [43] è normalmente assegnato con selezione in lista: in tale ipotesi la selezione è limitata ai primi 5 valori (da 0 a 5).

La possibilità di gestire anche il pistone 3 è invece da ritenersi utile in assegnazione diretta da parametrica: se il parametro [43] è assegnato in parametrica utensile con valore >0, viene ignorata la assegnazione in lavorazione.

In lavorazione è possibile assegnare anche il parametro di tipologia [44], per imporre una scelta forzata nella assegnazione dei pistoni. Si distinguono i casi:

- ✓ parametro [43] assegnato in parametrica con valore >0: OPTIMIZE ignora i parametri di gestione pistoni della lavorazione e per la scelta assegnata effettua i necessari controlli (verifica se l'assetto pistoni operato è compatibile con la geometria assegnata);
- ✓ parametro [43] non assegnato in parametrica o assegnato con valore =0: OPTIMIZE interpreta i parametri di gestione pistoni della lavorazione. In particolare:
 - ◇ parametro [43] >0 e parametro [44]=0: per la scelta assegnata effettua i necessari controlli (verifica se l'assetto pistoni operato è compatibile con la geometria assegnata);
 - ◇ parametro [43] >0 e parametro [44]=1: per la scelta assegnata non effettua alcun controllo.

8.3.6. Gestione setup con utensili multipli

Per una lavorazione di setup è possibile programmare un massimo di 4 utensili:

- ✓ in generale ciò rende necessaria la assegnazione di un codice custom di lavorazione;
- ✓ il primo utensile deve essere scritto in matrice pezzo alla colonna 9 (T1); i rimanenti tre utensili devono essere scritti rispettivamente alla colonna 10 (T2), 11 (T3) e 12 (T4);
- ✓ EDICAD non interpreta in alcun modo gli utensili aggiunti a quello di base.

Se è abilitata la gestione dei setup con utensili multipli (vedi EDICAD, settaggi per l'Ottimizzatore, voce "*Utensili Tn in Matrice pezzo*") OPTIMIZE aggiunge delle righe notevoli (records) notevoli in Matrice pezzo (vedi oltre):

- record di codice [3006]: riporta la sequenza dei cambi utensile relativi al primo utensile;
- record di codice [3007]: riporta la sequenza dei cambi utensile relativi al secondo utensile;
- record di codice [3008]: riporta la sequenza dei cambi utensile relativi al terzo utensile;
- record di codice [3009]: riporta la sequenza dei cambi utensile relativi al quarto utensile.

9. MATRICE PEZZO

È riportata la assegnazione delle righe notevoli della matrice pezzo, come prodotta dall'ottimizzatore generale (OPTIMIZE.EXE). Questa documentazione può servire solo per una indicazione di massima: per aggiunte o personalizzazioni della matrice si raccomanda comunque di consultare sempre il tecnico TPA di riferimento.

Occorre distinguere tra due formati gestiti (vedi: settaggi di ottimizzatore):

- formato mono-gruppo;
- formato multi-gruppo.

In caso di formato **mono-gruppo**, la matrice pezzo è assegnata sui singoli records (righe) come riportato di seguito:

- record di definizione pezzo (codice: 1);
- records di definizione utensili (codici: 3006, 3007, 3008, 3009): riportano la sequenza dei cambi utensile, ma solo per le lavorazioni di setup. Le righe non sono necessariamente presenti tutte e quattro;
- record di intestazione della 1° faccia che ha lavorazioni assegnate (codice: 2);
- records delle lavorazioni della faccia (ordinate secondo i criteri scelti dall'utente): il codice operativo di ogni record dipende dalla lavorazione;
-
- record di intestazione dell'ultima faccia che ha lavorazioni assegnate (codice: 2);
- records delle lavorazioni della faccia;
- record di chiusura della matrice pezzo (codice: 32000).

Nel caso di file di matrice pezzo in formato testo: ogni riga riporta i valori di ogni colonna separati dal `;`.

In caso di due `;` consecutivi: il valore della colonna è nullo (=0).

Nel caso di file di matrice pezzo in formato binario: ogni cella della matrice ha tipologia [float].

Esempio di matrice pezzo mono-gruppo in formato testo:

```

"EXT MxPezzo"
-----|-----|-----|-----|
1;6;;;0;.....1400;400;20;.....
3006;.....112;113;110;.....
3007;.....112;113;110;.....
2;1;.....0;0;20;1400;0;20;0;400;20;.....
89;1;1;0;...112;.....-100;100;10;...0;0;...7;
89;1;1;0;...113;.....-100;100;10;...0;0;...8;
89;1;1;0;...110;.....-100;100;10;...0;0;...9;
81;1;1;0;...1;.....-100;100;10;...0;0;...3;
81;1;1;1;0;...15;.....35;200;10;...0;0;...2;
2;3;.....0;0;0;1400;0;0;0;20;.....
81;3;1;1;0;...77;76;.....100;10;10;...0;0;...1;
2;4;.....1400;0;0;1400;400;0;1400;0;20;.....
81;4;1;1;0;...63;62;61;64;65;.....200;8;10;...0;0;...1;
32000;.....
  
```

← 1° riga di intestazione (*)
 ← 2° riga di intestazione (*)
 ← definizione pezzo
 ← definizione utensili
 ← definizione utensili
 ← definizione faccia 1
 ← 1° lavorazione di faccia 1

 ← definizione faccia 3

 ← definizione faccia 4

 ← chiusura matrice

In caso di formato **multi-gruppo**, la matrice pezzo è assegnata sui singoli records (righe) come riportato di seguito:

- records di definizione pezzo (codice: 1): è scritto un record per ogni macchina;
- records di definizione utensili (codici: 3006, 3007, 3008, 3009): riportano la sequenza dei cambi utensile, ma solo per le lavorazioni di setup. Le righe non sono necessariamente presenti tutte e quattro;
- record di intestazione della 1° faccia che ha lavorazioni assegnate (codice: 2);
- record di intestazione della 2° faccia che ha lavorazioni assegnate (codice: 2);
- ...
- record di intestazione dell'ultima faccia che ha lavorazioni assegnate (codice: 2);
- records delle lavorazioni assegnate per il pezzo: il codice operativo di ogni record dipende dalla lavorazione;
- record di chiusura della matrice pezzo (codice: 32000).

Esempio di matrice pezzo multi-gruppo in formato testo:

```

"EXT MxPezzo"
-----|-----|-----|-----|
1;6;1;1;...7;9;.....1500;1000;50;.....
1;6; 2;1;...7;9;.....1500;1000;50;.....
3006;.....112;113;110;.....
2;1;.....0;0;20;1400;0;20;0;400;20;.....
2;3;.....0;0;0;1400;0;0;0;20;.....
2;4;.....1400;0;0;1400;400;0;1400;0;20;.....
89;1;1;0;...112;.....-100;100;10;...0;0;...7;
81;1;1;0;...1;.....-100;100;10;...0;0;...3;
  
```

← 1° riga di intestazione (*)
 ← 2° riga di intestazione (*)
 ← definizione pezzo (mod0)
 ← definizione pezzo (mod1)
 ← definizione utensili
 ← definizione faccia 1
 ← definizione faccia 3
 ← definizione faccia 4
 ← lavorazioni pezzo

81;1;1;1;0;15;35;200;10;0;0;0;2;
81;3;1;1;0;77;76;100;10;10;0;0;0;1;
81;4;1;1;0;63;62;61;64;65;200;8;10;0;0;0;1;
89;1;1;0;113;-100;100;10;0;0;0;8;
89;1;1;0;110;-100;100;10;0;0;0;9;
32000;

← chiusura matrice

NOTA (*)

Le due righe di intestazioni indicate sono presenti solo nel file di matrice pezzo in formato testo.

	Nome colonna	Definizione pezzo [mono-gruppo]	Definizione pezzo [multi-gruppo]	Definizione utensili	Definizione faccia
1	CODOP	1	1	3006 – 3007 – 3008 -3009	2
2	Faccia	numero facce del pezzo			numero faccia
3	Piano	quadrante operativo (1- 4)			
4	Macchina		macchina	Macchina	
5	Gruppo			Gruppo	
6	Origine	1= origini in trasparenza			origine faccia o faccia simile (se faccia simile)
7	Gint1	release di PARTEC			orientamento terna (0=DX, 1= SX)
8	Gint2	release di PARTOOL			
9	T1		indice 1° lavorazione in gruppo 1	1° utensile utilizzato	
10	T2		idem in gruppo 2	idem 2° utensile	
11	T3		idem in gruppo 3	idem 3° utensile	
12	T4		idem in gruppo 4	idem 4° utensile	
13	T5		idem in gruppo 5	idem 5° utensile	
14	T6		idem in gruppo 6	idem 6° utensile	
15	T7		idem in gruppo 7	idem 7° utensile	
16	T8		idem in gruppo 8	idem 8° utensile	
17	T9		idem in gruppo 9	idem 9° utensile	
18	T10		idem in gruppo 10	idem 10° utensile	
19	T11	maschera gruppi abilitati	maschera gruppi abilitati	idem 11° utensile	
20	T12			idem 12° utensile	
21	T13			idem 13° utensile	
22	T14			idem 14° utensile	
23	T15			idem 15° utensile	
24	T16			idem 16° utensile	
25	T17			idem 17° utensile	
26	T18			idem 18° utensile	
27	qX	lunghezza pezzo			P0 X
28	qY	altezza pezzo			P0 Y
29	qZ	spessore pezzo			P0 Z
30	qA	offset x			P1 X
31	qB	offset y			P1 Y
32	qI	offset z			P1 Z
33	qJ	var0			P2 X
34	qK	var1			P2 Y
35	qRe	var2			P2 Z
36	qLav	var3			
37	qRu	var4			
38	vRe	var5			
39	vLav	var6			
40	vRu	var7			
41	gDoub1				
42	gDoub2				
43	gDoub3				
44	Diametro				
..					
N	Linea				

	Nome colonna	Puntuale	Setup verticale	Setup orientato	Setup lama	Lineare	Arco	Logica custom	
1	CODOP	codice operativo				2412	2413	cod. operativo	
2	Faccia	Faccia							
3	Piano	tipo & sottotipo (0 + sottotipo (0 - 4))	tipo & sottotipo(10+ sottotipo (0 - 4))				Piano: 0=xy 1=xz 2=yz	tipo & sottotipo (20+ sottotipo)	
4	Macchina	Macchina							
5	Gruppo	Gruppo							
6	Origine	Campo O							
7	Gint1						Rotazione: 0=orario 1=antiorario		
8	Gint2								
9	T1	Utensile 1							
10	T2	Utensile 2							
11	T3	Utensile 3							
12	T4	Utensile 4							
13	T5	Utensile 5 1=scelta forzata pistone							
14	T6	Utensile 6 X-min ingombro della fresata							
15	T7	Utensile 7 X-max ingombro della fresata							
16	T8	Utensile 8 Y-min ingombro della fresata							
17	T9	Utensile 9 Y-max ingombro della fresata							
18	T10	Utensile 10 Z-min ingombro della fresata							
19	T11	Utensile 11 Z-max ingombro della fresata							
20	T12	Utensile 12 correzione utensile (0=off/ 1=Sx/ 2=Dx)							
21	T13	Utensile 13 raggio di correzione utensile							
22	T14	utensile 14							
23	T15	utensile 15		scelta pistone (imposta o determinata in automatico): 1= pistone 1; 2= pistone 2 3=pistoni 1+2; 4=nessun pistone 8=pistone 3; 9=pistoni 1+3 10=pistoni 2+3; 11=pistoni 1+2+3					
24	T16	utensile 16 1= ultima interpolazione del profilo							
25	T17	utensile 17			1=doppia passata				
26	T18	utensile 18			1=calcolo corda				
27	qX	quota X							
28	qY	quota Y							
29	qZ	quota Z							
30	qA	angolo C						Raggio	
31	qB	angolo B						Angolo iniziale	
32	qI	centro X							
33	qJ	centro Y							
34	qK	centro Z							
35	qRe	q.rallentamento in entrata			quota Z		quota X1 (archi xyz)		
36	qLav	quota Z			quota Z di 1° passata		quota Y1 (archi xyz)		
37	qRu	q.rallentamento in uscita					quota Z1 (archi xyz)		
38	vRe	velocità di rallentamento in entrata							
39	vLav	Velocità entrata				Velocità di interpolazione			
40	vRu	velocità di rallentamento in uscita			velocità di 2° passata				
41	gDoub1	Velocità rotazione utensile							
42	gDoub2	campo M							

43	gDoub3	tipologia utensile			
44	Diametro	diametro utensile (in caso di programmazione per diametro)			
..					
N-2		Ottimizzazione monogruppo: numero sequenza primaria			
N-1		Ottimizzazione monogruppo: numero sequenza secondaria (per sviluppo di sottoprogramma)			
N	Linea	Ottimizzazione monogruppo: progressivo originale della lavorazione; Ottimizzazione multigruppo: indice di lavorazione successiva che lavora sullo stesso gruppo.			

La colonna indica con (Indice = N, Nome colonna = Linea) è l'ultima colonna della matrice, in base al dimensionamento dichiarato nei settaggi generali per l'Ottimizzatore.